

Patrik

Inštalácia

Príklady výhodného použitia aplikácie Patrik.

Ďo je to Patrik

Ďo Patrik umožňuje

Pri spustení aplikácie je možné špecifikovať Command Line parametre.

Hlavné okno aplikácie

Základné princípy aplikácie

Príručka k programu **PATRIK** je určená pre skúsených užívateľov a znalcov MS SQL servera. Predpokladá, že ste oboznámení so základnými koncepciami databáz a s ich spravovaním. (Viac informácií o spravovaní databáz viď Microsoft SQL Server Books Online.) Táto príručka tiež predpokladá, že ste oboznámení s Microsoft® Windows NT™.

Èo je to Patrik

PATRIK je databázový administratívny nástroj, ktorý bol vytvorený ako nevyhnutná pomôcka pri vývoji projektov, ktoré obsahujú viac ako 1000 SQL tabuliek, 800 tabu³/₄kových relácií, 1500 uložených procedúr a gigabajty dôležitých údajov. Celá štruktúra databázy bola od začiatku až do konca vyvinutá a spravovaná v PATRIK-ovi. Výsledkom toho je, že sa odstránili snáï všetky “detské choroby”. Aplikácia je nazvaná pod³/₄a svojho tvorca, Patrika Suenského.

PATRIK ponúka nasledujúce možnosti:

- browsovanie, editovanie a vytlaèenie relaèných modelov
- vytváranie a menenie štruktúry databázy
- browsovanie a editovanie tabuliek obsahujúcich údaje
- tvorenie, porovnávanie, editovanie a debug zdrojových objektov
- nástroj na dokumentovanie databázy
- online and offline prenos celej databázy alebo jej èastí
- client/server architektúru
- .. porovnávanie databázy
- .. nástroj na pohodlné vytváranie Queris - zostáv
- a mnoho ďalších

Ěo Patrik umožňuje

Použí [Query Editor](#) s lexikálním farbením textu a debuggerem

Kvůli lepší orientaci funguje v okně [Query Editor](#) lexikální farbení textu - farabne rozlišuje 10 syntaktických tříd slov. Jedna s tříd - **User defined** představuje uživatelem definované slova. Editor obsahuje integrovaný [debuger](#), v který pomáhá při odlaňování chýb v procedurách.

Graficky zobrazí [relaěné modely databáz](#)

PATRIK poskytuje možnost [browsování](#) a editování relaěných modelů (štruktúry databázy a relací mezi tabulkami). V okně Relational Model PATRIK zobrazuje tabulky s definovaným primárním klíčem, nebo tie, které sa vzáhujú k tabulke s definovaným primárním klíčem. PATRIK umožňuje tvorbu, tlaění a editování relaěných vzáhy mezi tabulkami. Okrem toho sa budete môcť lepšie orientovať v databázach, ktoré obsahujú veľké množstvo tabuliek, ako aj vzáhov medzi nimi, ak si na niekoľko strán vytlaěíte relaěný model danej databázy a zložíte tieto výtlaěky do jedného veľkého plagátu, ktorý môžete zavesiť na stenu. Viac informácií nájdete v kapitole [Relaěný model](#).

Definová [poháady](#), aby práca bola efektívnejšia

Āk databáza obsahuje veľa tabuliek, je veľami užitoěné vytvoriť v okne **Relational Model** poháady, z ktorých každý zobrazí len urěité tabulky. Umožní to omnoho úeinnejšie browsovať a editovať relaěné vzáhy medzi tabulkami v okne **Relational Model**. Viac informácií nájdete pod heslom [Príkazy Select, Deselect](#)

Porovnáva [a synchronizová štruktúry a údaje v dvoch databázach](#)

V databáze môžete porovnávať štruktúru tabuliek a ostatných objektů, ako aj obsahy tabuliek vybraných za účelom zistenia rozdielů medzi dvomi databázami. Okrem toho môžete nastaviť generovanie vyrovnávacích skriptů, ktoré budú synchronizovávať štruktúru alebo údaje databáz. Viac informácií nájdete v kapitole [Porovnávanie](#).

Pohodne modifiková [štruktúru databázy](#)

Keď robíte zmeny v databáze na SQL Serveri, je výhodné vytvoriť si takzvanú pracovnú alebo downsized databázu, ktorá je štruktúrovo identická s databázou na SQL Serveri. V tejto databáze môžete robiť nespočetné zmeny a keď skoněíte, spustíte upsíze pracovnej (downsized) databázy, čím sa uskotoěnia zmeny v reálnej databáze na SQL Serveri. Zmena skutoěnej databázy na SQL Serveri cez downsized databázu je prínosom v dvoch smeroch:

1. Implementovanie každej zmeny, ktorú robíte v databáze na SQL Serveri, môže trvať veľami dlho. Tým, že použijete pracovnú databázu, môžete urobiť niekoľko zmien naraz a potom spustiť upsíze pracovnej databázy do skutoěnej databázy na SQL Serveri. Môžete spustiť upsíze skutoěnej databázy a potom odísť, čím sa šetrí čas, ktorý by ste museli vždy stráviť čakáním na implementáciu zmien, ktoré ste vykonali v databáze na SQL Serveri.

2. Reálna databáza na SQL Serveri zostáva konzistentná - prejde s jedného konzistentného stavu do druhého behom Upsize-u, ktorý netrvá dlho.

Viac informácií nájdete pod heslom [Príkaz Downsize to Patrik](#) a [Príkaz Upsize to SQL](#).

Vyhnuť sa opakovaniu operácií použitím automatických nástrojov na modifikovanie štruktúry databázy a údajov

Príkaz **Save Downsize** slúži na uloženie zmien, ktoré ste vykonali v štruktúre databázy na jednom SQL Serveri a ich implementovanie do databázy na ďalšom SQL Serveri. Štruktúra databáz na oboch SQL Serveroch musí byť identická. Viac informácií nájdete pod heslom [Príkaz Save downsize](#).

Príkaz **Record Script** slúži na uloženie zmien, ktoré ste vykonali v údajoch databázy na jednom SQL Serveri a ich implementovanie do databázy na druhom SQL Serveri, za predpokladu, že údaje v oboch databázach na oboch SQL Serveroch sú identické. Viac informácií nájdete pod heslom [Príkaz Record script](#).

Podstatne modifikovať štruktúru databázy - viď [Design Table](#)

- a) Zmeniť následnosť atribútov v tabuľke.
- b) Zmeniť dátový typ atribútu.
- c) Vymazať atribút zo stredu tabuľky. Viac informácií pod heslom
- d) Pridať atribút k akejkoľvek časti tabuľky. Viac informácií pod heslom

Rýchlo diagnostikovať tabuľku

Môžete rýchlo zistiť počet záznamov a veľkosť tabuľky. Viac informácií pod heslom [Príkaz Select count\(*\)](#).

Preniesť databázu z jedného SQL Servera na druhý bez online pripojenia

Môžete exportovať niektoré, alebo všetky objekty aktuálnej databázy na jednom SQL Serveri do cieľovej databázy na druhom SQL Serveri, ktorý nie je napojený ani cez LAN (Local Area Network) ani WAN (Wide Area Network). Prenos offline sa uskutočňuje pomocou prenosných dátových zariadení. Viac informácií pod heslom [Príkaz OFF line export](#) a [Príkaz OFF line import](#).

Pridať vrstvu komentárov k databázovým tabuľkám, aby sa zvýšila prehľadnosť v databáze

Môžete pridať poznámky k relačným vzťahom medzi tabuľkami, databázovými objektami ako aj tabuľkami a ich atribútami. (Viac informácií pod heslom [Príkaz Edit notes](#), [Editovanie relačných vzťahov medzi tabuľkami](#)).

Vytvoriť dokumentáciu databázy

Môžete sa veľmi rýchlo a pohodlne vytvoriť dokumentáciu k databáze, ktorá bude obsahovať aj poznámky, ktoré ste urobili k databázovým objektom a tabuľkovým údajom. Viac informácií pod heslom [Príkaz Make document](#).

Inštalácia Patrika

V tejto kapitole sa dozviete, aké požiadavky kladie PATRIK na systém a aké kroky musíte vykonať pre úspešné nainštalovanie alebo odinštalovanie PATRIKA. Ďalej zistíte, čo robí, keď sa prvýkrát pripájate na SQL server.

Požiadavky na systém

Inštalácia

Odinštalovanie Patrika

Vytvorenie databázových zariadení

Prvé pripojenie na SQL server a vytvorenie databázy Patrik

Požiadavky na systém

PATRIK vyžaduje minimálne nasledovnú konfiguráciu softwaru a hardwaru:

- IBM PC kompatibilný s operačným systémom MS Windows NT 3.51, MS Windows 95 alebo novším
- 16 MB RAM
- MS-SQL CLIENT 6.0 alebo novší
- pripojenie na server s MS SQL SERVER 6.0 alebo novší
- 10 MB voľného miesta na hard disku pracovnej stanice
- 20 MB voľného miesta na hard disku každého SQL servera, na ktorom sa bude s Patrikom pracovať

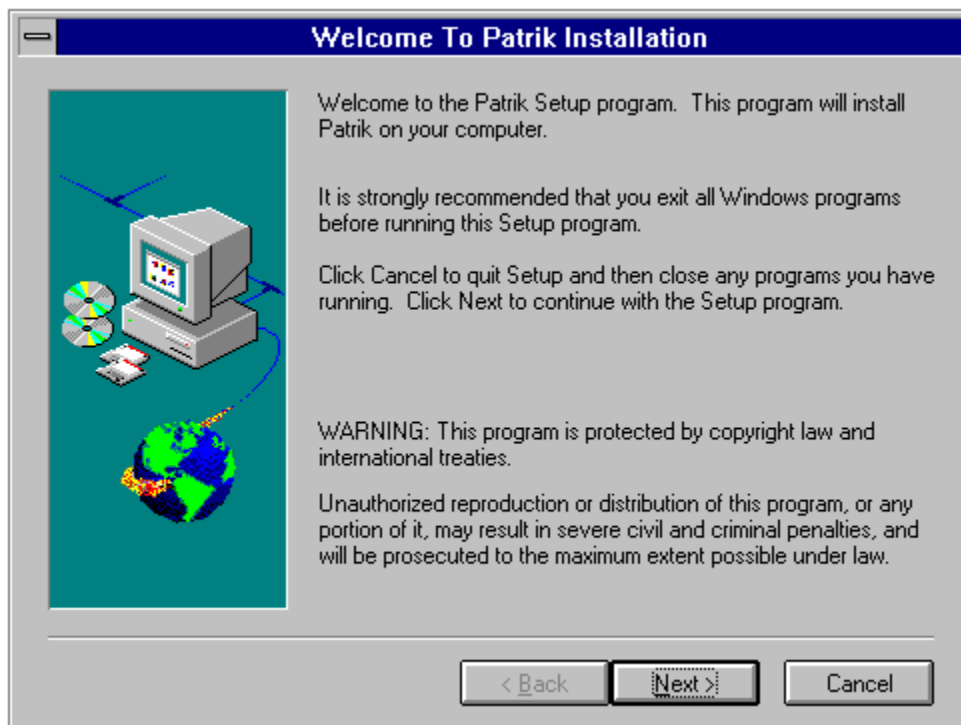
Inštalácia

Inštalovaný program PATRIKA sa dodáva na inštalovaných disketách. Názov inštalovaného programu je **SETUP.EXE** a poskytuje všetky nutné komponenty potrebné pre správnu inštaláciu.

Úspešná inštalácia PATRIKA

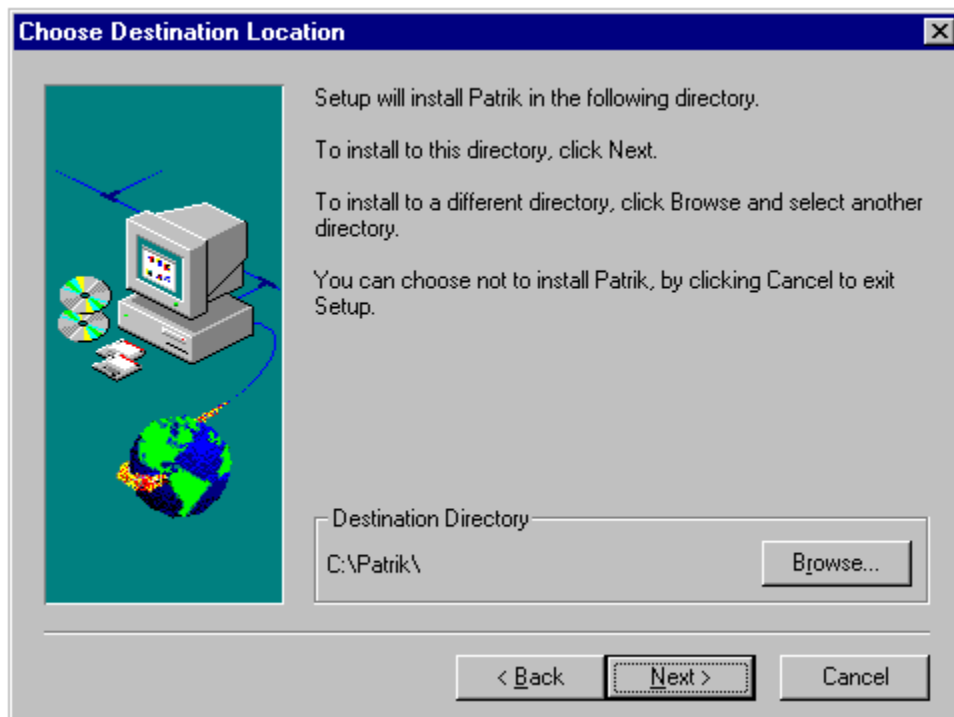
1. Prihláste sa na Windows NT pracovnú stanicu ako lokálny administrátor.
2. Vložte inštalovateľnú disketu do disketovej mechaniky A.
3. Vo File Manageri spustíte z inštalovateľnej diskety SETUP.EXE.

Zobrazí sa dialógové okno **Welcome to Patrik Installation**.

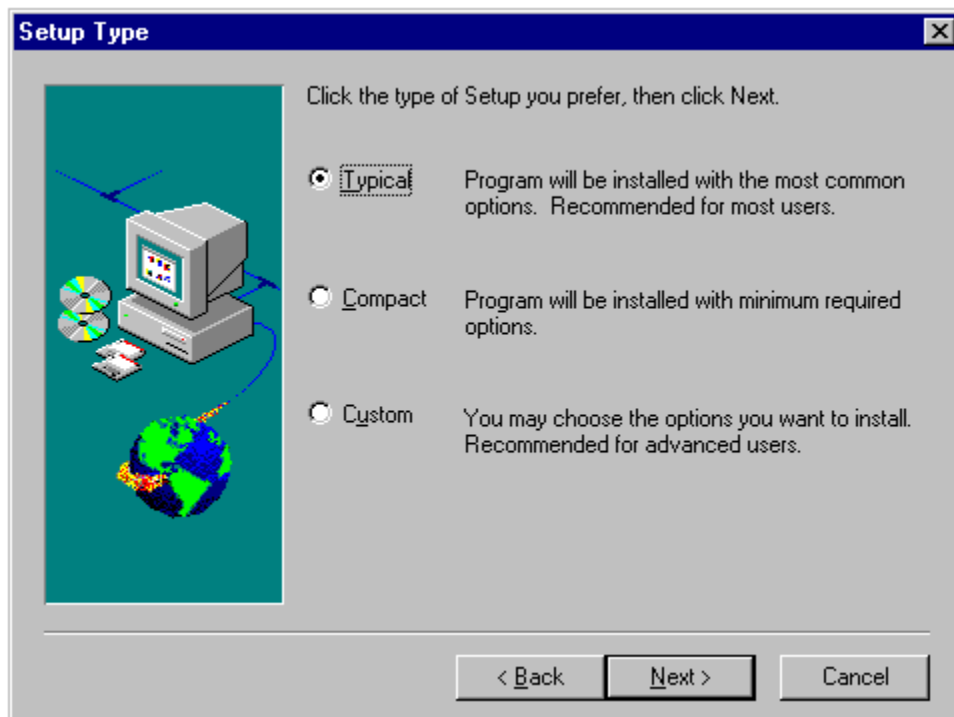


4. Potom ako si prečítate všetky dôležité informácie v dialógovom okne **Welcome To Patrik Installation**, stlačíte tlačidlo Next.

Zobrazí sa dialógové okno **Choose Destination Location**.

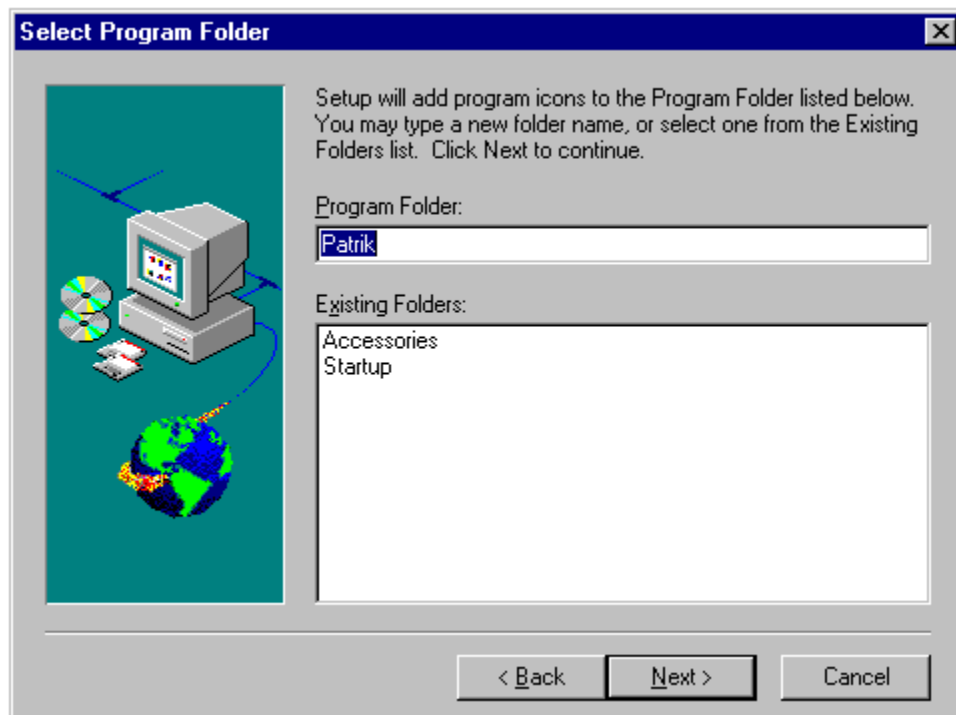


5. Setup štandardne navrhne inštaloanie aplikácie PATRIK do adresára Patrik. Ak chcete PATRIKa inštaloava do tohoto adresára, stlaète tlačítko **Next**. Ak chcete zmeni cie³/ový adresár, stlaète tlačítko **Browse**. Setup zobrazí dialógové okno **Choose Directory**, v ktorom musíte urèi na ktorú diskovú jednotku chcete PATRIKa inštaloava. Keï dokonèíte výber, stlaète tlačítko **Next**. Zobrazí sa dialógové okno **Setup Type**.



6. Vyberte typ inštalácie a stlačte tlačítko **Next**.

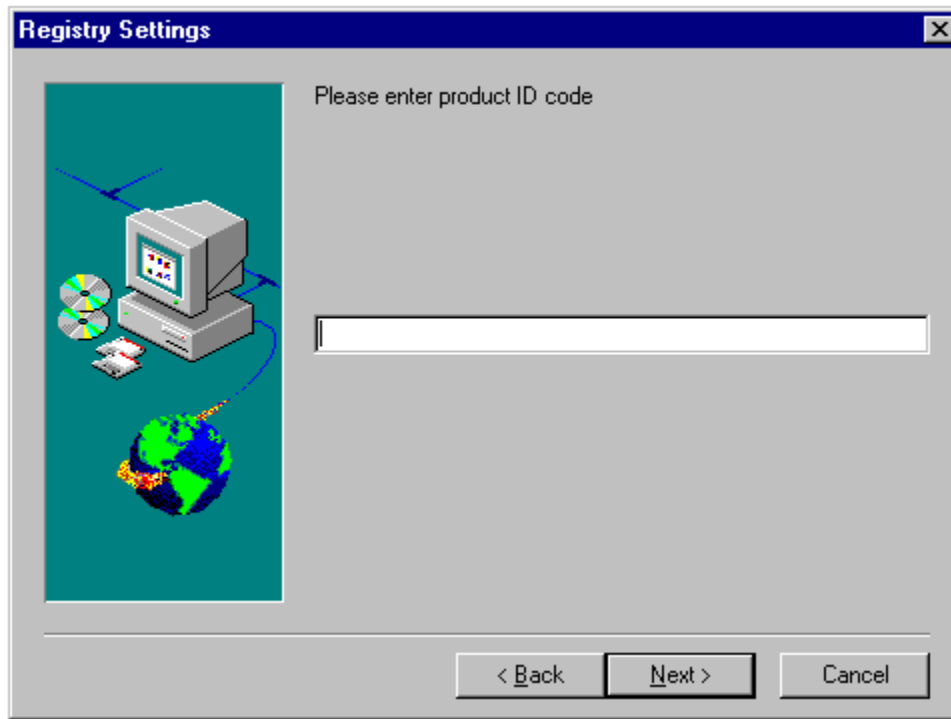
Zobrazí sa dialógové okno **Select Program Folder**.



7. Vyberte folder, do ktorého sa nainštaluje ikona programu PATRIK. Ak chcete inštalova

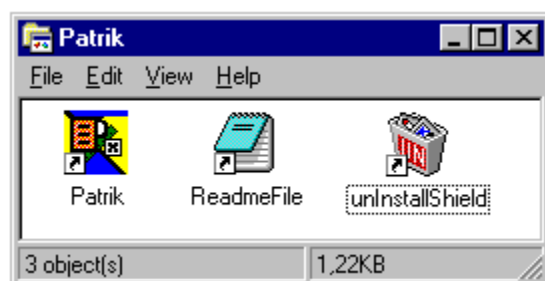
ikony do foldera **Patrika**, stlačte tlačítko **Next**.

Zobrazí sa dialógové okno **Registry Settings**.

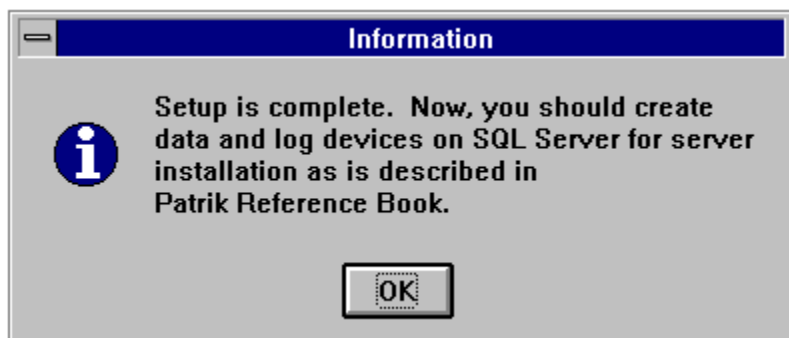


8. Do prázdneho poľa vpište ID číslo programu a stlačte tlačítko **Next**. Toto číslo nájdete na licencií vložených do inštalovaného balíka **PATRIKa**.

Setup vytvorí skupinu programu **PATRIK**, ktorá obsahuje ikony **PATRIKa**.



Po úspešnej inštalácii sa zobrazí nasledujúce dialógové okno.



1. Stlaète tlačítko **OK**, aby sa dokonèila inštalácia **PATRIKa**.

Okrem aplikácie a jej helpu a iných potrebných súborov sa nainštalujú aj [scripty plniace niektoré užitoènè funkcie](#).

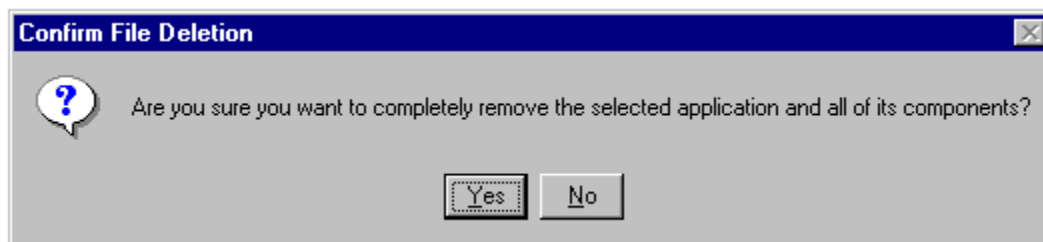
Odiňštalovanie Patrika

Ak chcete úplne odstrániť PATRIKA z hard disku pracovnej stanice:

1. Dvakrát kliknite na ikonu **UnInstallShield** v programovej skupine **PATRIKa**.

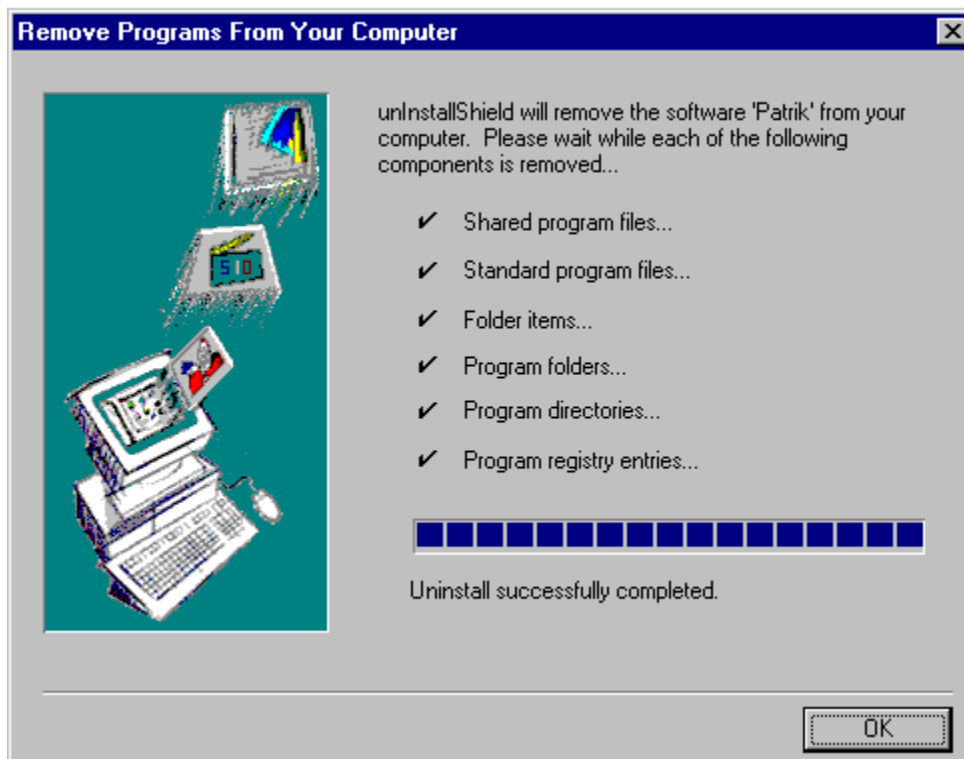


Zobrazí sa dialógové okno **Confirm File Deletion**.



2. Ak chcete, aby sa dokončilo odiňštalovanie **PATRIKa**, stlačíte tlačítko **Yes**.

Zobrazí sa dialógové okno **Remove Programs From Your Computer**.



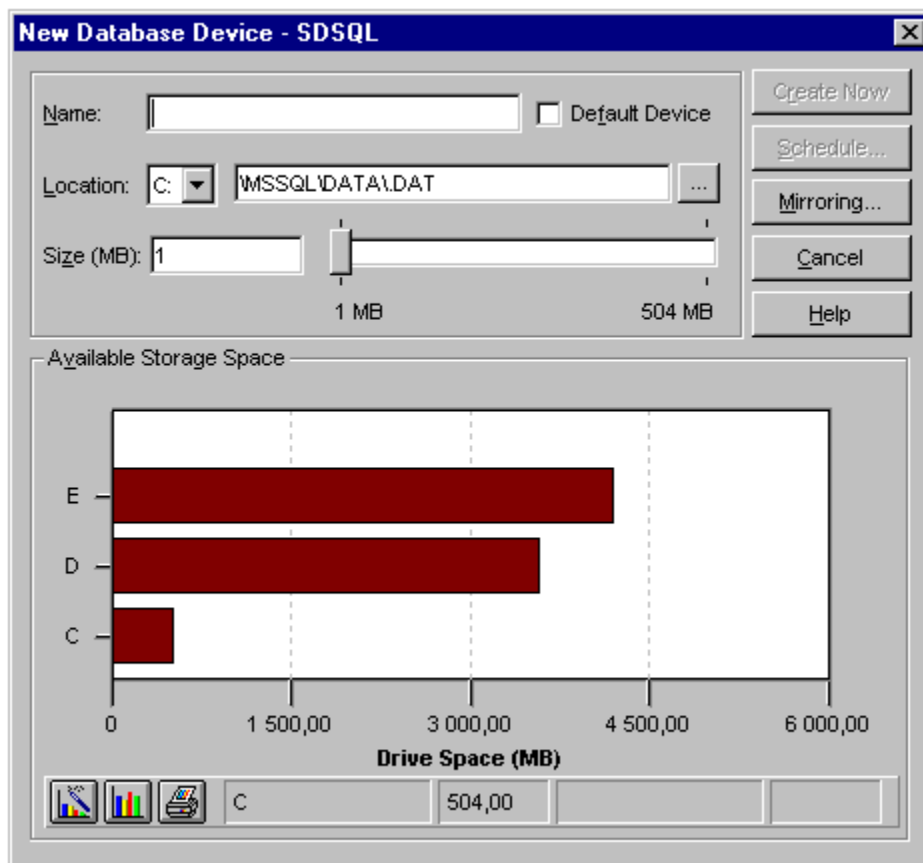
1. Stlaète tlaèitko **OK**.

Vytvorenie databázových zariadení

Skôr ako sa prvýkrát pripojíte na SQL Server, musia byť na dom vytvorené dve databázové zariadenia. Prvé zariadenie - Patrik_log - slúži ako transakčný log databázy a v druhom - Patrik_dat - sa uchovávajú údaje. Obe zariadenia by mali mať veľkosť aspoň 20 MB.

Vytvorenie databázových zariadení

1. Otvorte aplikáciu **Microsoft SQL Enterprise Manager**.
2. Vyberte SQL Server, na ktorý sa chcete v PATRIKovi pripojiť.
3. Kliknite na folder **Database Devices**.
4. Kliknite na pravé tlačítko myši.
Objaví sa **Shortcut menu**.
5. Vyberte príkaz **New Device**.
Zobrazí sa **New Database Device - názov_servera**.



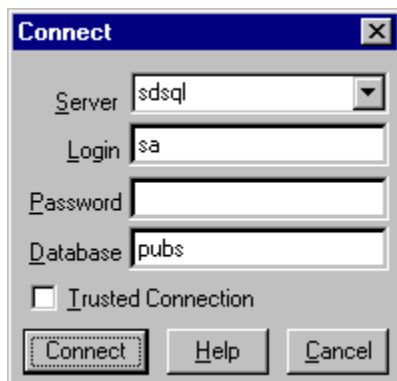
6. Do poľa **Name** napíšte názov zariadenia.

7. Do pol'ya **Size (MB)** urèite ve'kos' zariadenia.
1. Stlaète tlaèitko **Create Now**.

Prvé pripojenie na SQL server a vytvorenie databázy Patrik

Dvakrát kliknite na ikonu Patrika, čím spustíte aplikáciu **PATRIK**. **PATRIK** otvorí **hlavné okno aplikácie** a dialógové okno **Connect**.

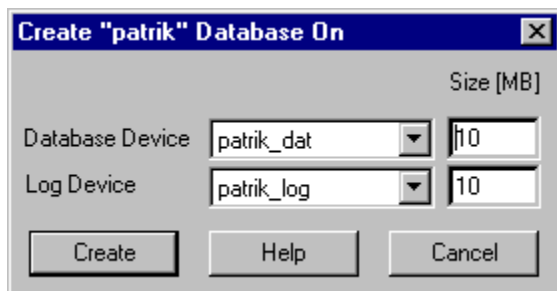
Aby ste sa pripojili na SQL Server, vypíšte v dialógovom okne Connect kolonky:



- **Server** - meno , alebo IP adres SQL Servra na ktorý sa má Patrik napojiť. Kolonka obsahuje názvy všetkých servrov, na ktoré sa Patrik s danej pracovnej stanice kedy úspešne napojil.
- **Login** - meno užívateľského účtu na SQL servri pod ktorým sa má napojenie uskutočniť
- **Password** - heslo daného Login-u (účtu)
- **Database** - meno databázy, do ktorej sa má Patrik po prihlásení na server nastaviť. Ak túto kolonku nevyplníte (alebo vyplníte invalidné meno databázy) tak sa patrik nastaví do default databázy daného Login-u (účtu).
- **Trusted Connection** - v prípade, že je na danom SQL servri nastavená **mixed security**, tak týmto **checkboxom** môžete nastaviť, či má byť uplatnená **integrated security** (checkbox zaškrtnutý), alebo **standard security**

Ak sa pripájate na SQL Server prvýkrát, PATRIK automaticky sa pokúsi na servri vytvoriť svoju vlastnú databázu **patrik**.

V dialógovom okne **Create Patrik Database On** musíte zadať názvy dvoch zariadení (**Datadevice** a **Logdevice**). Na týchto zariadeniach bude vytvorená databáza **patrik**.



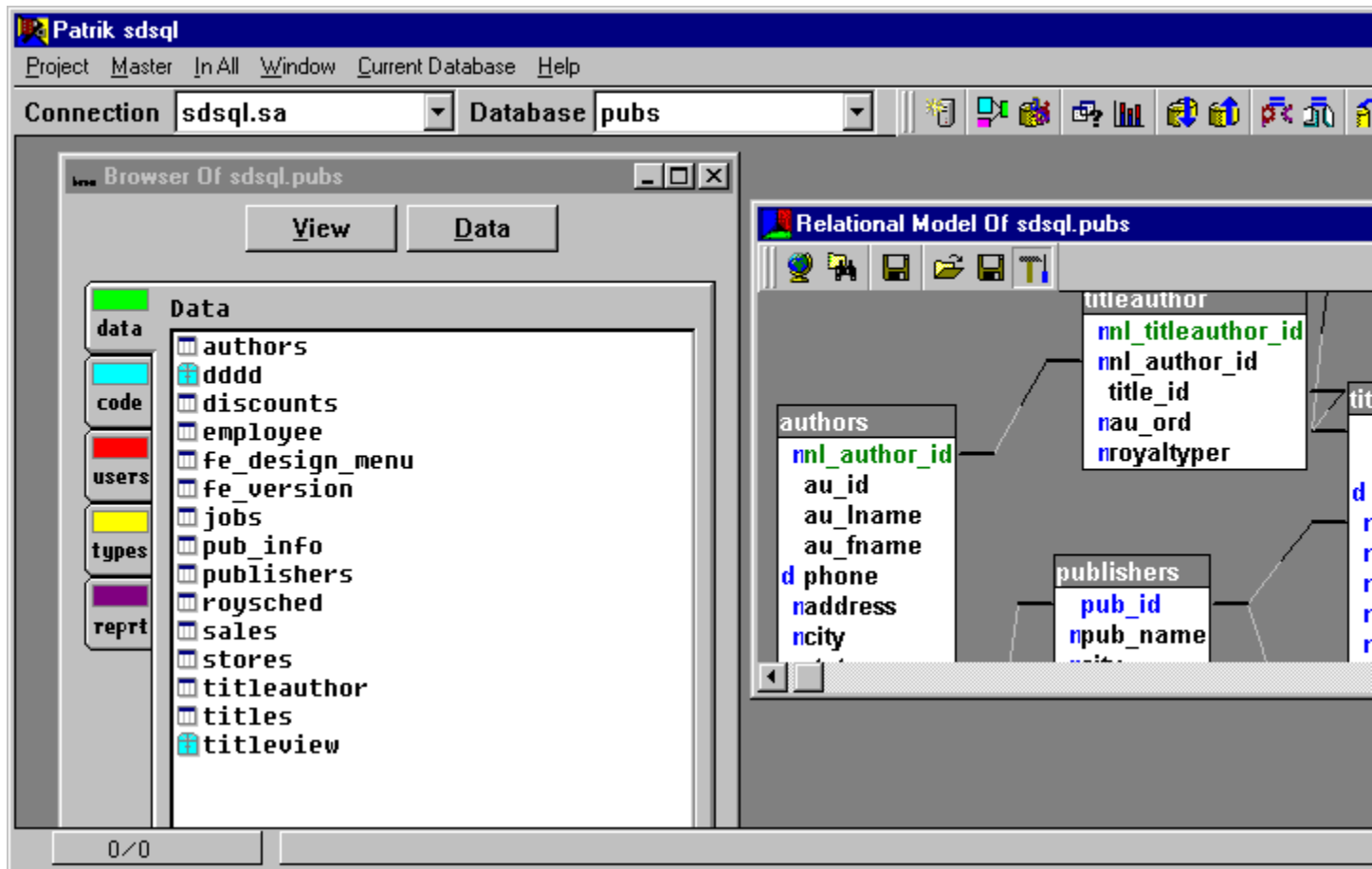
1. V combo boxe **Database Device** vyberte zariadenie **Patrik_dat**.
1. V combo boxe **Log Device** vyberte zariadenie **Patrik_log**.

1. Stlaète tlaèitko **Create**.

Hlavné okno aplikácie

Táto kapitola vysvetľuje položky, ktoré sa zobrazujú na obrazovke a poskytuje základné informácie o rôznych položkách v menu PATRIKa. Okrem toho v nej nájdete názvy zložiek **hlavného okna aplikácie**.

Po spustení aplikácie sa zobrazí jej **hlavné okno**. Pokiaľ v **Command Line-e** nebol špecifikovaný SQL server kam sa má Patrik napojiť, tak sa automaticky zobrazí dialógové okno **Connect** v ktorom je možné (a **nutné**) naviazať napojenie na nejaký SQL server.















Pracovná plocha pozostáva zo zložiek

- **titulok okna** - obsahuje okrem názvu aplikácie aj meno servera, na ktorý je momentálne Patrik napojený.
- **hlavné menu aplikácie**
- **Toolbar** - obsahuje Combo box connection(pripojenie), Combo box database(databáza) a Tlačítka hlavného okna aplikácie
- **informačnú lištu** - nachádza sa v na spodku okna a obsahuje
 - **číselné informácie** - ich význam sa mení s aktívnym oknom v užívateľskom prostredí aplikácie
 - **textové informácie** - sem sa vypisujú informačné hlásenia aplikácie

- èas

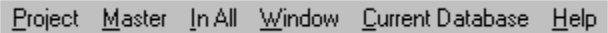
Tlaèítka toolbaru

V hlavnom okne aplikácie sa nachádzajú tlaèítka najèastejšie používaných príkazov:

-  Connect... - vykoná napojenie na (ia³/₄ší) SQL server.
-  Relational Model- otvorí okno relaèného modelu aktuálnej databázy.
-  Browser - otvorí databázový braowser
-  Query Editor . otvorí editor SQL scriptov
-  Table sysprocesses - slúži na mapvvanie aktivity procesov na servri
-  Downsize - príprava na modifikáciu databázy
-  Upsize - aplikovanie modifikácie databázy
-  Compare Schema - porovnanie štruktúry dvojice databáz
-  Compare Data - porovnanie dát vybraných tabuliek
-  Online Transfer- prenos (èasti) databázy do inej databázy
-  Offline Transfer - prenos (èasti) databázy na disk
-  Options - nastavenie správania aplikácie

Hlavné menu

Hlavné menu v hlavnom okne aplikácie obsahuje nasledovné položky:



Project Master In All Window Current Database Help

Project - všeobecné funkcie na ovládanie Patrik-a.

Master - obsahuje mená systémových tabuliek nachádzajúcich sa iba v databáze master (**System Catalog**).

In All - obsahuje mená systémových tabuliek nachádzajúcich sa v aktuálnej databáze (**Database Catalog**). Po zvolení konkrétnej funkcie (s menu **Master** alebo **In All**) s menom tabuľky sa dáta tabuľky zobrazia v jednoduchom tabuľkovom pohľade. Dáta systémových tabuliek sú podľa možnosti prikrášené tak, že sa za niektoré číselné hodnoty dosadia im adekvátne popisovo-textové hodnoty.

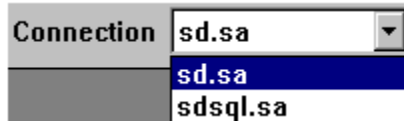
Windows - obsahuje funkcie, slúžiace na usporiadanie okien na ploche aplikácie.

Current Database - obsahuje príkazy pre operácie s aktuálnou databázou.

Combo box connection (pripojenie)

V combo boxe **Connection** je možné si vybrať SQL server, na ktorom chcete pracovať, zo zoznamu SQL serverov, na ktoré je PATRIK pripojený. Záznamy v combo boxe **Connection** majú nasledovnú syntax: *názov_serveru.meno_užívateľa*.

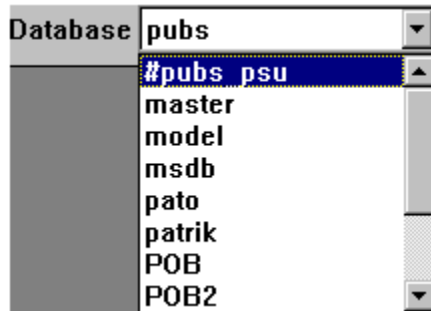
Aby ste mohli pridať nový SQL server do combo boxu **Connection**, musíte použiť príkaz **Connect...** na pripojenie v menu **Project** a pripojiť sa na nový SQL server.



Combo box database (databáz)

Databázy SQL servera, na ktorom práve pracujete, sú uvedené v combo boxe **Database**. Databáza, ktorej názov je určený v tomto combo boxe sa považuje za aktuálnu databázu, na ktorú sa vzťahujú všetky príkazy s menu Current Database.

Keď si chcete zvoliť nejakú databázu ako aktuálnu, vyberte názov databázy v combo boxe **Database**.



Niektoré mená databáz sa môže začínať znakom #. Jedná sa o pracovné databázy (vzniknulé funkciou Downsize), ktoré sa v skutočnosti nenachádzajú na SQL serveri, ale ich definícia sa nachádza v databáze patrik.

PATRIK poskytuje možnosť voliť veľa príkazov zo **Shortcut menu**, ktoré obsahuje príkazy vzťahujúce sa k položkám, s ktorými práve pracujete. Aby sa zobrazilo **Shortcut menu** konkrétneho okna alebo konkrétnej kolonky, umiestnite kurzor myši jej plochu a kliknite na pravé tlačítko myši.

Tabuška sysprocesses

Obsahuje záznamy systémovej tabušky sysprocesses. V tejto tabuške sa nachádza záznam pre každý proces, ktorý momentálne beží na serveri. V Shortcut Menu sa nachádzajú funkcie :

- Related Table - slúži na zobrazenie relačných záznamov
- Order By - slúži na utriedenie záznamov podľa vybraného stĺpca
- Edit WHERE - slúži na definovanie WHERE podmienky pre záznamy tabušky
- Refresh - opätovné načítanie záznamov tabušky
- Show Input Buffer - zobrazí posledný príkaz, ktorý vybraný proces spustil
- Send Message To Owners - pošle **net send** majiteľom označených procesov. Text správy je možné zadať v otvorenom sa dialógovom okne. Text správy môže obsahovať reťazec %s , ktorý pri posielaní Patrik nahradí menom aplikácie, ktorá má vybraný proces na serveri otvorený.

Dáta systémových tabuľiek

V okne sa nachádzajú záznamy danej systémovej tabuľky. V Shortcut Menu sa nachádzajú funkcie :

- Related Table - slúži na zobrazenie relačných záznamov
- Order By - slúži na utriedenie záznamov podľa vybraného stĺpca
- Edit WHERE - slúži na definovanie WHERE podmienky pre záznamy tabuľky
- Refresh - opätovné načítanie záznamov tabuľky

Menu project

Táto kapitola opisuje príkazy v menu **Project** a v **hlavnom menu**.

Connect - napojenie na (nový) **SQL server**

Disconnect - odpojenie z aktuálneho SQL Servra

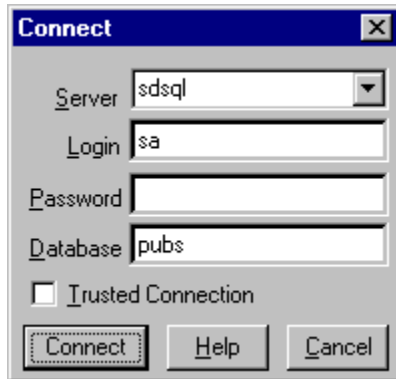
Query - editor SQL dotazov

Options

Exit

Pripojenie(connect)

Aby ste sa pripojili na SQL server, musíte v dialógovom okne **Connect** vyplniť potrebné informácie.



- **Server** - meno , alebo IP adres SQL Servra na ktorý sa má Patrik napojiť. Kolonka obsahuje názvy všetkých servrov, na ktoré sa Patrik s danej pracovnej stanice kedy úspešne napojil.
- **Login** - meno užívateľa ského účtu na SQL servri pod ktorým sa má napojenie uskutočniť
- **Password** - heslo daného Login-u (účtu)
- **Database** - meno databázy, do ktorej sa má Patrik po prihlásení na server nastaviť. Ak túto molonku nevyplníte (alebo vyplníte invalidné meno databázy) tak sa patrik nastaví do default databázy daného Login-u (účtu).
- **Trusted Connection** - v prípade, že je na danom SQL servri nastavená **mixed security**, tak týmto **checkboxom** môžete nastaviť, či má byť uplatnená **integrated security** (checkbox zaškrtnutý), alebo **standard security**

Ak sa prihlasujete na SQL server prvýkrát, pozrite heslo [Prvé pripojenie na SQL server a vytvorenie databázy Patrik](#).

Odpojenie(disconnect)

Zvolením tohoto príkazu PATRIK zruší spojenie s aktuálnym serverom bez akéhokoľvek varovania. Ak malo napojenie otvorené nejaké transakcie, Patrik na to upozorní, lebo odpojením budú všetky zrušené (Rollback).

Exit

Ak chcete ukonèi □ program **PATRIK**, zvo³/₄te príkaz **Exit** v menu **Project**. Ak malo niektoré z SQL napojení otvorené nejaké transakcie, Patrik na to upozorní, lebo odpojením budú všetky zrušené (Rollback).

Menu current database

Táto kapitola opisuje príkazy v menu **Current Database**, ktoré sa nachádza v hlavnom menu. Dozviete sa ako sa dá správne a pohodlne modifikovať databáza na SQL Serveri pomocou tzv. pracovnej (downsized) databázy, ako získať informácie o aktuálnej databáze a ako rýchlo a ľahko vytvoriť dokumentáciu databázy.

Aktuálna databáza

Všetky príkazy obsiahnuté v tomto menu pracujú s databázou nastavenou v Database Comboboxe

Príkazy menu je možné rozdeliť do štyroch skupín podľa ich charakteru.

- **Nástroje**
 - Browser
 - Relational Model
- **Príkazy modifikácie databázy**
 - Downsize to Patrik...
 - Príkaz Upsize to SQL...
 - Príkaz Save downsize...
 - Príkaz Load downsize
 - Touch...
- **Príkazy spravujúce databázu**
 - Options...
 - Compare
 - Synchronize...
 - Transfer
 - Rename
 - Drop...
- **Ďalšie príkazy**
 - Info...
 - Make document...

Príkaz Downsize to Patrik

Keď sa robia zmeny v databáze na SQL serveri, často je výhodnejšie vytvoriť takzvanú pracovnú alebo downsized databázu, ktorá je štruktúrou identická s databázou na SQL serveri. V databáze je možné urobiť mnoho zmien a po úpravách spustiť update databázy na SQL server.

Uplatňovanie zmien v databáze na SQL serveri pomocou pracovnej (downsized) databázy je výhodné v dvoch smeroch:

1. Implementovanie každej zmeny, ktorá sa urobí v databáze na SQL serveri, môže trvať veľmi dlho. Použitím pracovnej (downsized) databázy sa dá previesť niekoľko zmien naraz a potom spustiť upsizovací skript pracovnej databázy do reálnej databázy na SQL serveri.
2. Databáza na SQL serveri zostáva konzistentná.

Downsize databázy zo SQL servera do Patrika

1. Zo zoznamu v combo boxoch **Connection** a **Database** zvolíte server a databázu, z ktorej chcete vytvoriť pracovnú kópiu (downsized databázu).
2. Z menu **Current Database** zvolíte príkaz **Downsize to Patrik**.

Na obrazovke sa zobrazí dialógové okno v ktorom sa dotazuje na meno novej pracovnej databázy. Nie je potrebné vymýšľať meno, PATRIK automaticky predpripraví názov novej pracovnej databázy dosadením aktuálnych hodnôt do formátu

<meno aktuálnej databázy>_<meno NT užívateľa>

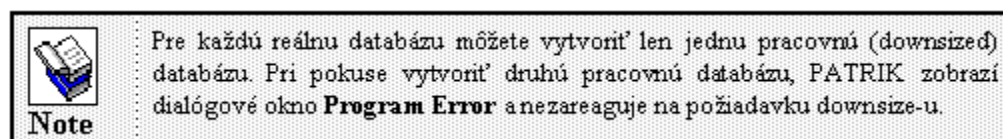


3. Ak súhlasíte s uvedeným názvom, stlačíte tlačidlo **OK**.
4. Ak ho chcete zmeniť, napíšete nový názov a stlačíte tlačidlo **OK**.

Pracovná databáza

PATRIK automaticky pridá znamienko # pred názov pracovnej (downsized) databázy. Následne pracovnú databázu daného názvu vytvorí skopírovaním štruktúry aktuálnej databázy. Do comboboxu **Database** v hlavnom okne aplikácie pridá názov vytvorenej databázy.

Pracovnú databázu je možné ubovočne modifikovať, chod originálnej databázy nebude narušený. Po vykonaní všetkých zmien v pracovnej databáze spustíte funkciu **Upsize To SQL script**, ktorý tie isté zmeny vykoná v originálnej databáze.

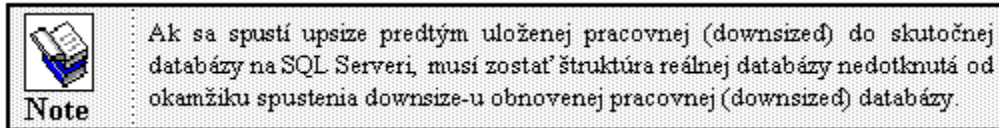


Zmeny, ktoré sa vykonali v pracovnej (downsized) databáze, je možné uložiť pomocou príkazu **Save downsize** v menu **Current Database**. Viac informácií nájdete pod heslom [Príkaz Save downsize](#).

Náročnejších užívateľov môže zaujímať popis [procesu downsize-u](#).

Príkaz Upsize to SQL server

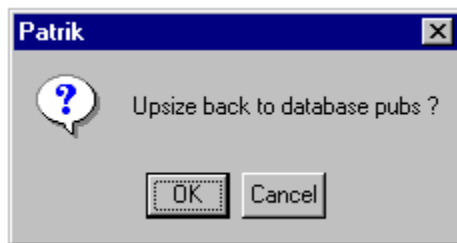
Je možné spustiť upsize pracovnej (downsized) databázy do reálnej databázy hneď po tom, ako sa vykonali zmeny v pracovnej (downsized) databáze, alebo sa dá spustiť upsize pracovnej (downsized) databázy, ktorá bola predtým uložená. Ak chcete obnoviť uloženú pracovnú (downsized) databázu zvoďte príkaz [Load Downsize](#) v menu **Current Database**.



Spustenie upsize pracovnej databázy

1. V combo boxe **Database** zvoďte pracovnú databázu, ktorú chcete upsize-núť.
2. V menu **Current Database** zvoďte príkaz **Upsize to SQL**.

Zobrazí sa toto dialógové okno.



3. Stlačte tlačítko **OK**.

Spustí sa proces upsize-u, ktorý na originálnej databáze aplikuje všetky zmeny vykonané na pracovnej databáze. Log (výsledky) upsize-u sa budú počas behu funkcie priebežne vypisovať do okna [Query Results](#). Posledný vypísaný text v okne bude : **Upsize Finished...** . Tento bude signalizovať skončenie priebehu funkcie. Následne sa pracovná databáza zmaže (**dropne**), lebo už nie je potrebná.

Náročnejších užívateľov môže zaujímať popis [procesu upsize-u](#) .

Príkaz Save downsize

Príkaz **Save Downsize** sa používa na uloženie zmien vykonaných v pracovnej databáze. Umožňuje spustiť upsize zmien neskôr, alebo do viacerých reálnych databáz na rôznych SQL serveroch, za predpokladu, že máte dve alebo viac identických reálnych databáz na dvoch alebo viacerých SQL serveroch a chcete implementovať zmeny na každom z nich.

Uloženie pracovnej databázy

1. Zvoľte príkaz **Save Downsize** v menu **Current Data**.
Zobrazí sa dialógové okno **Save As**.
2. Napíšte meno súboru a vyberte si diskovú jednotku a adresár, do ktorého chcete súbor uložiť.
 1. Stlačte tlačítko **OK**.
 2. Súbor s uloženým downsize-om je možné nahrať do identickej databázy (na inom servri) a Upsize-núť tam. Ušetrí to duplicitné vykonávanie rovnakých zmien na databázach s rovnakou štruktúrou (napr vo vašej databáze vo počítačovej firme a databáze u zákazníka)

Príkaz Load downsize

Otvorenie predtým uloženej downizovanej (pracovnej) databázy

1. Zvoľte príkaz **Load Downsize** v menu **Current Database**
Zobrazí sa dialógové okno **Open**.
2. Vyberte súbor, ktorý chcete otvoriť.
1. Stlačte tlačidlo **OK**.

Options

Zvolením príkazu **Options** v menu **Current Database** sa zobrazí dialógové okno **Database options**.



Dialógové okno **Database Options** pozostáva z dvoch podskupín.

- Options - pozostáva z checkboxov:

- Select Into/Bulkcopy
- Trunc. Log On Chkpt
- No Chkpt On Recovery
- Read Only
- Single User
- Dbo Use Only

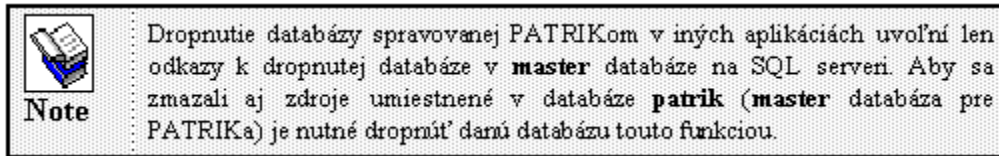
Výber akéhokoľvek check boxu definuje atribúty databázy na SQL serveri. Viac informácií nájdete v [SQL Server Books Online](#).

- Status - tu zobrazuje možné mimoriadne stavy aktuálnej databázy.

Možné stavy: **DATABASE IS IN RECOVERY**
 EMERGENCY MODE

Príkaz Drop database

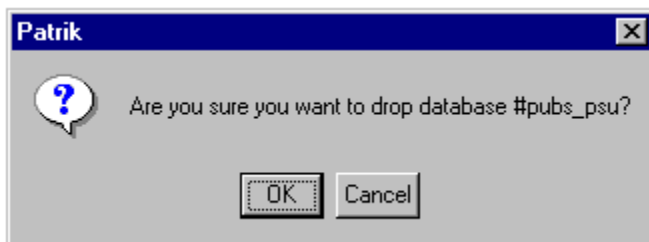
Použitím príkazu **Drop Database** sa vymaže celá aktuálna databáza. Dropnutie databázy odstráni všetky jej objekty a uvoľní sa priestor, ktorý databáza zaberala.



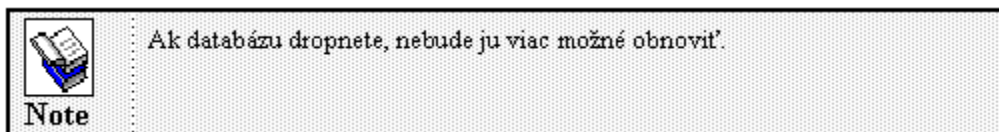
Dropnutie databázy

1. V combo boxe **Database** zvolíte databázu, ktorú chcete dropnúť. Vyberte príkaz **Drop Database** v menu **Current Database**.

Zobrazí sa nasledujúce dialógové okno.



2. Na potvrdenie voľby dropnúť databázu stlačíte tlačítko **OK**



Príkaz Rename

Zmenu názvu existujúcej databázy umožňuje príkaz **Rename**.

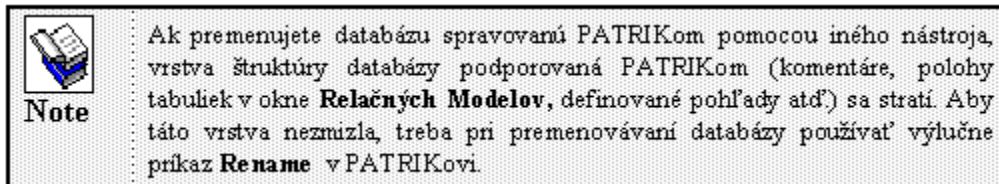
Premenovanie existujúcej databázy

1. V combo boxe **Database** zvolíte databázu, ktorú chcete premenovať.
2. Vyberte príkaz **Rename** v menu **Current Database**.

Zobrazí sa dialógové okno **Enter New DB Name**.



3. Napíšte nový názov databázy.
4. Stlačte tlačítko **OK**.



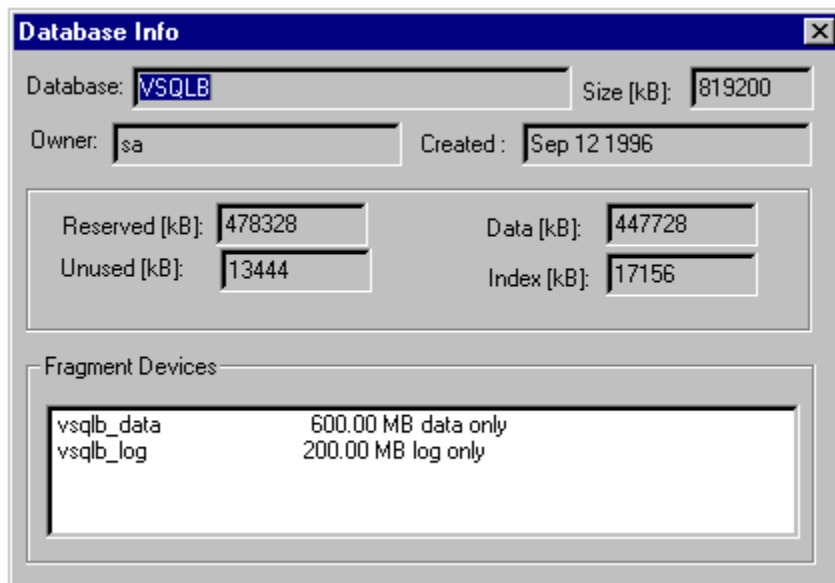
Príkaz Info databázy

Príkaz **Info** slúži na zobrazenie informácií o aktuálnej databáze.

Zobrazenie informácií o aktuálnej databáze

1. Zvoľte príkaz **Info** v menu **Current Database**.

Zobrazí sa okno **Database Info**.



- **Database** - názov aktuálnej databázy
- **Size (kB)** - veľkosť aktuálnej databázy v kilobytoch
- **Owner** - vlastník aktuálnej databázy
- **Created** - dátum vytvorenia aktuálnej databázy
- **Reserved (kB)** - priestor rezervovaný aktuálnej databáze na SQL Serveri
- **Unused (kB)** - nevyužitý rezervovaný priestor aktuálnej databázy na SQL Serveri
- **Data (kB)** - priestor využívaný údajmi na SQL Serveri
- **Index (kB)** - priestor využívaný indexami na SQL Serveri
- **Fragment devices** - zoznam zariadení, kde je umiestnená aktuálna databáza a ich veľkosť

Príkaz Make document

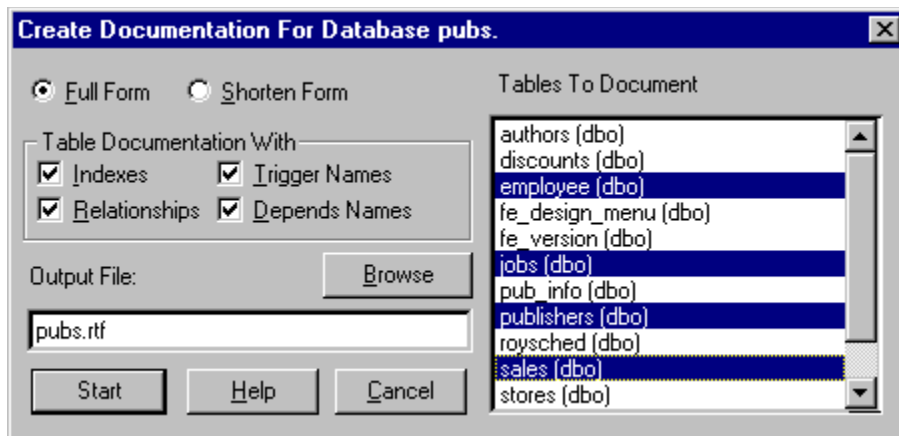
Príkaz **Make Document** slúži na vytvorenie textového dokumentu obsahujúceho informácie o tabuškách v aktuálnej databáze. Táto funkcia je užitočná nielen na vytvorenie elektronickej dokumentácie, ale aj tlačenej dokumentácie.

Dokument bude vytvorený vo formáte **rich text format** (názov_súboru.rtf). Bude prístupný v akomkoľvek textovom editore, ktorý podporuje tento formát (napr. MS Word, Wordpad).

Vytvorenie dokumentácie databázy

1. V combo boxe **Database** zvolíte databázu, pre ktorú chcete vytvoriť dokumentáciu.
2. Vyberte príkaz **Make Document** v menu **Current Database**.

Zobrazí sa dialógové okno **Create Documentation For Database** *názov_databázy*.



3. Pod skupinou **Table Documentation With** vyberte položku, ktorú chcete zahrnúť do dokumentácie.
4. Pod skupinou **Tables to Document** vyberte tabušky, ktoré chcete zahrnúť do dokumentácie.
5. Do políčka **Output File** zadajte názov a miesto, kde má byť dokument uložený. PATRIK automaticky navrhne umiestnenie dokumentu. Ak chcete zmeniť umiestnenie, napíšte path a názov dokumentu, alebo stlačte tlačítko Browse. PATRIK zobrazí dialógové okno Choose Directory, v ktorom zadáte súbor a adresár výstupu.
6. Radio voľbami **Full Form** a **Shorten Form** môžete nastaviť, či pôjde o štandardný typ dokumentácie (**Full Form**) - typ uvedený nižšie na obrázku, alebo o veľmi zjednodušenú dokumentáciu (**Shorten Form**), pozostávajúcu s jedinej tabušky so stĺpcami : Databáza, Tabuška, Meno stĺpca, Dátový typ stĺpca, komentár stĺpca, kde sú takto popísané všetky

7. stápage tabuliek vybraných v bode 4.
 7. Stlačte tlačítko **Start**.

Príklad dokumentovanej tabuľky zamestnancov:

Tabuľka : **employee (dbo)**

Ak ste zvolili check box **Trigger Names**, bude obsahovať aj túto časť.

Užívateľské meno vlastníka tabuľky

Triggers :insert- employee_insupd, update- employee_insupd

Depends :trigger- employee_insupd

Ak ste zvolili check box **Depends Names**, bude obsahovať aj túto časť.

Column name	Type	Rule	Default	Comment
emp_id	empid			employee identifier
fname	varchar(20)			employee first name
minit	char(1) NULL			
lname	varchar(30)			employee last name
job_id	smallint		DF__employee_job_id_2FAF1EF9	employee job
job_lvl	tinyint		DF__employee_job_lv_3197676B	employee job level
pub_id	char(4)		DF__employee_pub_id_328B8BA4	id of publisher where employee works
hire_date	datetime		DF__employee_hire_d_3473D416	the date when employee was hire

Ak ste zvolili check box **Indexes**, bude obsahovať aj túto časť.

Index name	Type	Sgmt	Indexed columns
employee_ind	Clustered	default	lname, fname, minit
PK_emp_id	Unique	default	emp_id

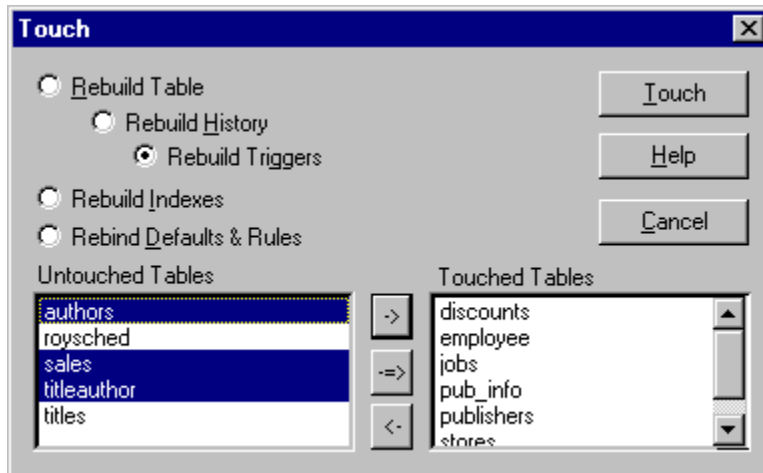
Related table	Relationship type	Linked columns	Comment
n->1 publishers	Constraint (Conditional Update, Delete)	pub_id - pub_id	
n->1 jobs	Opnstraint (Conditional Update, Delete)	job_id - job_id	

Ak ste zvolili check box **Relationships**, bude obsahovať aj túto časť.

Touch

Príkaz, ktorým je možné pre vybrané tabučky pracovnej databázy nasimulovať, že boli počas práce s databázou modifikované (napriek tomu, že v skutočnosti neboli). Týmto sa dá dosiahnuť pregenerovanie niektorých vlastností (prípadne celých) tabuliek použitím funkcie [Upsize](#).

V dialógovom okne



1. Vyberte typ pregenerovania ktorý sa má pre vybrané tabučky uskutočniť.
 - **Rebuild Table** - pregenerovanie všetkého, čo s tabučkami súvisí vrátane celých tabuliek.
 - **Rebuild History** - prebudovanie historických tabuliek pre dané tabučky a ich triggrov
 - **Rebuild Triggers** - pregenerovanie triggrov tabuliek
 - **Rebuild Indexes** - prebudovanie indexov tabuliek
 - **Rebind Defaults & Rules** - odpojenie a opätovné napojenie defaultov a rulov stápcov tabuliek
2. Z **Untouched Tables** presunte pomocou tlačítok -> {presun označených tabuliek) a ==> (presun všetkých tabuliek) tabučky, ktoré chcete pregenerovať, do **Touched Tables**.
3. Stlačte tlačítko **Touch**.

Popis pregenerovania nájdete v [podrobnom popise funkcie Upsize](#).

Transfer

Táto kapitola opisuje ako presunúť aktuálnu databázu na SQL serveri do inej databázy na tom istom alebo inom SQL serveri za predpokladu, že oba servery sú spojené cez LAN alebo WAN. Ďalej sa dozviete ako transferovať aktuálnu databázu SQL servera do cieľovej databázy na inom serveri pomocou prenosných dátových zariadení.

Online export... - priamy prenos databázy (prípadne jej časti)

Off-line export... - nepriamy prenos databázy (prípadne jej časti) na disk

Off-line Import... - prenos databázy z disku do databázy na SQL serveri

PatrikDatabase... - prenos vrstvy údajov uloženej v databáze patrik.

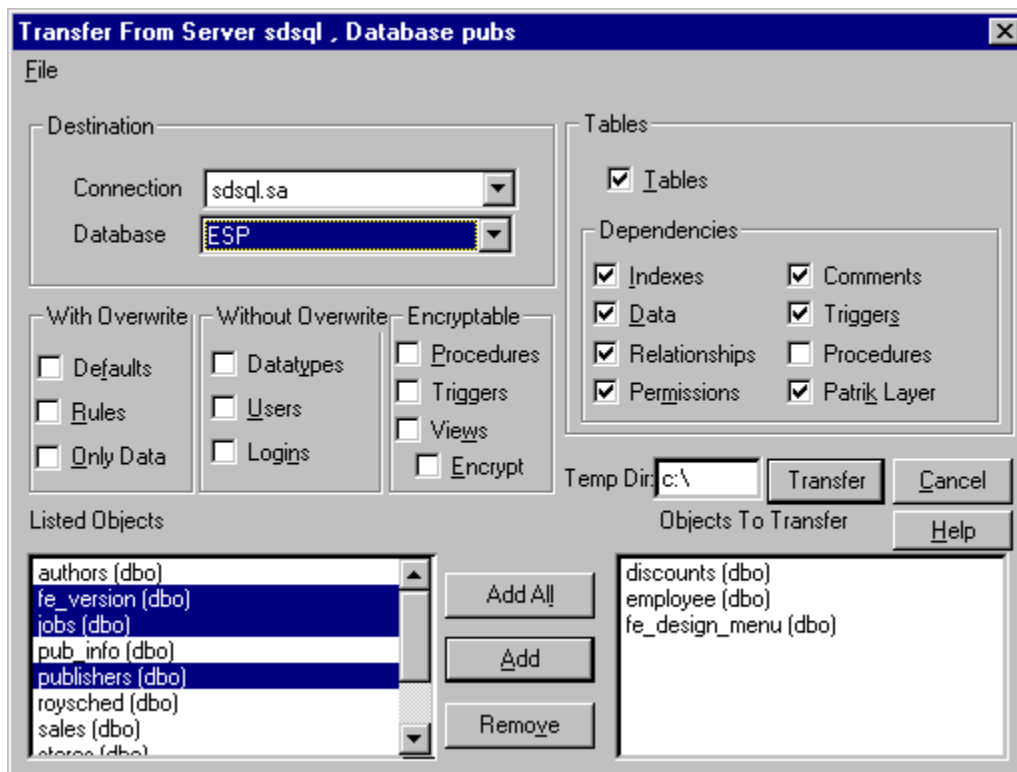
Príkaz Online export

Tento príkaz slúži na export niektorých alebo všetkých objektov aktuálnej databázy do cieľovej databázy, ktorá sa môže nachádzať na inom SQL serveri, za predpokladu, že oba servery sú pripojené cez LAN alebo WAN.

Online transfer databázy

1. Zvoľte databázu, ktorú chcete transferovať.
2. Zvoľte príkaz **Transfer** v menu **Current Database**.
3. Vyberte príkaz **Online Export**.

Zobrazí sa dialógové okno **Transfer From Server** *názov_server*, **Database** *názov_databázy*.



4. Skupina **Destination**
 - a) V combo boxe **Connection** vyberte názov cieľového servera, do ktorého budete aktuálnu databázu transferovať.
 - b) V combo boxe **Database** vyberte názov cieľovej databázy, do ktorej budete aktuálnu databázu transferovať.
5. Skupina **With Overwrite**
(Ak už zvolené objekty v cieľovej databáze existujú, PATRIK nahradí ich obsah)

objektami z aktuálnej databázy.)

Opakujte tieto kroky s každou položkou skupiny **With Overwrite**:

Vyberte check box objektov, ktoré chcete transferovať.

V poli **Listed Objects** sa zobrazí zoznam všetkých objektov nastaveného typu.

V zozname položiek **Listed Objects** označte objekt určený na transfer a stlačte tlačítko **Add**.

Pri transfere všetkých objektov stlačte tlačítko **Add All**.

Aby sa zabránilo transferu objektu, ktorý už bol pridaný do položky **Objects To Transfer**, vyberte objekt a stlačte tlačítko **Remove**.

6. Skupina **Without Overwrite**

(Ak sa vybrané objekty už v cieľovej databáze nachádzajú, nebudú nahradené objektami z aktuálnej databázy.)

Opakujte tieto kroky s každou položkou skupiny **Without Overwrite**:

Vyberte si check box objektov, ktoré chcete transferovať.

Zobrazí sa pole **Listed Objects** obsahujúce vybrané objekty..

V zozname položiek **Listed Objects** označte objekty určené na transfer a stlačte tlačítko **Add**.

Pri transfere všetkých objektov stlačte tlačítko **Add All**.

Aby sa zabránilo transferu objektu, ktorý už bol pridaný do položky **Objects To Transfer**, vyberte objekt a stlačte tlačítko **Remove**.

7. Skupina **Encryptable**

(Zvolením check boxu **Encrypt** sa kódujú zdrojové kódy **Procedúr, Triggrov, View-ov**, aby sa zabránilo zneužitiu princípov v zdrojovom kóde objektov. Ak sa nevyberie check box **Encrypt**, zdrojové kódy **Procedúr, Triggrov, View-ov** budú transferované bez zakódovania.)

Opakujte tieto kroky s každou položkou skupiny **Encryptable**:

Vyberte check box objektov určených na transfer.

V poli **Listed Objects** sa zobrazí zoznam všetkých objektov nastaveného typu.

V zozname objektov položky **Listed Objects** vyberte objekt, ktorý chcete transferovať a stlačte tlačítko **Add**.

Pri prenose všetkých objektov stlačte tlačítko **Add All**.

Aby sa zabránilo transferu objektu, ktorý už bol pridaný do položky **Objects To Transfer**, vyberte objekt a stlačte tlačítko **Remove**.

8. Skupina **Tables**

Vyberte check box **Tables**.

V poli **Listed Objects** sa zobrazí zoznam tabuliek aktuálnej databázy.

V zozname položiek **Listed Objects** označte tabuľky určené na transfer a stlačte tlačítko **Add**.

Pri transfere všetkých tabuliek stlačte tlačítko **Add All**.

Aby sa zabránilo transferu tabuliek, ktoré už boli pridané do položky **Objects To Transfer**, vyberte tabuľku a stlačte tlačítko **Remove**.

9. Skupina **Dependencies**

Vyberte check boxy vlastností, ktoré sa majú preniesť spolu s prenesenými tabuľkami.

Indexes - indexy tabuliek

Data - údaje tabuliek

Relationships - relačné vzťahy medzi tabuľkami (ak je vybraný check box **Patrik Layer** budú transferované aj atribúty relačných vzťahov a polohy tabuliek v relačnom modeli)

Permissions - práva užívateľov na určité tabuľky (pozri poznámku nižšie)

Comments - komentáre k tabuľkám a atribútom

Triggers - triggery


Procedures - procedúry

Patrik Layer - história tabuliek

V poli **Temp Dir.** zadajte adresár na svojom počítači, do ktorého sa počas transferu dočasne uložia údaje tabuliek. Dočasný adresár musí byť dostatočne veľký, aby bolo možné uložiť údaje najväčších tabuliek.

10. V poli **Temp Dir.** zadajte adresár na svojom počítači, do ktorého sa počas transferu dočasne uložia údaje tabuliek. Dočasný adresár musí byť dostatočne veľký, aby bolo možné uložiť údaje najväčších tabuliek.

11. Keď ste ukončili výber, stlačte tlačítko **Transfer.**

 Note	Aby sa transferovali aj práva užívateľov, je nutné zvoliť a prenos užívateľov (Users), ak prenášaný užívateľia už v cieľovej databáze existujú, tak sa ich profily neprenesú. Ak check box Users nie je vybraný, PATRIK zobrazí chybovú správu a bude pokračovať v transfere databázy.
--	--

Menu **File** dialógového okna.

Príkaz OFF-line export

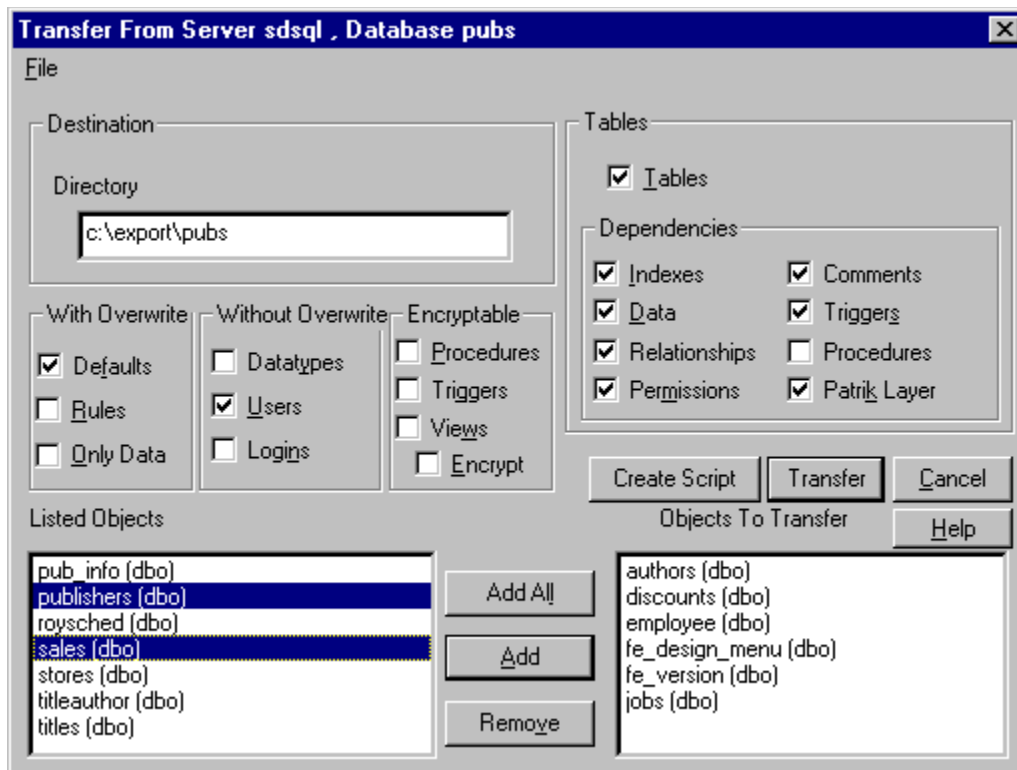
Tento príkaz je určený na exportovanie niektorých alebo všetkých objektov aktuálnej databázy na SQL serveri do cieľovej databázy na inom SQL serveri, ktoré nie sú pripojené ani cez LAN, ani cez WAN. Transfer sa zabezpečuje pomocou prenosných dátových zariadení.

Funkcia umožňuje aj vytvorenie scriptu štruktúry vybranej časti databázy.

Exportovanie niektorých alebo všetkých objektov offline

1. Označíte databázu určenú na transfer.
2. Zvoľte príkaz **Transfer** v menu **Current Database**.
3. Zvoľte príkaz **Off-line Export**.

Zobrazí sa dialógové okno **Transfer From Server** *názov_serveru*, Database *názov_databázy*.



1. Pod **Destination** v poli **Directory** zadajte miesto, kde bude databáza dočasne uložená. S tohoto adresára (alebo nejakej jeho kópie) je možné prenášať časť databázy rekonštruovať do nejakej databázy na nejakom SQL serveri funkciou **Offline Import**.
5. Skupina **With Overwrite**
(Ak vybrané objekty už v cieľovej databáze existujú, PATRIK nahradí ich obsah objektami z aktuálnej databázy.)

- Opakujte tieto kroky pri každej položke skupiny **With Overwrite**:
Vyberte check box objektov určených na transfer.
V poli **Listed Objects** sa zobrazí zoznam všetkých objektov nastaveného typu.
V zozname po³/₄a **Listed Objects** označte objekty, ktoré chcete transferovať a stlačte tlačítko **Add**.
Pri transfere všetkých objektov stlačte tlačítko **Add All**.
Aby sa zabránilo transferu objektu, ktorý už bol pridaný do po³/₄a **Objects To Transfer**, vyberte objekt a stlačte tlačítko **Remove**.
6. Skupina **Without Overwrite**
(Ak v cieľovej databáze vybrané objekty už existujú, nebudú nahradené objektami aktuálnej databázy.)
Opakujte tieto kroky pri každej položke skupiny **Without Overwrite**:
Vyberte check box objektov určených na transfer.
V poli **Listed Objects** sa zobrazí zoznam všetkých objektov nastaveného typu.
Z po³/₄a **Listed Objects** vyberte objekty určené na transfer a stlačte tlačítko **Add**.
Pri prenose všetkých objektov stlačte tlačítko **Add All**.
Aby sa zabránilo transferu objektu, ktorý už bol pridaný do po³/₄a **Objects To Transfer**, vyberte objekt a stlačte tlačítko **Remove**.
7. Skupina **Encryptable**
(Je možné zakódovať zdrojové kódy **Procedúr, Triggrov, View-ov**, ak vyberiete check box **Encrypt**, aby sa zabránilo zneužitiu princípov v zdrojovom kóde objektov. Ak check box **Encrypt** nie je vybraný, zdrojový kód **Procedúr, Triggrov, View-ov** bude transferovaný bez zakódovania.)
Opakujte tieto kroky pri každej položke skupiny **Encryptable**:
Vyberte check box objektov určených na transfer.
V poli **Listed Objects** sa zobrazí zoznam všetkých objektov nastaveného typu.
Zo zoznamu po³/₄a **Listed Objects** vyberte objekt, ktorý chcete transferovať a stlačte tlačítko **Add**.
Pri transfere všetkých objektov stlačte tlačítko **Add All**.
Aby sa zabránilo transferu objektu, ktorý už bol pridaný do po³/₄a **Objects To Transfer**, označte objekt a stlačte tlačítko **Remove**.
8. Skupina **Tables**
Vyberte check box **Tables**.
V poli **Listed Objects** sa objavia tabučky aktuálnej databázy.
V zozname po³/₄a **Listed Objects** označte tabučky určené na transfer a stlačte tlačítko **Add**.
Pri transfere všetkých tabuliek kliknite na tlačítko **Add All**.
Aby sa zabránilo transferu tabuliek, ktoré už sú pridané do po³/₄a **Objects To Transfer**, tabučky označte a stlačte tlačítko **Remove**.
9. Skupina **Dependencies**
Vyberte check boxy vlastností, ktoré sa majú preniesť spolu s prenesenými tabučkami.
Indexes - indexy tabuliek

Data - údaje v tabuľkách

Relationships - relačné vzťahy medzi tabuľkami (ak je vybraný check box Patrik Layer, budú transferované atribúty relačných vzťahov a polohy tabuliek v relačnom modeli)

Permissions - práva užívateľov na určité tabuľky (pozri poznámku nižšie)

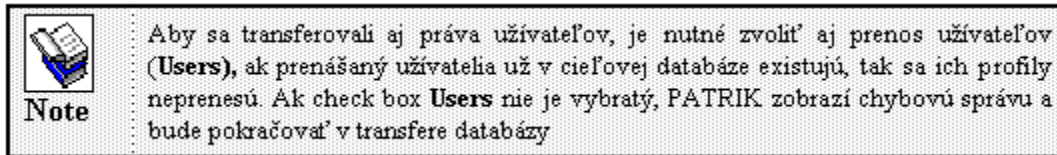
Comments - komentáre k tabuľkám a atribútom

Triggers - trigger vytvorené k tabuľkám

Procedures - procedúry odkazujúce na tabuľky

Patrik Layer - história tabuliek

10. Ak ste ukončili výber, stlačte tlačítko **Transfer** - ak chcete vykonať prenos na disk (aj s dátami - ak boli tieto nastavené na prenos). Ak Vám stačí vytvoriť SQL script prenášanej časti databázy, zvolte tlačítko **Create Script**. To ale musí byť v kolonke **Directory** uvedené meno súboru do ktorého sa má script uložiť (Nie teda meno adresára).



Menu **File** dialógového okna.

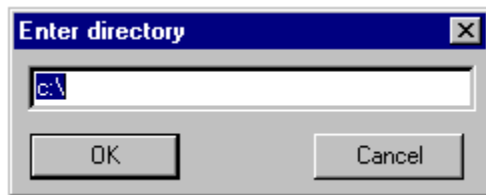
Príkaz OFF-line Import

Tento príkaz je určený na importovanie databázových objektov do aktuálnej databázy zo zdrojovej databázy na inom SQL serveri, ktorý nie je pripojený cez LAN, ani cez WAN. Offline transfer sa zabezpečuje pomocou prenosných dátových zariadení.

Importovanie objektov databázy z adresára na disku.

1. Zvoľte príkaz **Transfer** v menu **Current Database**.
2. Zvoľte príkaz **Off-line Import**.

Zobrazí sa dialógové okno **Enter Directory**.



3. Napíšte adresár zdrojovej databázy, z ktorej sa budú transferovať objekty do aktuálnej databázy.
1. Stlačte tlačítko **OK**.

Transfer databázy Patrik

Tento príkaz je určený na dodatočný transfer vrstvy, ktorú PATRIK pridáva ku každej spravovanej databáze. Ak sa databáza transferovala pomocou inej aplikácie, bola transferovaná bez vrstvy v databáze **patrik**, ktorá obsahuje komentáre k tabuľkám a atribútom, polohy okien v relačnom modeli, vlastnosti relačných vzťahov, informácie o histórii existujúcich tabuliek a relačných vzťahov view-ov.

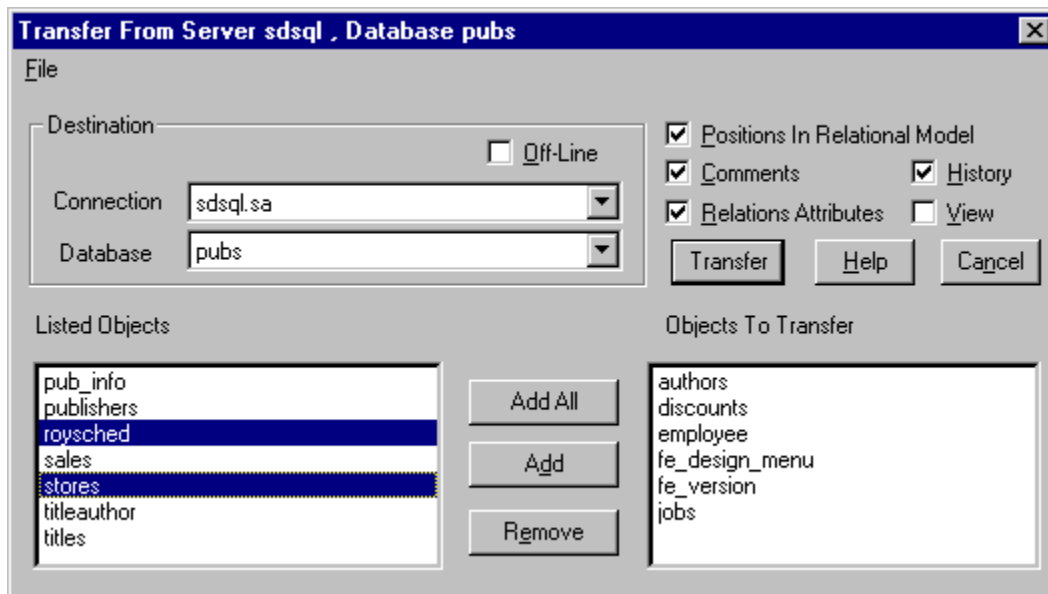
Tu sa nachádza popis štruktúry databázy **patrik**.

Databázu **patrik** je možné transferovať online aj offline.

Transfer databázy patrik online

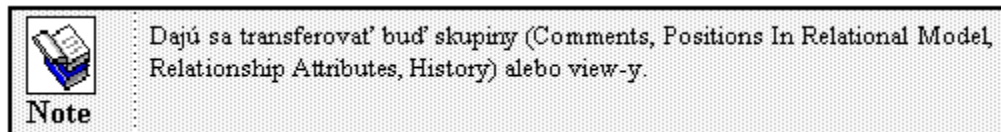
1. Zvoľte príkaz **Transfer** v menu **Current Database**.
2. Vyberte príkaz **Patrik Database**.

Zobrazí sa dialógové okno **Transfer From Server** *názov_servera*, **Database** *názov_databázy*.



3. Skupina **Destination**
 - a) combo boxe **Connection** vyberte názov cieľového servera, do ktorého bude databáza **patrik** transferovaná.
 - b) combo boxe **Database** vyberte meno cieľovej databázy, do ktorej bude databáza **patrik** transferovaná.
4. Vyberte skupiny (**Comments**, **Positions In Relational Model**, **Relationship Attributes**, **History**) alebo **View-y**, ktoré chcete transferovať (pozri poznámku nižšie).

- V zozname po¾a **Listed Objects** vyberte tabu¾ky (v skupine **Views** - mená urøených view-ov) urøené na transfer a stlaète tlačítko **Add**.
 Pri transfere všetkých objektov stlaète tlačítko **Add All**.
 Aby sa zabránilo transferu objektu, ktorý už bol pridaný do po¾a **Objects To Transfer**, objekt oznaète a stlaète tlačítko **Remove**.
 Stlaète tlačítko **Transfer**.



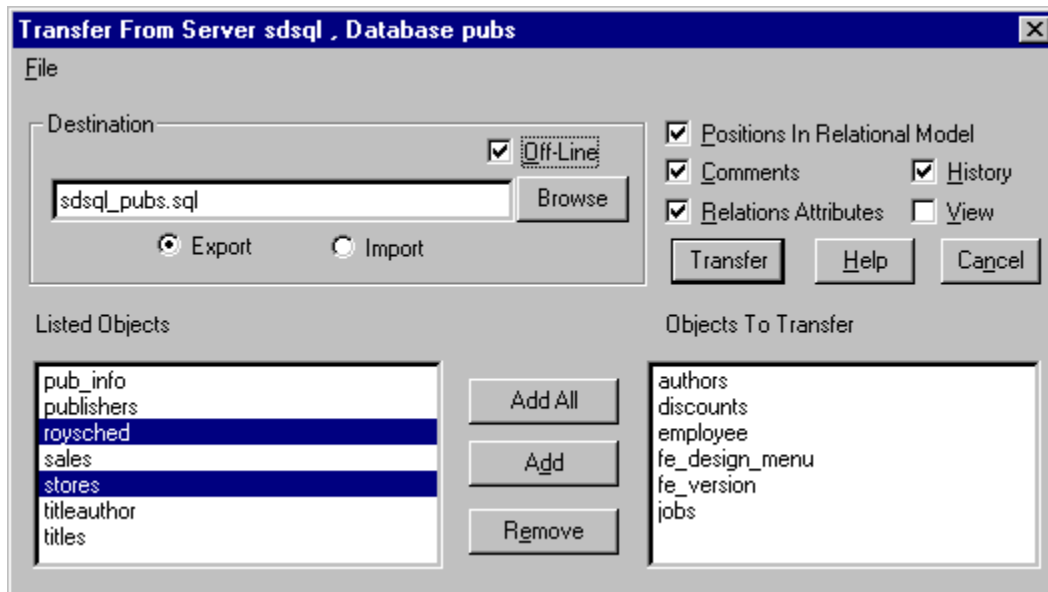
Export databázy patrik offline

- Zvo¾te príkaz **Transfer** v menu **Current Database**.
- Vyberte príkaz **Patrik Database**.

Zobrazí sa dialógové okno **Transfer From Server názov_servera, Database názov_databázy**.

- Skupina **Destination**
 Vyberte check box offline.

Zmení sa podokno.



- Pod **Destination**
 Vyberte rádio tlačítko **Export**.
 Stlaète tlačítko **Browse** a urèite miesto, kde bude databáza doèasne uložená.
- Vyberte skupiny (**Comments**, **Positions In Relational Model**, **Relationship Attributes**,

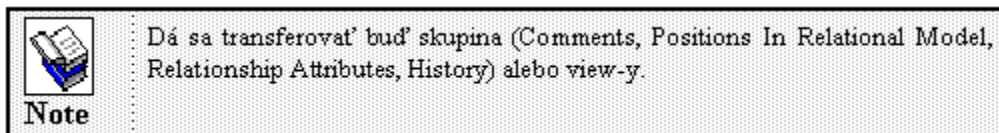
History) alebo **View-y**, ktoré chcete transferovať (pozri poznámku nižšie).

6. Zo zoznamu podľa **Listed Objects** vyberte tabuľky (v skupine **Views** - mená určených view-ov) určené na transfer a stlačte tlačidlo **Add**.

Pri prenose všetkých tabuliek (v skupine **Views** - určených view-ov) stlačte tlačidlo **Add All**.

Aby sa zabránilo prenosu tabuľky (v skupine **Views** - určenému view-u), ktorá už bola pridaná do podľa **Objects To Transfer**, tabuľku označte a stlačte tlačidlo **Remove**.

7. Stlačte tlačidlo **Transfer**.



Import databázy patrík offline

1. Stlačte tlačidlo **Transfer** v menu **Current Database**.

2. Vyberte príkaz **Patrík Database**.

Zobrazí sa dialógové okno **Transfer From Server** *názov_server*, Database *názov_databázy*.

3. Skupina **Destination**

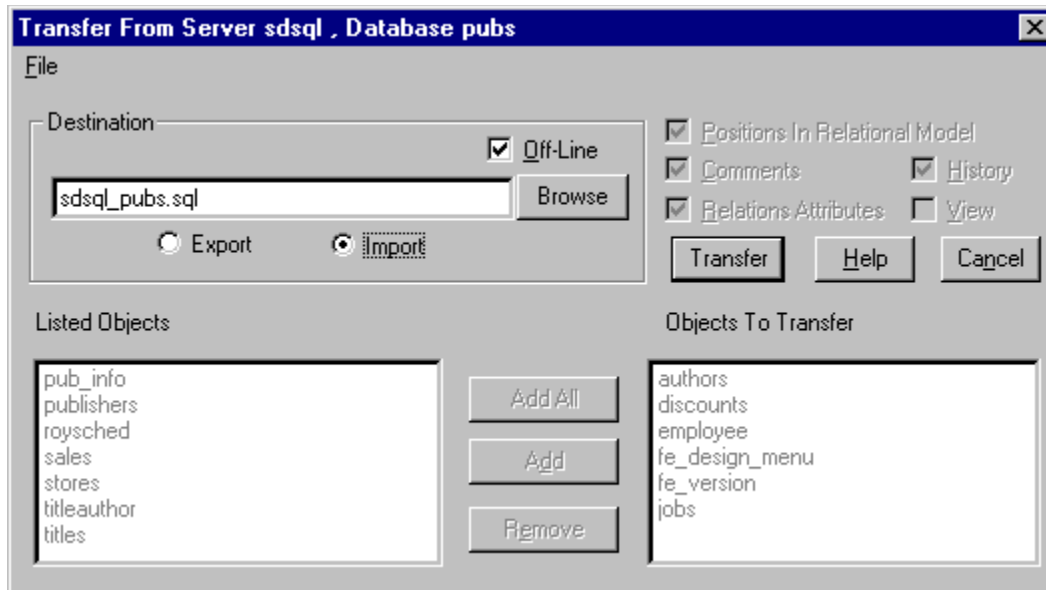
Vyberte check box **Offline**.

Zmení sa podokno.

4. Pod **Destination**

Vyberte radio tlačidlo **Import**.

Zobrazí sa dialógové okno **Transfer From Server** *názov_server*, Database *názov_databázy*.



5. Stlaète tlačítko **Browse**.
Zobrazí sa dialógové okno **Open**.
6. Vyberte databázu určenú na import.
7. Stlaète tlačítko **Transfer**.

Menu **File** dialógového okna.

Menu File dialógového okna

Nastavenie, ktoré ste vykonali v dialógovom okne môžete **uložiť** na disk a neskôr z disku nahráť. Ušetrí to duplicitné vypáďanie koloniiek dialógového okna pri opakovanom používaní rovnakého nastavenia okna.

Uloženie nastavenia dialógového okna, ktoré ste naďefinovali

1. Z menu **File** v dialógovom okne vyberte príkaz **Save Setting**.
Zobrazí sa dialógové okno **Save As**.
2. Napíšte názov súboru, zvoľte diskovú jednotku a adresár, do ktorého chcete súbor uložiť.
3. Stlačte tlačítko **OK**.

Obnovenie nastavenia, ktoré ste uložili predtým

1. Z menu **File** v dialógovom okne zvoľte príkaz **Load Settings**.
Zobrazí sa dialógové okno **Open**.
2. Vyberte súbor, ktorý obsahuje nastavenia.
3. Stlačte tlačítko **OK**.

Zahrnúť zjednotenia niekoľkých nastavení do dialógového okna

1. Z menu **File** v dialógovom okne zvoľte príkaz **Include Settings**.
Zobrazí sa dialógové okno **Open**.
2. Vyberte súbor, ktorý chcete priradiť k nastaveniu dialógového okna.
3. Stlačte tlačítko **OK**.

Opakujte kroky 1 až 3, kým nevyberiete všetky nastavenia, ktoré chcete priradiť do dialógového okna.

Porovnávanie

V tejto kapitole sa dozviete, ako sa zisťujú rozdiely medzi dvomi databázami a ako synchronizovať údaje a štruktúry dvoch databáz. Používaním príkazov v menu **Compare** môžete porovnávať štruktúru kódových objektov, tabuľkových štruktúr, históriu a údaje v tabuľkách.

Compare - With Another Database... - porovnanie štruktúry dvoch databáz

Compare - Structure Of Selected Tables... - porovnanie štruktúry vybraných tabuliek dvojice databáz

Compare - Source Codes... - porovnanie zdrojových kódov procedúr, triggrov, ...

Compare - Data In Tables... - porovnanie dát tabuliek a ich prípadná synchronizácia

Compare - Permissions... - porovnanie práv užívateľov na jednotlivé objekty

Príkaz Compare with another database

Príkaz **Compare With Another Database** slúži na porovnanie štruktúry databázy. Neporovnáva však obsahy (dáta) tabuliek ani zdrojové kódy procedúr, triggrov, defaultov a rule-ov. Tieto funkcie sa dajú realizovať pomocou :

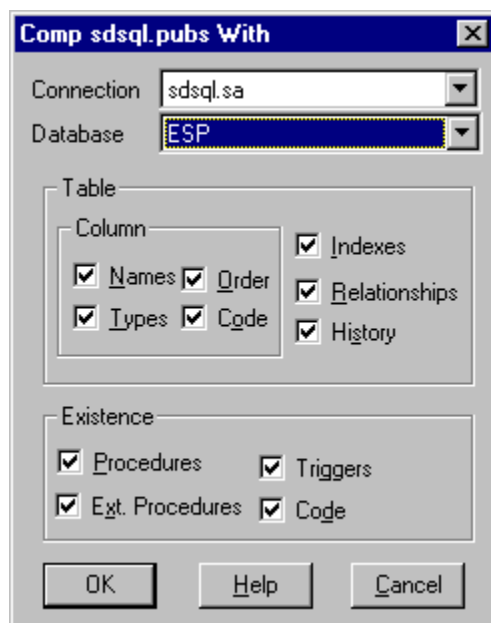
Compare - Data In Tables...

Compare - Source Codes...

Porovnanie štruktúry aktuálnej databázy s inou databázou

1. V combo boxe **Database** vyberte databázu, ktorú chcete porovnať s inou databázou.
2. Otvorte menu **Compare** v menu **Current Database**.
3. Zvoľte príkaz **With Another Database**.

Zobrazí sa dialógové okno *Comp názov_serveru.názov_databázy With*.



1. V combo boxoch **Connection** a **Database** vyberte server a databázu, s ktorou chcete porovnať aktuálnu databázu.
2. V skupine **Table** vyberte check boxy èat tabuliek urèených na porovnanie.
Indexes - zistenie rozdielov v indexoch tabuliek
Relationships - zistenie rozdielov v relaèných vzťahoch medzi tabuľkami
History - zistenie rozdielov v existencii histórie tabuliek
3. V skupine **Column** vyberte check boxy
Names - zistenie rozdielov v názvoch stĺpcov tabuľky

Types - zistenie rozdielov v dátových typoch stĺpcov tabuľky

Order - zistenie rozdielov v poradí stĺpcov tabuľky

Code - kontrola existencie pravidiel (rules) a defaultov tých istých stĺpcov tabuľky

4. V skupine **Existence** vyberte check boxy existencie databázových objektov, ktoré chcete porovnávať.
5. Po ukončení výberu stlačte tlačítko **OK**.

Okno **Query Results**, v ktorom sa zobrazia výsledky porovnávania je oknom typu Results.

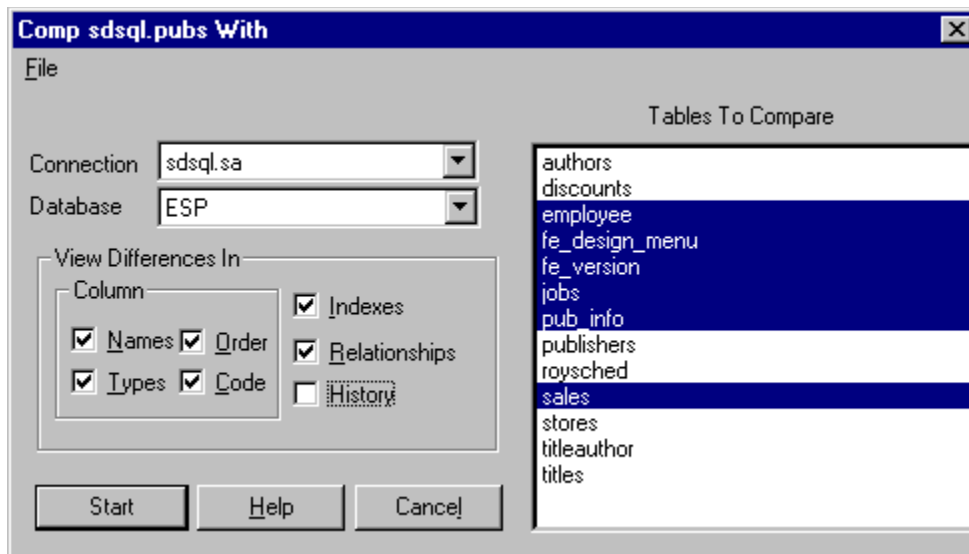
Príkaz Compare Structure Of Selected Tables

Príkaz **Compare Structure Of Selected Tables** slúži na porovnanie štruktúry vybraných tabuliek databázy s rovnomennými tabuľkami v inej databáze. Neporovnáva však obsahy (dáta) tabuliek. Tieto funkcie sa dajú realizovať pomocou funkcie Compare - Data In Tables.....

Porovnanie štruktúry vybraných tabuliek databázy s rovnomennými tabuľkami v inej databáze

1. V combo boxe **Database** vyberte databázu, ktorej tabuľky chcete porovnať s inou databázou.
2. Otvorte menu **Compare** v menu **Current Database**.
3. Zvoľte príkaz **Structure Of Selected Tables**.

Zobrazí sa dialógové okno



1. V combo boxoch **Connection** a **Database** vyberte server a databázu, s ktorej tabuľkami chcete porovnať vybrané tabuľky aktuálnej databázy.
2. V skupine **Table** vyberte check boxy èat tabuliek urèených na porovnanie.
Indexes - zistenie rozdielov v indexoch tabuliek
Relationships - zistenie rozdielov v relaèných vzťahoch medzi tabuľkami
History - zistenie rozdielov v existencii histórie tabuliek
3. V skupine **Column** vyberte check boxy
Names - zistenie rozdielov v názvoch stĺpcov tabuľky
Types - zistenie rozdielov v dátových typoch stĺpcov tabuľky
Order - zistenie rozdielov v poradí stĺpcov tabuľky
Code - kontrola existencie pravidiel (rules) a defaultov tých istých stĺpcov tabuľky

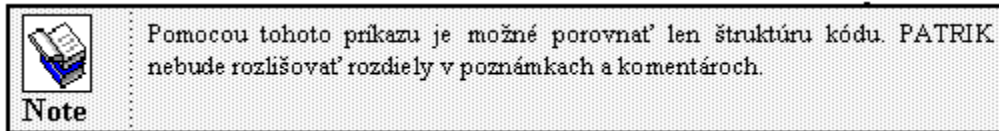
4. V listboxe **Tables To Compare** vyberte tabuľky, ktorých štruktúru chcete porovnávať s rovnomenými tabuľkami inej databázy.

5. Po ukončení výberu stlačte tlačítko **OK**.

Okno **Query Results**, v ktorom sa zobrazia výsledky porovnávania je oknom typu Results.
Menu File dialógového okna.

Príkaz Compare source codes

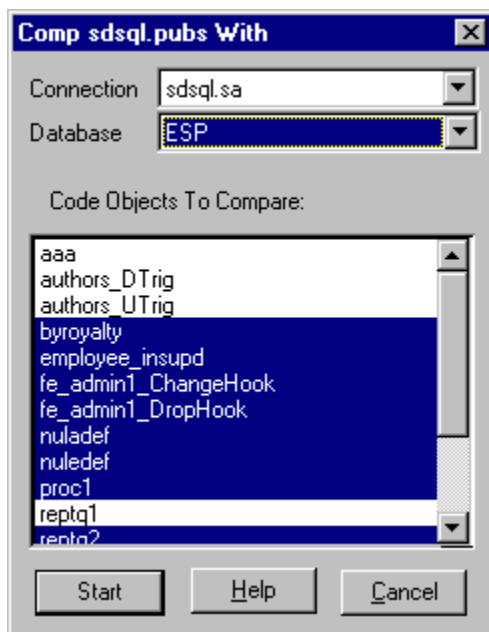
Funkcia **Compare Source Codes** slúži na porovnanie Transact SQL zdrojových kódov (procedúr) dvoch databáz.



Porovnávanie zdrojových kódov dvoch databáz

1. V combo boxoch **Connection** a **Database** vyberte server a databázu, ktorej zdrojové kódy chcete porovnávať so zdrojovými kódmi rovnomených objektov inej databázy.
2. Otvorte menu **Compare** v menu **Current Database**.

Zobrazí sa dialógové okno **Comp názov_server.názov_databázy With**.



1. V combo boxoch **Connection** a **Database** vyberte databázu, ktorej zdrojové kódy chcete porovnávať s aktuálnou databázou.
2. V zozname pod **The Code Objects To Compare** označte zdrojové objekty, ktoré chcete porovnávať.
3. Keď ste ukončili výber, stlačte tlačítko **Start**.

PATRIK zobrazí okno **Query Results**, kde sa zobrazí riadok, stápec a slovo prvého rozdielu

procedúry pre každú rozdielnu dvojicu procedúr. Táto funkcia zobrazí len prvý rozdiel a bude pokračovať v porovnávaní ďalšieho objektu.

Okno **Query Results** je oknom typu Results.

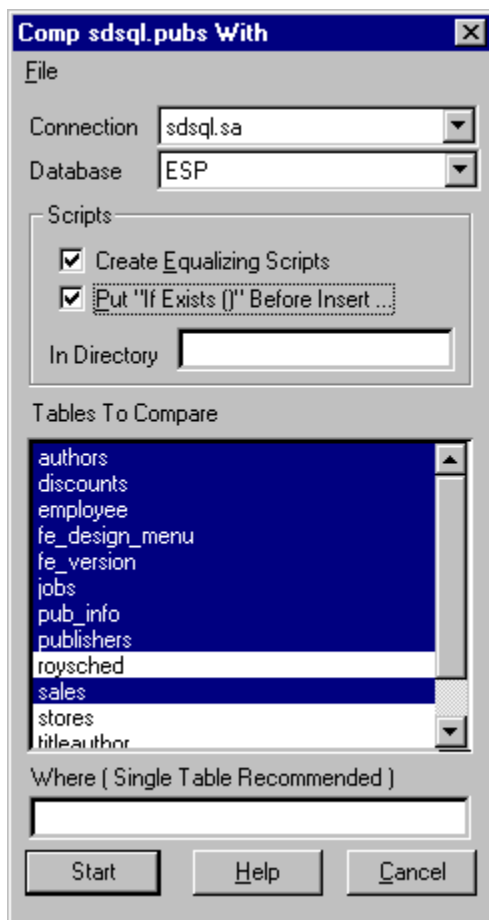
Príkaz Compare data in tables

Príkaz **Compare Data In Tables** slúži na porovnanie údajov v tabuľkách dvoch databáz.

Porovnávanie údajov v tabuľkách dvoch databáz

1. V combo boxoch **Connection** a **Database** vyberte server a databázu, ktorej údaje v tabuľkách chcete porovnať s údajmi rovnomenných tabuliek v inej databáze.
2. Otvorte menu **Compare** v menu **Current Database**.
3. Zvoľte príkaz **Data In Tables**

Zobrazí sa dialógové okno **Comp názov_server.názov_databázy With**.




1. V combo boxoch **Connection** a **Database** vyberte server a databázu, ktorej údaje v tabuľkách chcete porovnať s aktuálnou databázou.
2. V skupine **Scripts**
 - 1) k označíte check box **Create Equalizing Scripts**, PATRIK vytvorí 2 súbory , ktoré

sa môžu použiť neskôr pri synchronizácii údajov v tabuľkách porovnávaných databáz:

1. `server1_databáza1.sql` - obsahuje množinu príkazov Insert, Delete a Update. Po spustení tohoto súboru na serveri1 v databáze1 budú údaje porovnávaných tabuliek rovnaké ako údaje v tabuľkách na serveri2 v databáze2.
 2. `server2_databáza2.sql` - obsahuje množinu príkazov insert, delete a update. Po spustení tohoto súboru na serveri2 v databáze2 budú údaje porovnávaných tabuliek rovnaké ako údaje v tabuľkách na serveri1 v databáze1
- 2) v poli **In Directory** určíte adresár, do ktorého majú byť skripty uložené.
- 3) **Put "If Exists ()" Before Insert...** - ak je zaškrtnuté, tak sa v synchronizačných skriptoch bude pred každým príkazom INSERT vyskytovať podmienka zabraňujúca duplicitnému vloženiu záznamov do tabuliek pri prípadnom opakovanom spustení jedného synchronizačného skriptu viackrát.
3. V poli **Tables to Compare** vyberte tabuľky, ktoré chcete porovnávať.
4. V poli **Where (Single Table Recommended)** - zadajte podmienku na porovnanie tej časti tabuľky, ktorá obsahuje údaje vyhovujúce podmienke. Pretože podmienka **Where** platí pre každú vybranú tabuľku, odporúča sa porovnávať len jednu tabuľku.
5. Keď ste ukončili výber, stlačte tlačítko **Start**.

Po spustení query sa zobrazí okno **Query Results**, kde vidíte rozdiely medzi dátami porovnávaných tabuliek.

Okno **Query Results** je oknom typu Results.

	<p>Ak sa porovnáva viac ako jedna tabuľka, berie sa do úvahy referenčná integrita. Zmeny sú preto implementované postupnosťou, ktorá neporušuje referenčnú integritu.</p> <p>Porovnávané tabuľky musia mať definované PK.</p> <p>Ak niektorý atribút PK nie je integer, porovnávanie trvá približne 10 krát dlhšie.</p>
---	---

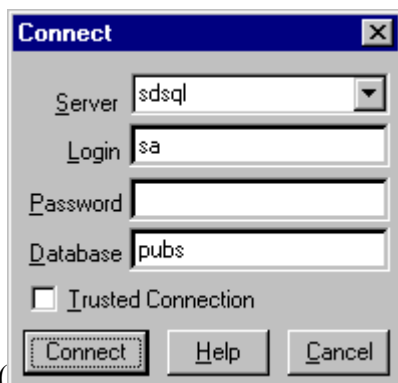
Synchronizácia údajov v tabuľkách



Menu **File** dialógového okna.

Synchronizácia údajov v tabuľkách

Synchronizácia údajov v dvoch databázach

1. Vyberte databázu, ktorej tabuľky chcete synchronizovať, ako aktuálnu databázu.



2. Stlačíte tlačítko **Query** () v hlavnom okne aplikácie. Zobrazí sa okno Query Editor.
3. Zvoľte príkaz **Load** zo Shortcut Menu. Zobrazí sa dialógové okno **Open**.
4. Označíte súbor, ktorý vytvorila funkcia porovnania dát.
Názov súboru, ktorý bude synchronizovať údaje v aktuálnej databáze s údajmi porovnáwanej databázy má nasledovnú syntax: *názov_serveru_názov_aktuálnej_databázy.sql*.
5. Stlačíte tlačítko **OK**.
Daný súbor sa nahrá do okna **Query Editor**.
6. Stlačíte tlačítko **Execute** () v okne **Query Editor**.

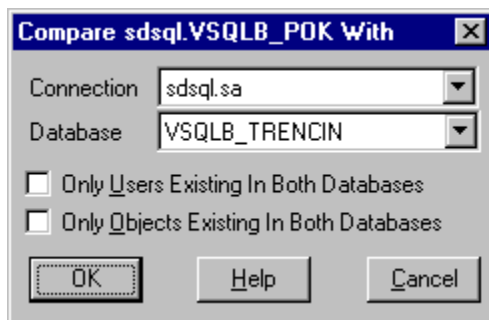
Porovnávanie práv (permissions)

Príkaz **Compare Permissions** slúži na porovnanie práv v dvoch databázach.

Porovnanie práv v dvoch databázach:

1. V combo boxoch **Connection** a **Database** vyberte server a databázu, ktorej práva chcete porovnávať s právami rovnomenných objektov v inej databáze.
2. Otvorte menu **Compare** v menu **Current Database**.
3. Zvoľte príkaz **Permissions**

Zobrazí sa dialógové okno **Compare meno_serveru.meno_databázy With**.



1. V combo boxoch **Connection** a **Database** vyberte server a databázu, ktorej práva chcete porovnať s právami v aktuálnej databáze.
2. Ak chcete, aby výstup porovnania neobsahoval hlásenia o právach užívateľov, ktorý existujú v len jednej s databáz, tak zaškrtnite check box **Only Users Existing In Both Databases**.
3. Ak chcete, aby výstup porovnania neobsahoval hlásenia o právach na objekty (tabuľky, view-y, procedúry), existujúce v len jednej s databáz, tak zaškrtnite check box **Only Objects Existing In Both Databases**.
4. Stlačte tlačítko **OK**.

PATRIK zobrazí okno **Query Results**, kde sa zobrazí riadok, pre každý rozdiel v tabuľkách práv v jednotlivých databázach.

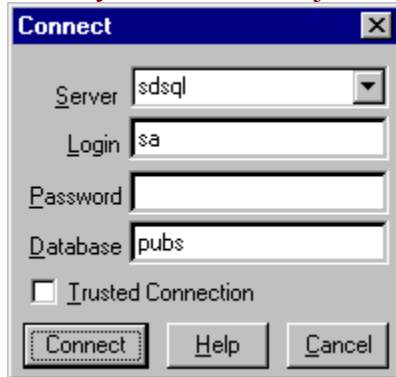
Relačný model

Táto kapitola opisuje možnosti grafického zobrazenia relačného modelu, editovanie relačných vzťahov priamo v okne **Relational Model**, definovanie view-ov, ktoré umožňujú zobrazovať len tie tabučky, ktoré Vás zaujímajú, hľadanie a prenášanie tabučky v okne **Relational Model** pomocou mapy relačných vzťahov a tlačenie relačných modely.

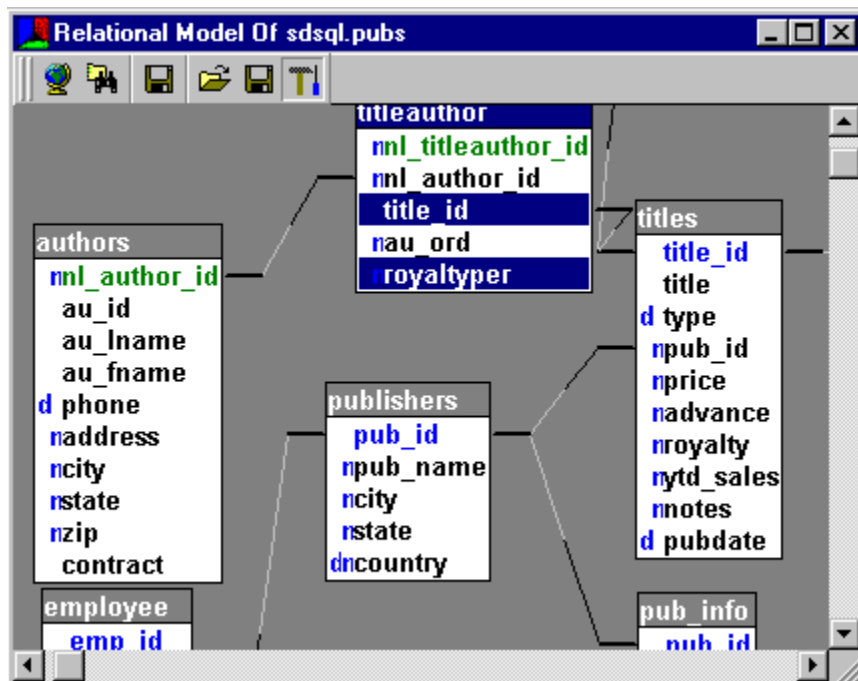
Okno relačného modelu

PATRIK umožňuje browsovať a editovať Relačný model (štruktúru databázy a relačné vzťahy medzi tabučkami). V okne **Relational Model** PATRIK zobrazí len tabučky s definovaným primárnym kľúčom, alebo tabučky s relačnými vzťahmi k tabučkám s definovaným primárnym kľúčom. PATRIK umožňuje vytvoriť, vytlačiť a editovať relačné vzťahy medzi tabučkami. Relačné vzťahy sú zobrazené v okne **Relational Model**.

Relačný model aktuálnej databázy sa zobrazí stlačením tlačítka **Relational Model...** (



) v toolbare aplikácie.



Tabuľky

PATRIK zobrazí tabuľky so stĺpcami aktuálnej databázy v podobe okien a relačné vzťahy spájajúce relačné tabuľky ako čiary. Okná s tabuľkami je možné podľa potreby posúvať. Rozmiestnenie tabuliek je možné ulžiť funkciou [Relations - Save Layout](#) v [shortcut menu](#)

Relácie

Primárna strana relácie je znázornená svetlejšou čiarou a foreign strana tmavšou čiarou. Na reláciu je možné kliknúť, čím sa vykreslí širšou čiarou. Potom shortcut menu bude bohatšie o nejaké funkcie, ktoré súvisia s editovaním relácie. Dvojkliknutím na reláciu sa priamo zobarí [Editor Relácií](#).

Stĺpce tabuliek

Stĺpce tabuľky sú zobrazené ako riadky okna s menom danej tabuľky. Stĺpce označujúce primárne kľúče tabuľky sú zobrazené




- modrou farbou - ak sa jedná o **Primary Key Constraint**
- zelenou farbou - ak sa jedná o obyčajný primárny kľúč (definovaný napríklad **systemovou procedúrou sp_primarykey**)
- tyrkysovou farbou - ak sa jedná o primárny kľúč oboch typov zároveň

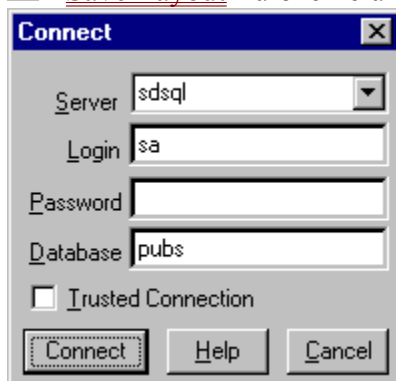
Pred jednotlivými stĺpcami tabuľky sa môžu vyskytovať modré písmenká signalizujúce :



- i - ak má stĺpec nastavený index
- d - ak má stĺpec nastavený default
- n - ak stĺpec môže obsahovať NULL hodnoty

Vypisovanie týchto písmienok (informácií) je možné nastaviť v [Options](#).

Toolbar v okne relačného modelu obsahuje funkcie

-  [Show Map](#) - zobrazí extrémnu zmenšeninu relačného modelu
-  [Find Table](#) - vyhľadávanie tabuliek v relačnom modeli
-  [Save Layout](#) - uloženie umiestnenia okien s tabuľkami do databázy



- [Load View](#) - načítanie pohľadu na relácie s databázy
-  [Save View](#) - uloženie pohľadu na relácie do databázy
-  [Edit Mode](#) - prepnutie módu editovania pohľadu na relačný model

Ďalšie funkcie sa nachádzajú v shortcut menu.

Príkaz Show map

Nakoľko databázy často obsahujú veľa tabuliek, nie je možné zobraziť ich na obrazovke všetky. Z toho dôvodu treba pohľad na relačný model pri browsovaní alebo editovaní posúvať pomocou scrollbarov okna relačného modelu. Pre lepšiu orientáciu v relačnom modeli PATRIK ponúka okno **Map Of Relational Model**. Tabuľky v tomto okne sa nachádzajú malé čierne štvorce, ktoré predstavujú tabuľky v okne **Relational Model** aktuálnej databázy.

Zobrazenie okna Map Of Relational Model

1. Zo shortcut menu **Relationship** v okne **Relational Model** zvolíte príkaz **Show Map**.



Tabuľky

Čierne obdĺžničky v okne mapy reprezentujú jednotlivé tabuľky relačného modelu. Kôli rýchlosti vykreslovania sa nevykresľujú relácie medzi tabuľkami. Ak zväčšíte okno mapy, tak sa proporčne zväčšia aj všetky tabuľky (čierne štvorčeky). Kliknutím na plochu nejakého obdĺžnika sa vypíše meno tabuľky, ktorú obdĺžnik reprezentuje do spodnej (**informačnej**) časti okna mapy. **Drag and drop** funkcionalitou je možné tabuľku v okne premiestniť.

Pohľad v okne relačného modelu.

Pozíciu na mape, ktorá je zobrazená v okne relácií reprezentuje biely obdĺžnik vykreslený v okne mapy. Dvojkliknutím na nejakom mieste v okne mapy presunie pohľad v okne relačného modelu na dané miesto.

V okne **Map Of Relational Model** je možné:

- prenášať skupinu tabuliek
- vytlačiť označené tabuľky

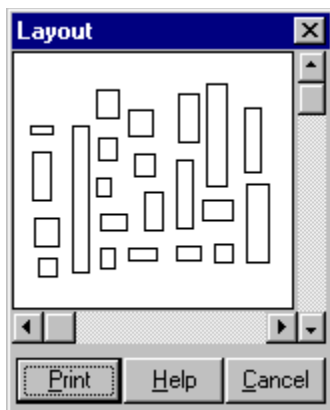
Preniesť skupinu tabuliek

1. Označíte tabučky, ktoré chcete preniesť bielym obdĺžnikom (pozri obrázok nižšie)
 - a) Umiestnite kurzor myši, na ktorékoľvek miesto v okne **Map Of Relational Model**.
 - b) Stlačíte ľavé tlačítko myši a ťahajte, kým všetky tabučky, ktoré potrebujete nie sú ohraničené bielym obdĺžnikom.
2. Keď pustíte ľavé tlačítko myši, zobrazí sa ponuka **Shortcut menu**.
3. Zvoľte príkaz **Move Area**.
4. Preneste štvorec na iné miesto.

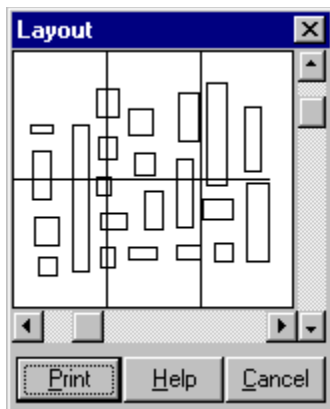
Vytlačenie skupiny tabuliek pomocou okna Map Of Relational Model

1. Vyberte tabučky, ktoré chcete vytlačiť bielym obdĺžnikom (pozri obrázok nižšie).
 - a) Umiestnite kurzor myši kdekoľvek v okne **Map Of Relational Model**.
 - b) Stlačíte ľavé tlačítko myši a ťahajte, kým všetky tabučky, ktoré chcete vytlačiť nie sú ohraničené bielym obdĺžnikom.
2. Keď pustíte ľavé tlačítko myši zobrazí sa **Shortcut menu**.
Zvoľte príkaz **Print Area**.

Zobrazí sa okno **Layout**, v ktorom môžete nastaviť formát tlače.




3. Posúvajte vertikálny rolovací pruh smerom dolu a horizontálny rolovací pruh doprava a nastavte formát tlače.



PATRIK vytlačí strany v nasledovnom poradí (toto je poradie pre 12 strán).

1	4	7	10
2	5	8	11
3	6	9	12

 Relaçný model databázy, ktorá obsahuje veľa tabuliek a relačných vzťahov medzi tabuľkami, môžete kvôli lepšej orientácii a prehľadnosti vytlačiť na niekoľkých stranách formátu A4 . Potom zlepiť tieto strany do jednej veľkej mapy a zavesiť na stenu.

Tip

Príkaz Save layout

Uloženie zmien v polohách tabuliek v Relaènom modeli

1. Zvoľte príkaz **Save Layout** v shortcut menu **Relationships** v okne **Relational Model**.

Pozície tabuliek sa uložia do databázy **patrik**

- . Dôsledkom toho bude rozmiestnenie okien relaèného modelu rovnaké pre všetkých užívateľov na všetkých pracovných staniach.
- Ak databázu prenesiete inak ako **transfer funkcionalitou Patrika**, tak sa v cieľovej databáze tabuľky nebudú nachádzať tak utriedené, ako v zdrojovej databáze. Dokonca budú absolútne neutriedené. Na dodatočný prenos pozícií okien v reláciách môžete použiť funkciu **Transfer - Patrik Database** .

Shortcut menu v okne relational model

Aby sa zobrazilo **Shortcut Menu** obsahujúce príkazy potrebné na prácu s relačným modelom, umiestnite kurzor myši na pracovnú plochu okna **Relational Model** a kliknite na pravé tlačítko myši.

Shortcut Menu v okne **Relational Model** SQL server (skutočnej) databázy obsahuje nasledujúce príkazy:

- **Relations**
 - **Show Map** - zobrazí aj zmenšenú mapu relačného modelu
 - **Save Layout** - uloží pozície okien v relačnom modeli do db.
- **View Relation...** - zobrazí definíciu označenej relácie. Funkcia je zaradená do menu, iba ak je označená nejaká relácia.
- **Find Table...** - vyhľadávanie tabuliek
- **Open Table** - otvorí okno s dátami tabuľky. Funkcia je zaradená do menu, iba ak je aktívne nejaké okno v relačnom modeli.
- **WHERE Open...** - otvorí okno s dátami tabuľky spájanými zadanú WHERE podmienku. Funkcia je zaradená do menu, iba ak je aktívne nejaké okno v relačnom modeli.
- **Select COUNT(*)** - vypíše počet riadkov aktívnej tabuľky. Funkcia je zaradená do menu, iba ak je aktívne nejaké okno v relačnom modeli.
- **Select** alebo **Deselect** - slúžia na editovanie pohľadu (Viewu). Voľba je aktívna iba ak je zapnutý **Edit Mode**.
- **Scale** - škálovanie okna relačného modelu
- **View** - sprosredkúva vstupnovýstupné operácie s pohľadmi.
 - **Load...**
 - **Save...**
 - **Delete...**

Ponuka **Shortcut menu** pracovnej (downsized) databázy v porovnaní s ponukou **Shortcut menu** SQL server skutočnej databázy je ukrátené o príkazy **Open table**, **WHERE open**, **Select COUNT**, lebo tieto pracujú s dátami, ktoré ovšem tabuľky pracovnej databázy neobsahujú. Menu je ale obohatené o príkazy umožňujúce modifikáciu relačného modelu databázy :

- **Príkaz Add log** - nastaví mapovanie (logovanie) zmien v dátach označenej tabuľky (históriu)
- **Príkaz Remove log** - zruší mapovanie (logovanie) zmien v dátach označenej tabuľky (históriu)
- **Edit Relation...** - zobrazí definíciu označenej relácie. Funkciou je možné reláciu predefinovať. Funkcia je zaradená do menu, iba ak je označená nejaká relácia.
- **Delete Relation** - zmaže označenú reláciu. Funkcia je zaradená do menu, iba ak je označená nejaká relácia.
- **Add Relation...** - pridá reláciu One to Many označenej tabuľky.

Príkaz Add Relation

1. Zobrazí dialógové okno obsahujúce combobox obsahujúci všetky tabučky, ktoré sa nachádzajú v relaènom modeli.



1. vyberte tabučku, ktorá sa bude nachádzať na **foreign strane** relácie
1. Zobrazí sa dialógové okno Edit Relation, kde sa nastaví ostatné podrobnosti relácie.

Reláciu je však možné pridať aj takto:

1. V okne Relational Model umiestnite kurzor myši na atribút primárneho kľúča tabučky na strane "one" relaèného vzťahu "one to many".
2. Stlačte ľavé tlačítko myši a ťahajte atribút primárneho kľúča k budúcu foreign atribútu v tabučke, s ktorou chcete vytvoriť relačný vzťah.
3. Zobrazí sa dialógové okno Edit Relation, kde sa nastaví ostatné podrobnosti relácie.

Príkaz Add log

Ak je potrebná história zmien tabuliek, je možné vytvoriť tabuľku histórie, v ktorej sa zaznamená každá zmena danej tabuľky.

Zapísanie histórie zmien do tabuľky

1. Umiestnite kurzor myši do tabuľky v okne **Relational Model** a kliknite na pravé tlačítko myši.
2. Zvoľte príkaz **Add Log**.

PATRIK doplní pred názov tabuľky symbol *.

Po spustení upsize-u pracovnej (downsized) databázy do reálnej databázy na SQL serveri, PATRIK pridá tabuľku histórie k skupine **Data** objektov v okne **Browser Of názov_serveru.názov_databázy**.

Prezeranie histórie zmien urobených v tabuľke databázy

1. Stlačte tlačítko **Browser** v hlavnom okne aplikácie. Zobrazí sa okno **Browser Of názov_serveru.názov_databázy**.
 2. Vyberte tabuľku histórie, ktorú chcete vidieť.
 3. Stlačte tlačítko **Data**.
- Názov tabuľky histórie v okne **Browser Of názov_serveru.názov_databázy** má nasledovnú syntax: `_názov_tabuľky`

id	n1_perukci_id	n1_znovecie_id	cimnov	col	data	d_system	user
int	int N	int N	tinyint N	tinyint N	char(255) N	datetime N	int N datetime N
1	11445	7805	1	3	PSN27	1996-08-18 08:34:19	1 9-18-1996 8:34:19
2	11446	7806	1	2	PSN27	1996-08-18 08:34:19	1 9-18-1996 8:34:19
3	11463	97	1	4	11463	1996-08-18 08:34:20	1 9-18-1996 8:34:20
4	11462	97	1	3	údaj	1996-08-18 08:34:20	1 9-18-1996 8:34:20
5		97	1	3		1996-08-18 08:34:20	1 9-18-1996 8:34:20
6		97	1	3		1996-08-18 08:34:20	1 9-18-1996 8:34:20
7		97	1	3		1996-08-18 08:34:20	1 9-18-1996 8:34:20
8		96	1	3		1996-08-18 08:34:20	1 9-18-1996 8:34:20
9		96	1	3		1996-08-18 08:34:20	1 9-18-1996 8:34:20
10		96	1	3		1996-08-18 08:34:20	1 9-18-1996 8:34:20
11		96	1	3		1996-08-18 08:34:20	1 9-18-1996 8:34:20

Príkaz Remove log

Ak umiestniete kurzor myši na tabuľku histórie, ktorá má pred názvom *, objaví sa v ponuke **Shortcut menu** príkaz **Remove Log** namiesto príkazu **Add Log**. Príkaz **Remove Log** slúži na:

1. zrušenie zapisovania histórie zmien dát pre danú tabuľku
2. dropnutie tabuľky obsahujúcej históriu zmien pre danú tabuľku (Táto tabuľka má rovnaký názov ako tabuľka s dátami, no má znak “_” ako predpnu)

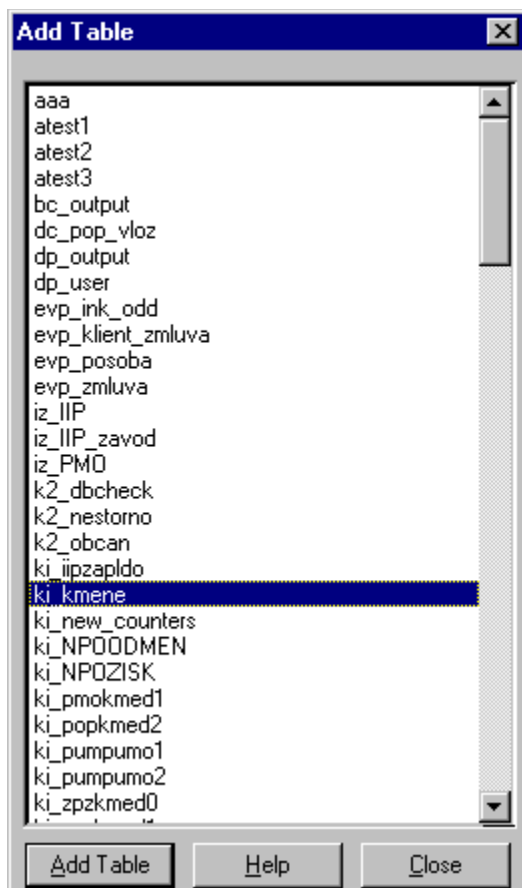
Príkaz Add table

V okne **Relational Model** sú zobrazené len tabučky s definovaným primárnym kľúčom, alebo tabučky s relačnými vzťahmi k inej tabučke. Tabučky bez primárneho kľúča a bez relačných vzťahov nie sú v okne **Relational Model** zobrazené, ale je možné ich doplniť pomocou príkazu **Add Table**.

Doplnenie tabučky do okna Relational Model

1. Umiestnite kurzor myši do akejkoľvek tabučky v okne **Relational Model** a kliknite na pravé tlačítko myši.
2. Zvoľte príkaz **Add Table**.

Zobrazí sa dialógové okno **Add Table** obsahujúce tabučky, ktoré sa nenachádzajú v okne relačného modelu.



3. Označíte tabučku, ktorú chcete pridať do relačného modelu.
4. Stlačte tlačítko **Add Table**.

PATRIK pridá označenú tabuľku do okna **Relational Model**. Priamo v okne **Relational Model** je možné určiť relačné vzťahy tabuliek. Viac informácií nájdete pod heslom [Editovanie relačných vzťahov medzi tabuľkami](#).

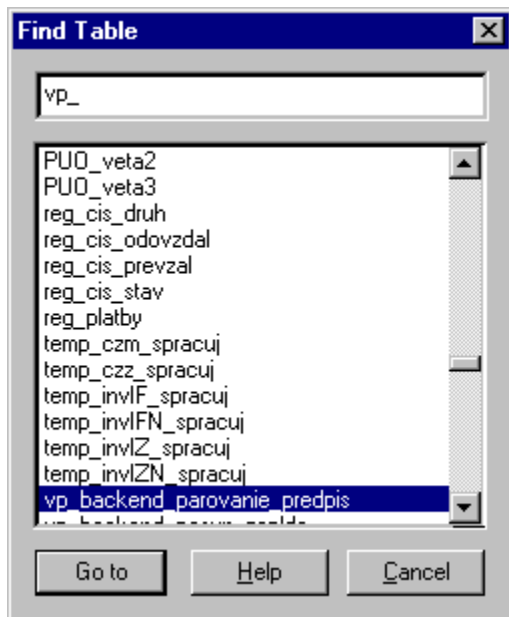
Príkaz Find table

Týmto príkazom je možné veľmi rýchlo nájsť v okne **Relational Model** tabuľku, ktorú potrebujete.

Vyhľadanie tabuľky v okne Relationship Model

1. Umiestnite kurzor myši do akejkoľvek tabuľky v okne **Relational Model** a kliknite na pravé tlačítko myši.
2. Zvoľte príkaz **Find Table** z ponuky **Shortcut Menu**.

Zobrazí sa dialógové okno **Find Table**.



3. Urobte nasledovné:
 - a) Do prázdneho poľa v dialógovom okne **Find Table** napíšte názov tabuľky, ktorú hľadáte.
 - b) V poli so zoznamom tabuliek označte potrebnú tabuľku.
4. Stlačte tlačítko **Go To**.


PATRIK označí hľadanú tabuľku a pohľad v okne relačného modelu nastaví tak, aby sa hľadaná tabuľka nachádzala v strede.

Príkaz Open table

Tento príkaz je určený na otvorenie celej hľadanej tabučky alebo jej časti (čo závisí od nastavenia v poli **Memory Usage (KB)** dialógového okna **Options** a veľkosti tabučky). Ak hodnota podľa **Memory Usage (KB)** v dialógovom okne **Options** je nižšia ako je veľkosť tabučky, PATRIK otvorí len časť tabučky.

1. Umiestnite kurzor myši do tabučky v okne **Relational Model** a kliknite na pravé tlačítko myši.
2. Zvoďte príkaz **Open Table** z ponuky **Shortcut menu**.
1. Patrik zobrazí dáta tabučky v novootvorenom okne.

Príkaz Select count



Tip

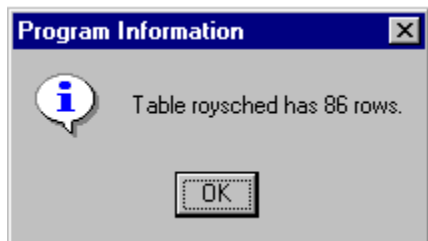
Príkaz pre situáciu, keď sa otvára tabuľka obsahujúca veľké množstvo údajov. Nahratie celej tabuľky môže trvať niekoľko minút, čím sa stráca čas. Aby sa predišlo takej situácii, treba najskôr pomocou príkazu **Select Count(*)** zistiť veľkosť tabuľky. Ak je tabuľka príliš veľká, je možné definovať podmienku príkazom **Open Where** a určiť, ktoré údaje chceme browsovať alebo editovať.

Príkazom **Select Count(*)** sa počet záznamov vybratej tabuľky.

Zistenie veľkosti vybratej tabuľky

1. Umiestnite kurzor myši do tabuľky v okne **Relational Model** a kliknite na pravé tlačítko myši.
2. Zvoľte príkaz **Select Count(*)** z ponuky **Shortcut menu**.

Zobrazí sa dialógové okno **Program Information**.

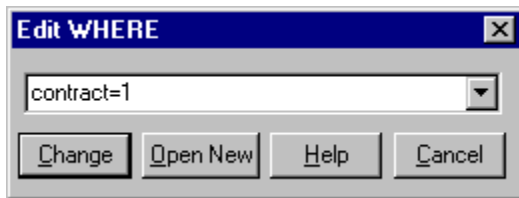


Príkaz Where open

Zobrazenie údajov tabučky pod podmienkou, ktorú ste definovali.

1. Umiestnite kurzor myši do tabučky v okne **Relational Model** a kliknite na pravé tlačítko myši.
2. Zvoľte príkaz **WHERE Open...** z ponuky **Shortcut menu**.

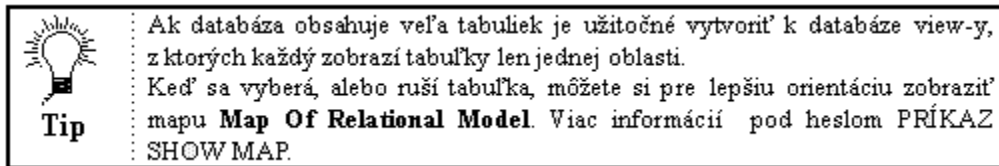
Zobrazí sa dialógové okno **Type Where**.



3. Zadajte WHERE podmienku.
4. Stlačte tlačítko **OK**.

PATRIK zobrazí časť tabučky obsahujúcu údaje, ktoré spĺňajú definovanú podmienku v novootvorenom okne.

Príkazy Select, Deselect



Definovanie view-u

1. Umiestnite kurzor myši do tabuľky v okne **Relational Model** a kliknite na pravé tlačítko myši.
 2. Zvoľte príkaz **Select** z ponuky **Shortcut menu**.
 - a) danú tabuľku vyberiete pomocou príkazu **Alone**.
 - b) Skupinu tabuliek pripojených k danej tabuľke s relačným vzťahom "one to many", kým zvolená tabuľka je na strane "one", vyberiete príkazom **With Children**
 - c) Skupinu tabuliek pripojených k danej tabuľke s akýmkoľvek relačným vzťahom vyberiete príkazom **With Whole Family**.
- Opakovaním sa dajú vybrať tabuľky, ktoré sa majú nachádzať vo **View-e**
3. Príkazom **Save As** v shortcut menu je možné definíciu **Viewu** uložiť do databázy

Zrušenie výberu zvolenej tabuľky pomocou príkazu Deselect.

1. Umiestnite kurzor myši do tabuľky v okne **Relational Model** a kliknite na pravé tlačítko myši.
 2. Zvoľte príkaz **Deselect** z ponuky **Shortcut menu**.
 - a) Na zrušenie výberu danej tabuľky vyberte príkaz **Alone**.
 - b) Na zrušenie výberu skupiny tabuliek pripojených k danej tabuľke relačným vzťahom "one to many", keď vybraná tabuľka je na strane "one", zvoľte príkaz **With Children**.
 - c) Na zrušenie výberu tabuliek pripojených k danej tabuľke s akýmkoľvek relačnými vzťahmi zvoľte príkaz **With Whole Family**.
1. Zvoľte príkaz **Deselect** z ponuky **Shortcut menu**.

Príkaz Load

Nahratie definovaného view-u

1. Zvoľte príkaz **Load** v menu **View** v okne **Relationship**. Zobrazí sa dialógové okno **Choose View To Load**.
2. Vyberte view, ktorý chcete nahráť.
1. Stlačte tlačítko **OK**.
4. View sa nahrá a relačný model sa vypne s **Edit Módu**.

Príkaz Save

Uloženie zmien, ktoré ste urobili vo view-e

1. Zvoľte príkaz **Save** v menu **View** v okne **Relationship**.
Zobrazí sa dialógové okno **Choose View To Save As**.
2. Napíšte názov view-u, ktorý ste zmenili.
1. Stlaète tlačítko **OK**.

Príkaz Delete

Vymazanie definovaného view-u

1. Zvoľte príkaz **Delete** v menu **View** v okne **Relationship**.
Zobrazí sa dialógové okno **Choose View To Delete**.
2. V poli so zoznamom view-ov označíte view, ktorý chcete vymazať.
1. Stlačíte tlačítko **OK**.

View dialógové okno

Používanjú funkcie

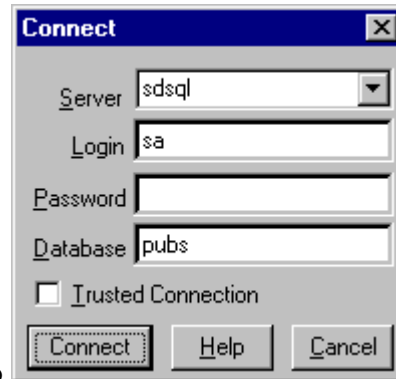
- Save
- Load
- Delete

pracujúce s pohľadmi v relaènom modeli.

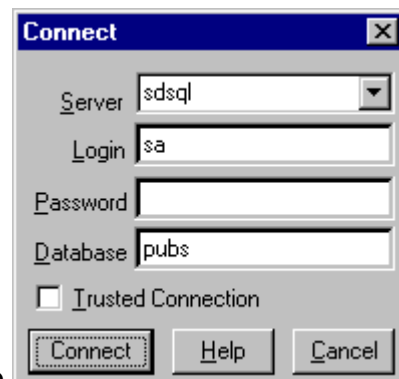
Tlačítko Edit mode

Služi na prepínanie medzi dvoma módmi.

Relačný model sa môže nachádzať v dvoch módoch.



- keď je zapnutý Edit Mode - je stlačené tlačítko **Connect** v toolbare - Za takýchto okolností sa zobrazujú príkazy Select, Deselect v **shortcut menu** a za označené tabučky sa považujú tie s modrým pozadím a za neoznačené tie s bielym pozadím.



- keď je vypnutý Edit Mode - nie je stlačené tlačítko **Connect** v toolbare - Za takýchto okolností sa nezobrazujú príkazy Select, Deselect v **shortcut menu** a zobrazené sú len tie tabučky, ktoré boli pri zapnutom Edit móde označené (s modrým pozadím).

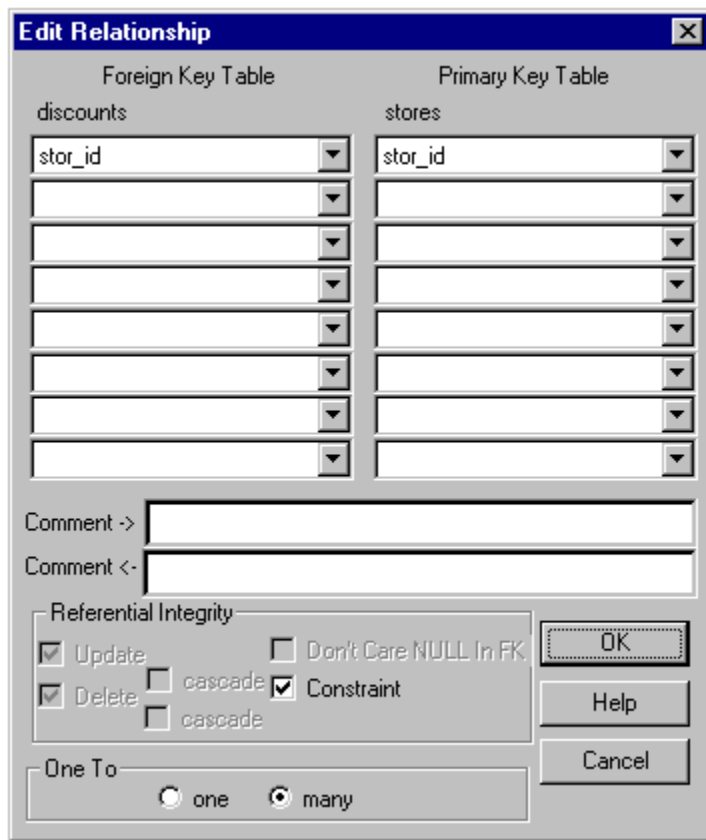
Po otvorení relácií je Edit Mode zapnutý a žiadna tabučka nie je označená.

Editovanie relačných vzťahov medzi tabuškami

Relačné vzťahy medzi tabuškami v relačnom modeli predstavujú čiary spájajúce relačné tabušky. V závislosti od relačného vzťahu môžu byť konce čiar rôznej hrúbky. Ak je koniec čiar tenký, pripojená tabuška je na strane "one" relačného vzťahu. Ak je koniec čiar hrubý, pripojená tabuška je na strane "many" relačného vzťahu (pozri obrázok nižšie).

Editovanie relačného vzťahu v okne Relational Model

1. V okne **Relational Model** dvakrát kliknite na relačný vzťah, ktorý chcete editovať. Zobrazí sa dialógové okno **Edit Relationship**.



2. V jednom z ôsmich combo boxov na ľavej strane zvolíte atribút PK tabušky.
3. V relačnom combo boxe na pravej strane zvolíte atribút FK tabušky. Opakujte kroky 3 a 4, aby sa vytvorili zložené relačné vzťahy.
4. V skupine **One To** vyberte vzťah relácie.
5. V skupine **Referential Integrity** urobte nasledujúce:
 - Zvoľte check box **Constraint** a určíte deklaratívnu referenčnú integritu (foreign key constraints).
 - Aby sa oživil check boxy **Update**, **Delete** a návazne **Cascade**, čím by ste zvolili

referenčnú integritu sprostredkovanú triggami, zrušte výber check boxu **Constraint**.

-Vyberte check box **Update**, aby sa neskôr po T-SQL príkaze UPDATE nezmenila hodnota primárneho kľúča v tabuľke primárneho kľúča, ak tento záznam má relačné záznamy.

-Po vybratí check boxu **Update** sa check box **Cascade** vedľa check boxu **Update** oživí.

-Vyberte check box **Cascade**, aby sa aktivoval update **Cascading Update** v triggr, ktorý kaskádovo spustí update všetkých relačných záznamov.

-Vyberte check box **Delete**, aby sa neskôr po T-SQL príkaze DELETE zabránilo vymazaniu záznamu z primárnej tabuľky, ak v relačnej tabuľke existujú rovnaké záznamy.

-Po vybratí check boxu **Delete**, sa oživí check box **Cascade** vedľa check boxu **Delete**.

-Vyberte check box **Cascade** aby sa aktivoval **Cascading Delete** v triggr, ktorý kaskádovo vymaže všetky relačné záznamy pre každý mazaný záznam prípadnej tabuľky.

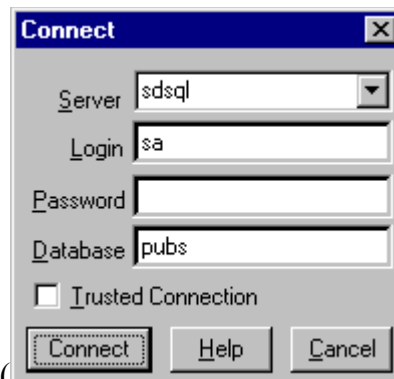
-Vyberte **Don't Care Null In FK** - foreign key môže obsahovať hodnotu NULL (v tomto prípade dopustíme existenciu **sirôt**, ktoré majú NULL foreign kľúč)

6. Do polí **Comments** a **Comments** je možné napísať komentár k relačnému vzťahu.
7. Stlačte tlačítko **OK**.

Browser databázových objektov

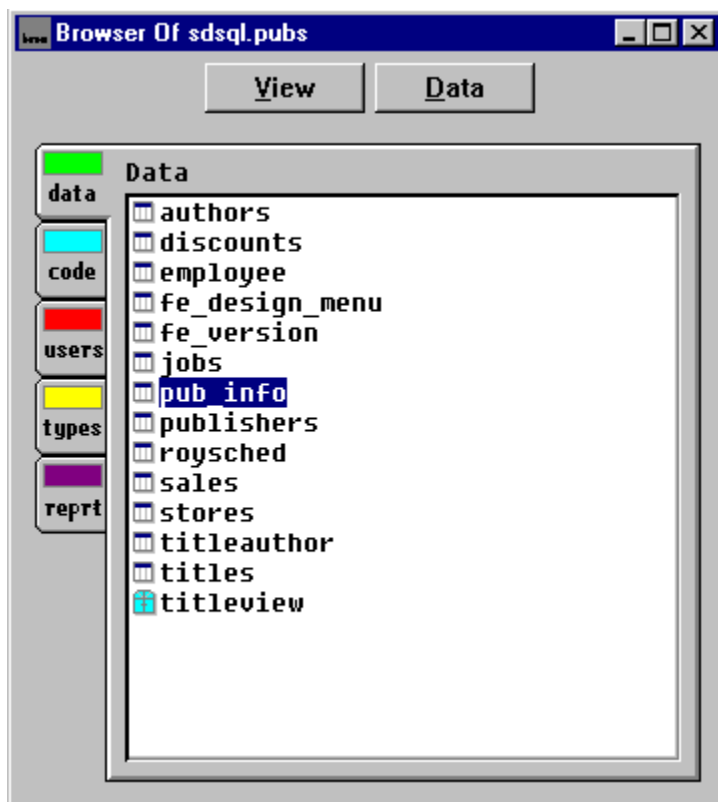
Táto kapitola opisuje okno *Browser Of názov_servera.názov_databázy* a postup ako rýchlo nájsť databázové objekty, ktoré potrebujete.

Zobrazenie okna *Browser Of názov_servera.názov_databázy*



1. Stlačíte tlačítko **Browser** () v toolbare **hlavného okna aplikácie**.

Zobrazí sa okno *Browser Of názov_servera.názov_databázy* pre aktuálnu databázu .



Browser databázy umožňuje browsovať databázové objekty. Tieto sú kôli prehľadnosti

rozdelené v piatich skupinách reprezentovaných piatimi záložkami okna.

- Data - tabuľky, view-y
- Code - stored procedúry, trigger, defaulty, rules. Implicitne je zobrazovanie triggerov vypnuté, lebo nás väčšinou nezaujímajú. V prípade záujmu je možné ich zobrazovanie zapnúť prostredníctvom funkcie Edit Filter.
- Users - užívatelia databázy
- Types - Užívateľské dátové typy
- Report - zostavy

Špecifické funkcie a tlačítka

Pre každú so skupín objektov sú prístupné špecifické funkcie, ktoré sú opísané v samostatných kapitolách. (pozri vyššie referencie)

Všeobecné funkcie browsera

Vyhľadanie objektu v okne zozname objektov

1. Stlačíte klávesu ALT a napíšete prvé písmená hľadaneho objektu (počas písania musí byť Alt stlačený).
Text, ktorý píšete sa zobrazuje v **Informaènom pruhu** aplikácie.
2. V prípade preklepu sa dá počas písania v texte vrátiť klávesou **Space** (teda **Alt space**).

PATRIK automaticky skočí na objekt začínajúci textom, ktorý píšete.

Pozn. : V skupine Users funkcia nefunguje.

Refresh

Opätovné načítanie obsahu okna sa realizuje klávesou F5.

Reporty

Táto kapitola sa venuje zostavám. Jedná sa o akési pohľad na existujúce dáta. Pre zostavy sa dajú vytvárať aj grafy, ktoré slúžia na prezentáciu dát a štatistických údajov s dátami súvisiacimi.

V skupine **Report** sa nachádzajú názvy zostáv – reportov, ktoré sú v databáze vytvorené. Na rozdiel od ostatných skupín (**data**, **code**, atd.), zostavy nie sú štandardné databázové objekty a preto ich ani nie je možné „vidieť“ v iných aplikáciách. Zostavy slúžia iba na **prezentáciu dát**, nedá sa do nich teda zapisovať. **Práva** na editovanie sa nedajú nastaviť explicitne, sú však dedené po tabuľkách nad ktorými je zostava vytvorená.

Tlačítka v Browseri pre skupinu Report

Zostavy a downsizované databázy

Nakoľko zostavy zobrazujú dáta tabuliek, tak nemá o nich význam hovoriť v **downsize**-nutej databáze. Preto celá funkcionálnosť v zostavách funguje iba v skutočnej databáze s dátami. Browser ponúka tieto tlačítka:

- **New**
- **Design**
- **Execute**

Tlačítka **New** a **Design** otvoria designer zostáv, v ktorom si môžete navrhnuť novú zostavu (**New**), alebo modifikovať existujúcu (**Design**). Samozrejme modifikácia existujúcej zostavy nie je nutná, v designeri sa dá iba prezrieť konštrukcia existujúcej zostavy. Je to vlastne najlepší pohľad do štruktúry zostavy. Tieto tlačítka slúžia viacmenej pre designera zostavy, na rozdiel od tretieho tlačítka – **Execute**, ktoré je určené skôr pre koncového užívateľa.

Tlačítko Execute slúži na prezeranie dát zostavy bez ďalších zbytočných informácií o jej štruktúre.

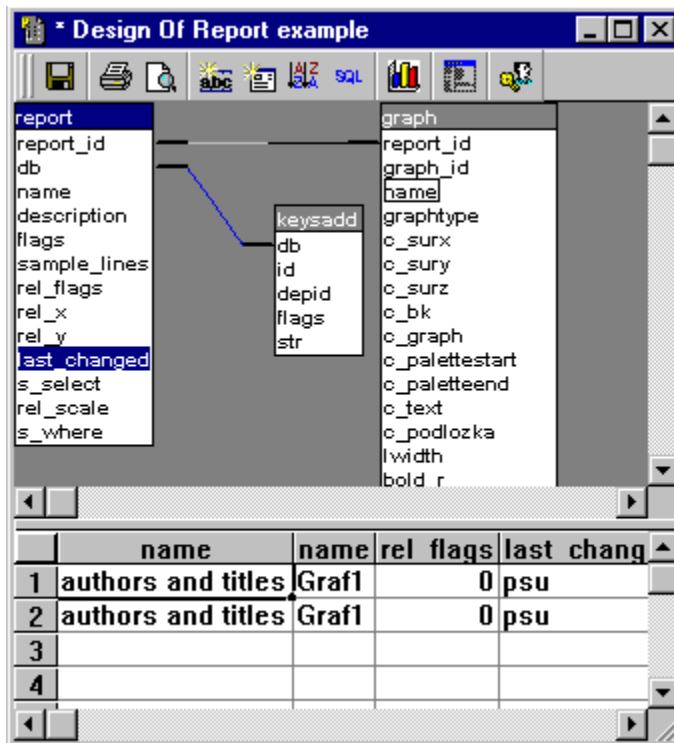
Shortcut Menu pre skupinu Report v Browseri.

Po stlačení pravého tlačítka myši nad nejakou kurzorom označenou zostavou ponúkne menu pozostávajúce s funkcií

- Edit filter - filtrácia objektov v okne browsra
- Drop object - zmazanie procedúry
- Edit notes - vytvorenie poznámky k procedúre

Designér zostáv

Zostava je vlastne „obyčajný“ SELECT (výber) s nejakých tabuliek databázy. Tento SELECT nie je však nutné (a ani možné) písať ručne. V prípade, že sa jedná o design novej zostavy, automaticky sa zobrazí dialógové okno funkcie [Add Table](#), v ktorom je možné vybrať tabuľku (prípadne viac tabuliek) s ktorej sa výber – zostava bude robiť. Zostavy nemusia byť postavené iba na dátach nejakých tabuliek. Môžu zobrazovať aj dáta [View-ov](#) a [iných zostáv](#). Týmto je umožnená akási [štruktúrovaná definícia zostáv](#). V takomto prípade je SELECT zostavy élenitejší a na miestach za klauzulou FROM sú namiesto tabuliek uvedené [alias-ované Sub-SELECT-y](#).



Okno designéra je vertikálne rozdelené na dve časti. V hornej časti sa nachádzajú [Tabuľky zostavy](#) a v spodnej časti sa nachádzajú [stĺpce zostavy](#).

Príznak Modifikácie


Titulok okna obsahuje text [Design Of Report <meno zostavy>](#). Pred týmto textom sa nachádza znak *, ak bola zostava modifikovaná.


Toolbar


Toolbar obsahuje nasledovné funkcie




[Save](#) – uloží zostavu do databázy


 [Print](#) – tlače zostavy


 [Preview](#) – nastavenie tlače zostavy


 [Add Column](#) – pridá nový stápec do tabučky a otvorí dialógové okno [Column](#)

Options


 [Add Table](#) – pridá tabučku do zostavy

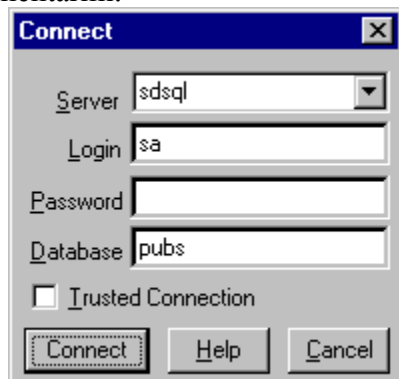
 [Order By](#) – definuje utriedenie dát zostavy

 [Show SQL](#) – zobrazí SELECT uskutočňujúci výber dát pre zostavu

 [Open New Graph](#) – otvorí nový [graf](#). Ak boli pred zvolením funkcie niektoré stápece zostavy označené, tak iba tie budú zobrazené v grafe. Ak neboli označené žiadne stápece zostavy, tak sa v grafe zobrazia všetky jej stápece. Rozlišujú sa dva typy stápcov – **číselné** a **textové**.

Číselné stápece predstavujú vektory s dátami a **textové** reprezentujú názvy bodov. V prípade že sa má v grafe zobraziť viacero textových stápcov, tak sa ich hodnoty (texty) pospájajú do jedného textu (s medzerou medzi nimi).

 [Auto Read Comments](#) – načíta popisy (komentáre) všetkých stápcov všetkých tabuliek v [relačnom modeli](#). Každá pridaná tabučka do zostavy bude automaticky načítaná aj s komentármi.



[Properties](#)

Uloženie zostáv

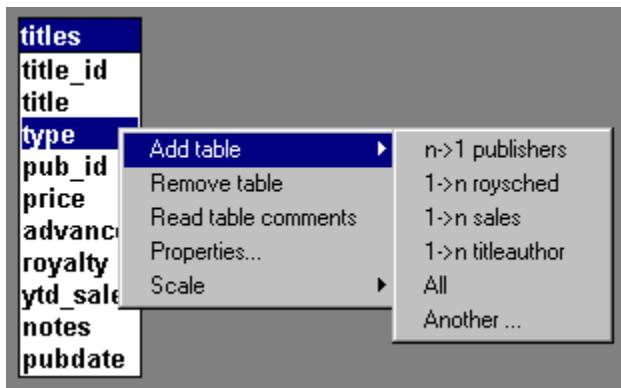
Všetky zostavy sú uložené na SQL Servi v databáze patrik. Všetko, čo sa v zostave dá nastaviť sa do databázy ukladá vrátane [grafov](#) a [preview](#) informácií. Každá zostava môže mať ľubovoľný počet grafov. Každý graf má svoj názov, ktorý by môže obsahovať medzery a akékoľvek iné znaky, aby názov mohol byť výstižný.

Tabuľky zostavy – relačný model.

V hornej časti designéra je zobrazený **relačný model** tabuliek, s ktorých pozostáva zostava. Ak pozostáva s viacerých tabuliek a je správne urobená, tak by všetky tabuľky mali byť spojené reláciami do jedného celku. Ďi by sa už jednalo o relácie **skutočné** (zobrazované **čiernou farbou**), alebo o relácie vytvorené len pre účely danej zostavy (zobrazované **modrou farbou**). Oba typy relácií hrajú v zostave úlohu skutočných relácií, farebne sú odlišené iba preto, aby neskutočné relácie nezmatli užívateľa.

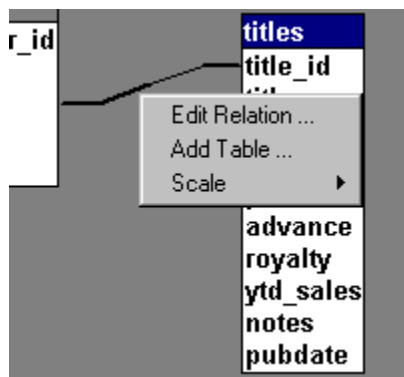
Shortcut Menu v Relačnom Modeli Designéra Zostáv.

Pravé tlačítko myši vyvolá tradične **shortcut menu**. Taktiež tradične je toto shortcut menu rozlíšené za rôznych okolností – podľa umiestnenia myši.



Ak je myš umiestnená **na ploche nejakej tabuľky**, tak sa po stlačení pravého tlačítka na myši ponúknú funkcie s nasledovnej množiny funkcií:

- **Add Table** – pozostáva zo submenu, ktoré obsahuje všetky mená tabuliek, ktoré majú s danou tabuľkou vytvorenú **reláciu** a s príkazov **All** a **Another...**. Po výbere ktoréhokoľvek mena tabuľky sa do relačného modelu pridá tabuľka daného mena. Pri výbere **All** sa do relačného modelu pridajú všetky tabuľky uvedené v menu. Ak chcete medzi tabuľky pridať takú, ktorá nie je v menu (takú, čo **nemá reláciu s danou tabuľkou**) zvolíte funkciu **Another...**, ktorá plní univerzálnu funkciu pridania tabuľky (tabuliek).
- **Remove Table** – voľba tejto funkcie zmaže tabuľku z relačného modelu zostavy, čím tabuľka prestane figurovať v zostave. Ak nejaký stápec danej tabuľky je zároveň aj stápcom zostavy, **Patrik** na to upozorní a umožní proces stornovania. Ak však budete trvať na jeho realizácii, automaticky sa z reportu zrušia aj stápcy mazanej tabuľky.
- **Read Table Comments | Hide Table Comments** - tu sa zobrazí vždy iba jedna s týchto funkcií, lebo, lebo funkcie vykonávajú navzájom inverzné operácie. Funkcia **Read Table Comments** slúži na načítanie poznámok k stápcom tabuľky. Tieto poznámky sa dajú zadať v okne **Design Table**. Nie vždy môže byť režim, v ktorom sú zobrazené poznámky výhodný, lebo texty s poznámkami môžu niekedy zaberať väčšinu plochy relácií a potom sa tam nemusí zmestiť veľa tabuliek. Obyčajne sa načítajú poznámky iba pri kompozícii zostavy a potom sa vypnú, aby tam zbytočne nezaberali miesto.
- **Properties** – funkcia slúži na doladenie a zoptimalizovanie SELECT-u.
- **Scale**



Ak sa myš nenachádza na ploche žiadnej tabuľky, ale je **aktívna nejaká relácia** (relácia sa aktivuje tak, že sa na ňu klikne ľavým tlačítkom myši), tak sa zobrazí menu s funkciami

- **Delete Relation** – iba ak je aktívna neskutočná relácia (relácia zobrazená modrou farbou). Skutočné relácie sa dajú mazať iba v **module Relácií**.
- **Edit Relation** – slúži na prezretie/zmenu vlastností relácie súvisiacich s vytváraním SELECT-u.
- **Add Table...**
- **Scale**

A za ostatných okolností sa zobrazí menu pozostávajúce s funkcií:

- **Add Table...**
- **Scale**

Presun tabuliek

Tabuľky v relačnom modeli je možné ľubovoľne presúvať, nemá to žiaden vplyv na funkčnosť zostavy.

Vytváranie nových (neskutočných relácií)

Niekedy je potrebné spojiť stĺpce dvoch tabuliek reláciou, ale bez akejkoľvek referenčnej integrity a prezistencie relácie mimo zostavy. Designér podporuje takéto fiktívne relácie a zobrazuje ich modrou farbou. Postup vytvorenia novej relácie:

1. Označíte v jednej s tabuliek stĺpce, ktoré majú byť napojené reláciou.
2. Stlačíte ľavé tlačítko myši (nepúšťajte) a prejdite myšou na plochu druhej s tabuliek.
3. Pustíte ľavé tlačítko myši.
4. Zobrazí sa dialógové okno **Edit Relation**, v ktorom nastavíte dvojicu stĺpcov, ktoré majú byť navzájom napojené a typ napojenia (**join-u**).

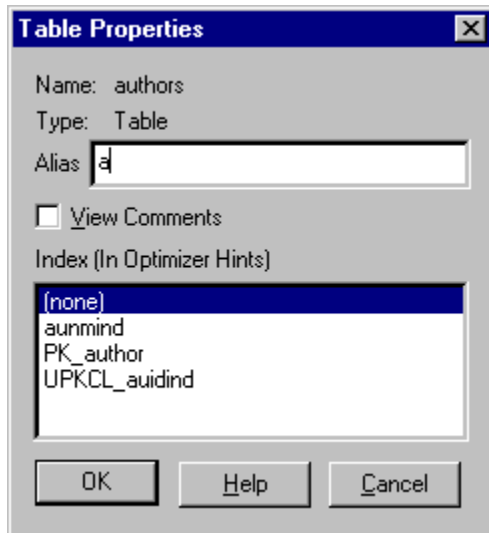
Pridávanie stĺpcov do zostavy

Stĺpce zostavy sú uvedené v **spodnej časti** designéra zostáv. Pridanie stĺpca sa realizuje buď dvojkliknutím (ľavého tlačítka myši) na daný stĺpec tabuľky v relačnom modeli, alebo **drag and drop** pretiahnutím stĺpca (prípadne viacerých stĺpcov naraz) medzi stĺpce zostavy.

Properties tabuľky

Funkcia sa vyvolá tak, že sa na ploche nejakej tabuľky v relačnom modeli designéra zostávajú stlačením pravého tlačítka myši a zvolí sa funkcia **Properties**.

Po tejto voľbe sa otvorí dialógové okno



V kolonke **Name** je zobrazené meno danej tabuľky.

V kolonke **Type** sa nachádza typ objektu. Je to na 99 percent tabuľka (**Table**), ale môže to byť aj **View** alebo **Report** - iná zostava.

Kolonka **Alias** je už vyplniteľná a je do nej možné zadať **alias** názov, pre skrátenú identifikáciu tabuľky v SELECT-e. Toto sa zvykne používať v prípadoch, keď chce užívateľ vytvoriť SELECT pomocou designer-a a následne ho chce použiť v nejakej **Procedúre** alebo **Query**.

Check Box **View Comments** reprezentuje stav načítanosti komentárov k stĺpcom tabuľky. Tento stav je pochopiteľne možné zmeniť.

Listbox **Index (In Optimizer Hints)** slúži na optimalizáciu SELECT-u vracajúceho dáta zostavy. Pre danú tabuľku je možné určiť **index**, ktorý má byť v SELECT-e explicitne navrhnutý optimalizérom na použitie. Listbox obsahuje riadok **(none)** a **mená indexov** danej tabuľky. Ak chceme nechať výber indexov SQL optimalizérom (ktorý by to koniec koncov mal sám najlepšie určiť), tak nastavíme **(none)**.

Táto funkcia je podporovaná Patrikom, lebo niekedy SQL Server neurčí správne najoptimálnejšie indexy.

Nastavenie vlastností relácie.

Funkcia sa vyvolá pri pridávaní novej relácie, dvojkliknutí na reláciu, alebo voľbou funkcie **Edit Relation** s **shortcut menu**, nad aktivovanou reláciou, alebo pri pridávaní relácie. Vykreslí sa nasledovné dialógové okno :

The dialog box is titled "Edit Relationship" and contains two columns: "Foreign Key Table" and "Primary Key Table".

- Foreign Key Table:** "titleauthor". The selected field is "nl_author_id".
- Primary Key Table:** "authors". The selected field is "nl_author_id".

Below the columns, there is a section for "Outer join" with two radio buttons: "Left" (unchecked) and "Right" (checked). To the right of the radio buttons is the text "Equation sample: =*".

At the bottom right of the dialog are three buttons: "OK", "Cancel", and "Help".

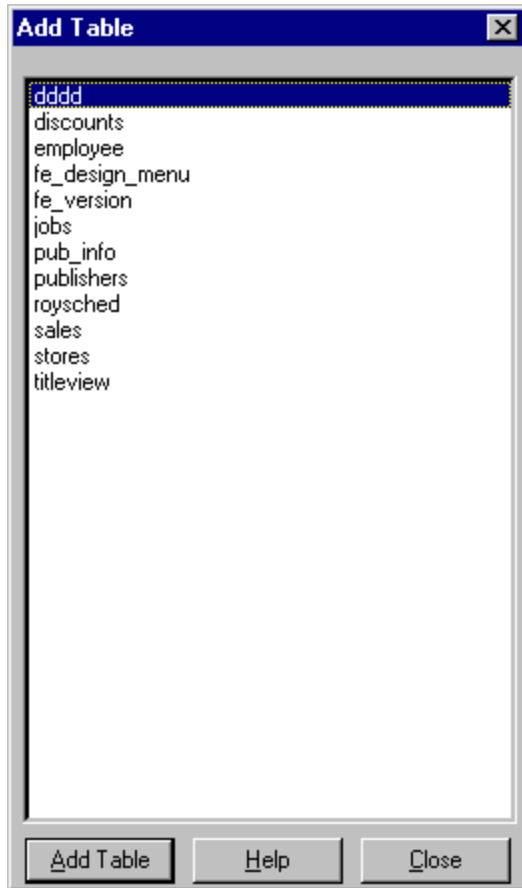
Sú tu zobrazené mená stĺpcov v oboch tabuľkách, ktoré sú na seba napojené. Toto napojenie je možné zmeniť iba pre **neskutočnú** reláciu (reláciu vykreslenú **modrou farbou**), alebo pre novopridanú reláciu. Skutočné relácie sa pochopiteľne nedajú zmeniť, v tomto dialógovom okne, ale v dialógovom okne **Edit Relation**.

V sekcii **Outer Join** je zobrazený typ napojenia (**join-u**). Toto je možné zrealizovať pre ľubovoľný typ relácie, lebo to nesúvisí s jej deklaráciou, ale iba s budovaním **JOIN-u** v **SELECT-e**.

Add Table

Funkcia sa automaticky vyvoláva pri vytváraní novej zostavy, ale dá sa vyvolať aj neskôr tlačítkom  na toolbare Designéra zostáv.

Zobrazí sa **dialógové okno**



Obsahuje zoznam všetkých **tabuliek**, **view-ov** a **zostáv** (spolu – **objektov**), ktoré sa momentálne nenachádzajú v relačnom modeli designéra. **Objekty** (tabuľky) sa do relačného modelu pridajú buď dvojkliknutím na ich mená, alebo označením a následným stlačením tlačítka **Add Table**.

Scale

Shortcut menu **Scale** obsahuje položky

- 100 %
- 75 %
- 50 %
- 25 %

Zvoľte položku s číslom reprezentujúcim **zmenšenie**, pri ktorom chcete mať **relatívny model** zobrazovaný.

Stápece zostavy

V spodnej èasti designéra zostáv je zobrazená tabuška zostavy so vzorkou dát.

Tabuškové vlastnosti

Tabuška vykazuje väčšinu štandardných tabuškových vlastností, ako je

- oznaèovanie stápcov / riadkov / obdážnikových oblastí buniek
- funkcionálna **clipboard**
- rozširovanie/zužovanie riadkov/stápcov
- triedenie stápcov dvojkliknutím v prvom riadku tabušky

Zmena poradia stápcov.

Funkcionalitou **drag and drop** je možné presúvať stápece v rámci zostavy nasledovne:

1. Oznaíte stápec (viacero stápcov vedľa seba), ktorý chcete presúvať a stlaíte 3. avé tlačítko myši.
2. Presunte myš na miesto stápcu, pred ktorý chcete presúvaný stápec vložiť.
3. Pustíte 3. avé tlačítko myši.

Mazanie stápcov zo zostavy

1. Oznaíte stápec (stápece), ktoré chcete zmazať.
2. Stlaíte klávesu **Delete**.

Shortcut menu

Stlaením pravého tlačítka myši sa zobrazí menu s nasledujúcimi funkciami

- **Open Graph** – obsahuje submenu s názvami **grafov**, ktoré sú vytvorené pre zostavu a príkaz **< New >**. Vošbou niektorého z existujúcich grafov sa **graf zobrazí**. Nemusí sa to vždy podariť, lebo stápece zostavy, ktoré graf zobrazoval mohli byť medzièasom zo zostavy zrušené. **Patrik** v takomto prípade vypíše varovné hlásenie a umožní invalidný graf **zmazať**. Vošbou **< New >** sa zobrazí nový graf. Ak boli pred zvolením funkcie niektoré stápece zostavy oznaené, tak iba tie budú zobrazené v grafe. Ak neboli oznaené žiadne stápece zostavy, tak sa v grafe zobrazia všetky jej stápece. Rozlišujú sa dva typy stápcov – **èíselné** a **textové**. **Èíselné** stápece predstavujú vektory s dátami a **textové** reprezentujú názvy bodov. V prípade že sa má v grafe zobraziť viacero textových stápcov, tak sa ich hodnoty (texty) pospájajú do jedného textu (s medzerou medzi nimi).
- **Column Options** – umožní nastaviť vlastnosti stápcu zostavy. **Menu obsahuje túto položku, ak je pri vyvolaní shortcut menu oznaený nejaký stápec zostavy.**
- **Delete Column** – zruší stápec zo zostavy. **Menu obsahuje túto položku, ak sú pri vyvolaní shortcut menu oznaené nejaké stápece zostavy.**
- **Add Column** – pridá nový stápec do tabušky a otvorí dialógové okno **Column Options**
- **Refresh** – opätovne načíta vzorku dát zostavy
- **Show Cursor Script** - vygeneruje a zobrazí script, browsujúci jednotlivé záznamy SELECT-u zostavy pomocou kurzora. Na názov kurzora sa najskôr opýta.

Column Options

Funkcia umožňuje nastaviť podrobné vlastnosti stĺpca designovanej zostavy v dialógovom okne:

The screenshot shows the 'Report column options' dialog box. The 'Column Name' field contains 'cena knihy'. The 'Table Column' dropdown shows 'titles.price'. The 'Aggregate' dropdown is set to 'ALL'. The 'if NULL' field contains '0'. The 'Align' dropdown is set to 'Auto'. The 'Group by' checkbox is unchecked. There are two sections for exception conditions: 'Top Exception Condition' and 'Bottom Exception Condition'. The 'Top Exception Condition' section has radio buttons for 'None', 'Top M', 'Top M %', and 'At or Above Value of M'. The 'Bottom Exception Condition' section has radio buttons for 'None', 'Bottom N', 'Bottom N %', and 'At or Below Value of N'. Below these sections are input boxes for 'M = 10' and 'N ='. At the bottom of the dialog are 'OK', 'Help', and 'Cancel' buttons.

- **Column Name** - obsahuje text s názvom stĺpca zostavy.
- **Table Column** - ktorý stĺpec ktorej tabuľky zobrazuje daný stĺpec zostavy. V combo boxe **table column** sa nachádzajú mená stĺpcov všetkých tabuliek nachádzajúcich sa v zostave. Ak si nevyberiete s uvedených stĺpcov môžete tu napísať nejaký vlastný text obsahujúci nejakú funkciu (prípadne znak *). Tento text potom bude figurovať v SELECT - liste SELECT-u zostavy.
- **Aggregate** - combo box obsahujúci mená všetkých agregátnych funkcií Transakt SQL-u a pseudo funkciu ALL, ktorá znamená, že sa zobrazia všetky dáta stĺpca tabuľky, takže sa vlastne v tomto prípade nepoužije žiadna agregátna funkcia. **Ak pre nejaký stĺpec (nejaké stĺpce) použijete nejakú agregátnu funkciu, tak vytvorený SELECT bude automaticky Grupovaný ostatnými stĺpcami zostavy. Je to totiž jediná možnosť ako vytvoriť korrektný SELECT.**
- **Group By** - reprezentuje fakt, že je v SELECT-e zostavy použité **Grupovanie** podľa daného stĺpca.
- **if NULL** - hodnota, ktorá bude uvedená namiesto NULL hodnôt stĺpca. **Hodnota musí mať rovnaký dátový typ, ako má daný stĺpec.**
- **Align** - určuje zarovnanie hodnôt stĺpca (**vľavo**, **vpravo** alebo **do stredu**). Voľba **Auto** reprezentuje automatické zarovnanávanie (texty **naľavo** a čísla **napravo**).
- **Top Exception Condition** - funkcia slúži na vyznačovanie najväčších hodnôt stĺpca. Môžeme tu nastaviť, či chceme zvýrazniť (**červenou farbou**)
 - **Top M** - M najvyšších hodnôt stĺpca.
 - **Top M %** - najvyšších M percent hodnôt stĺpca
 - **At Or Above Value Of M** - hodnoty stĺpca, ktoré sú väčšie alebo rovné hodnote M.

Táto voľba je možná iba pre celočíselné stĺpce. Pre ostatné by totiž nemala význam, lebo M je celočíselná hodnota.

- **None** - ak nechceme zvýrazniť žiadne veľké hodnoty
- **M =** - celočíselná hodnota M

· **Bottom Exception Condition** - funkcia slúži na vyznačovanie najnižších hodnôt stĺpca.

Môžeme tu nastaviť, či chceme zvýrazniť (zelenou farbou)

- **Bottom N** - N najnižších hodnôt stĺpca.
- **Top N %** - najnižších N percent hodnôt stĺpca
- **At Or below Value Of N** - hodnoty stĺpca, ktoré sú menšie alebo rovné hodnote N. Táto voľba je možná iba pre celočíselné stĺpce. Pre ostatné by totiž nemala význam, lebo N je celočíselná hodnota.
- **None** - ak nechceme zvýrazniť žiadne malé
- **N =** - celočíselná hodnota N

Uloženie zostavy do databázy

Ak sa nejedná o **novovytvorenú** zostavu, tak sa táto uloží do databázy pod pôvodné meno. Ak sa o **novovytvorenú** zostavu jedná, tak sa otvorí dialógové okno

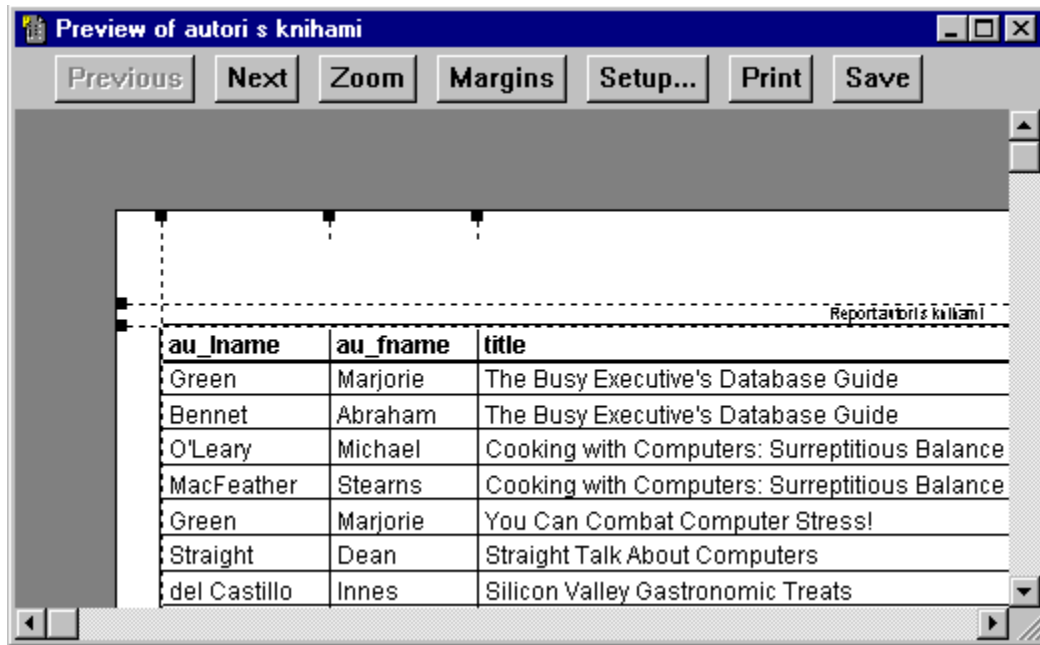


s kolonkami:

- **Name** - sem napíšete meno ukladanej zostavy
- **Existing Reports** - tu sú uvedené mená všetkých zostáv, ktoré sa nachádzajú v databáze. Za meno ukladanej zostavy je možné určiť aj niektoré s týchto mien. Potom sa zostava s uvedeným menom nahradí ukladanou zostavou. **Patrik sa optýa, či chcete ozaj zostavu prepísať (overwrite).**

Preview

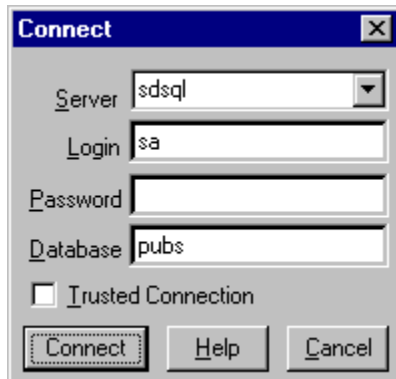
Umožňuje nastaviť výzor tlačenej zostavy.



- **Previous, Next** - tlačítka slúžia na prepínanie zobrazenej stránky v **Preview**.
- **Zoom** - tlačítkom je možné prepínať medzi **zmenšeným pohľadom** - v okne vidno celú stránku a **zväčšeným pohľadom** - v okne je vidno iba časť stránky, kde pohľad na vyvolenú časť stránky je možné nastaviť scrollbarmi okna **Preview**.
- **Margins** - zobrazí čiarkované čiary, ktorými je možné nastaviť veľkosti okrajov a všetkých stĺpcov zostavy. Čiarkované čiary sa dajú zrušiť opätovným stlačením tlačítka **margins**.
- **Setup** - umožní nastaviť ďalšie detaily (fonty, hlavičku, pätičku) tlačenej zostavy.
- **Print** - vytlačí zostavu
- **Save** - uloží nastavenia, ktoré boli v **preview** zostavy nastavené. Tlačítko je stlačené, ak bola urobená v preview nejaká zmena. Uložiť nastavenia je možné iba pre zostavu, ktorá už je uložená v databáze.

Preview Setup

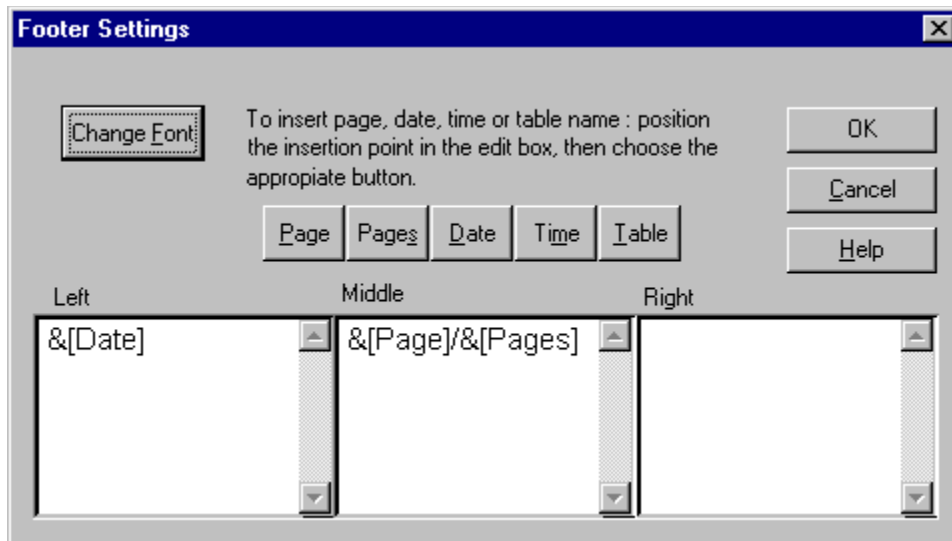
Otvorí dialógové okno, v ktorom je možné nastaviť charakter tlaenej zostavy



v kolonkách

- **Table Font - Change** - otvorí dialógové okno v ktorom je možné nastaviť font, ktorým budú tlaené bunky tabuľky. Týmto fontom je v dialógovom okne vypísaný text **Table Font**.
- **Horizontal Lines** - či majú byť tlaené linajky medzi riadkami tabuľky.
- **Vertical Lines** - či majú byť tlaené linajky medzi jednotlivými stĺpcami tabuľky.
- **Table Border** - či má byť tlaený okraj (širšia linajka) okolo tabuľky
- **Table Header** - nastavenia týkajúce sa hlavičky tabuľky (obsahujúcej názvy stĺpcov)
 - **On First Page** - hlavička tabuľky sa bude nachádzať iba na vrchnej stránke zostavy.
 - **On All Pages** - hlavička tabuľky sa bude nachádzať na každej stránke zostavy. **Ak nie je nastavený ani jeden z dvojice check boxov, tak sa hlavička tabuľky nebude nachádzať na žiadnej stránke zostavy.**
 - **Separation Line** - či bude hlavička od buniek tabuľky (v prípade, že bude tlaená) oddelená širšou čiarou.
 - **Table Header Font - Change** - otvorí dialógové okno v ktorom je možné nastaviť font, ktorým bude tlaená hlavička tabuľky. Týmto fontom je v dialógovom okne vypísaný text **Table Header Font**.
- **Header - Edit** - v dialógovom okne **Header Settings** umožní nastaviť hlavičku strán.
- **Footer - Edit** - v dialógovom okne **Footer Settings** umožní nastaviť pätičku strán. **Biele úseky pod oboma týmito funkciami načítajú preview hlavičky a pätičky.**
- **Over Then Down , Down Then Over** - nastavenie poradia tlače stránok, v prípade, že sa zostava šírkou nezmesť na jednu stranu.
- **Printer Settings** - otvorí dialógové okno **drivera** tlačiarnie.

Header / Footer Settings



Dialógové okno, umožňujúce nastaviť hlavičku / pätičku tlačených strán zostavy. Všetky tlačené strany majú pochopiteľne tie isté definície hlavičky / pätičky (**ďalej len hlavičky**). Definícia hlavičky pozostáva z troch častí **Left** - na ľavo zarovnaná časť, **Middle** - časť zarovnaná v strede, **Right** - na pravo zarovnaná časť. Každú z týchto častí reprezentuje príslušný textový editor, kde je možné napísať text, ktorý bude v danej sekcii hlavičky vypisovaný. Ak na určitej časti textu má byť nejaká **substitúcia za hodnotu**, tak umiestnite kurzor do textu na dané miesto a stlačte niektoré z tlačítok:

- Page - vloží číslo tlačenej strany
- Pages - vloží číslo označujúce počet tlačených strán
- Date - vloží dátum, v ktorom bude zostava tlačená
- Time - vloží čas, v ktorom bude zostava tlačená
- Table - vloží meno tlačenej zostavy (tabuľky)

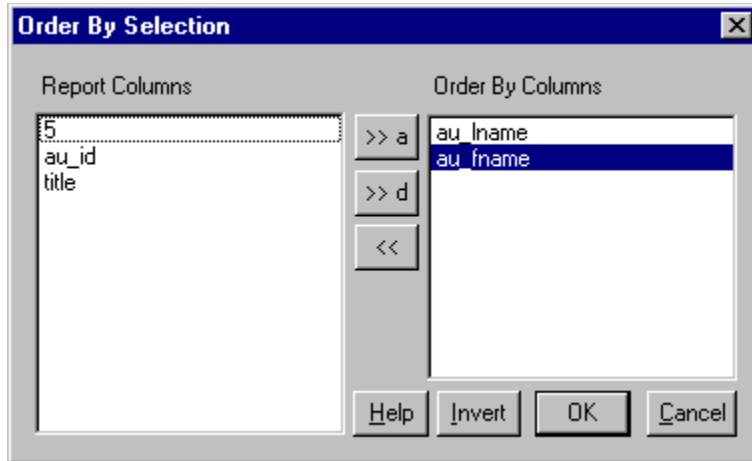
Tlačítko **Change Font** otvorí dialógové okno v ktorom je možné nastaviť font sekcie, v ktorej sa nachádza textový kurzor.

Print

Vytlaeí zozstavu. Vzhľad vytlaèenej zozstavy je možné pred samotnou tlaèou nastaviť a/alebo vidieť v [Preview](#).

Order By

V dialógovom okne



umožňuje nastaviť **utriedenie** dát v zostave.

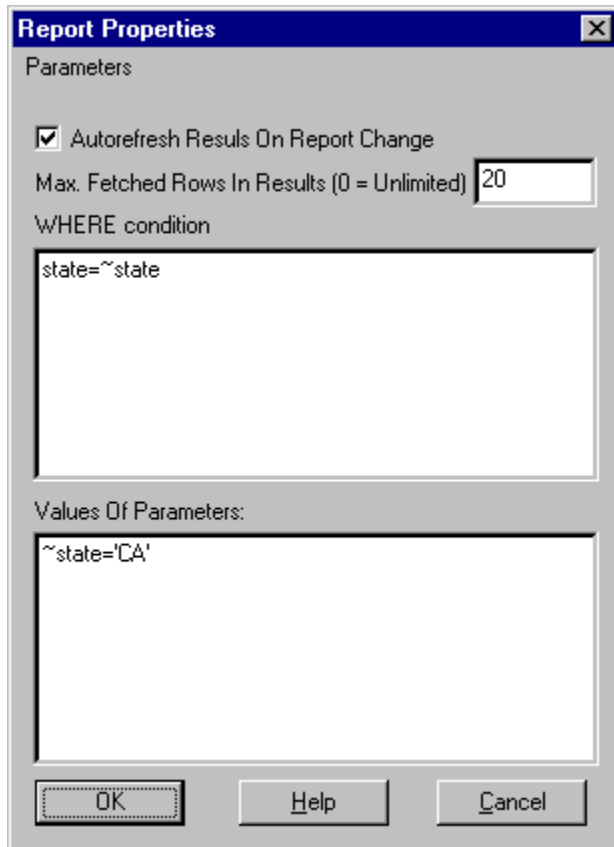
- v listboxe **Report Columns** sa nachádzajú názvy stĺpcov zostavy (abecedne utriedené), pod ktorých neprebíha triedenie. Môžu sa to vyskytovať aj čísla. Jedná sa o poradové čísla stĺpcov zostavy, ktoré sú vypočítavané (teda nie priamo vybrané s tabuľkami).
- v listboxe **Order By Columns** sa nachádzajú názvy stĺpcov, pod ktorých prebieha triedenie. Stĺpce sa tu nachádzajú v poradí definujúcom prioritu triedenia. Ak sa za názvom stĺpca vyskytuje text “- Desc”, tak je triedenie pod daného stĺpca zostupné (inak je triedenie vzostupné).
- tlačítko **>> a** pridá na koniec listboxu **Order By Columns** mená stĺpcov označené v listboxe **Report Columns**. Triedenie pod pridaných stĺpcov bude prebiehať vzostupne.
- tlačítko **>> d** pridá na koniec listboxu **Order By Columns** mená stĺpcov označené v listboxe **Report Columns**. Triedenie pod pridaných stĺpcov bude prebiehať zostupne.
- tlačítko **<<** vyhodí s listboxu **Order By Columns** mená označených stĺpcov späť do **Report Columns**.
- tlačítko **Invert** zamení **vzostupné** a **zostupné** triedenie pod stĺpcov označených v **Order By Columns**.

Show SQL

Otvorí dialógové okno s **read-only** editorom, v ktorom je zobrazený text SELECT-u reprezentujúceho zostavu. Text zostavy je možné označiť, skopírovať do **clipboardu (Ctrl C)** a následne vložiť povedzme do Query Editoru.

Properties

Služi na konfiguráciu WHERE podmienky a na nastavenie chovania designéra.



- **Autorefresh Results On Report Change** - pri každej zmene zostavy sa automaticky načíta z SQL servra nová vzorka dát zostavy. Toto je možné **checkboxom** zrušiť/nastaviť. Ak je **Autorefresh** zrušený je možné obnovovať dáta vzorky manuálne funkciou **Refresh** v **shortcut menu** medzi stápcami zostavy.
- **Max Fetched Rows In Results** - kolonka obsahuje číslo zhora ohraničujúce počet riadkov vzorky dát zostavy. 0 znamená, že počet riadkov vzorky zostavy nebude obmedzený.

Where podmienka

Where podmienka SELECT-u zostavy pozostáva z dvoch častí.

1. Časť obsahujúca **join-y** relačne spojených tabuliek. Táto časť je generovaná do SELECT-u designérom automaticky.
2. Užívateľsky definovaná časť WHERE podmienky. Táto časť je editovateľná v kolonke **WHERE condition** dialógového okna.

Obe časti sú spojené logickým operátorom **AND**.

Parametre

Užívateľsky definovaná časť **WHERE** podmienky môže obsahovať identifikátory začínajúce

znakom ~ . Tieto identifikátory sú designérom chápané ako parametre. Hodnoty sa dajú priradiť parametrom v kolonke **Values Of Parameters** napísaním textu priradení napr:

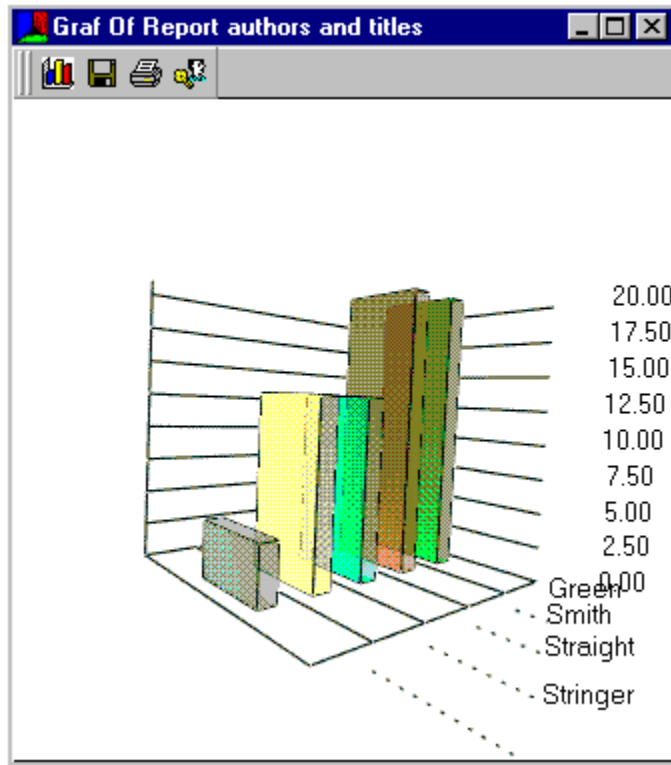
~state='CA'

~x=8

Menu Parameters dialógového okna

- **File** - kolonka **Values Of Parameters** obsahuje text, ktorý sa dá uložiť do textového súboru a neskôr s ním opätovne načítať.
 - **Load...** - umožňuje nahráť text do kolonky **Values Of Parameters** s textového súboru.
 - **Save As...** - uloží text napísaný v kolonke **Values Of Parameters** do súboru.
- **SQL** - hodnoty parametrov sa dajú uchovávať v tabuľke **patrik.param**. Každá zostava môže mať ľubovoľný počet **sád parametrov**. Každá sada parametrov môže obsahovať definície hodnôt pre viacero parametrov. Každá **sada parametrov** má svoje identifikačné meno.
 - **Load** - slúži na načítanie sady parametrov s tabuľky **param**.
 - **Save As** - slúži na uloženie sady parametrov do tabuľky **param**. Zobrazí dialógové okno, v ktorom sú zobrazené mená všetkých **sád parametrov** danej zostavy a umožňuje zadať nové meno pre **sadu** alebo prepísať už existujúcu sadu.
 - **Save** - to isté ako **Save As**, ale ak už bola raz počas práce s **Properties** sada parametrov uložená, tak sa teraz nebude dotazovať na meno, ale uloží sa pod menom, pod ktorým bola uložená posledne.

Graf



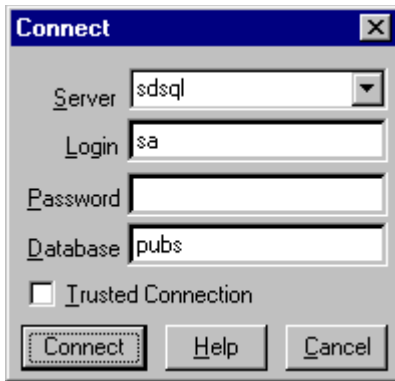
Graf slúži na grafickú prezentáciu údajov zostavy. Za hodnoty jednotlivých vrcholov grafu sa berú údaje s aspoň jedného číselného stĺpca zostavy (ak ide o jeden stĺpec jedná sa o **dvojmerný** graf a ak o viac, tak o **trojrozmerný**). Za popisy vrcholov sa berú dáta aspoň jedného textového stĺpca zostavy.

Shortcut Menu

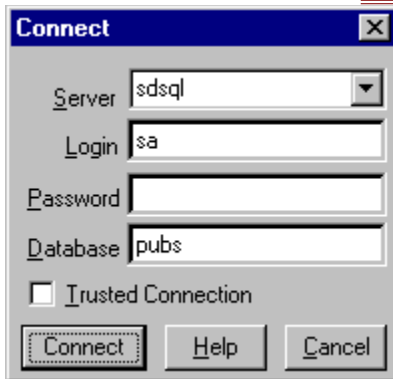
Stlačením pravého tlačítka myši na ploche grafu vykreslí **shortcut menu** s nasledovnými funkciami.

- **Graph Type** - slúži na zmenu typu grafu.
- **Options**
- **Rotation** - otvorí dialógové okno umožňujúce nastaviť natočenie 3D grafu. **Voľba je možná iba pri type grafu Bar 3D, alebo Pie 3D.**
- **Size Of Graph** - umožňuje zmeniť veľkosť grafu. Táto je nezávislá od veľkosti okna s grafom.
- **Save Graph "<meno grafu>"** - funkcia uloží modifikácie vykonané v grafe späť do databázy
- **Save Graph As**
- **Print Graph ...** - otvorí **Printer Setup** dialógové okno, v ktorom je možné nastaviť typ tlače a vytlačiť graf.

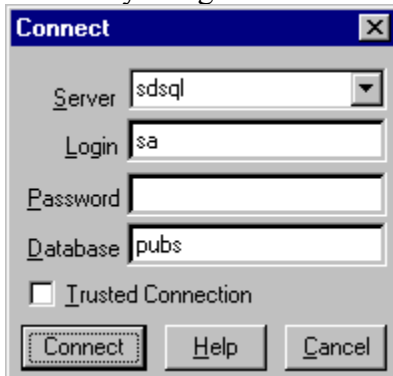
Toolbar obsahuje nasledovné funkcie : (všetky sú obsiahnuté v **Shortcut Menu**)



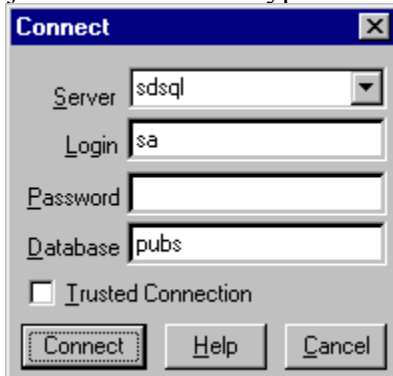
[Graph Type](#) - slúži na zmenu typu grafu



[Save Graph](#) - funkcia uloží modifikácie vykonané v grafe späť do databázy. Ak graf ešte nebol uložený, zavolá [Save Graph As](#)



[Print Graph ...](#) - otvorí [Printer Setup](#) dialógové okno, v ktorom je možné nastaviť typ tlače a vytlačiť graf.



[Options](#).

Farby grafu je možné nastaviť vyvolanie funkcie [Colors](#) v [Options](#).

Save graph As

Otvorí dialógové okno



s kolonkami:

- **Name** - sem napíšete meno ukladaého grafu.
- **Existing Graphs** - tu sú uvedené mená všetkých grafov, ktoré sú vytvorené pre danú zostavu. Za meno ukladaného grafu je možné urèi aj niektoré s týchto mien. Potom sa graf s uvedeným menom nahradí ukladaným grafom. **Patrik sa opýta, èi chcete ozaj graf prepísa (overwrite).**

Uvedením nepoužitého mena grafu pridá nový graf k zostave.

Graf je možné uloži , iba ak je už uložená zostava.

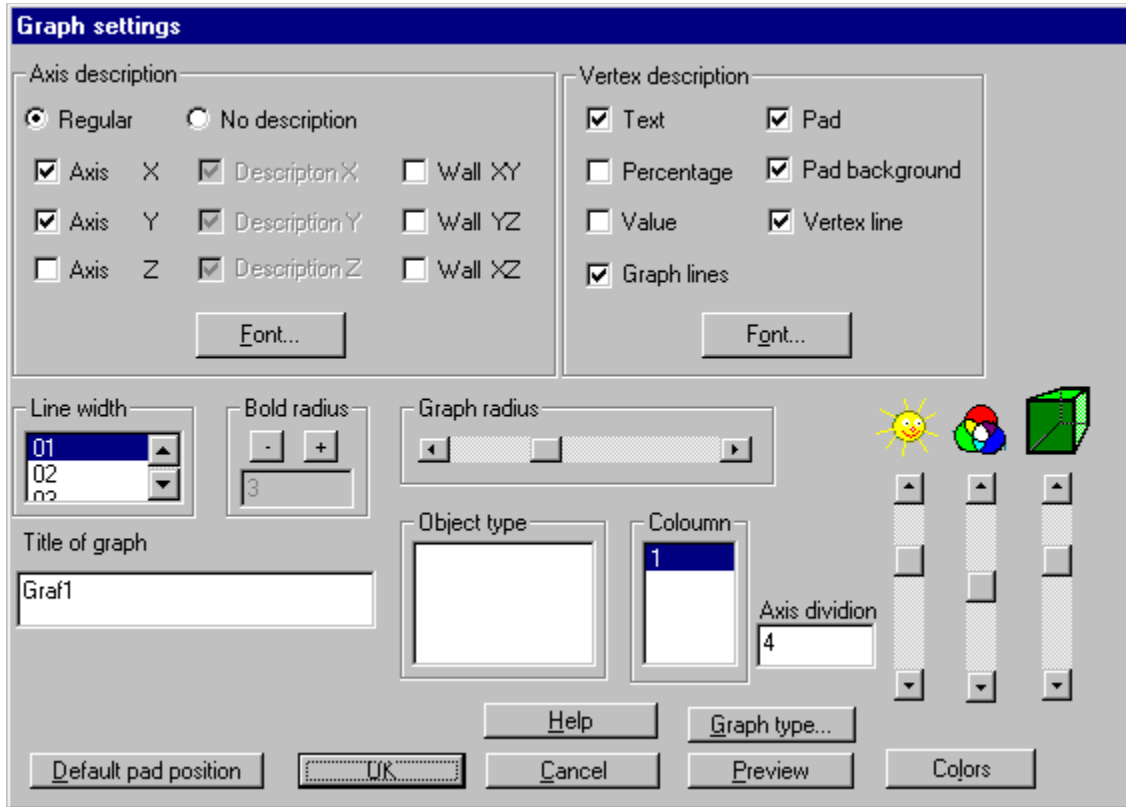
Typy Grafu

Dialógové okno Graph Type obsahuje rádio tlačítka s názvami typov grafov:

- **Simple Line** - jednoduchý 2d graf, kde hodnoty vrcholov reprezentujú body v súradnicovej sústave pospájané čiarou. V grafe sú, aj pre viacstĺpcové grafy, zobrazené hodnoty jedného stĺpca. Zobrazený stĺpec je možné nastaviť v [Options](#).
- **Multi Line** - jednoduchý 2d graf, kde hodnoty vrcholov reprezentujú body v súradnicovej sústave pospájané čiarou. V grafe sú sú zobrazené všetky stĺpce viacstĺpcového grafu.
- **Bar 2D** - stĺpcový graf. V grafe sú, aj pre viacstĺpcové grafy, zobrazené hodnoty jedného stĺpca. Zobrazený stĺpec je možné nastaviť v [Options](#).
- **Multi Bar** - stĺpcový graf. V grafe sú sú zobrazené všetky stĺpce viacstĺpcového grafu.
- **Bar 3D** - trojrozmerný stĺpcový graf. V grafe sú sú zobrazené všetky stĺpce viacstĺpcového grafu. Tento typ grafu je najideálnejší na viacstĺpcové grafy.
- **Pie 2D** - dvojrozmerný koláčový graf. V grafe sú, aj pre viacstĺpcové grafy, zobrazené hodnoty jedného stĺpca. Zobrazený stĺpec je možné nastaviť v [Options](#).
- **Pie 3D** - dvojrozmerný koláčový graf. V grafe sú, aj pre viacstĺpcové grafy, zobrazené hodnoty jedného stĺpca. Zobrazený stĺpec je možné nastaviť v [Options](#).

Options

Dialógové okno



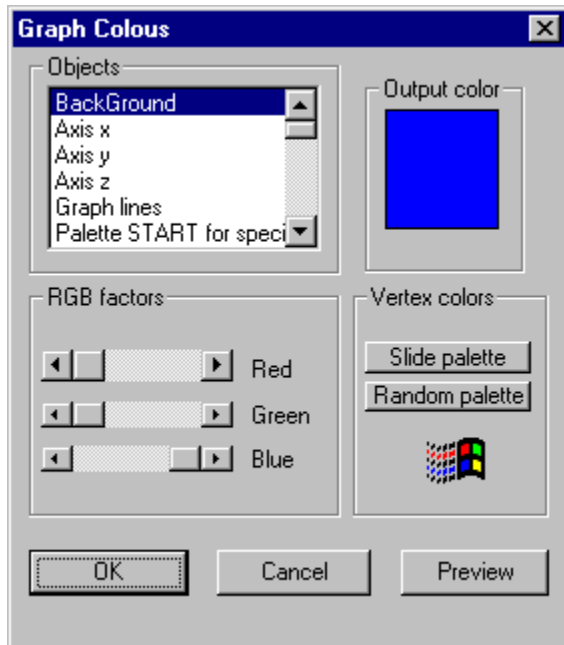
umožňujúce nastavenie grafu v kolonkách:

- **Axis Description**
 - **Regular** a **No Description** - popis osí Áno / Nie
 - **Axis X, Y, Z** - zobrazenie osi x,y,z
 - **Description X, Y, Z** - popis vybraných osí
 - **Wall XY, YZ, XZ** - farba na jednotlivých stenách (Bar 3d)
 - **Font** - nastavenie fontu osí
- **Vertex Description** - pri 2D grafoch (nie multi grafy) a koláèových grafoch je možné používať popisové rámčeky, ktoré obsahujú: **názov vrcholu**, (pri koláèovom grafe **percentá**), **hodnotu vrcholu**. **Názov grafu používa rovnaké nadstavenia ako popis vrcholov**.
 - **Pad** - popis vrchola v rámčeku
 - **Pad Background** - farebné pozadie rámčeka
 - **Vertex line** - čiara, ktorá spája vrchol a rámček (ak sa posunie)
 - **Text** - názov vrcholu.
 - **Percentage** - percentuálne vyjadrenie hodnoty rezov pri koláèovom grafe.
 - **Value** - hodnota vrcholu.
 - **Graph lines** - obrysy vrcholov.
 - **Font** - výber fontu pre popis vrcholov.
- **Line Width** - šírka čiary použitých v obrysoch grafu

- **Bold Radius** - veľkosť bodov v grafe **Simple Line** a **Multi Line**
- **Graph radius** - Pri barových grafoch hodnota polomeru vrcholov, pri koláčových grafoch posun častí koláča od stredu.
- **Title of graph** - zobrazovanie názvu grafu.
- **Object Type** - typ použitých objektov
 - pri **Line** grafoch :
 - **Sphere** - guľka
 - **Triangle** - trojuholník
 - **Quad** - štvorec
 - pri **Bar 3D** grafe
 - **Box** - kváder
 - **Cylinder** - valec
 - **Pyramid** - kužeľ
 - **Spiral** - špirála
 - **Flower** - kvetinka
- **Coloumn listbox** - označuje stápec dát, ktorý sa má zobraziť. Pri **Bar 3D**, **Multi Bar** a **Multi Line** grafoch je tato položka ignorovaná - tieto grafy zobrazujú všetky stápece.
- **Axis Division** - na koľko častí majú byť rozdelené (a hodnotami popísané) osi
- **Scrollbar**
 - Intenzita svetla (Ikona so slniečkom)
 - Priesvitnosť materiálu (Ikona s tromi kruhmi RGB)
 - Perspektíva (Ikona s kockou)
- **Tlačítka**
 - **Default Pad Position** - nastaví všetky popisové rámčeky na pozície vrcholov ku ktorým patria - platí iba pre aktuálny stápec.
 - **Graph type**
 - **Colours**
 - **Preview** - render grafu.

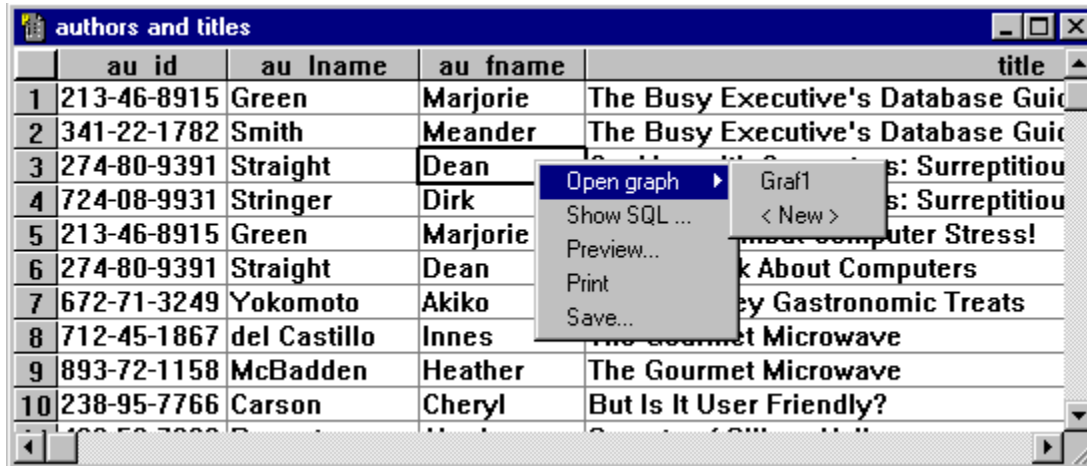
Colors

Dialógové okno, v ktorom je možné nastaviť farby grafu



- umožňuje nastaviť farby pre :
 - **Background** - pozadie
 - **Axis X, Y, Z** - jednotlivé osi grafu
 - **Graph Lines** - čiary grafu
 - **Palette START** - vii **Slide Palette**
 - **Palette END**
 - **Text** - fara textu
 - **Remark Background** - pozadie rámečkov s popismi vrcholov.
 - **Wall 1, 2, 3** - jednotlivé steny Bar 3D grafu
 - **Vertex 1, ..., N** - jednotlivé vrcholy.
- farby pre jednotlivé vrcholy je samozrejme zdâhavé urèovať jednotlivô, preto je možné použiť funkcie
 - **Slide Palette** - vygeneruje plynulý prechod od **Palette START** po **Palette END**
 - **Random Palette** - vygeneruje náhodné farby pre vrcholy

Okno s dátami zostavy



	au id	au lname	au fname	title
1	213-46-8915	Green	Marjorie	The Busy Executive's Database Guide
2	341-22-1782	Smith	Meander	The Busy Executive's Database Guide
3	274-80-9391	Straight	Dean	Surreptitious Database Access: The Art of the Computer Hack
4	724-08-9931	Stringer	Dirk	Computer Stress!
5	213-46-8915	Green	Marjorie	How to Succeed in Computer Business: Real-World Lessons from the Experts
6	274-80-9391	Straight	Dean	Ask About Computers
7	672-71-3249	Yokomoto	Akiko	Key Gastronomic Treats
8	712-45-1867	del Castillo	Innes	The Gourmet Microwave
9	893-72-1158	McBadden	Heather	The Gourmet Microwave
10	238-95-7766	Carson	Cheryl	But Is It User Friendly?

Tabuľkové vlastnosti

Tabuľka vykazuje väčšinu štandardných tabuľkových vlastností, ako je

- označovanie stĺpcov / riadkov / obdĺžnikových oblastí buniek
- funkcionálna **clipboard**
- rozširovanie/zužovanie riadkov/stĺpcov
- triedenie stĺpcov dvojkliknutím v hlavičke tabuľky

Shortcut menu

Stlačením pravého tlačítka myši sa zobrazí menu s nasledujúcimi funkciami

- **Open Graph** – obsahuje submenu s názvami **grafov**, ktoré sú vytvorené pre zostavu a príkaz **< New >**. Voľbou niektorého z existujúcich grafov sa **graf zobrazí**. Nemusí sa to vždy podariť, lebo stĺpce zostavy, ktoré graf zobrazoval mohli byť medzičasom zo zostavy zrušené. **Patrik** v takomto prípade vypíše varovné hlásenie a umožní invalidný graf **zmazať**. Voľbou **< New >** sa zobrazí nový graf. Ak boli pred zvolením funkcie niektoré stĺpce zostavy označené, tak iba tie budú zobrazené v grafe. Ak neboli označené žiadne stĺpce zostavy, tak sa v grafe zobrazia všetky jej stĺpce. Rozlišujú sa dva typy stĺpcov – **číselné** a **textové**. **Číselné** stĺpce predstavujú vektory s dátami a **textové** reprezentujú názvy bodov. V prípade že sa má v grafe zobraziť viacero textových stĺpcov, tak sa ich hodnoty (texty) pospájajú do jedného textu (s medzerou medzi nimi).
- **Show SQL** – zobrazí SELECT uskutočňujúci výber dát pre zostavu
- **Preview** – nastavenie tlače zostavy
- **Print** – tlače zostavy
- **Save** - uloží dáta zostavy do súboru, ako tabuľkou delimitovaný text.

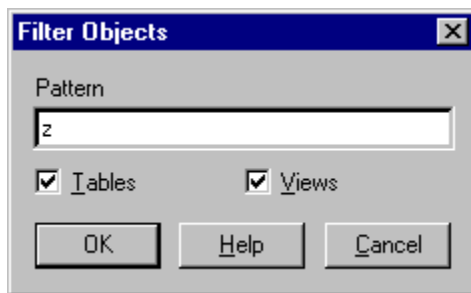
Príkaz Edit filter

Použitím príkazu **Edit Filter** je možné prezrieť len tie tabučky, o ktoré máte záujem. Príkaz **Edit Filter** slúži na filtrovanie databázových objektov tiež v skupinách objektov **Code** a **Report**.

Filtrovanie objektov v skupine objektov Data

1. Zvoľte príkaz **Edit Filter** z **Shortcut menu** browsera.

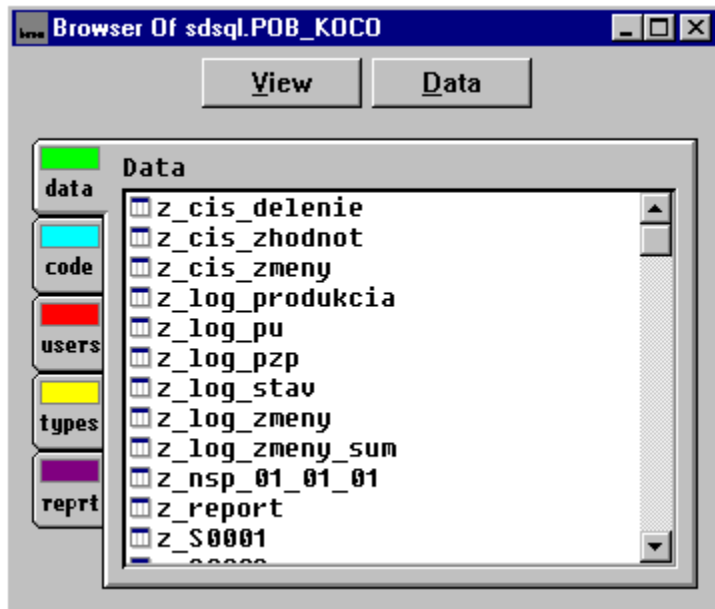
Zobrazí sa dialógové okno **Filter Objects**.



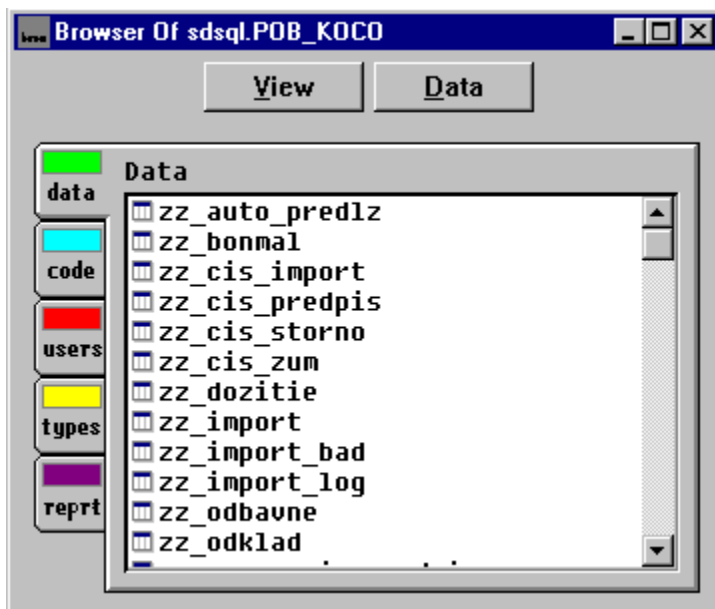
3. Do poľa **Pattern** napíšte prvé písmená (prefix) objektov, ktoré majú byť zobrazené v okne browsera.
4. Vyberte check boxu tých typov objektov, ktoré chcete zobrazovať v okne browsera.
5. Stlačte tlačítko **OK**.

Príklad.

Napríklad, ak napíšete “z” v poli **Pattern**, **PATRIK** bude v okne browsera zobrazovať iba tabučky, ktorých názov začína písmenom “z”.





Napríklad, ak napíšete “zz” v poli Pattern, PATRIK bude v okne browsera zobrazovať iba tabuľky, ktorých názov začína písmenom “zz”.



Zrušenie definovaného filtra

Aksi nejjednoduchšie je zatvoriť a opäť otvoriť **Browser** pre danú databázu.

Data - ako skupina objektov v Browsri.

Táto kapitola opisuje možnosti browsovania štruktúry dátových objektov - **tabuliek** () a **view-ov** () (Ďalej len tabuliek, lebo aj tak sa najčastejšie jedná o tabuľky), tvorenie a editovanie poznámok k tabuľkám. Ďalej sa dozviete ako sa dajú pohodlne editovať a browsovať údaje v tabuľkách databázy.

Tlačítka okna sa líšia v závislosti od typu browsovanej databázy.

- Pri browsovaní **skutočnej** databázy **SQL servra** sa nachádzajú v okne tlačítka
 - **View** - zobrazí **definíciu (štruktúru) označenej tabuľky**, alebo označeného **viewu**.
 - **Data** - zobrazí okno s dátami označenej tabuľky
- Pri browsovaní **pracovnej (downsized)** databázy sa v okne nachádzajú tlačítka
 - **New** - vytvorenie novej tabuľky (nie view-u)
 - **Design** - editor štruktúry tabuľky (nie view-u). **View-y je možné editovať a vytvárať iba v skutočnej databáze.**

Shortcut menu ponúka prídavné funkcie na prácu s tabuľkami. Je odlišné v browseri **skutočnej** a **pracovnej (downsized)** databázy.

Shortcut Menu skutočnej databázy obsahuje funkcie :

- **Edit Filter...** - nastavenie podmnožiny tabuliek, ktorú chcete v browseri zobrazovať.
- **Drop Object...** - zmazanie označenej tabuľky
- **Edit Notes** - editácia poznámok k označenej tabuľke.
- **View All Notes** - zobrazenie poznámok všetkých užívateľov k označenej tabuľke.
- **Save Data...** - uloženie dát označenej tabuľky do súboru
- **Load Data...** - načítanie dát do označenej tabuľky zo súboru
- **Select COUNT(*)** - diagnostikovanie veľkosti označenej tabuľky
- **Last Line WHERE...** - zobrazenie posledného záznamu tabuľky
- **WHERE Open...** - zobrazenie dát tabuľky splývajúcich zadanú WHERE podmienku.

Shortcut menu browsera **pracovnej (downsized)** databázy neobsahuje položky narábajúce s dátami tabuliek (**Save Data...**, **Load Data...**, **Select Count(*)**, **Last Line WHERE** a **WHERE Open**), lebo tabuľky pracovnej databázy nikdy neobsahujú dáta. Do menu je pridaná funkcia

- **Clone table** - replikácia označenej tabuľky

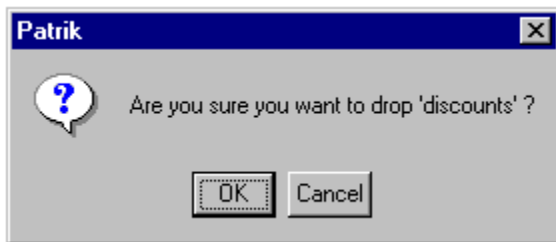
Príkaz Drop object

Tento príkaz slúži na vymazanie vybraného objektu.

Dropnutie objektu

1. Označíte objekt, ktorý chcete vymazať z databázy, v okne **browsra**. Stlačením pravého tlačítka myši vyvolajte **Shortcut Menu**.
2. Zvoľte príkaz **Drop Object** z ponuky **Shortcut menu**.

Zobrazí sa nasledujúce dialógové okno.



3. Stlačte tlačítko **OK** a potvrdíte rozhodnutie dropnúť objekt.

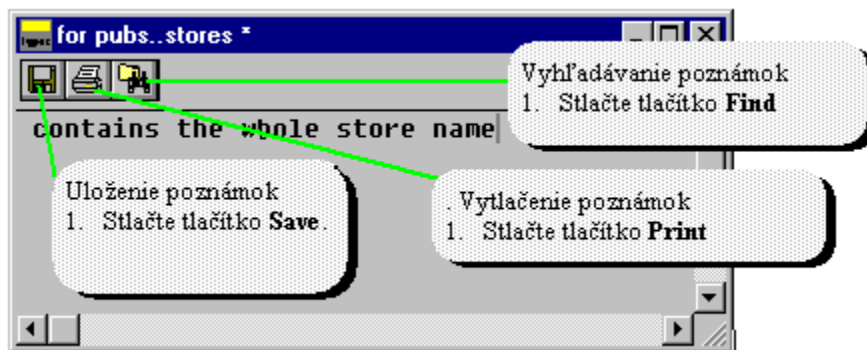
Príkaz edit notes

Ku každému databázovému objektu môžete pridať svoju poznámku.

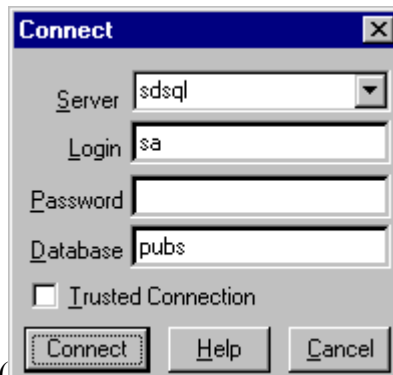
Pridávanie poznámok k databázovému objektu

1. Označíte objekt, ku ktorému chcete zadať poznámku (prípadne zmeniť, už vytvorenú poznámku), v okne **browsera**. Stlačením pravého tlačítka myši vyvolajte **Shortcut Menu**.
2. Zvoľte príkaz **Edit Notes** z ponuky **Shortcut menu**.

Zobrazí sa okno **Edit Notes for názov_tabuľky**.



3. Napíšte novú alebo zmeňte vytvorenú poznámku.



4. Stlačte v **toolbar** tlačítko **Save** (akejkoľvek odozvy uloží). Poznámka sa bez

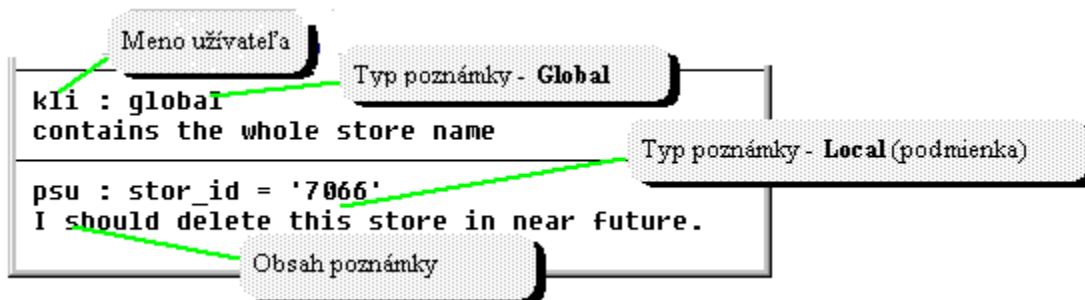
Zobrazenie všetkých poznámok k objektu

Je možné zobrazíť všetky poznámky, t.j. aj poznámky aj ostatných užívateľov, ale editovať len svoje vlastné.

Zobrazenie všetkých poznámok

1. Označíte objekt, pre ktorý chcete zobrazíť poznámky, v okne **browsersa**. Stlačením pravého tlačítka myši vyvolajte **Shortcut Menu**.
2. Zvoľte príkaz **View All Notes** z ponuky **Shortcut menu**.

Zobrazí sa okno s poznámkami. Toto okno je typu Results.



Príkaz Save data

Údaje vybratej tabuľky môžete uložiť a neskôr nahráť. Napríklad v prípade, že skúšobne meníte údaje v tabuľke, ale zmenu nechcete uskutočniť.

Uloženie údajov.

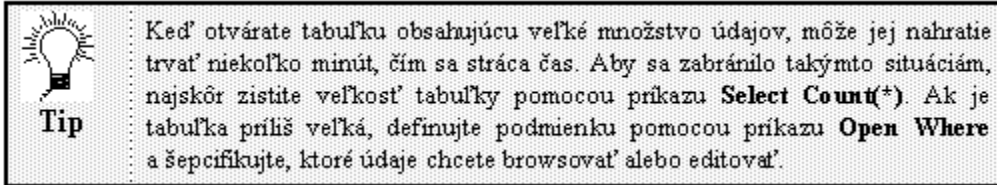
1. Označíte tabuľku, ktorej dáta chcete uložiť do súboru, v okne **browsers**. Stlačením pravého tlačítka myši vyvolajte **Shortcut Menu**.
2. Zatvorte príkaz **Save Data...** z ponuky **Shortcut menu**. Zobrazí sa dialógové okno **Save As**.
3. Napíšte názov súboru, vyberte diskovú jednotku a adresár, do ktorého chcete súbor uložiť.
PATRIK súbor uloží s príponou **.tab**.

Príkaz Load data

Nahratie uložených údajov zo súboru (súbor musel byť vytvorený funkciou Save Data...) do tabuľky.

1. Označíte tabuľku, do ktorej chcete načítať dáta zo súboru, v okne **browsersa**. Stlačením pravého tlačítka myši vyvolajte **Shortcut Menu**.
2. Zvoľte príkaz **Load Data** z ponuky **Shortcut menu**.
Zobrazí sa dialógové okno **Open**.
3. Vyberte súbor, ktorý chcete nahráť do aktuálnej tabuľky.
4. Stlačte tlačítko **OK**.

Príkaz Select count(*)



Na zistenie veľkosti tabuľky je možné použiť príkaz **Select Count(*)**, aby ste zistili veľkosť vybratej tabuľky.

Zistenie veľkosti tabuľky

1. Označíte tabuľku, ktorej veľkosť chcete zistiť, v okne **browsra**. Stlačením pravého tlačítka myši vyvolajte **Shortcut Menu**.
2. Zvoľte príkaz **Select Count(*)** z ponuky **Shortcut menu**. PATRIK zobrazí počet riadkov a veľkosť tabuľky v kilobytoch v **Informačnom pruhu** v hlavnom okne aplikácie.

```
Table fe_cinnost has 623 rows, used 746 KB
```

Príkaz WHERE Open

Zobrazenie údajov tabuľky splôujúce Vami definovanou WHERE podmienkou.

1. Označíte tabuľku, ktorej podmnožinu záznamov chcete zobraziť, v okne **browsra**. Stlačením pravého tlačítka myši vyvolajte **Shortcut Menu**.
2. Zvoľte príkaz **WHERE Open...** z ponuky **Shortcut menu**. Zobrazí sa dialógové okno **Type Where**.



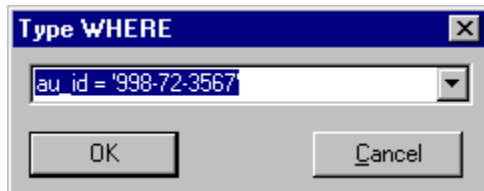
3. Zadajte WHERE podmienku.
4. Stlačte tlačítko **OK**.

PATRIK zobrazí záznamy tabuľky spôôajúce zadanú WHERE podmienku v novootvorenom okne.

Príkaz Last Line WHERE

Zobrazenie posledného riadku tabuľky - funguje iba pre tabuľky s primárnym kľúčom.

1. Označíte tabuľku, ktorej posledný záznam chcete zobraziť, v okne **browsers**. Stlačením pravého tlačítka myši vyvolajte **Shortcut Menu**.
2. Zvoľte príkaz **Last Line WHERE...** z ponuky **Shortcut menu**. Zobrazi sa dialógové okno **Type Where**. V ktorom je predpísaná WHERE podmienka, ktorú spáda jediný záznam tabuľky. Je to záznam obsahujúci maximálne hodnoty v stĺpcoch primárneho kľúča - väčšinou sa tabuľky plnia tak, že sa vkladajú do stĺpca primárneho kľúča hodnoty (MAX + 1), takže uvedená podmienka bude pravdepodobne podmienka pre posledný riadok tabuľky:



3. Stlačte tlačítko **OK**.

PATRIK zobrazí posledný záznam tabuľky v novootvorenom okne.

Príkaz Clone table

Príkaz **Clone Table** sa používa na rýchle vytvorenie štruktúry tabučky, ktorá je podobná alebo identická s už existujúcou tabučkou.

Klonovanie tabučky

1. Označíte tabučku, ktorú chcete klonovať, v okne browsra. Stlačením pravého tlačítka myši vyvolajte **Shortcut Menu**.
2. Zvoďte príkaz **Clone Table**.

Zobrazí sa dialógové okno **Clone To**.



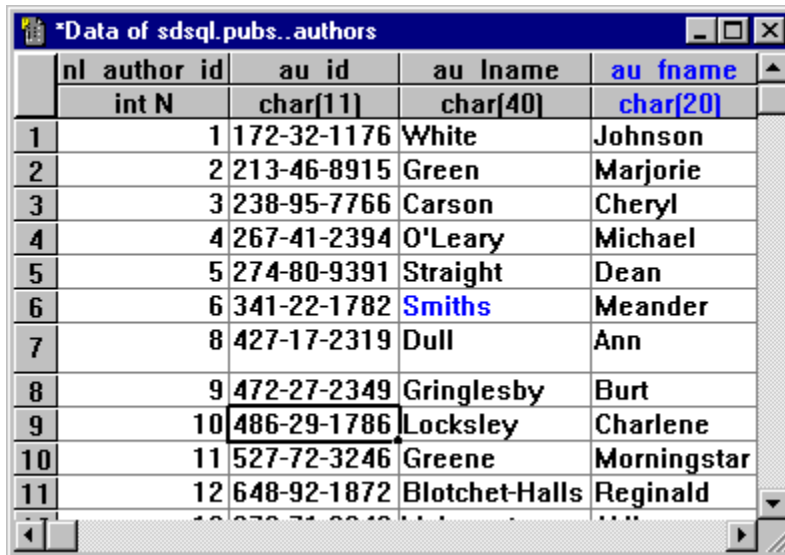
5. Napíšte názov novej tabučky a stlačte tlačítko **OK**.

PATRIK otvorí okno **Design Table názov novej tabučky**, kde je možné editovať štruktúru novej tabučky.

Tabuška s Dátami SQL tabušky.

Vešké tabušky.

V prípade že sú v okne načítané údaje rozsiahlej SQL tabušky, tak je pravdepodobné, že sa v okne nenachádzajú všetky záznamy danej tabušky. Načítavaná časť tabušky je ohraničená veľkosťou nastavenou v kolonke Memory Usage v Options. V prípade že nie je tabuška načítaná celá, tak sa nachádza v spodnej lište aplikácie (v jej ľavej časti) znak *. V takomto prípade je rozumé použiť funkciu Edit WHERE, kde špecifikujete podmienku, ktorú budú spádať len záznamy tabušky, ktoré chcete vidieť.



nl	author id	au id	au lname	au fname
	int N	char[11]	char[40]	char[20]
1	1	172-32-1176	White	Johnson
2	2	213-46-8915	Green	Marjorie
3	3	238-95-7766	Carson	Cheryl
4	4	267-41-2394	O'Leary	Michael
5	5	274-80-9391	Straight	Dean
6	6	341-22-1782	Smiths	Meander
7	8	427-17-2319	Dull	Ann
8	9	472-27-2349	Gringlesby	Burt
9	10	486-29-1786	Locksley	Charlene
10	11	527-72-3246	Greene	Morningstar
11	12	648-92-1872	Blotchet-Halls	Reginald

Titulok okna

Názov okna obsahuje popis SQL tabušky, ktorej dáta sú v okne zobrazené, v tvare : **Data of <server>.<databáza>.<tabuška>**. Ak okno obsahuje iba časť tabušky, splývajúcu WHERE podmienku, tak je aj táto podmienka uvedená v názve okna.

Hlavička tabušky

Prvý riadok tabušky (so šedým pozadím) obsahuje mená stĺpcov zobrazenej SQL tabušky. Ak je v Options nastavené **Show Columns Datatype**, tak hlavička tabušky obsahuje ešte aj druhý riadok, pozostávajúci s dátových typov jednotlivých stĺpcov tabušky. Za menom každého dátového typu sa môžu vyskytovať znaky:

- D - ak má stĺpec tabušky nastavený Default.
- N - ak stĺpec tabušky môže obsahovať NULL hodnoty

Číslo riadkov tabušky

SQL tabuška pochopiteľne nemá riadky číslované, ale pre väčší prehľad v tabuške a kôli funkcionalite **zmeny výšky riadkov** a **označovanie riadkov**, sú pred jednotlivými riadkami tabušky uvedené poradové čísla riadkov.

Triedenie údajov

Tabušku je možné utriediť podľa ľubovoľného stĺpca - dvojkliknutím na jeho **hlavičku**.

Èíselné stápcy sa triedia podľa veľkosti, dátumové podľa časového sledu dátumov a textové podľa abecedy. Záznamy sa utriedia vzostupne. Opakovaným dvojkliknutím na hlavičku stápcy sa záznamy utriedia zostupne.

Oznaèovanie riadkov, stápcov alebo výseku buniek

Niektoré funkcie (napríklad clipboardové operácie) pracujú s oznaenými riadkami/stápcami/bunkami.

Riadičky sa oznaujú kliknutím ľavého tlačítka myši na ich èíslo. Na oznaenie viacerých riadkov slúži klávesa **Ctrl** a na oznaenie intervalu riadkov slúži klávesa **Shift** - tak ako to štandardne funguje vo **Windows**.

Stápcy sa oznaujú kliknutím ľavého tlačítka myši na ich hlavičku. Na oznaenie viacerých stápcov slúži klávesa **Ctrl** a na oznaenie intervalu stápcov slúži klávesa **Shift** - tak ako to štandardne funguje vo **Windows**.

Výsek buniek sa oznauje kliknutím myši so stlaèeným ľavým tlačítkom, alebo klávesami **Shift+šípky**.

Oznaenie celých riadkov pre riadky, v ktorých je oznaený výsek buniek sa vykonáva klávesou **F6**. Niekedy je totiž nepohodlné siahnuť po myši a oznaovať riadky, preto sa za pomoci klávesnice oznaí **Výsek buniek** oznaujúci riadky, ktoré chcete oznaíť a stlaèí sa **F6**.

Oznaenie všetkých riadkov a všetkých stápcov sa uskutoòuje klávesami **Ctrl A**.

Zmena šírky stápcov a výšky riadkov je identická ako v tabuľkách vo **Windows**.

Zmena dát v tabuľke

V tabuľke je možné meniť hodnoty záznamov tabuľky (**UPDATE**), mazať oznaené riadky klávesou **Del** (**DELETE**) a meniť hodnoty neobsadených buniek, čím pridáte záznamy do tabuľky (**INSERT**). Na orientáciu v texte slúži kurzor (obdážnik so širším rámom) ktorým je možno pohybovať šípkami na klávesnici a klávesami **Tab** a **Shift Tab**, prípadne kliknutím myšou na žiadané miesto. Klávesou **F2** je možné prepnúť kurzor do editovacieho módu a zmeniť tak obsah bunky. Obsahy modifikovaných buniek sú zobrazené modrou farbou, aby ste mali prehľad, èo ste zmenili. Zmeny ktoré vykonávate v tabuľke sa nevykonávajú priamo aj v **SQL** tabuľke. Tieto sa všetky vykonajú až voľbou funkcie **Update changes** v **shortcut menu**. Po aplikácii zmien sa dáta tabuľky opätovne naèítajú z databázy.

Dlhé textové polia

Dáta v stápcoch typu dátového typu **text** alebo **image** môžu byť dosť rozsiahle. Patrik naèítava iba ich prvé **4KB**. Nedoèítané bunky (väčšie ako **4KB**) sú v tabuľke zobrazené **šedou farbou**. Ak je na takejto bunke nastavený kurzor, tak stlaèením klávesy **F3** sa naèíta celý obsah bunky a plocha bunky sa dočasne (pokiaľ nestlaèíte jednu s kláves **Esc** - **undo**, **F3** - zrušenie veľkého pohľadu, **F2** - koniec editácie podľa) zväčší na **veľkosť celého okna**, čím je možné pohodlne prehliadnuť / editovať dlhý text.

Príznak modifikácie

Ak zmeníte údaje v tabuľke, **PATRIK** zobrazí symbol * v ľavom rohu jej názvu.

Replikácia vyplnených buniek.

Ak vkladáte do tabuľky nové riadky, ktoré sa do istej miery budú podobať poslednému riadku,

tak je možné označiť **výsekom** úsek buniek riadku a potiahnutím guličky v pravej dolnej časti textového kurzoru myšou do ľavého dolného rohu novovytváraných riadkov sa text označených buniek zreplikuje do všetkých novovytváraných riadkov.

Clipboard + import/export s MS Office

V tabuľke je možné používať všetky funkcie clipboardu :

- Cut - Ctrl X
- Delete - Del
- Copy - Ctrl C
- Paste - Ctrl V

Pomocou Clipboardu sa dajú kopírovať dáta medzi Patrikom a tabuľkami v **MS Office-e**.

Zmena poradia stĺpcov.

Funkčnosťou **drag and drop** je možné presúvať stĺpce v rámci tabuľky nasledovne:

1. Označíte stĺpec (viacero stĺpcov vedľa seba), ktorý chcete presúvať a stlačíte ľavé tlačidlo myši.
2. Presunúte myš na miesto stĺpca, pred ktorý chcete presúvaný stĺpec vložiť.
3. Pustíte ľavé tlačidlo myši.

Shortcut menu

Ďalšie funkcie, ktoré je možné vykonávať v tabuľke sú začlenené do Shortcut Menu .

- Related Table - zobrazenie relačných záznamov tabuľky, ktorá má reláciu s aktuálnou tabuľkou

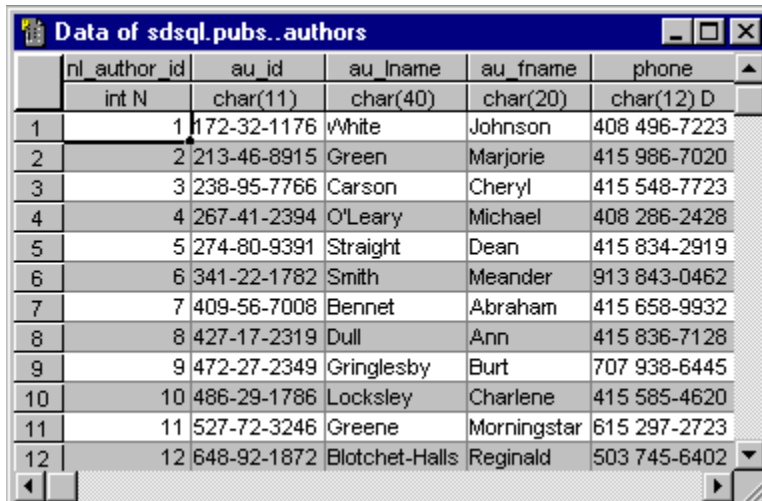
Nasledovné dve položky sa zobrazia v menu iba ak bolo vyvolané nad stĺpcom, ktorý je cudzineckým kľúčom (foreign key) nejakého jednoduchého (teda nie zloženého) relačného vzťahu. Tabuľku na primárnej strane relácie označujeme ako **číselník**.

- Open <meno tabuľky> - Otvorí tabuľku - číslo <meno tabuľky> naviazanú na aktuálny stĺpec.
- Expand <meno tabuľky> - nahradí (zväčša číselné) hodnoty v aktuálnom stĺpci textovými hodnotami s príslušného číselníku.
- Edit WHERE... - filtrácia zobrazených dát
- Refresh - opätovné načítanie dát tabuľky
- Edit Notes - vytvorenie poznámky k tabuľke
- View All Notes - zobrazenie poznámok k tabuľke
- Save... - uloží dáta tabuľky do súboru ako tabuľkou delimitovaný text.
- Save Table Options - uloží vzhľad tabuľky (autofilter, where, expand, poradie a šírka stĺpcov) ako jej default vzhľad.
- Record Script... - logovanie zmien v tabuľke
- Update Changes - aplikácia vykonaných zmien v dátach do SQL tabuľky
- Auto Filter <meno stĺpca> - pohodlná filtrácia dát stĺpca. Funkcia sa nachádza v menu iba ak sa pri jeho vyvolaní nachádzala myš na hlavičke nejakého stĺpca.

Výzor tabuľky

V Options je možné nastaviť nasledovné prvky výzoru tabuliek.

- Horizontal Stripes - ak je nastavené, budú párne záznamy v tabuľkách zobrazované s tmavšou farbou pozadia
- Change Font - otvorí dialógové okno v ktorom je možné nastaviť font v tabuľkách.



The screenshot shows a window titled "Data of sdsq1.pubs..authors". The window contains a table with 12 rows and 5 columns. The columns are labeled: "nl_author_id", "au_id", "au_lname", "au_fname", and "phone". The data is as follows:

nl_author_id	au_id	au_lname	au_fname	phone
1	1	White	Johnson	408 496-7223
2	2	Green	Marjorie	415 986-7020
3	3	Carson	Cheryl	415 548-7723
4	4	O'Leary	Michael	408 286-2428
5	5	Straight	Dean	415 834-2919
6	6	Smith	Meander	913 843-0462
7	7	Bennet	Abraham	415 658-9932
8	8	Dull	Ann	415 836-7128
9	9	Gringlesby	Burt	707 938-6445
10	10	Locksley	Charlene	415 585-4620
11	11	Greene	Morningstar	615 297-2723
12	12	Blotchet-Halls	Reginald	503 745-6402

Save Table Options

Funkcia **Save Table options** vyvolaná so shortcut menu uloží informácie o výzore tabuľky spolu s **menom užívateľa** (NT login) do databázy **patrik** . Pri nasledujúcom otvorení dát danej tabuľky (tým istým užívateľom) sa otvorí tabuľka uloženého výzoru.

Informácie o výzore tabuľky zahŕňajú:

- autofiltre na všetky stĺpce
- Expand stĺpcov
- WHERE podmienku pre tabuľku
- poradie stĺpcov (v prípade, že bolo toto zmenené)
- šírku stĺpcov (v prípade, že bola zmenená s implicitne **best to fit**)

Ignorácia uloženého výzoru

V prípade, že máte uložený výzor pre nejakú tabuľku chcete otvoriť tabuľku so štandardným výzorom, otvorte ju stlačením tlačítka Data v Browsri za držania stlačenej klávesy **Ctrl** .

Ukladanie informácií

Informácie sa ukladajú do tabuľky **patrik.tabexpand** .

Expand <meno tabu³ky>

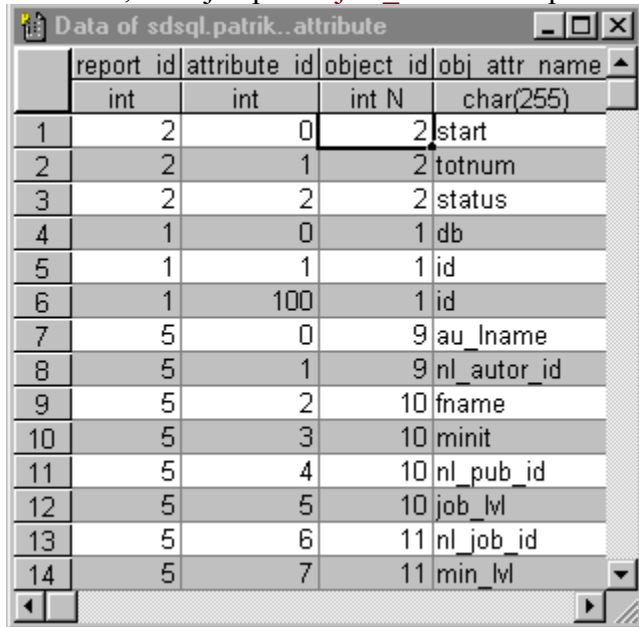
Èíselníky

V relaèných databázach sa èasto používajú tabu³ky (èíselníky), ktoré vznikli dôsledkom dekompozície relácií kôli zvýšeniu priestorovej efektivity. Èíselníky obsahujú zväèša nejaký èíselný a textový stápec, kde text v textovom stápci popisuje èíselné hodnoty, ktoré sa zvyknú potom používaŕ v ostatných tabu³kách naviazaných na èíselníky vzŕahom N-1.

Ak je na aktuálny stápec naviazaný nejaký èíselník, tak shortcut menu vyvolané nad týmto stápcom obsahuje položku **Expand <meno èíselníka>**. Zvolením tejto položky sa otvorí dialógové okno s jediným comboboxom obsahujúcim mená všetkých stápcov èíselníka. Pokia³ nejaký z názvov stápcov obsahuje substring “name”, “meno”, alebo “popis”, tak sa tento stápec automaticky v comboboxe vyberie. V comboboxe by mal užívate³ vybraŕ textový stápec èíselníku. Stlaèením tlačítka OK po vykonaní výberu, sa hodnoty v aktuálnom stápci tabu³ky nahradia hodnotami s vybraného stápcu èíselníka. Editovanie hodnôt v aktuálnom stápci tabu³ky bude od tohoto okamihu prebiehaŕ v comboboxe (teda už nie v editboxe) obsahujúcom všetky texty s popisného stápcu èíselníka.

Najlepšie bude asi uviesŕ príklad:

Tabu³ka, ktorej stlpec **object_id** ideme expandovaŕ



	report id	attribute id	object id	obj attr name
	int	int	int N	char(255)
1	2	0	2	start
2	2	1	2	totnum
3	2	2	2	status
4	1	0	1	db
5	1	1	1	id
6	1	100	1	id
7	5	0	9	au_lname
8	5	1	9	nl_autor_id
9	5	2	10	fname
10	5	3	10	minit
11	5	4	10	nl_pub_id
12	5	5	10	job_lvl
13	5	6	11	nl_job_id
14	5	7	11	min_lvl

Èíselník

Data of sdsq1.patrik..object				
	object id	report id	type	table name
	int	int	int	char(255)
1	1	1	0	keysadd
2	2	2	0	spt_committab
3	6	1	0	report
4	7	4	2	keysadd
5	8	4	0	graph
6	9	5	0	authors
7	10	5	0	employee
8	11	5	0	jobs

Výsledok

Data of sdsq1.patrik..attribute				
	report id	attribute id	object id	obj attr name
	int	int	int N	char(255)
1	2	0	spt_committab	start
2	2	1	spt_committab	totnum
3	2	2	spt_committab	status
4	1	0	keysadd	db
5	1	1	keysadd	id
6	1	100	keysadd	id
7	5	0	authors	au_lname
8	5	1	authors	nl_autor_id
9	5	2	employee	fname
10	5	3	employee	minit
11	5	4	employee	nl_pub_id
12	5	5	employee	job_lvl
13	5	6	jobs	nl_job_id
14	5	7	jobs	min_lvl

Uloženie dát na disk

Údaje aktívnej tabuľky môžete uložiť do súboru s koncovkou **.txt** ako tabulátorom delimitovaný text. Text je možné importnúť v programe MS Excel.

Uloženie údajov tabuľky do súboru

1. Otvorte sa **Shortcut menu**.
2. Zvoľte príkaz **Save** z ponuky **Shortcut menu**.
Zobrazí sa dialógové okno **Save As**.
3. Napíšte názov súboru, vyberte diskovú jednotku a adresár, do ktorého chcete súbor uložiť.
PATRIK uloží súbor s koncovkou **.txt**.

Menu Related Table

Príkaz **Related Table** slúži na zobrazenie dát tabuľkí, ktoré sú v relácii s aktuálnou tabuľkou. Záznamy týchto tabuľkí sa nezobrazia všetky, ale iba tie, ktoré sú napojené na označené záznamy aktuálnej tabuľky.

Menu Related Table obsahuje submenu obsahujúce riadok pre každú tabuľku, ktorá je s aktuálnou tabuľkou spojená reláciou. Ak existujú aspoň dve tabuľky napojené na aktuálnu tabuľku reláciou 1->n, tak menu ešte obsahuje príkaz **All 1->n** (rovnako je to s n->1). Voľbou niektorého z mien tabuľkí sa zobrazia záznamy danej tabuľky relačne súvisiace s označenými záznamami aktuálnej tabuľky. Voľbou príkazu **All 1->n (n->1)** sa zobrazia záznamy všetkých tabuľkí spojených s tabuľkou reláciou 1->n (n->1), relačne súvisiace s označenými záznamami aktuálnej tabuľky.

PATRIK otvorí okno *Data of názov_servera.názov_databázy..názov_tabuľky WHERE foreign (primary) key value=primary (foreign) key - hodnota označených záznamov v aktuálnej tabuľke*, obsahujúce údaje relačnej tabuľky odkazujúce na predtým označený riadok (riadky) v aktuálnej tabuľke.

Aplikovanie zmien vykonaných v tabuľke na SQL databáze

Ak zmeníte údaje v tabuľke, PATRIK zobrazí symbol * v ľavom rohu jej názvu - je nastavený príznak modifikácie.

Uloženie zmien údajov na SQL server sa vykoná príkazom Update Changes v shortcut menu.

Nie všetky zmeny v tabuľke sa vždy uložia. Ak ste vykonali nejakú zmenu porušujúcu referenčnú, doménovú, alebo entitnú integritu v databáze, tak sa vypíše príslušné chybové hlásenie a zmena sa nevykoná.

Po vykonaní zmien v SQL tabuľke sa opätovne načítajú záznamy SQL tabuľky, čo má za následok zafarbenie všetkých modrých (predtým modifikovaných) buniek na čierne a vrátenie zmien, ktoré sa nevykonali. V tabuľke sa môžu objaviť riadky zobrazené červenou farbou, to sú také riadky, ktoré sa do SQL tabuľky nedali vložiť. Chyby v nich je možné opraviť a opätovne spustiť funkciu Update Changes.

Funkcia po dobehnutí zruší **príznak modifikácie**.

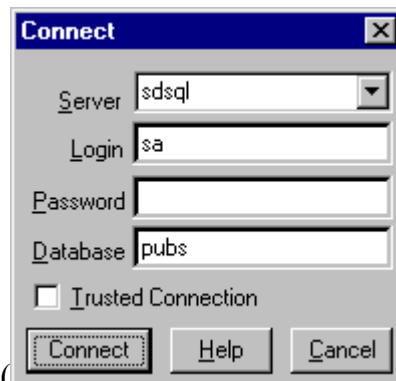
Príkaz Record script

Príkaz **Record script** môžete použiť na uloženie zmien, ktoré ste urobili v údajoch v databáze na jednom SQL serveri a implementovať ich do databázy na ďalšom SQL serveri, za predpokladu, že údaje v databázach na oboch SQL serveroch sú identické.

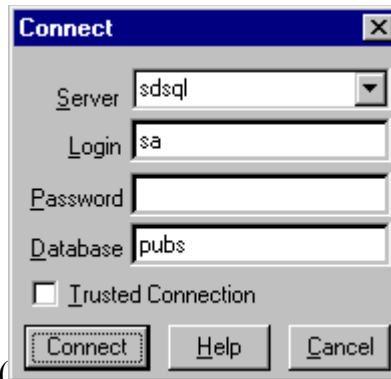
Uloženie zmien, ktoré sa vykonali v údajoch databázy v okne **Data of názov_servera.názov_databázy..názov_tabuľky**

1. Zvoďte príkaz **Record Script** z ponuky **Shortcut menu**.
Zobrazí sa dialógové okno **Save As**.
3. Napíšte názov skriptu, vyberte diskovú jednotku a adresár, do ktorého chcete skript uložiť.
4. Urobte potrebné zmeny v dátach tabuľky (tabuliek na danom SQL serveri).
5. Kliknite na pravé tlačítko myši a zvoďte príkaz **Update Changes** z ponuky **Shortcut menu**.
6. Kliknite na pravé tlačítko myši a zvoďte príkaz **Stop Recording** z ponuky **Shortcut menu**.
Tým sa súbor do ktorého sa ukladá skript s vykonávanými zmenami.

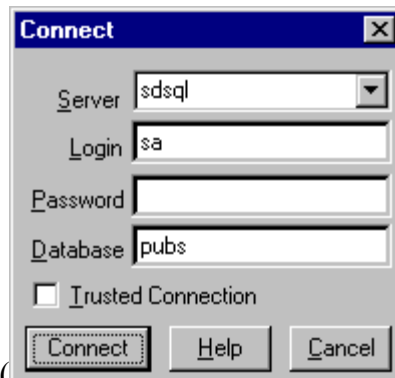
Implementovanie zmien v databáze na inom SQL Serveri



1. Stlačte tlačítko **Query Editor** (v toolbare hlavného okna aplikácie).
Zobrazí sa okno **Query Editor**.



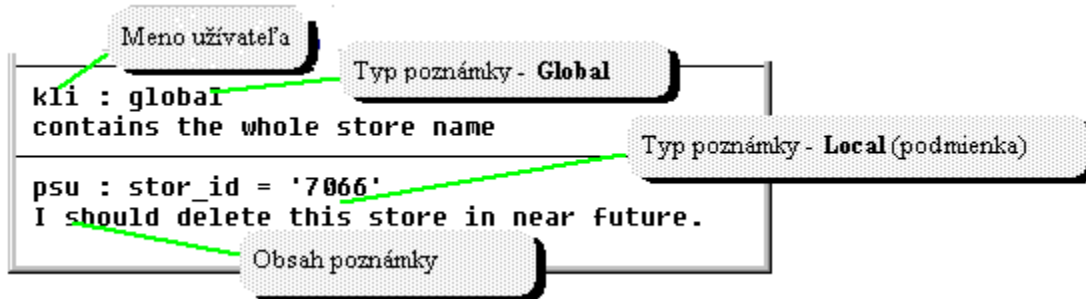
2. Zvo³4te v toolbar okna query Load (Zobrazí sa dialógové okno Open).
3. Vyberte skript, ktorý ste predtým uložili.



4. Stlaèete tlačítko Execute () v okne Query Editor.

Zobrazenie všetkých poznámok

Poznámky k aktuálnej tabuľke môžete zobraziť použitím príkazu **View My Notes** v shortcut menu. Zobrazí sa okno, v ktorom sú vypísané poznámky, ktoré kto kedy k tabuľke napísal. Okno s poznámkami je typu **Results**:

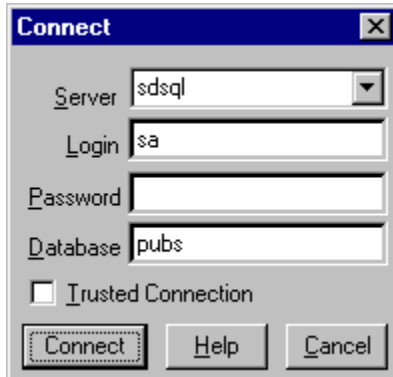


Editovanie poznámok

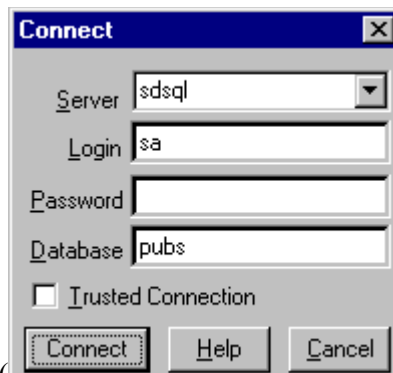
Použitím príkazu **Edit Notes** sa dajú editovať poznámky k tabuľke.

Editovanie poznámok vybratej tabuľky

1. Zvoľte príkaz **Edit Notes** z ponuky **Shortcut menu**.
Zobrazí sa okno **Edit Notes for *názov_tabuľky***.



3. Napíšte novú alebo zmeňte vytvorenú poznámku.



4. Stlačte v **toolbar** tlačítko **Save** (). Poznámka sa bez akejkoľvek odozvy uloží.

Príkaz Refresh

Ak pracujete s tabuľkou databázy, s ktorou pracuje niekto iný užívateľ, môže sa stať, že údaje, ktoré máte dlhšie zobrazené na obrazovke, sú staré (už neaktuálne).

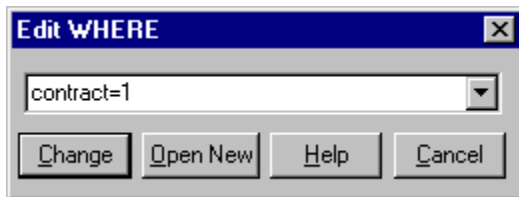
Funkcia **Refresh** v shortcut menu slúži na opätovné načítanie údajov zo SQL Servera

Príkaz Edit WHERE

Podmienku WHERE platnú pre záznamy tabuľky zobrazené v okne ja môžete zmeniť, pomocou príkazu **Edit WHERE** zo shortcut menu.

Zmenenie podmienky Where

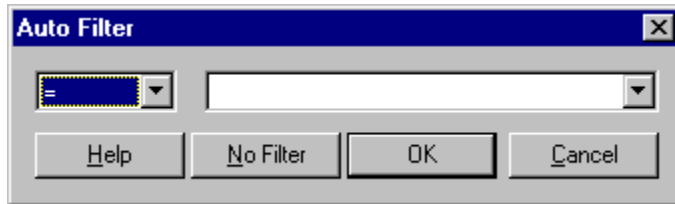
1. Zvoľte príkaz **Edit WHERE** z ponuky **Shortcut menu**. Zobrazí sa dialógové okno **Edit Where**.



2. Napíšte novú podmienku WHERE, alebo vyberte **cahe-ovanú** WHERE podmienku z combo boxu.
4. Urobte nasledovné:
 - a) Stlačte tlačítko **Change**, aby sa zobrazili údaje spôôajúce novú podmienku v aktuálne otvorenom okne tabuľky
 - b) Stlačte tlačítko **Open New**, aby sa zobrazili údaje spôôajúce novú podmienku do novootvoreného okna.

Auto Filter

Služi na pohodlnú filtráciu záznamov tabuľky. Funkcia sa vždy vyvolá pre nejaký stĺpec tabuľky. Po jej vyvolaní sa otvorí Dialógové okno



V tomto okne sa nachádzajú dva comboboxy

- **Operátory** - (=, <, >, <=, >=, <>, in, like) tu si zvolíte operátor, ktorý chcete použiť na definovanie filtra
- **Hodnoty** - tu sa nachádzajú v utriedenom poradí všetky hodnoty ktoré sa v okamžiku zavolania funkcie nachádzali v danom stĺpci. Je možné vybrať si nejakú hodnotu z nich, prípadne napísať nejakú inú hodnotu.

Po tom ako vyplníte oba comboboxy, tak stlačíte **OK** v dialógovom okne. Tabuľka sa opätovne načíta, ale budú v nej zobrazené iba filtrované záznamy - iba tie záznamy ktorých hodnoty v danom stĺpci spĺňajú zadanú podmienku filtra. Hlavička daného stĺpca bude zobrazovaná modrou farbou - čo označuje, že záznamy tabuľky sú filtrované podľa tohoto stĺpca. Filtre môže byť nastavené aj na viacerých stĺpcoch.

Zrušenie Filtra.

V prípade, že chcete aby na stĺpci nebolo nastavené filtrovanie, tak stlačíte tlačítko **No Filter**. V tomto prípade sa obsahy vyššie uvedených comboboxov neberú do úvahy.

Skupina objektov Code




Táto kapitola opisuje možnosti tvorenia, editovania a browsovania štruktúry zdrojovkódových objektov (zväčša procedúr) , ako aj tvorenia a editovania poznámok k zdrojovkódovým objektom v okne **Browser of názov_servera.názov_tabučky**. Dozviete sa ako vytvoriť a získať nápovedu k procedúram.

Typ objektov

V záložke **Code** browsra môžete vidieť päť typov objektov. Dajú sa rozoznať podľa symbolu umiestneného naľavo pred názvom objektu.

-  - stored procedúra
-  - trigger. Zobrazovanie triggerov je implicitne vypnuté, a je ho možné zapnúť funkciou

Edit filter

-  - default
-  - rule
-  - extended procedúra

V okne Browsera sa nachádzajú nasledujúce tlačítka



- New - slúži na vytvorenie nového objektu (napr. procedúry). Tlačítko sa nachádza v okne iba ak browsujete skutočnú databázu.
- Design - otvorí designér procedúr s definíciou označeného objektu
- Execute - spustenie procedúry

Shortcut Menu obsahuje príkazy

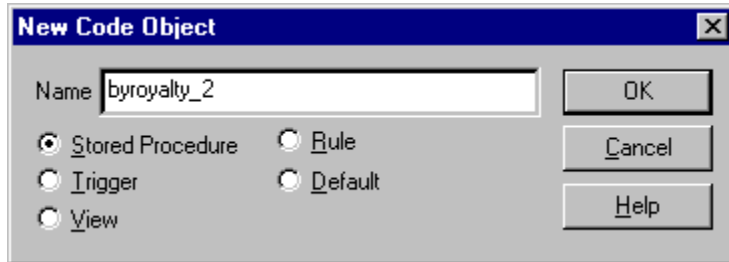
- Edit filter - filtrácia objektov v okne browsra
- Drop object - zmazanie procedúry
- Edit notes - vytvorenie poznámky k procedúre
- Plan size - diagnostika veľkosti procedúry
- View help - zobrazenie prvého komentára v tele procedúry

Tlačítko New

Toto tlačítko slouží na vytvoření nového objektu v skupině objektů **Code**.

Vytvoření nového kódového objektu

1. Stlačíte tlačítko **New** v okně skupiny objektů **Code** v okně **browsers**.
Zobrazí se dialogové okno

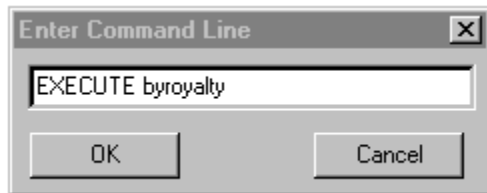


2. Napíšete do kolonky **Name** meno vytváraného objektu, zvolíte príslušné tlačítko typu vytváraného objektu a stlačíte tlačítko **OK**.
3. Otvorí sa okno [designéra procedúr](#), v ktorom môžete objekt naprogramovať.

Tlačítko Execute

Spustenie procedúry.

1. V browsri označíte procedúru, ktorú chcete spustiť.
2. Stlačíte tlačítko **Execute**.
Zobrazí sa dialógové okno **Enter Command Line** s nastavením: EXECUTE - názov_procedúry.



3. Doplňte hodnoty pre parametre procedúry a stlačíte tlačítko **OK**.
4. PATRIK spustí procedúru a výsledky zobrazí v okne Query Results.

Pred samotným spustením procedúry sa otvorí na SQL serveri nový proces, v ktorého kontexte sa daná procedúra spustí.

Príkaz Plan size

Príkaz **Plan Size** slúži na diagnostiku veľkosti procedúry po kompilácii. Pomocou tejto funkcie sa vám podarí udržať veľkosť procedúry pod 64 KB a vyhnúť sa tak chybnému fungovaniu kódového objektu.

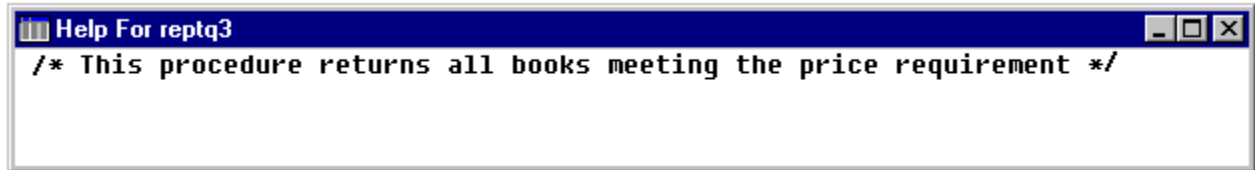
1. V browsri vyberte procedúru, ktorej veľkosť chcete zistiť a kliknite na pravé tlačítko myši.
2. Zvoľte príkaz **Plan Size** z ponuky **Shortcut menu**.

PATRIK zobrazí veľkosť procedúry v bytoch po skompilovaní.

```
Plan size of procedure jobs_ITrig is 4335.
```

Príkaz View help

Voľbou **View Help**, sa zobrazí prvý komentár v zdrojovom kóde vybranej procedúry. Ak zdrojový kód procedúry žiaden komentár neobsahuje, tak sa nezobrazí nič. Pri písaní procedúry obyčajne prvý komentár obsahuje popis činnosti procedúry.



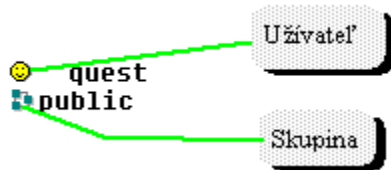
```
Help For reptq3
/* This procedure returns all books meeting the price requirement */
```

Otvorené okno **Help For názov_tabuľky** je typu Results.

Skupina užívateľov

Táto kapitola opisuje možnosti tvorenia, editovania a browsovania užívateľov ako aj editovania poznámok k užívateľom v okne **browsra**.

V okne skupiny **Users** nájdete dva druhy položiek. Môžete ich rozoznať podľa symbolu umiestneného vľavo pred ich menami



Tlačítka okna browsra pre skutočnú databázu

- View - zobrazenie definície označeného užívateľa.

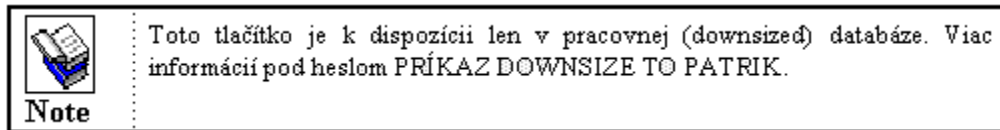
Tlačítka okna browsra pre pracovnú (downsized) databázu

- New - vytvorenie nového užívateľa.
- Design - modifikácia označeného užívateľa. Vytváranie a editovanie grupy nie je možné.

Shortcut menu v okne browsra obsahuje funkciu :

- Drop object - zmazanie užívateľa

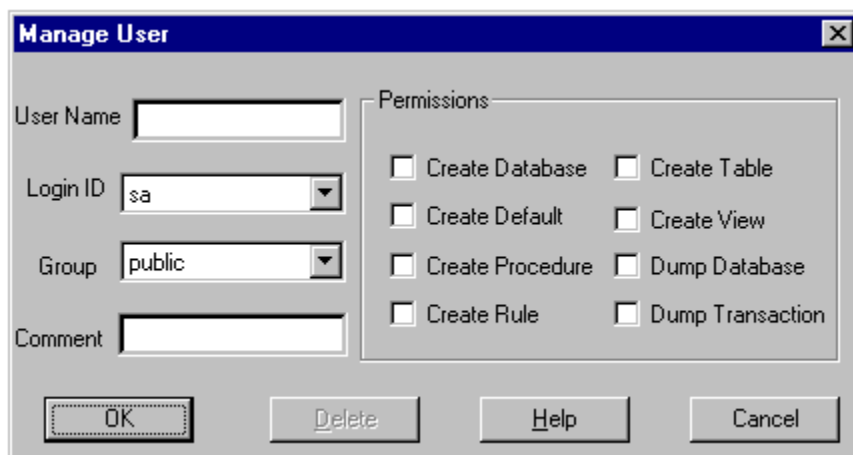
Tlačítko New



Toto tlačítko použijete, ak chcete pridať nových užívateľov do databázy.

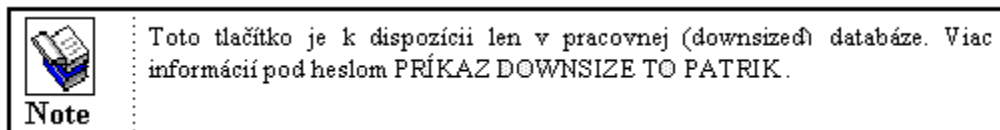
Vytvorenie nového užívateľa

1. V browsri stlačíte tlačítko **New**.



2. Do poľa **User Name** napíšete meno nového užívateľa.
3. V combo boxe **Login ID** vyberete Login ID, pod ktorým sa bude užívateľ hlásiť na server.
4. V combo boxe **Group** vyberete skupinu, do ktorej bude nový užívateľ patriť.
5. Do poľa **Comment** môžete napísať komentár o užívateľovi.
6. V skupine **Permissions** definujete práva nového užívateľa.
7. Stlačíte tlačítko **OK**.

Tlačítko Design

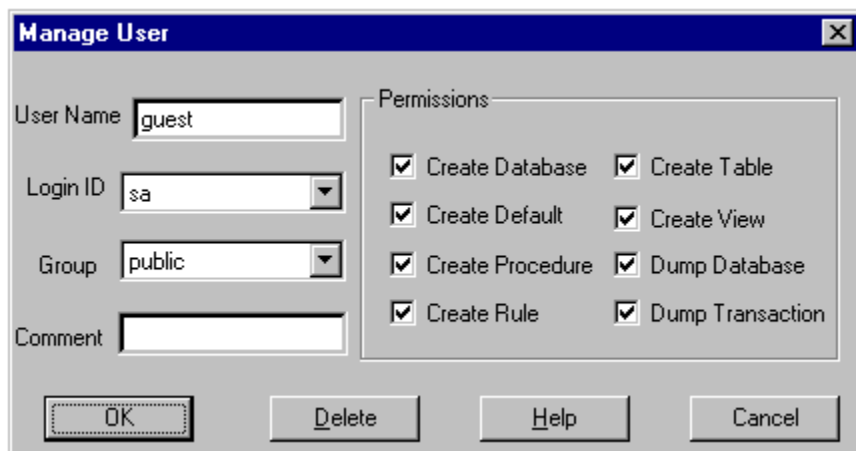


Toto tlačítko slúži na editovanie definície užívateľa.

Editovanie definície užívateľa

1. V browsri vyberte užívateľa, ktorého definíciu chcete editovať.
2. Stlačíte tlačítko **Design**.

Zobrazí sa dialógové okno **Manage User**.

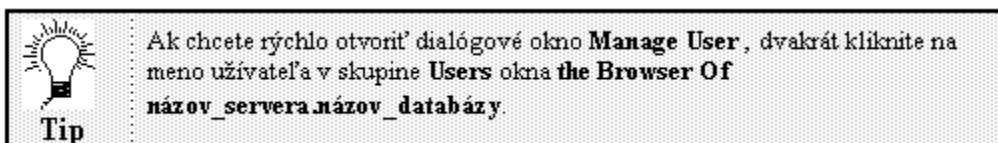


pozostávajúce s koloniek

- User Name - meno užívateľa
- Login ID - účet, pod ktorým sa hlási užívateľ do databázy
- Group - skupina, do ktorej užívateľ patrí
- Comment - popis užívateľa
- Permissions - práva užívateľa

Tlačítkom **Delete** je možné užívateľa zmazať.

3. Urobte potrebné zmeny a stlačíte tlačítko **OK**.

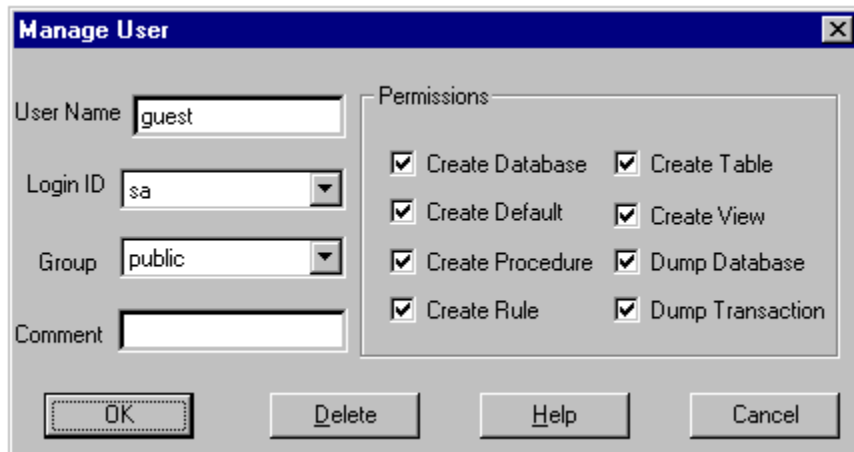


Tlačítko View

Prezeranie definície užívateľa

1. V Browse vyberte užívateľa, ktorého definíciu chcete vidieť.
2. Stlačíte tlačítko **View**.

Zobrazí sa dialógové okno **Manage User**.



pozostávajúce s koloniek

- User Name - meno užívateľa
- Login ID - účet, pod ktorým sa hlási užívateľ do databázy
- Group - skupina, do ktorej užívateľ patrí
- Comment - popis užívateľa
- Permissions - práva užívateľa

Tlačítkom **Delete** je možné užívateľa zmazať.

Užívateľské dátové typy

Táto kapitola opisuje možnosti tvorby, editovania a browsovania štruktúry užívateľských dátových typov, ako aj tvorbu a editovanie poznámok k užívateľským dátovým typom v okne *Browser of názov_server.názov_tabučky*

Tlačítka v okne browsra pre skutočnú databázu

- View - zobrazenie definície dátového typu

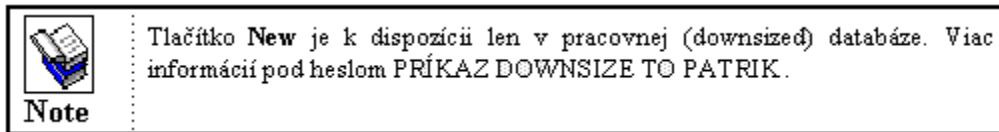
Tlačítka v okne browsra v pracovnej (downsized) databáze

- New - vytvorenie nového dátového typu
- Design - modifikácia označeného užívateľského dátového typu

Shortcut menu v okne browsra obsahuje funkciu :

- Drop object - zmazanie užívateľa

Tlačítko New

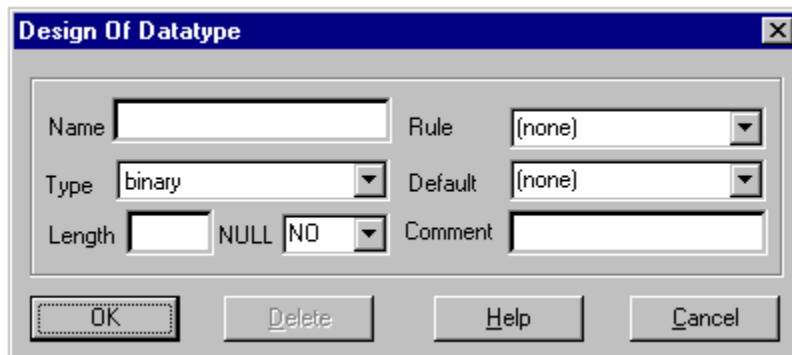


Toto tlačítko môžete použiť na definovanie nových užívateľských dátových typov v skupine **Types**.

Definovanie nových užívateľských dátových typov

1. Stlačíte tlačítko **New** v okne skupiny **Types** v okne **Browser Of názov_servera.názov_databázy**.

Zobrazí sa dialógové okno **Design Of Datatype**.



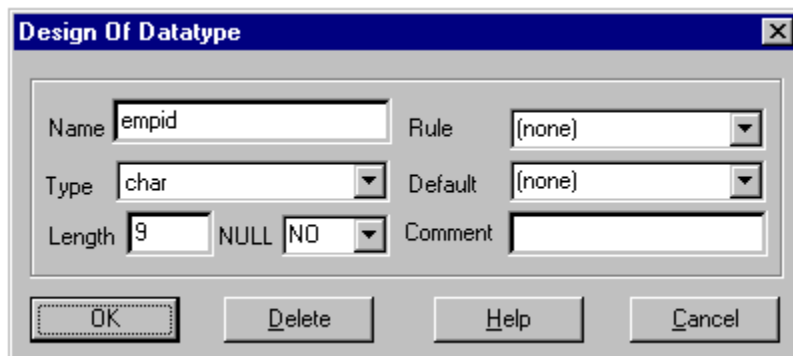
2. Do poľa **Name** napíšete meno nového dátového typu.
3. V combo boxe **Type** vyberiete systémový dátový typ, ktorý bude základný k novému dátovému typu.
4. Do poľa **Length** napíšete veľkosť nového dátového typu.
5. V combo boxe **Null** určíte, či dátový typ môže obsahovať hodnotu null.
6. V combo boxe **Rule** vyberiete validačné pravidlo.
7. V combo boxe **Default** môžete vybrať DEFAULT generujúci štandardnú hodnotu pre nový dátový typ.
8. Do poľa **Comment** môžete napísať komentár k definovanému dátovému typu.
9. Stlačíte tlačítko **OK**.

Tlačítko Design

Toto tlačítko můžete použít na změnu existujícího datového typu.

Editování definice uživatelského datového typu

1. Vyberte datový typ, kterého definici chcete editovat, v skupině **Types** v okně **Browser Of názov_servera.názov_databázy**.
2. Stlačte tlačítko **Design**.
Zobrazí se dialogové okno **Design Of Datatype**.



so záložkami

- Name - meno datového typu
- Type - systémový datový typ, ktorý je bazový k danému datovému typu.
- **Length** - veľkosť nového datového typu.
- **Null** - či datový typ môže obsahovať hodnotu null.
- **Rule** - validačné pravidlo pre hodnoty datového typu.
- **Default** - default generujúci štandardnú hodnotu datového typu.

Tlačítkom **Delete** je možné datový typ vymazať.

3. Urobte potrebné zmeny a stlačte tlačítko **OK**.

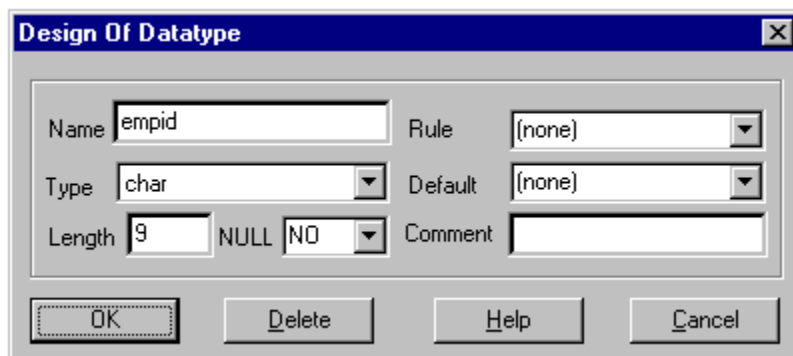
Tlačítko view

Toto tlačítko můžete používat jen na prezeranie dátových typov.

Prezeranie definície dátových typov

1. Vyberte dátový typ, ktorého definíciu si chcete pozrieť, v skupine **Types** v okne **Browser Of názov_servera.názov_databázy**.
2. Stlačíte tlačítko **View**.

Zobrazí sa dialógové okno **Design Of Datatype**.



so záložkami

- Name - meno dátového typu
 - Type - systémový dátový typ, ktorý je základný k danému dátovému typu.
 - **Length** - veľkosť nového dátového typu.
 - **Null** - či dátový typ môže obsahovať hodnotu null.
 - **Rule** - validačné pravidlo pre hodnoty dátového typu.
 - **Default** - default generujúci štandardnú hodnotu dátového typu.
- Tlačítkom **Delete** je možné dátový typ vymazať.

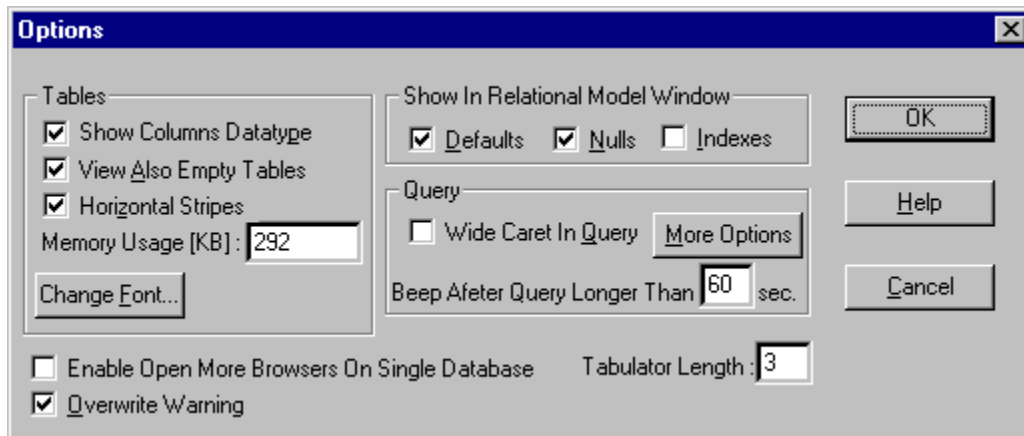
Options

Táto kapitola opisuje nastavovanie volieb v PATRIKovi a obsahuje ich krátky popis.

Nastavenie tabuľky volieb v PATRIKovi



1. Stlačíte tlačítko **Options**
2. Zobrazí sa dialógové okno **Options**.



1. Skupina **Tables**
 - a) Ak zvolíte check box **Show Column Datatype**, PATRIK bude zobrazovať dátové typy stĺpcov v hlavičke okien s datami a vo výsledkoch query.
 - b) Ak zvolíte check box **View Also Empty Tables**, PATRIK bude otvárať aj okná s dátami napriek tomu, že tieto nebudú obsahovať žiadne záznamy
 - c) **Horizontal Stripes** - ak je nastavené, budú párne záznamy v tabuľkách s dátami zobrazované s tmavšou farbou pozadia
 - d) **Change Font** - otvorí **dialógové okno** v ktorom je možné nastaviť font v dátových tabuľkách.
2. V skupine **Show In Relationship Window** vyberte check box vlastností stĺpcov tabuliek, ktoré chcete zobrazovať v tabuľkách okna Relational Model
3. Skupina **Query**

- a) Ak zvolíte check box **Wide Caret In Query**, PATRIK bude zobrazovať široký kurzor namiesto úzkeho v okne **Query Editor**.
 - b) Do polí **Beep after query longer than** napíšete počet sekúnd. Ak bude query po **spustení** bežať ako určený počet sekúnd, PATRIK na jeho konci pípne.
 - c) **More Options** - nastavenie globálnych query options.
4. V okne **Query Editor** v poli **Tab Length** určíte počet medzier, ktoré budú reprezentovať tabulátor.
5. Ak zvolíte check box **Overwrite Warning**, PATRIK zobrazí varovnú správu vždy, keď sa pokúsite pri ukladaní údajov na disk prepísať existujúci súbor
6. **Enable Open More Browsers On Single Database** - ak nie je zaškrtnuté, tak Patrik neumožňuje otvoriť viac Browse okien pre jednu databázu.
7. Keď ukončíte výber, stlačíte tlačítko **OK**

Informácie o options sa automaticky ukladajú na disk, takže pri ďalšom otvorení Patrika ich už nie je nutné opäť nastavovať.

Zobrazenie a editovanie štruktúry tabučky

	Name	Type	Comment
	nl_author_id	int	new id
	au_id	id	old id
1	au_lname	varchar	last name
2	au_fname	varchar	first name
→	phone	char	phone number
	address	varchar	
	city	varchar	
	state	char	
	zip	char	zip code
	contract	bit	has contract?

Table Name: authors
Length: 12
Default:
Rule:
Allow Nulls: No
Constraint: CK_authors_at
Cnst.Column: au_id

Index: aunmind
PK_author
UPKCL_auind

Fill Factor: 100
default

Use As Primary Key
 Clustered
 Unique
 Ignore Dup. Key
 Ignore Dup. Row
 Allow Dup. Rows

Add Remove Default

Okno *Design Table názov_tabučky* sa dá rozdeliť do dvoch častí:

- Horná časť - obsahuje názvy atribútov, definície dátových typov, komentáre atď
- Dolná časť obsahuje
 - Bližší popis aktívneho atribútu
 - Default, Check Constrainty tabučky
 - Indexy tabučky

Ak je v okne zobrazená tabučka pracovnej (*downsizenutej*) databázy a vykonáte v nej nejaké zmeny, tak sa pri zavretí okna Patrik opýta, či chcete zmeny uložiť. Ak ich dáte uložiť, tak sa overí korektnosť vykonaných zmien. Ak sa nájdu nejaké nezrovnalosti, tak ich Patrik vypíše, neuloží zmeny a nezavrie okno s tabučkou, čím umožní užívateľovi vykonané chyby opraviť.

Horná časť okna design table názov_tabučky

Horná časť okna Design Table názov_tabučky obsahuje päť stĺpcov:

The screenshot shows the 'Design Table' dialog box for the table 'spt_datatype_info'. The table has five columns: Name, type, Comment, and two unlabeled columns. The first column is labeled '1. stĺpec' (Arrow pointing to active attribute), the second '2. stĺpec' (Index attribute and priority), the third '3. stĺpec' (Attribute name), the fourth '4. stĺpec' (Attribute data type), and the fifth '5. stĺpec' (Comment on attribute).

	Name	type	Comment		
1	sstype	tinyint			
2	fixlen	int			
3	ODBCVer	tinyint			
	TYPE_NAME	varchar			
	DATA_TYPE	smallint			
	data_type	int			
	numeric_precision	smallint			
	RADIX	smallint			
	length	int			
	LITERAL_PREFIX	varchar			
	LITERAL_SUFFIX	varchar			
	CREATE_PARAMS	varchar			

Table Name: spt_datatype_info
Index: datatypeinclust
Fill Factor: 100
default
 Use As Primary Key
 Clustered
 Unique
 Ignore Dup. Key
 Ignore Dup. Row
 Allow Dup. Rows

Zmena poradia atribútov v okne design table názov_tabučky

Zmena dátového typu atribútu

Shortcut menu obsahuje funkcie

- **Add column** - pridaná nový stĺpec na koniec tabučky
- **Delete column** - zmaže označeného stĺpca tabučky. Ak sú na mazaný stĺpec napojené najaké relácie, tak sa pri uložení zmien v tabučke tieto relácie zrušia s databázy (Predtým na to Patrik ovšem upozorní).
- **Find Column** - o novootvorenom dialógovom okne umožní zadať meno alebo poradové číslo stĺpca tabučky a po voľbe **OK** daný stĺpec v tabučke nájde a označí.

Konvencia názvov atribútov.

Pri editovaní mena nového atribútu sa automaticky vyplní aj kolonka dátového typu podľa tabučky:

predpona mena atribútu

dátový typ

s_	char, varchar
nl_	int
ni_	<i>smallint</i>
nb_	<i>tinyint</i>
d_	datetime
nd_	decimal,float,number
c_	money
ts_	timestamp
bn_	binary, varbinary
m_	text, image
b_	bit

V prípade, že nechtete dodržiavať stanovenú konvenciu môžete doplnený dátový typ podľa potreby zmeniť.

Zmena poradia atribútov v okne design table *názov_tabu^{3/4}ky*



Note

Zmeny, ktoré ste vykonali v pracovnej databáze sa uplatnia, až keď spustíte upsíze pracovnej databázy do reálnej databázy na SQL server. Viac informácii pod heslom PRÍKAZ UPSIZE TO SQL.

[oprava stranu vymazať](#)

Zmena poradia atribútov v okne Design Table *názov_tabu^{3/4}ky*

1. Kliknite na meno stĺpca, ktorý chcete premiestni . Modrá šípka v prvom stĺpci okna sa presunie do riadku s daným menom stĺpca, čím sa stĺpec sa označí za aktívny.
2. Kliknite ^{3/4}avým tlačítkom myši na modrú šípku, nechajte ^{3/4}avé tlačítko myši a posuňte kurzor myši na meno stĺpca, pred ktorý chcete daný stĺpec vložiť .

Zmena dátového typu atribútu

Na rozdiel od mnohých nástrojov spravovania databáz, PATRIK umožňuje zmeniť dátový typ atribútu bez akejkoľvek straty dát.

Zmena dátového typu atribútu

1. Kliknite na stĺpec dátových typov (4. stĺpec) do riadku s atribútom, ktorého dátový typ chcete zmeniť.
2. Vyberte dátový typ v combo boxe **Datatype**.

Skupina urèujúca bližší popis aktívneho atribútu

Táto skupina definuje ģalšie vlastnosti aktívneho atribútu (atribútu oznaèeného v tabu³4ke modrou šípkou) obsahuje nasledovné položky:

Table name	- názov tabu³4ky
Length	- ve³4kos¼ atribútu
Default	- procedúra (default) definujúca štandardnú hodnotu atribútu
Rule	- procedúra (rule) urèujúca validaèné pravidlo atribútu
Allow Nulls	- No - atribút nemôže obsahova¼ null hodnotu - Yes - atribút môže obsahova¼ null hodnotu

Z estetických dôvodov je v tejto èasti aj informácia o názve tabu³4ky v kolonke **Table Name**:

Table Name	authors
Length	
Default	
Rule	
Allow Nulls	Yes

Constrainty

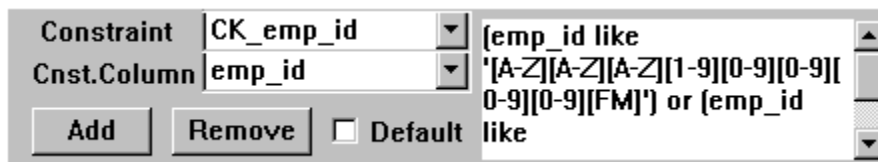
Constraint je obmedzenie hodnoty, ktorá sa zapisuje do stĺpca tabuľky. Slúži na udržanie doménovej integrity.

Existujú dva typy constraintov

1. **Check Constraint** - určuje formou validačného pravidla hodnoty údajov, ktoré sú v stĺpci prijateľné. Môžete aplikovať **CHECK constrainty** na viaceré stĺpce a môžete aplikovať viaceré **CHECK constrainty** na jeden stĺpec. Keď je tabuľka dropnutá, všetky CHECK constrainty sú dropnuté tiež.
2. **Default Constraint** - určuje hodnotu, ktorá bude automaticky pridaná do nového záznamu, pokiaľ užívateľ žiadnu hodnotu nedefinoval.

Ďalej **Constraint** obsahuje nasledujúce položky:

- **Constraint** - combo box obsahuje zoznam constraintov určený ich menami.
- **Cnst.Column** - atribút tabuľky vzťahujúci sa k validačnej podmienke alebo default hodnote.



Constraint	CK_emp_id	[emp_id like
Cnst.Column	emp_id	'[A-Z][A-Z][A-Z][1-9][0-9][0-9][0-9][0-9][FM]') or (emp_id
		like

Default

Ďalej **Constraint** obsahuje tlačítka **Add** a **Remove** slúžiace na pridanie nového, resp zrušenie aktuálneho constraintu.

Tlačítko Add

Tlačítko Remove

Tlačítko Add

Pridanie nového constraintu

1. Stlačíte tlačítko **Add** v okne **Design Table** *názov_tabuľky*.

Zobrazí sa dialógové okno **New constraint name**.



2. Napíšete názov nového constraintu a stlačíte tlačítko **OK**.
3. Ak sa jedná o **default constraint** vyberte v poli **Cnst. Column** atribút (stápec tabuľky), do ktorého sa bude nová default hodnota vkladať.
4. Vyberte check box **Default** ak sa má jedna o **Default Constraint** (ináč pôjde o **Check Constraint**).
5. Do poľa **Text Constraintu** napíšete default hodnotu, ak sa jedná o **Default Constraint**, alebo validačné pravidlo, ak sa jedná o **Check Constraint**.

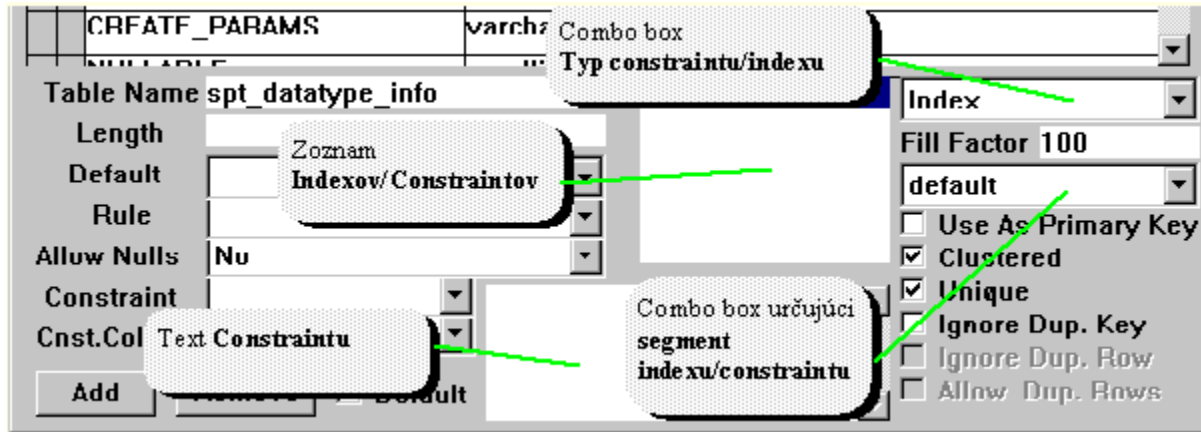
Tlačítko Remove

Vymazanie predtým nadefinovaného constraintu

1. V **combo boxe Constraint** vyberte constraint, ktorý chcete zrušiť.
2. Kliknite na tlačítko **Remove**.

Skupina indexov

Štruktúra skupiny **Indexov** sa mení v súvislosti s typom indexu, ktorý si vyberiete v combo boxe **Index Type**.



Typy indexov v combo boxe **Index Type**.

- Identity
- Primary key (primárny kľúč)
- Index
- Unique

Shortcut menu sa vyvoláva stlačením pravého tlačítka myši v kolonke **Zoznam Indexov/Constraintov**. Obsahuje príkazy

- Add index... - pridanie nového indexu
- Delete index - zrušenie označeného indexu
- Rename... - premenovanie označeného indexu
- Add identity - pridanie IDENTITY

Stĺpce tabuľky indexované označeným indexom sú označené v **hornej časti** okna (v druhom stĺpci) poradovými číslami. V uvedenom poradí sú stĺpce zahrnuté do indexu.

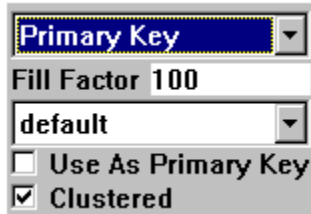
Identity

Identity je vlastnosť SQL tabuľky nazývaná **IDENTITY**. Je doplnená dvomi konštantami **Seed** a **Increment**, ktoré PATRIK zobrazuje v rovnomených poliach.

Identity	
Seed	2
Increment	5

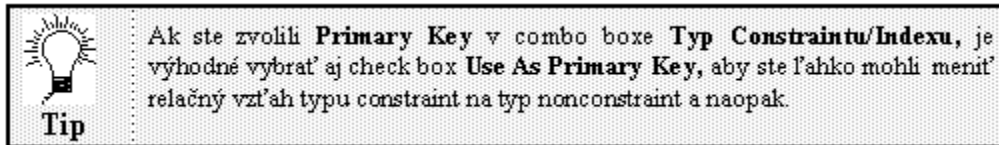
Primary key (primárny kľúč)

Primary key je stápec, alebo kombinácia stápcov, ktoré jedinečne identifikujú každý záznam v tabuľke. Stápe primárneho kľúča nemôžu obsahovať hodnotu NULL. Primary key sa používa na spojenie reláciou typu constraint s foreign tabuľkou.



Primary Key
Fill Factor 100
default
 Use As Primary Key
 Clustered

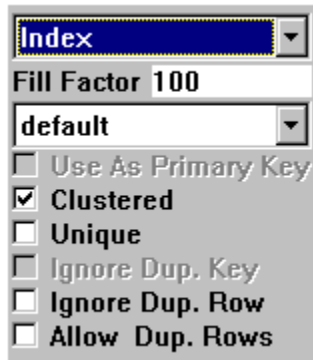
- **Fill Factor** - špecifikuje fill factor indexu primárneho kľúča.
- **Segment** - označuje segment, v ktorom je primary key umiestnený
- **Use As Primary Key** - tento check box vyberte, ak chcete mať tabuľku na strane "one" z "one to many" nonconstraintového relačného vzťahu (zabezpečeného triggerami). Tento typ primary key sa používal v MS SQL 4.2 a je v PATRIKovi kvôli spätnej kompatibilitate.



- **Clustered** - určuje, či je index primárneho kľúča CLUSTERED.

Index

Index zvyšuje usporiadanosť údajov v tabuľke.

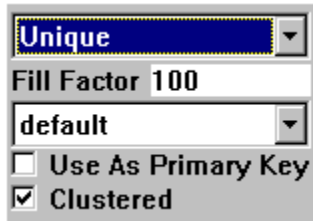


- **Fill Factor** - špecifikuje fill factor indexu.
- **Segment** - označuje segment, v ktorom je index umiestnený
- **Use As Primary Key** - tento check box vyberte, ak chcete mať tabuľku na strane "one" z "one to many" nonconstraintového relačného vzťahu (zabezpečeného triggerami). Tento typ primary key sa používal v MS SQL 4.2 a je v PATRIKovi kvôli spätnej kompatibilitate.
- **Clustered** - určuje, či je index CLUSTERED.
- **Unique** - určuje jedinečnosť indexu.
- **Ignore Dupl. Row, Allow Dup. Row a Allow Dup. Row**

Viac informácií o týchto voľbách sa viac dozviete v SQL Server Books Online.

Unique

Unique constraint je index, ktorý nedovoľuje opakujúce sa hodnoty v indexovaných stĺpcoch tabuľky.



- **Fill Factor** - špecifikuje fill factor indexu.
- **Segment** - označuje segment, v ktorom je index umiestnený
- **Use As Primary Key** - tento check box vyberte, ak chcete mať tabuľku na strane "one" z "one to many" nonconstraintového relačného vzťahu (zabezpečeného triggerami). Tento typ primary key sa používal v MS SQL 4.2 a je v PATRIKovi kvôli spätnej kompatibilitate.
- **Clustered** - určuje, či je index CLUSTERED.

Viac informácií viť SQL Server Books Online.

Príkaz Add index

Pridanie nového indexu

1. Prvým tlačítkom myši kliknite na **Zoznam indexov/constraintov**.
Zobrazí sa menu **Shortcut menu**.
2. Zvoľte príkaz **Add Index**.
Zobrazí sa dialógové okno **Enter New Index Name**.



3. Napíšte názov nového indexu a stlačte tlačítko **OK**.
4. V combo boxe **Typ constraintu/indexu** vyberte jeden zo štyroch **inde** opísaných typov.
5. Nastavte špecifické vlastnosti vybraného typu **constraintu** alebo **indexu** (napr. fill factor, clustered, increment, seed)
6. V druhom stĺpci **tabuľky** (pred názvom atribútu) kliknite na atribút (atribúty), na ktoré sa má index vzťahovať. Ak pôjde o viacero atribútov, môžete kliknutím určiť prioritu poradia atribútov.

Príkaz Delete index

Vymazanie indexu

1. Zvoľte index, ktorý chcete vymazať v **Zozname indexov/constraintov**.
2. Kliknite na pravé tlačítko myši.
Zobrazí sa **Shortcut menu**.
3. Zvoľte príkaz **Delete Index**.

Príkaz Rename

Premenovanie indexu

1. Zvoľte index, ktorý chcete premenovať v **Zozname indexov/constraintov**.
2. Kliknite na pravé tlačítko myši.
Zobrazí sa **Shortcut menu**.
3. Zvoľte príkaz **Rename Index**.
4. Zobrazí sa dialógové okno **Enter New Name**.



5. Napíšte nový názov a stlačte tlačítko **OK**.

Príkaz Add identity

Pridávanie identity tabuľke

1. Umiestnite kurzor myši do **Zoznamu indexov/constraintov** a kliknite na pravé tlačítko myši.
2. Zvoľte príkaz **Add Identity**.
3. Vyplňte hodnotu **Increment** a **Seed**.
4. V druhom stĺpci hornej časti tabuľky kliknite na riadok atribútu, na ktorý sa bude IDENTITY vzťahovať.

Vytvorenie novej tabučky databázy

Zobrazí sa okno Design Table neobsahujúce žiadne atribúty.

Name	Type	Comment

Table Name:

Length:

Default:

Rule:

Allow Nulls:

Constraint:

Cnst.Column:

Index:

Fill Factor:

Use As Primary Key

Clustered

Unique

Ignore Dup. Key

Ignore Dup. Row

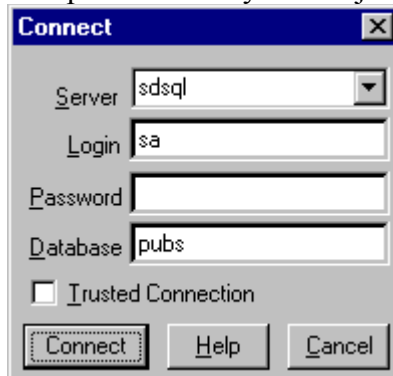
Allow Dup. Rows

Add Remove Default

Vyplňte potrebné informácie a zavrite okno, čím sa tabučka uloží do databázy.

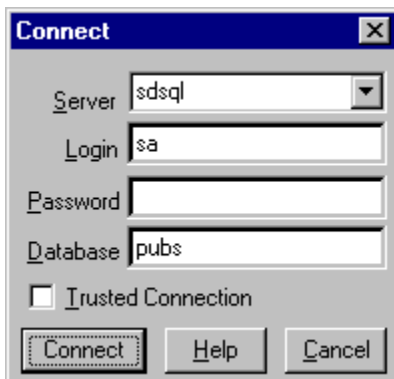
Query Editor

Služi na editovanie (písanie, prezeranie, vytváranie, spúšanie) voľných dotazov (**Free Query**). Na editovanie procedúr, viewov, triggrov, defaultov, rule-ov služi **designér** na to určený. Tento designér bere ohľad na štruktúru editovaného objektu a poskytuje pri práci väčší komfort, akoby sa mal objekt (napr. procedúra) editovať v **Query Editore**. Samozrejme, že voľný dotaz môže byť aj príkaz (množina príkazov), ktorého spustením sa vytvorí nejaká procedúra.

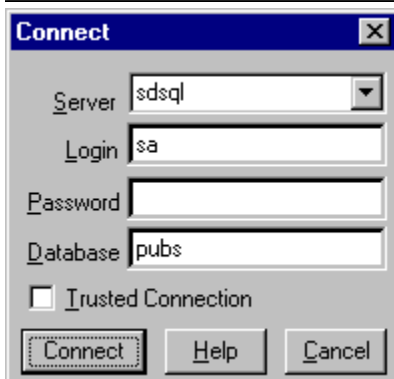


Query Editor sa dá otvoriť pomocou tlačítka **Connect** na toolbare aplikácie, alebo v menu Project funkciou Query.

V názve okna je vidno že sa jedná o editovanie query, pod akým menom je query uložená na disku (pre novootvorenú query je vymyslené meno unnamed) a či bola query modifikovaná od okamžiku posledného uloženia - ak bola, tak sa pred názvom query vyskytuje znak *). Toolbar **Query Editora** obsahuje nasledovné príkazy:



Load... - nahrá do editora obsah vyvoleného súboru.



Save - uloží napísanú Query do súboru.

Print... - tlačidlo textu.

Find - vyhľadávanie (Alt F3)


Find Next - nájde nasledujúci výskyt hľadaného slova (F3)

Properties

Execute - spustí napísanú **Query**, alebo jej časť označenú v bloku (Alt E)

Go - spustí **debugovanie** query (F5)

Insert/Remove Breakpoint - v **editore** vloží/zruší breakpoint v riadku s kurzorom. (F9)

-  **Remove All Breakpoints** - zruší všetky nastavené breakpointy

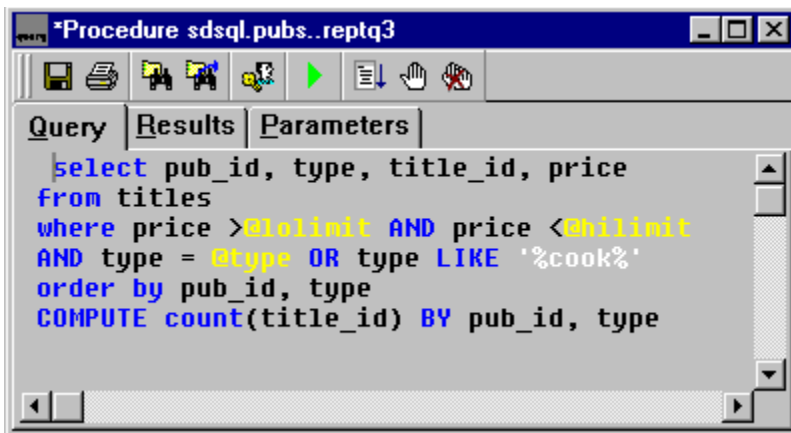
Pod toolbarom sa nachádza priestor so záložkami. Po otvorení sa tu nachádza iba záložka **Query**, reprezentujúca **editor**, ale už po prvom spustení (funkcia **Execute**) pribudne aj záložka **Results**, obsahujúca množinu výsledkov (a prípadných chybových hlásení), ktoré vrátil SQL server pri poslednom spustení **Query**. Ak je v **Options** nastavená voľba **Show Plan**, tak pribudne aj záložka **Show Plan** s informáciami o optimálnosti query. Medzi záložkami sa prepína pomocou klávesy **Alt + <podčiarknuté písmenko v názve záložky>**.

Funkcie, ktoré nie sú vyvedené do toolbaru, je možné vyvolať stlačením pravého tlačítka v samotnom editore. Otvorí sa shortcut menu s ponukou funkcií:

- **Save As...** - umožní uložiť **Query** do iného súboru ako bola ukladaná doteraz .
- **Find Prev** - nájde predchádzajúci výskyt hľadaneého slova (Shift F3)
- **Replace** - funkcia **zámeny textu**.
- **Colors...** - nastavenie farieb syntaktických skupín v editore
- **Define Vocabulary** - zdefinovanie vlastnej, farebne odlišenej, skupiny slov
- **Run To Begin** - spustí **debuger** a zastaví sa pred vykonaním prvého príkazu príkazu query.
- **Run To Cursor** - spustí **debuger** a vykonáva query, pokiaľ nenarazí na nejaký breakpoint, alebo na riadok, v ktorom sa nachádzal kurzor, keď bola funkcia vyvolaná.

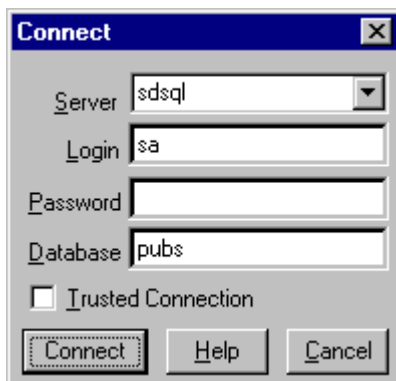
Designér Procedúr

Služi na prezeranie, editovanie a vytváranie **stored procedúr, view-ov, triggrov, default-ov a rule-ov**. Opticky sa moc nelíši od **Query Editor**a, no designér robí niečo navyše. V **editore** designéra sa nenachádza inicializačný text procedúry (CREATE PROCEDURE) ani definícia parametrov. Na parametre je určená ďalšia záložka - **Parameters**, ktorá reprezentuje tabuľku, v ktorej sú uvedené parametre procedúry spolu s ich vlastnosťami. To samozrejme platí, ak sa edituje procedúra. Ak sa edituje **Default, Rule, alebo Trigger**, tak žiadna záložka s parametrami nie je potrebná. Pri editácii **View-u** tiež nie je pochopiteľne možné zadávať parametre, no je možné zadávať názvy stĺpcov **View-u**, takže záložka **Parameters** sa v designéri v tomto prípade nachádza a reprezentuje tabuľku s názvami stĺpcov **View-u**.



V hornej lište (titulku) **Designéra** sa nachádza text informujúci o názve objektu, jeho type a lokácii. Ak bol objekt (procedúra) v designéri modifikovaný od posledného uloženia do databázy, tak sa nechádza pred názvom procedúry znak *.

Toolbar **Designéra** obsahuje nasledovné príkazy:



Save- zoskriptuje procedúru (alebo iný editovaný objekt) do súboru.

Print... - tlačidlo textu.

Find - vyhľadávanie (Alt F3)

Find Next - nájde nasledujúci výskyt hľadaného slova (F3)



Properties - umožňuje zadať prídavné vlastnosti

Procedúry, View-u, Triggra, Default-u, alebo Rule-u

Execute - spustenie a/alebo uloženie objektu späť do

databázy. Funkcia sa správa odlišne pre Procedúru, View a pre Trigger, Default a Rule (Alt E)
Nasledujúce tlačidlá sú zobrazené iba pri editovaní procedúry:

Go - spustí debugovanie procedúry (F5)

-  **Insert/Remove Breakpoint** - v editore vloží/zruší breakpoint v riadku s kurzorom. (F9)
-  **Remove All Breakpoints** - zruší všetky nastavené breakpointy

Pod toolbarom sa nachádza priestor so záložkami. Môžu sa tu nachádzať záložky:

- **Query** - predstavuje editor so zdrojovým kódom procedúry (Alt q)
- **Results** - vytvorí sa až pri prvom spustení, alebo uložení procedúry a obsahuje výsledky/stav po poslednom spustení/uložení. (Alt r)
- **Parameters** - obsahuje definíciu parametrov procedúry, alebo názvov stápcov **View-u**. (Alt p)

Medzi záložkami sa prepína pomocou klávesy Alt + <podčiarknuté písmenko v názve záložky>.

Funkcie, ktoré nie sú vyvedené do toolbaru, je možné vyvolať stlačením pravého tlačítka v samotnom editore. Otvorí sa shortcut menu s ponukou funkcií:

- **Save As...** - zoskriptuje procedúru iného súboru, ako tomu bolo posledne.
- **Find Prev** - nájde predchádzajúci výskyt hľadadaného slova (Shift F3)
- **Replace** - funkcia zámeny textu.
- **Colors...** - nastavenie farieb syntaktických skupín v editore

Nasledovné položky sa nachádzajú v menu iba pri editovaní procedúry.

- **Run To Begin** - spustí debuger a zastaví sa pred vykonaním prvého príkazu príkazu procedúry.
- **Run To Cursor** - spustí debuger a vykonáva procedúru, pokiaľ nenarazí na nejaký breakpoint, alebo na riadok, v ktorom sa nachádzal kurzor, keď bola funkcia vyvolaná.

V menu sa ešte zvykne nachádzať aj položka **Dependencies** ktorá v submenu obsahuje položky typu : **Called By** <meno procedury> a **Calls** <meno procedúry> s názvami procedúr volajúcich editovanú procedúru a s názvami procedúr volaných s editovanej procedúry. Voľbou niektorej s uvedených položiek sa otvorí vybraná procedúra.

Editor

V editor je možné písať nejaké príkazy, prípadne kód procedúr, ktoré sa pri spustení pošlú SQL serveru. Editor písané slová a výrazy automaticky analyzuje a zadeľuje do tzv. **syntaktických skupín**, ktoré sú odlišené rôznymi farbami. Editor rozoznáva nasledovné syntaktické skupiny:

- **Kľúčové slová** (napr SELECT, from, Where - tu nezáleží na veľkosti jednotlivých písmen)
- **Reťazce** (uzatvorené v úvodzovkách, či v apostrofoch)
- čísla
- **premenné a temporárne tabuľky** (napr. @@rowcount, #tmp)
- **komentáre**
- **užívateľom nadefinované slová**
- **ostatné slová** - sem sú zaradené slová, ktoré nepatria do žiadnej s vyššie uvedených skupín

Farebný text

Farby pre jednotlivé syntaktické skupiny sa pochopíte ľahšie dajú **nastaviť**.

Pri písaní Vám môže byť nápomocná funkcionálna **Write-Ahead** pre kľúčové slová. Ak chcete napríklad napísať slovo SELECT, stačí, ak napíšete jeho prvých pár písmen (povedzme SE) a následne stlačíte klávesu **Esc**. Patrik načaté slovo doplní o zostávajúce písmená. Ak slovo nedoplnil správne, tak postupným stláčaním klávesy **Esc**, sa dopracujete k žadanému slovu (v tomto prípade slovu SELECT).

Online Help

V texte funguje **Online Help** na kľúčové slová, tabuľky a procedúry. Stačí sa kurzorom v texte postaviť na nejaké slovo, ku ktorému by ste chceli help a stlačíte klávesu **F1**. Editor rozozná či ide o meno tabuľky, procedúry alebo kľúčové slovo a podľa toho zobrazí príslušné shortcut menu. Ak ide o tabuľku (povedzme *pubs..authors* - aj s takto referencovanými tabuľkami vie help nájsť), zobrazí sa menu s jedinou položkou **View Table** *pubs..authors*. Pre procedúru sa zobrazí menu s položkou **Design Procedure** <meno procedúry>. V oboch prípadoch je možné položku zvoliť, alebo zrušiť klávesou **Esc**. Ak však ide o kľúčové slovo, je to komplikovanejšie. Nemusí ísť totiž vždy o celé kľúčové slovo a aj ak ide nemusí celé kľúčové slovo vystihovať príkaz, ku ktorému chcete vidieť help (napr. slovo *CREATE* neznamena vôbec nič, ale už dvojice slov *CREATE TABLE*, *CREATE DATABASE*, *CREATE RULE* znamenajú konkrétne príkazy). Patrik túto nejednoznačnosť rieši tak, že v zobrazenom shortcut menu zobrazí všetky možné témy, ktoré sa zaoberajú na žiadané slovo (prípadne na žiadanú časť slova) a užívateľ si už vyberie, k čomu chce help vidieť. Nejedná sa o nejaký honostný help s obrázkami a početnými príkladmi. Text helpu (a vlastne všetkých helpov príkazov) sa nachádza v tabuľke *helpsql* v databáze *master* na každom SQL serveri. Patrik tento text „iba“ zobrazí.

Skok na párovú zátvorku (úvodzovku, komentárovú značku)

Prostredníctvom klávesovej kombinácie **Ctrl {** sa dá pohodlne vyjsť z ľavej zátvorky k príslušnej pravej a naopak. Funkcia funguje nielen pre všetky typy zátvoriek (), { }, [] ale aj pre úvodzovky, apostrofy, či diakritiku a konce komentárov (/* , */) a kľúčové slová **BEGIN** a **END**.

Write Ahead

Funkcionalita urýchľujúca písanie kľúčových slov. Ak po napísaní prvých pár písmen kľúčového slova stlačíte klávesu **Esc**, tak sa doplní koniec kľúčového slova, začínajúceho uvedenými písmenami. Opakovaným stláčaním klávesy **Esc**, sa cyklicky striedajú začínajúce sa na napísanými písmenami.

Indent

Viacriadkový text **označený v bloku** je možné celý posunúť o šírku tabulátoru **doprava**, klávesou **Tab**, alebo **doprava**, klávesou **Shift+Tab**.

Auto Text

Pri opakovanom písaní niektorých programových segmentov je možné využiť funkcionality **Auto Text**. Označený segment je možné uložiť do súboru (napr. klávesami **Ctrl Shift 1**) a niekedy neskôr vložiť do textu klávesou (**Ctrl 1**). Je možné uložiť a neskôr použiť viac segmentov scriptu (procedúry). Podobne ako **Ctrl Shift 1** a **Ctrl 1** funguje aj **Ctrl Shift 2** a **Ctrl 2**, **3**, **4**, ..., **9**, **0**.

Návestia v query editore

Ak si chcete pro editovaní dlhej procedúry v texte niekam "odskočiť" a neskôr sa vrátiť na pôvodné miesto, dajú sa na toto použiť:

- **Ctrl F2** – nastavenie návestia v aktuálnom riadku
 - **Ctrl F2** - zrušenie návestia v aktuálnom riadku
 - **F2** – skok na najbližšie nižšie (a ak neexistuje nižšie, tak na najvzdialenejšie vyššie) návestie
- Návestie sa zobrazuje ako znak * tyrkisej farby na ľavej strane textu.

Číslo riadku a stápa kurzora v editore je zobrazené v spodnej (informačnej) lište aplikácie. V editore fungujú nasledovné štandardné kombinácie kláves:

- **Ctrl C** - **copy** - skopírovanie označenej časti textu do clipboardu
- **Ctrl V** - **paste** - vloženie obsahu clipboardu do textu
- **Ctrl X** - **cut** - skopírovanie označenej časti textu do clipboardu a jej následné zmazanie z textu.
- **Ctrl Z** - **undo** - návrat o "krok" späť v písaní
- **Ctrl A** - **redo** - návrat kroku späť
- **Alt F3** - Find - vyhľadávanie
- **F3** - Find next (nájdí nasledujúci výskyt hľadaného slova)
- **Shift F3** - Find previous (nájdí predchádzajúci výskyt hľadaného slova)
- **Ctrl]** - skok na párovú zátvorku
- **Esc** - **Write Ahead**
- **Ctrl F2** - vlož/zruš návestie
- **F2** - skok na návestie
- **F9** - nastavenie/zrušenie breakpointu
- **Ctrl B** - editor breakpointov

Execute

Spúšťanie Bloku

Funkcia slúži na spustenie **Query** alebo jej časti označenej v bloku. Ak je označená nejaká časť textu v bloku, tak sa spustí iba táto časť a ak nie, tak sa spustí celý obsah **Query** okna.

Viacero batch-ov v jednom texte

Patrik pri spúšťaní dodržiava konvenciu súvisiacu s batch-mi príkazom **GO** tak ako je to popísané v príručkách k **MS SQL** Servru. Pseudopríkaz **GO** slúži na oddelenie dvojice batch-ov v rámci jedného textu.

Automatické ukladanie pri spúšťaní.

Keďže spúšťanie dotazov môže niekedy viesť k zrušeniu editora, tak sa pred spustením **Query** všetky modifikované **Query** automaticky uložia do adresára **\$temp** (ktorý sa obvykle nachádza v c:\temp alebo d:\temp) do súborov **query1.sql** až **query<n>.sql**.

Databáza a server.

Implicitne sa ako cieľová databáza a cieľový server určí **aktívna databáza** a server. Užívateľ si však v **Properties** môže nastaviť aby sa **Query** spúšťala v nejakej konkrétnej databáze na nejakom konkrétnom servri, bez ohľadu na to, čo je nastavené ako aktívne.

Samostatný proces a thread.

Patrik pred spustením **Query** naviaže nové napojenie (**connection**) na SQL Server, ktoré je identické s napojením, na ktorom sa mala **Query** pôvodne vykonať. Aplikácia spustí **Query** v samostatnom thread-e, čo zabezpečí zastaviteľnosť bežiacей **Query** užívateľom. Pri spustení sa v toolbare zobrazí tlačítko **Stop** (■), ktorým je možné bežiacu **procedúru** zastaviť (zastavenie sa dá zrealizovať aj klávesou **Pause**). Technicky sa to zrealizuje uzaverom napojenia (**disconnect**) a zastavením threadu.

Napojenie sa však uzavre aj pri štandardnom dobehnutí **Query**, takže nie napríklad možné písať **Query** na viackrát s postupným spúšťaním (napríklad, že by sa pri jednom spustení vytvorila temporárna tabuľka a pri druhom by sa do nej niečo išlo vkladať).

Množina výsledkov

Po spustení sa automaticky prepne do popredia záložka výsledkov **Results**. Tu sa zobrazujú priebežne získavané výsledky. Nakoľko výsledkov môže byť veľa, je každá vrátená tabuľka redukovaná na takú veľkosť, aby sa zmestila do pamäte určenej v **Options** v kolonke **Memory Usage**.

Results

nl_author_id	au_id	au_lname
int	char(11)	char(40)
1	172-32-1176	White
2	213-46-8915	Green
3	238-95-7766	Carson
4	267-41-2394	O'Leary
5	274-80-9391	Straight
7	409-56-7008	Bennet
8	427-17-2319	Dull
9	472-27-2349	Gringlesby
10	486-29-1786	Locksley
13	672-71-3249	Yokomoto
16	724-08-9931	Stringer
17	724-80-9391	MacFeather
18	756-30-7391	Karsen
20	846-92-7186	Hunter
21	893-72-1158	McBadden

(15 rows affected)
hallo

Záložka results obsahuje posledne vrátené výsledky od SQL Servra po spustení **Query / Procedúry** . Výsledky pozostávajú s dvoch typov informácií:

- tabučky - zobrazované čiernou farbou. Tabučka obsahuje okrem dát aj dodatočne informácie. V prvom riadku sa nachádzajú mená stĺpcov tabučky (samozrejme sa nemusí jedna o skutočnú tabučku, ale iba o akúsi fiktívnu, ktorá nemá ani mená stĺpcov - takú tabučku dostaneme napríklad do **Results** tak, že spustíme **Query SELECT I**), v druhom riadku sa nachádzajú dátové typy jednotlivých stĺpcov a až za predelovacou čiarou sa nachádzajú samotné dáta (ak ovšem príslušný SELECT vrátil neprázdnu tabučku). Počet dátových riadkov je tu redukovaný na takú veľkosť, aby sa dáta zmestili do pamäte určenej v **Options** v kolonke **Memory Usage**. Za dátami tabučky sa ešte nachádza informácia o počte riadkov a ak tabučka nebola načítaná celá, tak je tu uvedená aj zmienka o tejto skutočnosti.
- informačné a chybové hlásenia - zobrazované tmavočervenou farbou.

Results a Clipboard

Okno **Results** slúži ako **Read Only Edit Control**, čo znamená, že sa tu nachádza kurzor, ktorým je možno pohybovať a časť textu označovať do bloku a kopírovať ich pomocou klávesovej kombinácie **Ctrl C** do **clipboardu**.

Tabučky do Table View

Dáta tabuliek sú síce zobrazené a parciálne kopírovateľné do **clipboardu**, ale postrádajú komfort, ktorý majú ostatné tabučkové pohľady v **Patrikovi** (triedenie podľa rôznych stĺpcov,

clipboardové operácie na tabuľkovej úrovni). Dáta tabuľky sa však dajú do týchto efektívnejších tabuľkových pohľadov transformovať tak, že sa nad dátami stlačí pravé tlačítko myši, čím sa vyvolá **shortcut menu** obsahujúce jedinú položku : **View In Table**, po ktorej zvolení sa otvorí **tabuľkový pohľad** so žiadanými dátami.

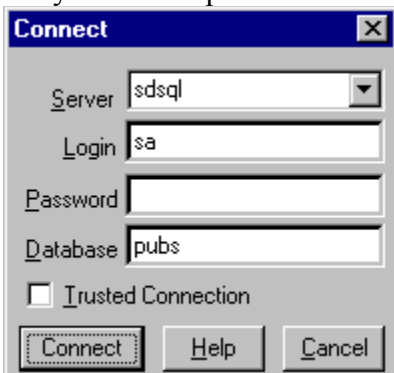
Táto funkcia je prístupná iba v prípade, že je v **Query Options** v kolonke **Result Output Format** nastavená voľba **Another Delimiter** .

Go To Error

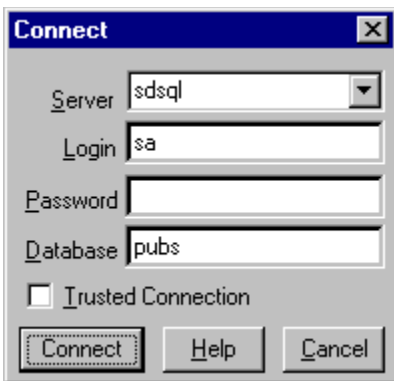
Ak pri spustení došlo ku chybovým hláseniam, tak nie je zväčša ťažké určiť na ktorom mieste **Query / Procedúry** nastali, lebo chybové hlásenia obsahujú číslo riadku. Ak ste však spustili query pozostávajúce s viacero batch-ov, tak je už ťažké určiť pozíciu chyby ťažké, lebo číslovanie riadkov sa pre každý batch začína od jednotky a nie teda od skutočného čísla riadku v texte. Ak sa však kurzorom postavíte na riadok obsahujúci chybové hlásenie a stlačíte klávesu F4, tak sa prepne do popredia záložka **Query** a kurzor sa v texte nastaví na riadok v ktorom chyba nastala (samozrejme aj s ohľadom na batch-e). Takýmto spôsobom sa dajú diagnostikovať chyby vo veľkých scriptoch.

Save, Print, Find, Find Next

Ak je záložka **Results** v popredí, tak vyššieuvedené tlačítka toolbaru slúžia na vykonávanie rovnomených funkcií pre **Results** a nie pre **Query**.




Save - uloží obsah okna do textového súboru.



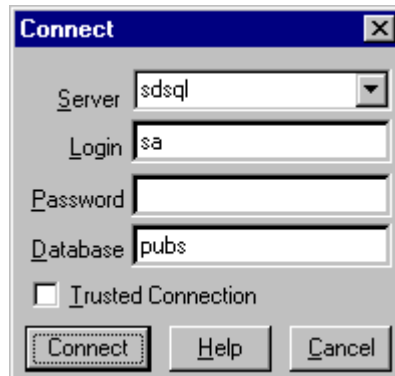
Print... - tlačí textu.


A screenshot of a 'Connect' dialog box. It has a title bar with 'Connect' and a close button. The dialog contains several fields: 'Server' with a dropdown menu showing 'sdsq', 'Login' with a text box containing 'sa', 'Password' with an empty text box, and 'Database' with a text box containing 'pubs'. Below these fields is a checkbox labeled 'Trusted Connection' which is unchecked. At the bottom are three buttons: 'Connect', 'Help', and 'Cancel'.

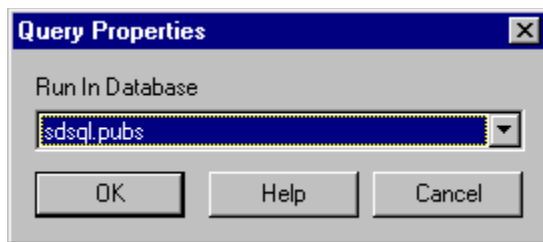
Find - vyhľadávanie (Alt F3)

 Find Next - nájde nasledujúci výskyt hľadaneého slova (F3)

Properties



Funkcia sa vyvoláva stlačením tlačítka  v toolbare Query Editora. Otvorí sa dialógové okno

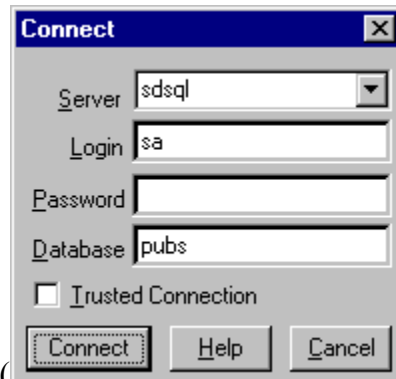


v ktorom je možné určiť, v akej databáze a na akom servri sa bude Query spúšťať. Obyčajne je tu nastavený text **<Auto>**, čo predstavuje, že sa query spustí v tej databáze a na tom servri, ktoré sú nastavené v hlavnom toolbare aplikácie. V combo boxe je možné vybrať si zo všetkých momentálne dostupných databáz.

Tlačítkom Query Options ja možné nastaviť ďalšie špecifické options Query Editora.

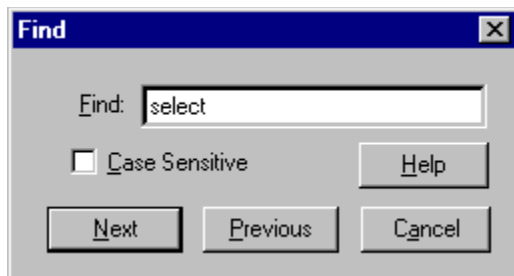
Find

Hľadanie textu v okne Query Editora



1. Zvoľte príkaz **Find** () z toolbaru.

Zobrazí sa dialógové okno **Find**.



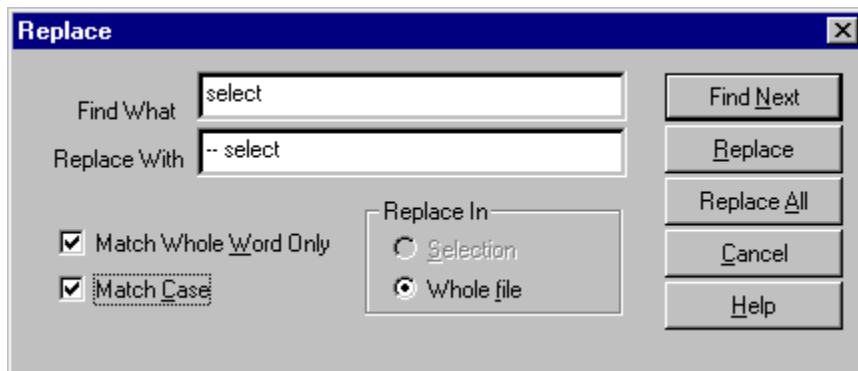
2. Do poľa **Find What** napíšte slovo, ktoré chcete nájsť.
3. Vyberte check box **Case Sensitive**, aby sa našiel text s rovnakými veľkými a malými písmenami, aké má text v poli **Find**.
4. Stlačte tlačítko **Next**, aby sa hľadal nasledujúci výskyt textu.
5. Stlačte tlačítko **Previous**, aby sa našiel predchádzajúci výskyt textu.

Replace

Vyhľadanie a nahradenie určitého slova alebo skupiny slov v okne Editoru.

1. Zvoľte príkaz **Replace** v shortcut menu editora.

Zobrazí sa dialógové okno **Replace**.



2. Do poľa **Find What** napíšte slovo, ktoré hľadáte
3. Do poľa **Replace With** napíšte slovo ktorým ho chcete nahradiť.
4. Vyberte check box **Match Whole Word Only**, aby sa našli celé slová, nie ich časti.
5. Vyberte check box **Match Case**, aby sa našiel text s rovnakými veľkými aj malými písmenami ako text v poli **Find What**.
6. Skupina **Replace In**
 - a) Zvoľte rádio tlačítko **Selection**, keď chcete nahradiť text vo výbere, ktorý ste predtým určili.
 - b) Zvoľte rádio tlačítko **Whole file**, ak chcete nahradiť text celého okna **Query Editoru**.
7. Urobte nasledovné:
 - a) Kliknite na tlačítko **Find Next**, čím sa text nezmení a hľadá sa ďalší text.
 - b) Kliknite na tlačítko **Replace**, čím sa nahradí text a hľadá ďalší text.
 - c) Kliknite na tlačítko **Replace All**, čím sa zmenia všetky texty bez potvrdenia.

Define vocabulary

Funkcia umožňuje nadefinovať slovnú zásobu, ktorá bude vypisovaná v Query Editore samostatnou farbou.

Keď sa nad tým zamyslíme, čo iné by mohla táto slovná zásoba obsahovať okrem názvov databáz, tabuliek, ich stĺpcov, prípadne dátových typov. Všetky tieto slová sa totiž v query editore nezvykne písať. Keďže táto slovná zásoba môže byť dosť rozsiahla, bolo by to asi dosť zdĺhavé všetko písať. Preto sa slovná zásoba definuje ako jeden, alebo viac SELECT-ov, ktoré sami vrátia množinu požadovaných slov. Všetky tieto slová sa totiž nachádzajú v **systemových tabuľkách** SQL servera, stačí len napísať správne SELECT-y.

1. Napíšte v **query editore** Transact SQL definíciu (SELECT-y) budúcej slovnej zásoby.

Príklady:

- a) **SELECT name FROM systypes**
určuje množinu všetkých dátových typov definovaných v aktuálnej databáze
- b) **SELECT name FROM master..sysdatabases**
určuje množinu všetkých databáz na SQL serveri
- c) **SELECT 'go'**
definuje slovo 'go'

3. Zvoľte príkaz **Define Vocabulary** z **Query menu** v menu **Query Editora**.

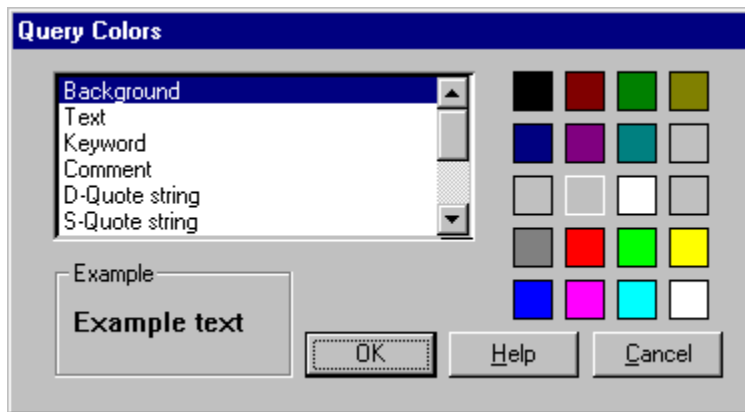
4. Slová určené výsledkami query jej spustením budú od tohoto okamžiku vypisované v Query Editore odlišnou farbou. Túto farbu je možné určiť príkazom **Colors...** v shortcut menu Query Editora. Jedná sa o syntaktickú skupinu **User Defined**.

Colors

Nastavenie farieb v Query editore.

1. Zvoľte príkaz **Color...** zo shortcut menu v **Query Editore**.

Zobrazí sa dialógové okno **Query Colors**.



2. Vyberte názov syntaktickej skupiny (triedy) slov a kliknite na farebý štvorček, aby sa mu pripísala farba. (Môžete vybrať z dvadsiatich farieb.)

3. Je možné zvoliť jednu z nasledujúcich tried slov:

Text - trieda nikde nezaradených slov

Keyword - kľúčové slová

Comment - text komentára

D-Quote String - reťazce ohraničené úvodzovkami

S-Quote String - reťazce ohraničené apostrofmi

Number - čísla

Variable - názvy premenných a temporárnych tabuliek

User Defined - užívateľom definovaná trieda slov

Ďalej je možné zmeniť:

Background - farba pozadia

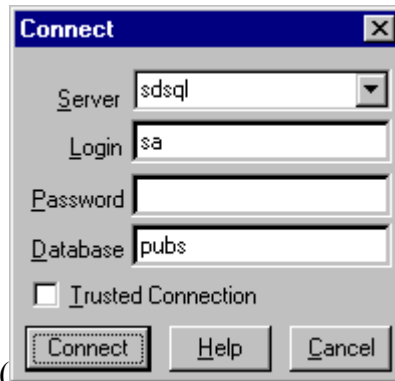
Block text - farba textu v označenom blok

Block Background - pozadie označeného textu

4. Keď ste ukončili výber, stlačte tlačítko **OK**.

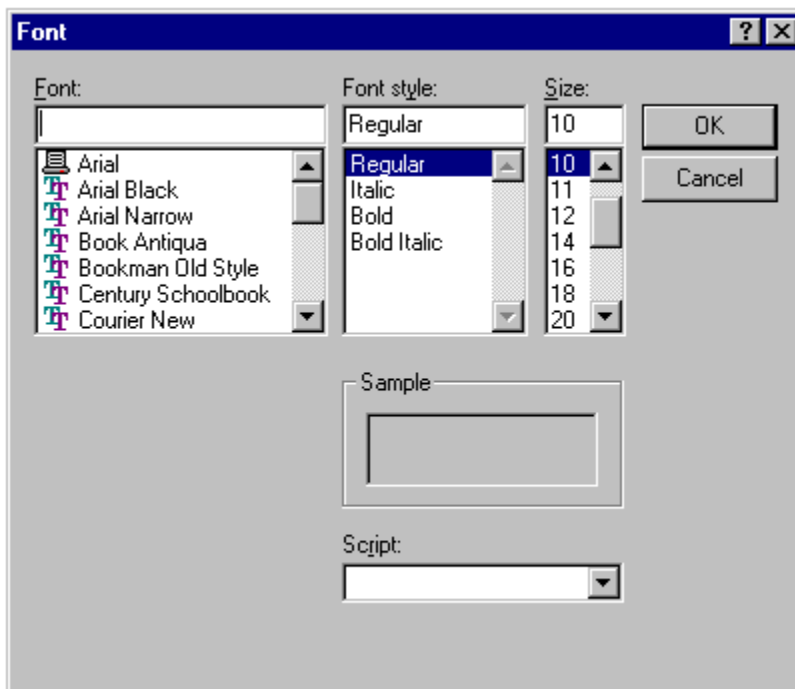
Print

Tlačenie textu napísaného v Query Editore alebo textu Resultov.



1. Zvoľte v toolbare tlačítko **Print** (

Zobrazí sa dialógové okno **Font**.



2. Vyberte font, font style a veľkosť fontu.
3. Stlačte tlačítko **OK**.

Parametre

Záložka Parameters Designéra Procedúr obsahuje definície parametrov procedúry.

	Name	Datatype	Output	Default Value	Running Value
1	@lolimit	money	No		
2	@hilimit	money	No	11.5	
3	@type	varchar[255]	No		
4					
5					
6					
7					
8					
9					

Parametre sa nachádzajú v tabuľke so stĺpcami:

- **Name** - meno premennej, ktorá reprezentuje parameter
- **Datatype** - dátový typ premennej
- **Output** - hodnote Yes/No definujúca, či je parameter výstupný, alebo nie.
- **Default Value** - default hodnota, ktorá sa parametru priradí, ak je procedúra zavolaná bez explicitnej špecifikácie hodnoty parametra
- **Running Value** - hodnota parametra, s ktorou sa procedúra spúšťa, ak je spúšťaná s Designéra ([Execute](#))
- **Description** - popis parametra, uchováva sa ako komentár v zdrojovom texte procedúry za definíciou parametra

Nie všetky stĺpce definujú štruktúru parametra. Stĺpec **Running Value** umožňuje zadať hodnoty pre parametre, ak sa procedúra s Designéra aj spúšťa - odlaňuje. Taktiež zmena nejakej hodnoty v tomto stĺpci sa nepočíta ako zmena v štruktúre procedúry (Nevyskočí teda hviezdička do ľavej časti titulku okna designéra).

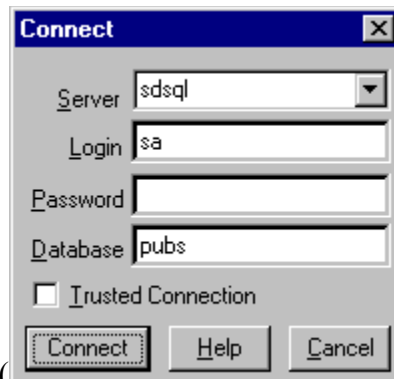
Tabuľkové vlastnosti

Tabuľka dodržiava väčšinu štandardných tabuľkových vlastností, ako je

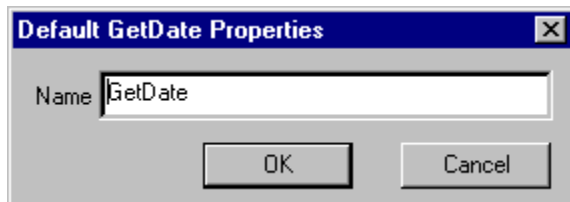
- funkcionálna **clipboard**
- rozširovanie/zužovanie riadkov/stĺpcov
- replikácia buniek

ale nepodporuje napríklad triedenie podľa stĺpca - dvojkliknutím na jeho názov.

Default, Rule - Properties

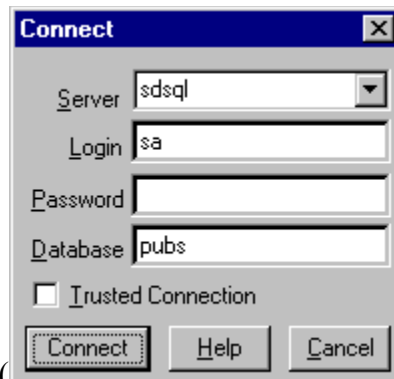


Po stlačení tlačítka **Propertis** (**Connect**) toolbaru designéra **default-u** alebo **rule-u** sa zobrazí dialógové okno

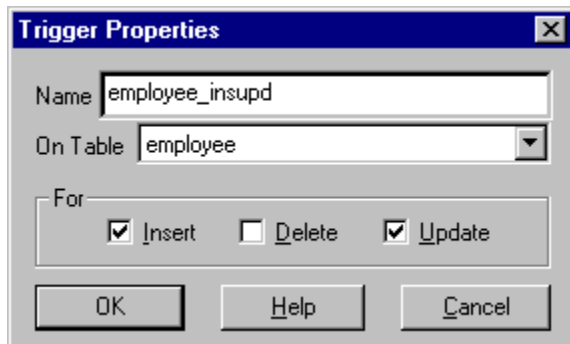


v ktorom je možné zmeniť názov **default-u**, alebo **rule-u**.

Properties Triggra



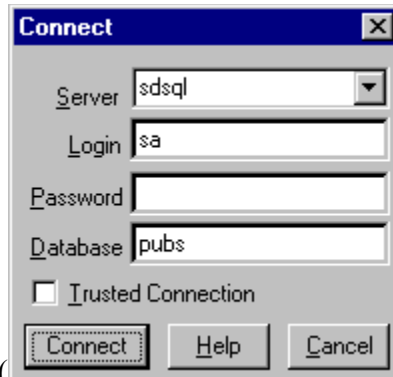
Po stlaèení tlačítka **Propertis** (**Connect**) toolbaru designéra triggra sa zobrazí dialógové okno



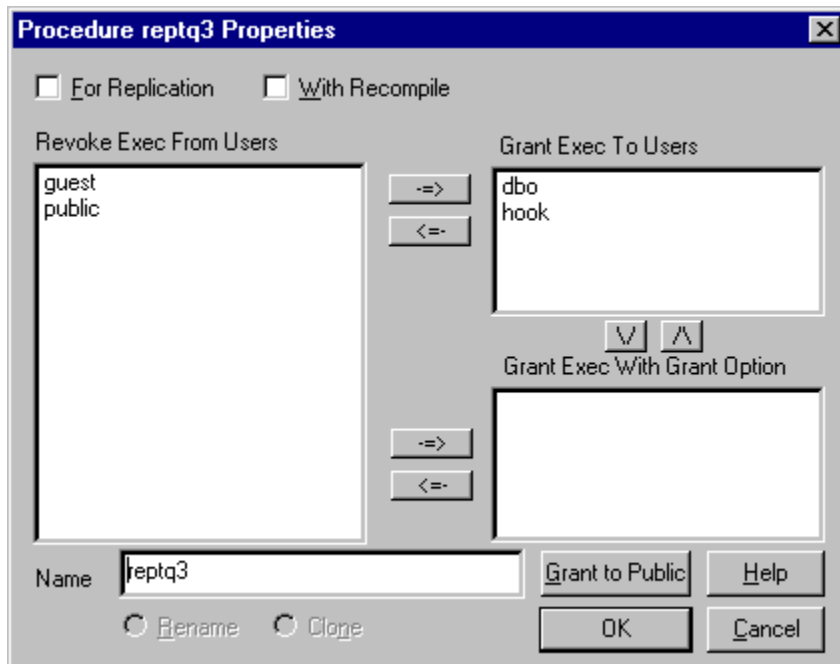
v ktorom je možné vidie/zmeni

- meno **triggra** - kolonka **Name**
- Tabu¾ku nad ktorou je trigger vytvorený - v **On Table**
- Typ akcie (**Update**, **Delete** a/alebo **Insert**), ktorú trigger validuje
- Tlačítkom Query Options ja možné nastavi¾ ía¾šie špecifické options editora.

Properties Procedúry a View-u.



Po stlaèení tlačítka **Propertis** (toolbaru designéra procedúry alebo **view-u** sa zobrazí dialógové okno



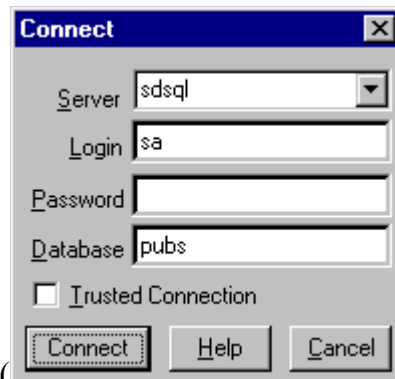
v ktorom je možné vidie/zmeni


- práva užívateľov na danú procedúru / view - v zjednotenú troch listboxov : **Revoke Exec From Users**, **Grant Exec To Users** a **Grant Exec With Grant Option** sa nachádzajú mená všetkých užívateľov (a grup) databázy. Pomocou tlačítok so šípkami je možné presunúť užívateľov tak, ako je potrebné. Tlačítko **Grant To Public** je iba shortcut na prehodenie grupy **public** do **Grant Exec To Users** - to sa totiž dosť často používa.
- meno procedúry / view-u. Ak meno zmeníte, povolia sa dve tlačítka - **Rename** a **Clone**. Voľba **Rename** zabezpečí premenovanie procedúry pri najbližšom uložení. Ak však chcete vytvoriť novú - podobnú procedúru, a nechcete ju písať od začiatku, tak zvolíte **Clone**. Pri najbližšom uložení sa vytvorí procedúra s novým menom. Počas ďalšej práce s designérom budete pracovať už iba s novou procedúrou.
- voľby **For Replication** a **With Recompile** . Tieto voľby sú zobrazené iba pre procedúru.

View sa s uvedenými vlastnosťami nedá vytvoriť.

- Tlačítkom Query Options je možné nastaviť ďalšie špecifické options editora.

Trigger, Default, Rule - Execute



Stlačením tlačítka **Execute** () toolbaru designéra triggra, default-u alebo rule-u, sa objekt uloží do pôvodnej databázy na pôvodný SQL Server.

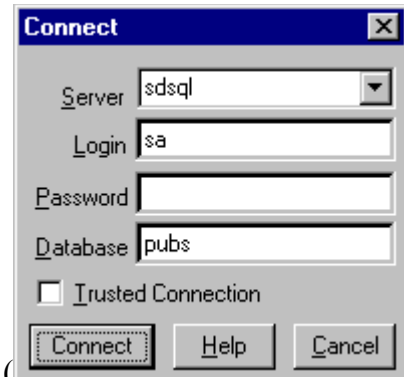
Množina výsledkov

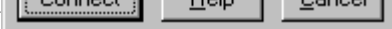
Pri ukladaní sa automaticky prepne do popredia záložka výsledkov **Results**. Tu sa zobrazujú prípadné chybové hlásenia. Ak sa objekt uloží bez chyby, ostane okno **Results** prázdne.

Transakcia

Ak pracujete na SQL Servri verzii aspoň 6.5 , tak sa objekt ukladá v transakcii, takže ak sa ho nepodarí uložiť, tak sa pôvodný objekt nedropne (nezmaže).

View - Execute



Ak bol **view** modifikovaný, tak stlačením tlačítka **Execute** () toolbaru designéra sa **view** uloží do pôvodnej databázy na pôvodný SQL Server.

Množina výsledkov

Pri ukladaní sa automaticky prepne do popredia záložka výsledkov **Results**. Tu sa zobrazujú prípadné chybové hlásenia. Ak sa **view** uloží bez chyby, ostane okno **Results** prázdne.

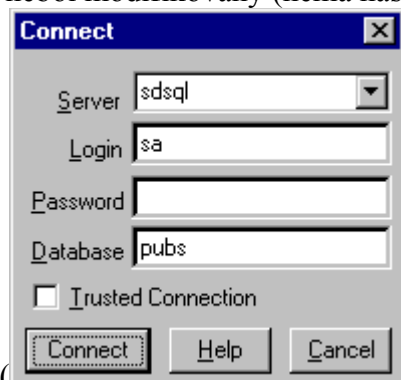
Transakcia

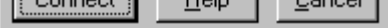
Ak pracujete na SQL Servri verzie aspoň 6.5 , tak sa **view** ukladá v transakcii, takže ak sa ho nepodarí uložiť, tak sa pôvodný **view** nedropne (nezmaže).

Príznak Modifikácie

Ak sa **view** podarilo úspešne uložiť, tak sa zruší **príznak modifikácie designéra** (zmizne hviezdička z 3/4avej časti titulku okna designéra).

Ak view nebol modifikovaný (nemá nastavený príznak modifikácie), tak stlačením tlačítka

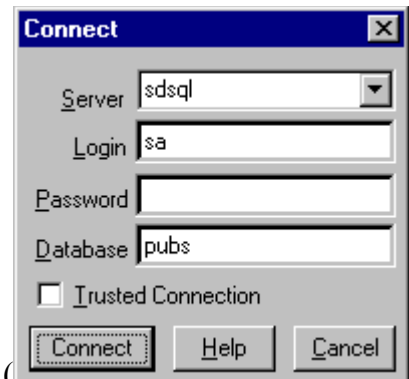


Execute () toolbaru designéra sa spustí

`SELECT * FROM <meno view-u>`

čím sa zobrazia do **Results** dáta, ktoré view reprezentuje.

Execute Procedúry



Ak bola **procedúra modifikovaná**, tak stlačením tlačítka **Execute** () **toolbaru** designéra sa **procedúra** uloží do pôvodnej databázy na pôvodný SQL Server.

Množina výsledkov

Pri ukladaní sa automaticky prepne do popredia záložka výsledkov **Results**. Tu sa zobrazujú prípadné chybové hlásenia. Ak sa **procedúra** uloží bez chyby, ostane okno **Results** prázdne.

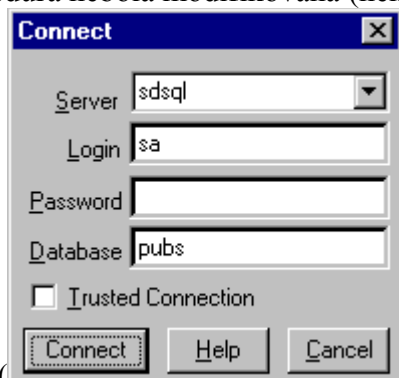
Transakcia

Ak pracujete na SQL Servri verzie aspoň 6.5 , tak sa **procedúra** ukladá v transakcii, takže ak sa ju nepodarí uložiť, tak sa pôvodná **procedúra** nedropne (nezmaže).

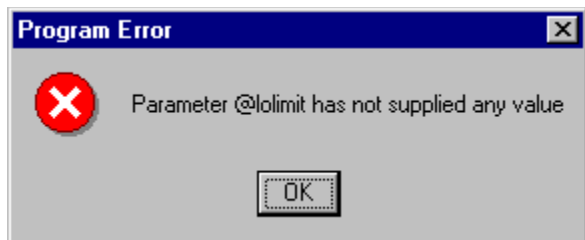
Príznak Modifikácie

Ak sa **procedúru** podarilo úspešne uložiť, tak sa zruší **príznak modifikácie designéra** (zmizne hviezdička z ľavej časti titulu okna designéra). Pri opätovnom spustení sa teda už neuloží, ale :

Ak **procedúra** nebola modifikovaná (nemá nastavený príznak modifikácie), tak stlačením tlačítka



Execute () **toolbaru** designéra sa spustí **procedúra** SQL príkazom **EXEC**. Ak má **procedúra** nejaké parametre (v záložke **Parameters**), tak musia mať špecifikované **Running Values - hodnoty**, ktoré sa **procedúre** pošlú ako parametre. Parametre ktoré nemajú špecifikované **Running Values** musia byť buď typu **OUTPUT** (výstupné parametre), alebo musia mať definovanú hodnotu v **Default Values**. Ak táto podmienka nie je splnená, **Príkaz** vypíše varovné hlásenie



a procedúru nespustí.

Ak sú definované hodnoty pre všetky parametre, tak sa procedúra spustí a výsledky procedúry sa priebežne zobrazujú do [Results](#). Nakoľko výsledkov môže byť veľa, je každá vrátená tabuľka redukovaná na takú veľkosť, aby sa zmestila do pamäte určenej v [Options](#) v kolonke [Memory Usage](#).

Output Parametre

Pri procedúrach s výstupnými parametrami sa používa na spúšťanie konštrukcia

```
DECLARE @i int
EXEC Procedure1 @i=@i OUTPUT
PRINT 'Values Of Output Parameters'
SELECT '@i'=@i
```

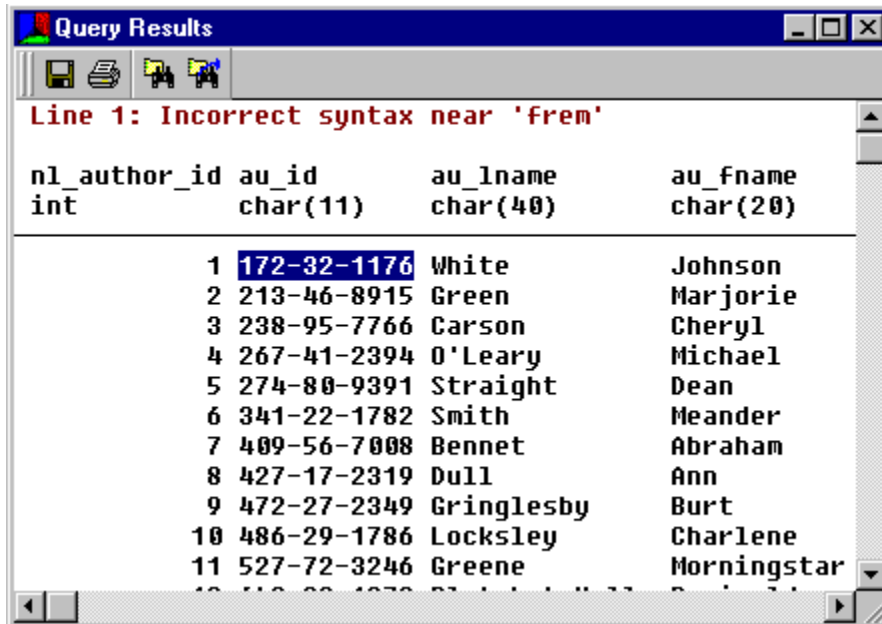
ktorá umožní zobraziť po spustení hodnoty týchto parametrov.

Samostatný proces a thread.

Patrik pred spustením procedúry naviaže nové napojenie ([connection](#)) na SQL Server, ktoré je identické s napojením, na ktorom sa mala procedúra pôvodne spustiť. Aplikácia spustí procedúru v samostatnom thread-e, čo zabezpečí zastaviteľnosť bežiacich procedúr užívateľom. Pri spustení sa v toolbare zobrazí tlačidlo **Stop** (■), ktorým je možné bežiacu procedúru zastaviť (zastavenie sa dá zrealizovať aj klávesou **Pause**). Technicky sa to zrealizuje uzaverom napojenia ([disconnect](#)) a zastavením threadu.

Okno Results

Okno slúži na prezentáciu vrátených výsledkov získaných spustením scriptu na SQL Servri alebo na zobazovanie iných textových výsledkov (napr. výsledky porovnania databáz).



The screenshot shows a window titled "Query Results" with a toolbar at the top. Below the toolbar, a red error message reads "Line 1: Incorrect syntax near 'frem'". Below the error message is a table with four columns: "nl_author_id", "au_id", "au_lname", and "au_fname". The first row of the table is highlighted in blue.

nl_author_id	au_id	au_lname	au_fname
int	char(11)	char(40)	char(20)
1	172-32-1176	White	Johnson
2	213-46-8915	Green	Marjorie
3	238-95-7766	Carson	Cheryl
4	267-41-2394	O'Leary	Michael
5	274-80-9391	Straight	Dean
6	341-22-1782	Smith	Meander
7	409-56-7008	Bennet	Abraham
8	427-17-2319	Dull	Ann
9	472-27-2349	Gringlesby	Burt
10	486-29-1786	Locksley	Charlene
11	527-72-3246	Greene	Morningstar

Výsledky môžu pozostávať s dvoch typov informácií:

- tabučky - zobrazované čiernou farbou. Tabučka obsahuje okrem dát aj dodatočné informácie. V prvom riadku sa nachádzajú mená stĺpcov tabučky (samozrejme sa nemusí jedna o skutočnú tabučku, ale iba o akúsi fiktívnu, ktorá nemá ani mená stĺpcov - takú tabučku dostaneme napríklad do **Results** tak, že spustíme **Query SELECT I**), v druhom riadku sa nachádzajú dátové typy jednotlivých stĺpcov a až za predelovacou čiarou sa nachádzajú samotné dáta (ak ovšem príslušný SELECT vrátil neprázdnu tabučku). Počet dátových riadkov je tu redukovaný na takú veľkosť, aby sa dáta zmestili do pamäte určenej v **Options** v kolonke **Memory Usage**. Za dátami tabučky sa ešte nachádza informácia o počte riadkov a ak tabučka nebola načítaná celá, tak je tu uvedená aj zmienka o tejto skutočnosti.
- informačné a chybové hlásenia - zobrazované tmavočervenou farbou.

Results a Clipboard

Okno **Results** slúži ako **Read Only Edit Control**, čo znamená, že sa tu nachádza kurzor, ktorým je možno pohybovať a časť textu označovať do bloku a kopírovať ich pomocou klávesovej kombinácie **Ctrl C** do **clipboardu**. Označenie celého textu je možné pomocou **Ctrl A**.

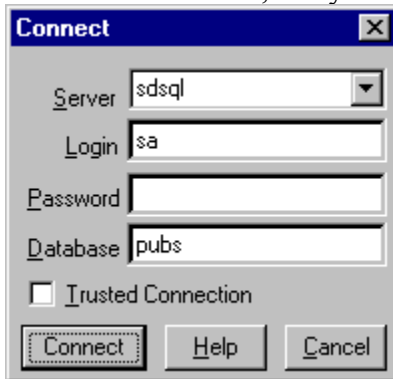
Tabučky do Table View

Dáta tabuliek sú síce zobrazené a parciálne kopírované do **clipboardu**, ale postrádajú komfort, ktorý majú ostatné tabučkové pohľady v **Patrikovi** (triedenie podľa rôznych stĺpcov, **clipboardové** operácie na tabučkovej úrovni). Dáta tabučky sa však dajú do týchto

efektívnejších tabuľkových pohľadov transformovať tak, že sa nad dátami stlačí pravé tlačítko myši, čím sa vyvolá **shortcut menu** obsahujúce jedinou položku : **View In Table**, po ktorej zvolení sa otvorí tabuľkový pohľad so žiadanými dátami.

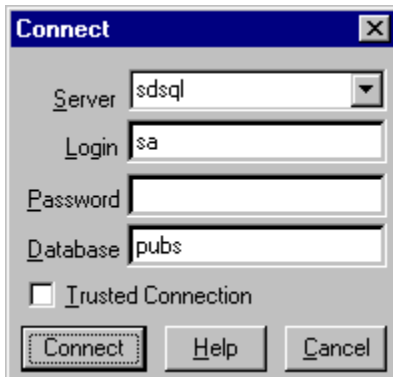
Save, Print, Find, Find Next

V okne sa nachádza toolbar, ktorý obsahuje tlačítka :



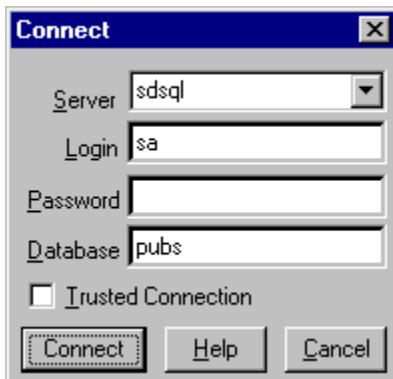
The screenshot shows a 'Connect' dialog box with the following fields: Server (dropdown menu with 'sdsq' selected), Login (text box with 'sa'), Password (empty text box), and Database (text box with 'pubs'). There is an unchecked checkbox for 'Trusted Connection' and three buttons at the bottom: 'Connect', 'Help', and 'Cancel'. The 'Connect' button is highlighted with a dotted border.

Save - uloží obsah okna do textového súboru.



The screenshot shows the same 'Connect' dialog box as above. The 'Print...' button is highlighted with a dotted border.

Print... - tlačítko textu.



The screenshot shows the same 'Connect' dialog box as above. The 'Find' button is highlighted with a dotted border.

Find - vyhľadávanie (Alt F3)

A screenshot of a 'Connect' dialog box. The title bar is blue with the text 'Connect' and a close button. The dialog has a light gray background. It contains four input fields: 'Server' with a dropdown menu showing 'sdsq', 'Login' with the text 'sa', 'Password' which is empty, and 'Database' with the text 'pubs'. Below these fields is a checkbox labeled 'Trusted Connection' which is unchecked. At the bottom are three buttons: 'Connect' (highlighted with a dashed border), 'Help', and 'Cancel'.

Find Next - nájde nasledujúci výskyt hľadaneého slova

(F3)

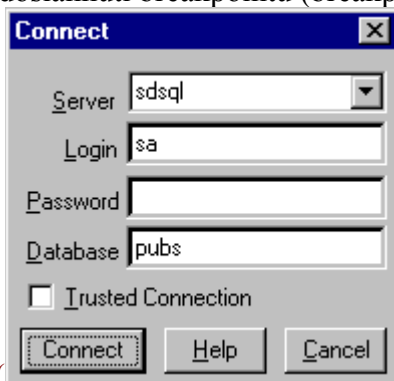
Debugger

Debugger je možné spustiť klávesami

- **F11** – spustenie debugera a zastavenie sa hneď pred prvým príkazom query/procedúry.



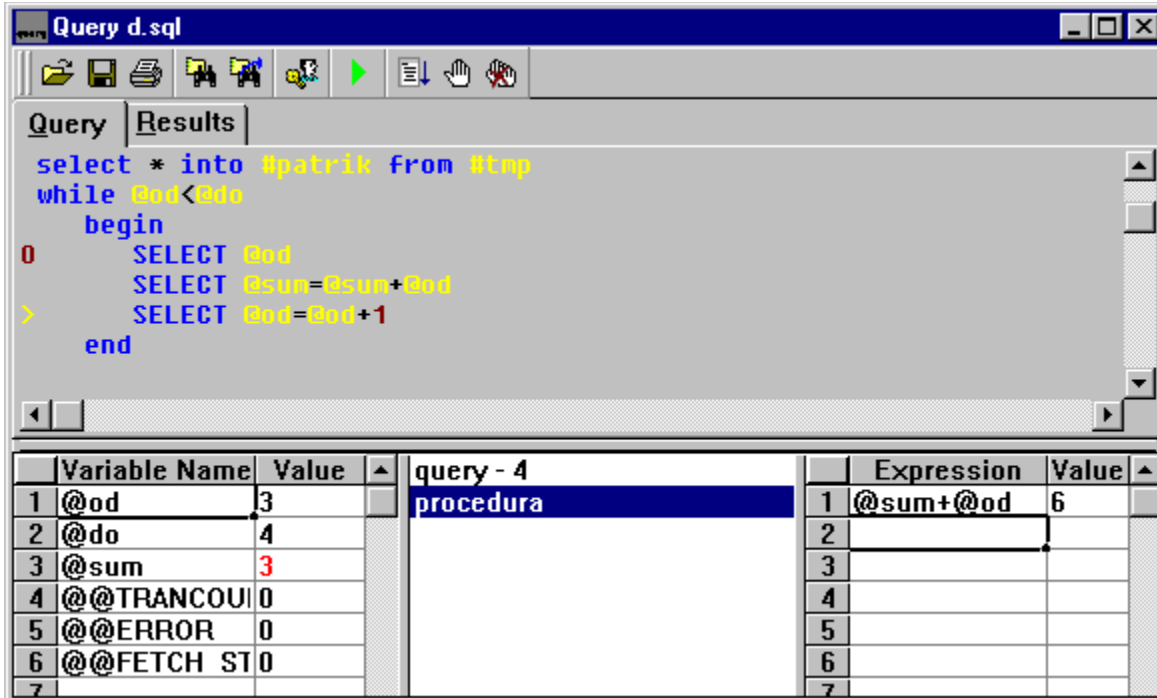
- **F5** () – spustenie query/procedúry (ďalej v texte budem používať len : procedúry) v debugeri. Procedúra sa spúšťa krok po kroku a pri volaní ďalšej procedúry, debuguje aj volanú procedúru. Beh sa **zastaví** v troch prípadoch:
 - Pri dosiahnutí konca procedúry
 - Pri dosiahnutí breakpointu (breakpointy sa v daných riadkoch pridávajú/rušia klávesou



- **F9** () prípadne sa dá vyvolať [editor breakpointov](#) klávesou **Ctrl B**.
- Pri manuálnom stopnutí klávesou **Pause/Break** alebo stlačením tlačítka **Stop** (■) v toolbare.
- **F7** - ako **F5** ale vykonávanie sa zastaví aj pri dosiahnutí riadku, v ktorom sa nachádzal kurzor, keď bola funkcia vyvolaná.

Ak je debugger spustený, ale výkon procedúry/query je zastavený, tak riadok v ktorom sa nachádza je označený žltým symbolom “>” na 3/4avej strane textu.

V spodnej časti Query okna sa počas debugovania nachádzajú 3 podokná.



V prvom (popisujem z³/₄ava doprava) sa nachádza zoznam parametrov procedúry, použitých premenných a užitočných globálnych premenných spolu s hodnotami, ktoré sú v danom okamihu premenným priradené.

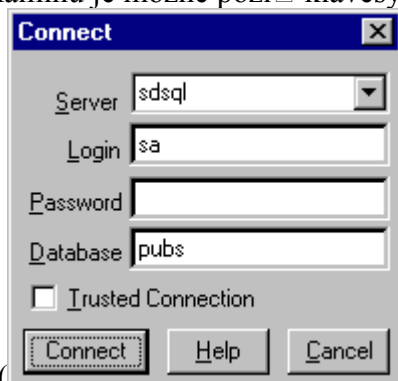
V druhom podokne sa nachádza stack volaní procedúr.

Tretie podokno slúži na **Watch** výrazov. Pre každý výraz uvedený v prvom stĺpci tabuľky sa v jej druhom stĺpci nachádza jeho aktuálna hodnota. Hodnota výrazu sa zisťuje konštrukciou :

SELECT výraz . Uvádzam príklady výrazov :


- @a+@b
- COUNT(*) FROM authors
- 1



V tomto okamihu je možné požiť klávesy / tlačítka toolbaru:




· Alt F5 () – ukončenie debugovania



· **F5** () – pokračovanie vykonávania procedúry (vyššie opísané)

· **F8** (), **F11** () – vykonanie aktuálneho príkazu procedúry. Ak sa jedná o príkaz *EXEC procedura* tak podš4a klávesy vlezie do procedúry (**Step In** – **F8**), alebo spustí procedúru (**Step Over** – **F11**) bez toho, aby do nej vlezol. Pri každom takto vykonanom segmente sa refeesh-nú v okne obsahy všetkých premenných a hodnoty všetkých sledovaných výrazov (**Watch**) – toto môže trochu spomalova□, takže skúste odtiaľ zbytoèné výrazy priebežne vyhadzova□.

· **F7** () - pokračovanie vykonávania so zastavením na riadku s kurzorom.

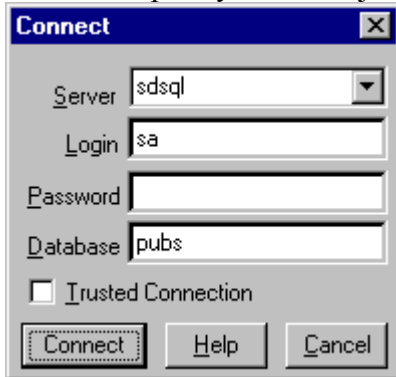
Obmedzenia

1. Triggre nie je možné debugova□.
2. Pri konštrukcii *INSERT .. EXEC* sa nedá urobi□ **Step In** .
3. Nie je možné robi□ **Step In** pri volaní systémových procedúr a pri volaní procedúr na inom servri.
4. Na rozdiel od klasicky spúš□aných procedúr, debugované procedúry nebežia na klonovanom procese, takže je možné browsova□ dáta tabuliek bez obáv s **block**-nutia **procesu**. Poèas debugovania sa neodporúèa používa□ zložitejšie funkcie (upsized, compare apod.)

Editor Breakpointov

Patrik podporuje tri druhy breakpointov

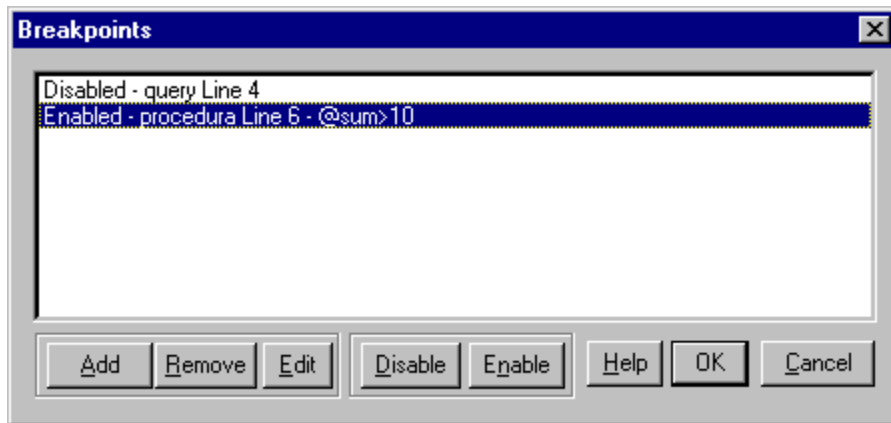
- **Pozičné** - výkon debugovanej procedúry sa zastaví **pred** vykonaním **riadku** s breakpointom. Tieto breakpointy sa vkladajú/rušia v texte klávesou **F9** alebo tlačítkom



v **toolbar** editora. Pozičné breakpointy sú v texte zobrazené symbolom “O” bordovej farby na $\frac{3}{4}$ avej strane textu.

- **Kondičné** - množina podmienok (**boolových** výrazov). Výkon procedúry sa zastaví, ak má **aspoň** jeden výraz logickú hodnotu **TRUE**. Každý kondičný breakpoint sa musí viazať na nejakú **procedúru** v ktorej (a len v ktorej) sa bude jeho platnosť testovať.
- **Obojaké** - **kondičné** breakpointy viazané na konkrétne **miesto** konkrétnej procedúry.

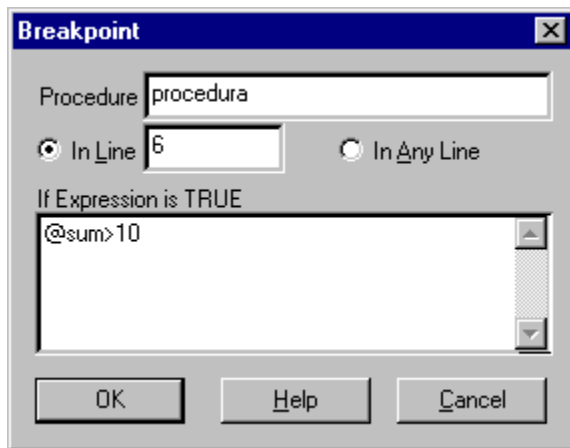
Všetky tieto typy breakpointov je možné spravovať v **editore breakpointov**, ktorý sa dá vyvolať klávesovou kombináciou **Ctrl B**.



Vybraný breakpoint je možné dočasne zakázať a neskôr povoliť tlačítkami **Enable**, **Disable**. Tlačítkami Add, Edit, Remove je možné pridávať nové, editovať/mazať existujúce breakpointy.

Editor Breakpointu

Pridanie nového breakpointu alebo edit existujúceho sa realizuje v **dialógovom okne**



s kolonkami

- **Procedure** - meno procedúry v ktorej bude breakpoint aktívny (ak chceme aby bol breakpoint aktívny v debugovanej query tak sem uvedieme text “**query**”).
- **In Line** - ak sa jedná o breakpoint viazaný na konkrétny riadok procedúry/query, tak je toto radio tlačítko zaškrtnuté a v edit kolonke je uvedené číslo daného riadku.
- **In Any Line** - ak breakpoint nie je viazaný na konkrétny riadok procedúry, tak je toto tlačítko zaškrtnuté.
- **If Expression Is TRUE** - obsahuje podmienku (ak sa ovšem jedná o kondičný , alebo obojaký breakpoint) . Ak sa v kolonke nenachádza žiaden text, tak sa jedná o pozičný breakpoint.

Príklady podmienok

- `0<(SELECT COUNT(*) FROM #tab WHERE a=@a)`
- `@@TRANCOUNT>2`
- `@A+@B<@C`

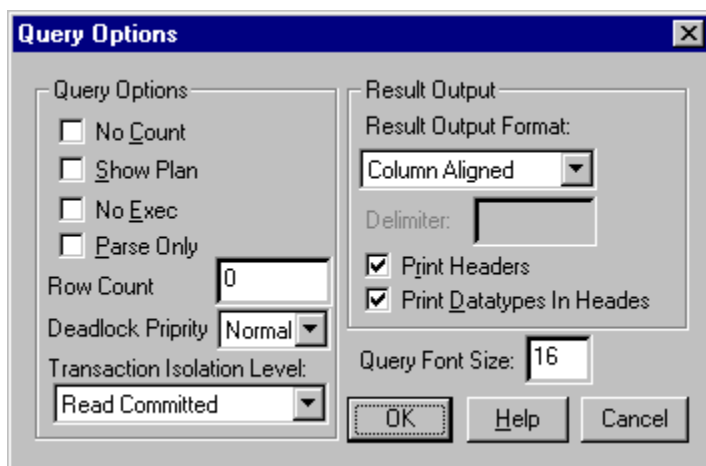
Query - More Options

Lokálne a Globálne query options

Každé otvorené **Query Editor** okno má svoje lokálne options. Modifikácia lokálnych options je prístupná cez Properties v editore query (procedúry).

Globálne query options sa dajú nastaviť v Options. Lokálne options Query editorov otvorených po prestavení globálnych options, budú mať lokálne options rovnako modifikované. Pri prestavení globálnych options sa Patrik opýta, či má rovnako prestaviť aj lokálne options všetkých momentálne otvorených query editorov.

Dialógové okno Query Options



Kolony query options:

- **No Count** - potlačenie výpisu počtu riadkov za jednotlivými tabuľkami výstupu
- **Show Plan** - optimalizačný výstup jednotlivých SQL dotazov
- **No Exec** - pri spúšťaní prebehne iba kompilácia nie však samotné spustenie a používa sa to spolu s **Show Plan**.
- **Parse Only** - namiesto spustenia prebehne iba syntaktická analýza
- **Row Count** - obmedzenie počtu spracovávaných riadkov v dotazoch. 0 = neobmedzený počet.
- **Deadlock priority** - priorita procesu pri dosiahnutí deadlocku.
- **Transaction Isolation Level** - viď SQL Books Online.

Kolony Result Output:

- **Column Output Format** - formát tabuliek v Results.
 - **Column Aligned** - jednotlivé stĺpce sú odsadené. Šírku stĺpca sa určuje textovo najdlhšia hodnota v stĺpci.
 - **Tab Delimited** - hodnoty v jednotlivých stĺpcoch sú oddelené tabulátormi.
 - **Another Delimiter** - hodnoty jednotlivých stĺpcov sú oddelené znakom uvedeným v kolonke Delimiter popísanej nižšie.
- **Delimiter** - znak, delimitujúci hodnoty v stĺpcoch v okne Results.
- **Print Headers** - tabuľky v options obsahujú hlavičku s menami jednotlivých stĺpcov a

prípadne aj s ich dátovými typmi.

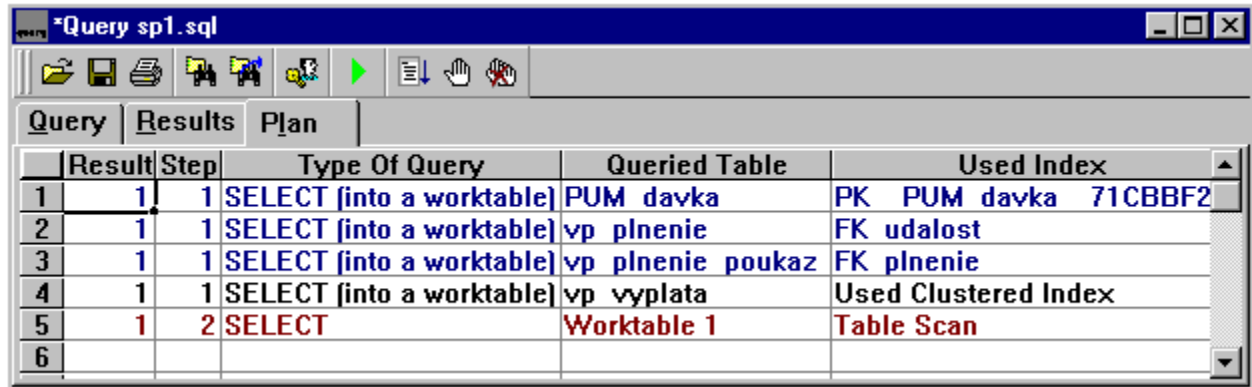
- **Print Datatypes In Headers** - výpis dátových typov jednotlivých stĺpcov v hlavičkách tabuliek.

Kolónka Font Size

- **Query Font Size** - veľkosť fontu v query editore.

Show Plan

Záložka pozostáva s tabuľky s informáciami o optimálnosti posledne spustenej query / procedúry.



Result	Step	Type Of Query	Queried Table	Used Index
1	1	1 SELECT [into a worktable]	PUM davka	PK PUM davka 71CBBF2
2	1	1 SELECT [into a worktable]	vp plnenie	FK udalost
3	1	1 SELECT [into a worktable]	vp plnenie poukaz	FK plnenie
4	1	1 SELECT [into a worktable]	vp vyplata	Used Clustered Index
5	1	2 SELECT	Worktable 1	Table Scan
6				

Každý (data modification language) príkaz (**vracajúci Result**) je spracovávaný v jednom, alebo vo viacerých krokoch (**Step**). Príkaz je typu, uvedenom v stĺpci **Query Type** (netreba byť zmatený informáciou že SELECT môže mať query type INSERT, je to v dôsledku postupného spracovávania dotazu prechodom cez rôzne pracovné tabuľky).

Dôležitá informácia je uvedená v stĺpci **Used Index**. Typ tejto hodnoty určuje farbu celého riadku. Možné hodnoty sú (utriedené podľa optimálnosti):

- **Used Clustered Index** - na optimalizáciu bol použitý Clustered Index, čo znamená maximálnu optimálnosť daného kroku query.
- **<Meno Indexu>** - na optimalizáciu bol použitý index <meno indexu> .
- **Table Scan** - na optimalizáciu nebol použitý žiadny index, a teda musela byť prescanovaná celá tabuľka.
- **Dynamic Index** - pri spracovávaní dotazu bol vytvorený pomocný dynamický index, ktorý bol po vyhodnotení zrušený.

Prvé dve možnosti znamenajú optimálny krok dotazu a posledné dve neoptimálne (použitá červená farba v riadku).

Command Line

Pri spúšaní Patrik-a s **Command Line** sa dajú v 3/4ubovo3/4nom poradí použiť nasledovné parametre:

Automatické napojenie na SQL Server

-Sserver_name0 [-Ddatabase_name0 -Uuser_name0 -Puser_password0 -T] zabezpečí automatické napojenie na SQL server. Parameter **-T** je potrebné použiť, ak sa bude jednať o trusted connection. Ak nešpecifikujete **-D**, **-U**, **-P**, tak sú použité default hodnoty: **user=sa**, **password** prázdny reťazec za aktuálnu **databázu** bude nastavená **default databáza** užívateľa **sa**. Ďalej sa dajú použiť parametre:

-S1server_name1 [-D1database_name1 -U1user_name1 -P1user_password1 -T1], pre ktoré platí to, čo pre tie vyššie uvedené.

Ak sa obe napojenia na server prebehnú úspešne, tak Patrik zvolí za aktívne (aktuálne) napojenie **server0.database0**.

Po napojení na server aplikuje ostatné Command Line parametre, a síce:

Načítanie súbrů do query editora.

-Qfile_name otvorí query okno a do neho nahrá súbor **file_name**

-Lsaved_downsize loadne pre aktuálnu databázu (!!!! Treba ju nastaviť parametrom **-D** !!!!) uložený downsize a upsized-ne ho bez akéhoko3/4vek pýtania sa a bez Save Downsize (v prípade, že je v options nastavené **Save Downsizes**). Ak pri generovaní Upsize scriptu nastane nejaká chyba tak sa vypíše okno s chybou a Patrik ostane otvorený. Ak však žiadna chyba nenastane, tak sa Patrik zavrie. Parameter **-L** je výhodné pre jednoduchú synchronizáciu štruktúry produkčnej databázy s vývojom, pri ktorej sa automaticky ukladajú Downsize-nuté databázy pred funkciou Upsize. Nie je totiž potrebné sedieť pri Patrikovi a postupne ručne spúšťať funkcie **Load downsize, Upsize, Load downsize, Upsize, Load downsize, Upsize** Stačí vygenerovať povelový (.BAT) súbor, ktorý bude spúšťať Patrik-a s Parametrom **-L**

Nasledovné parametre slúžia na zautomatizovanie istých činností, ktoré môže vykonať aj osoba neznajúca Patrik-a spustením nejakého povelového súboru, v ktorom sa Patrik spustí s jedným z nasledujúcich parametrov.

Porovnanie dát vybraných tabuliek dvoch databáz

-Cddata_compare_settings_file_name.tse Slúži na command line porovnanie dát vybraných tabuliek. Vykoná činnosť, ktorú by užívateľ vykonal nasledovnou sériou úkonov:

1. Nastavil si ako aktuálny server a databázu **server_name0, database_name0**.
2. Z menu **Current Database/Compare** zvolil Data in tables ...
3. V otvorivšom sa Dialog Boxe by v Menu **File** zvolil **Load settings** a nahral súbor **data_compare_settings_file_name.tse** (inak špecifikovaný za **-Cd** parametrom)
4. Spustil porovnávanie tlačítkom **Start**.
5. Vygenerované výsledky by uložil do súboru **server_name0_database_name0_dserver_ddb.txt** v adresári, ktorý bol špecifikovaný v Compare Dialog Boxe v kolonke **In Directory**. Ak nebol špecifikovaný žiaden adresár, súbor sa vytvorí v aktuálnom adresári. Server **dserver** a databáza **ddb** špecifikuje databázu a server s ktorou bola aktuálna databáza porovnávaná. Boli špecifikované v **.tse** súbore, ktorý

bol v kroku 3. aplikovaný.
6. Zavrel by Patrik-a.

Porovnanie štruktúr dvoch databáz

-CS Slúži na porovnanie štruktúr dvoch databáz. Bude sa jednať o **server0.database0** a **server1.database1**. Výsledky budú uložené v súbore **strdiff_server0_database0__server1_database1.txt** aktuálneho adresára. Po porovnaní sa Patrik automaticky zavrie.

Porovnanie procedúr dvoch databáz

-CP Slúži na porovnanie zdrojových kódov všetkých stored procedúr dvoch databáz. Bude sa jednať o **server0.database0** a **server1.database1**. Výsledky budú uložené v súbore **codediff_proc_server0_database0__server1_database1.txt** aktuálneho adresára. Pre každú procedúru ktorá je rozdielna v nejakom syntakticky dôležitom mieste bude súbor obsahovať jeden riadok s popisom prvého takého miesta. Po porovnaní sa Patrik automaticky zavrie.

Porovnanie triggrov dvoch databáz

-CT Slúži na porovnanie zdrojových kódov všetkých triggrov dvoch databáz. Bude sa jednať o **server0.database0** a **server1.database1**. Výsledky budú uložené v súbore **codediff_trig_server0_database0__server1_database1.txt** aktuálneho adresára. Pre každý trigger ktorý je rozdielny v nejakom syntakticky dôležitom mieste bude súbor obsahovať jeden riadok s popisom prvého takého miesta. Po porovnaní sa Patrik automaticky zavrie.

Porovnanie scriptu obsahujúceho procedúry s procedúrami v databáze

-CF **script_file_name.sql** Slúži na porovnanie zdrojových textov procedúr a iných zdrojovo-kódových objektov, zoskriptovaných v súbore, s tými, ktoré sa nachádzajú na serveri **server_name0** v databáze **database_name0**. Patrik vygeneruje súbor **codeffiff_server0_database0.txt** ktorý bude obsahovať výpisy o nezrovnalostiach podobne ako súbor generovaný pri -CT alebo -CP.
Po vykonaní sa Patrik zavrie. Táto funkcia nemá svoj non-command line equivalent.

Synchronizácia databáz

-CY **synchronize_settings_file_name.tse** Slúži na synchronizáciu celej, alebo časti databázy. Vykoná činnosť, ktorú by užívateľ vykonal nasledovnou sériou úkonov:

1. Nastavil si ako aktuálny server a databázu **server_name0, database_name0**.
2. Z menu **Current Database** zvolil Synchronize
3. V otvorivšom sa Dialog Boxe by v Menu File zvolil **Load settings** a nahral súbor **synchronize_settings_file_name.tse** (inak špecifikovaný za -CY parametrom)
4. Spustil synchronizáciu tlačítkom **Start**.
5. Zavrel by Patrik-a.

Offline export databázy

-CI **offline_export_settings_file_name.tse** Slúži na offline export celej, alebo časti databázy. Vykoná činnosť, ktorú by užívateľ vykonal nasledovnou sériou úkonov:

1. Nastavil si ako aktuálny server a databázu **server_name0, database_name0**.
2. Z menu **Current Database/Transfer** zvolil Offline Export ...

3. V otvorenom sa Dialog Boxe by v Menu File zvolil **Load settings** a nahral súbor **offline_export_settings_file_name.tse** (inak špecifikovaný za -CI parametrom)
4. Spustil offline export tlačítkom **Transfer**.
5. Zavrel by Patrik-a.

Offline import databázy

-CJ**directory** spustí offline import s adresára **directory**.

Textový export a import údajov

Funkci Offline Exopr t ukladá dáta tabuliek do bulkcopy (. **bcp**) súborov , ktoré zase Offline Import importuje do vytváraných tabuliek. Keiže sa jedná o binárne súbory, tak problém nastane, ak sa takto prenášajú údaje medzi serverami s **Pentium** a s **Alphou** , lebo uvedené procesory podporujú rozdielne kódovanie binárnych dát. Vo¾ba **-FC** v command line-e zabezpeí, že **.bcp** súbory sa nebudú generova a chápa ako binárne , ale ako textové , èo vyrieši naèrtnutý problém.

Debug režim

Ak je aplikácia spustená so switchom **-debug**, tak sa funkcia Upsize najsôr vždy opýta, èi má vytvori iba **modifikaèný script**, alebo ho aj aplikova a následne zmaza pracovnú databázu.

Help

Posledný podporovaný parameter je **-?**, ktorý vypíše zoznam možných použite¾ných parametrov.

Patrik Server Options

Databáza **patrik** na SQL servri obsahuje tabuľku **options**, v ktorej sa nachádzajú dva riadky. Oba špecifikujú správanie sa Patrik-a na danom servri.

Riadok **Save downsizes** špecifikuje, či sa pred každým **Upsize-om** má automaticky uložiť downsize-nutá databáza funkciou **Save Downsize** do súboru **dwms_server_databaza.pdbxxxx** kde **databáza** je meno databázy, do z ktorej vznikol downsize a **xxxx** je štvorciferné číslo, ktoré označuje verziu súboru. Začína sa číslom 0001 a pri každom ďalšom upsize-e danej databázy sa vytvorí súbor s číslom o jedno väčším ako ten posledný. Súbor sa vytvorí do adresára, ktorý je špecifikovaný v stĺpci **opt_text** v riadku **Save downsizes**. Automatické ukladanie downsize-ov bude vykonávané vtedy a len vtedy, ak stĺpec **opt_int** obsahuje nenulové číslo.

Riadok **Don't rollback (in trigger) in nested transaction** špecifikuje, ako sa majú generovať trigger, ktoré zabezpečujú referenčnú integritu. Ak je v tomto riadku a v stĺpci **opt_int** nenulová hodnota, tak sa pri generovaní zdrojového textu triggeru pred každý **ROLLBACK TRAN** vloží podmienka **IF @@TRANCOUNT=1**. Tým nastane, situácia, že Patrikom generovaná časť triggerov nerollback-ne pokus o porušenie referenčnej integrity, ak tento bežal v Transakcii. Ukázalo sa totiž nevýhodné aby trigger rollback-oval nested transakcie, lebo **ROLLBACK TRAN** s triggeru vykoná rollback až po prvý **BEGIN TRAN**, čo si nemusí uvedomiť procedúra, ktorá spustila trigger, čím môže zapríčiniť nekonzistentný stav tabuliek. Riadok **Historic Tables Indexes** špecifikuje aké indexy sa generujú pre historické tabuľky. Táto informácia je zadaná v stĺpci **opt_text** tabuľky **reazcom**, ktorý pozostávajúcim zo znakov :

- p - generovať index pre primárny kľúč historickej tabuľky
- i - generovať index pre dedený primárny kľúč sledovanej tabuľky
- d - generovať index pre dátum zmeny
- c - generovať index pre stĺpec s poradovým číslom modifikovaného atribútu

Ak reazec pozostáva so všetkých uvedených znakov, tak sa budú pre historické tabuľky generovať všetky indexy a ak iba niektoré, tak iba indexy korešpondujúce s uvedenými znakmi.

Stĺpec **last_update** v tabuľke **options** obsahuje NT user name užívateľa, ktorý nejakú hodnotu v riadku zmenil ako posledný.

Tip:

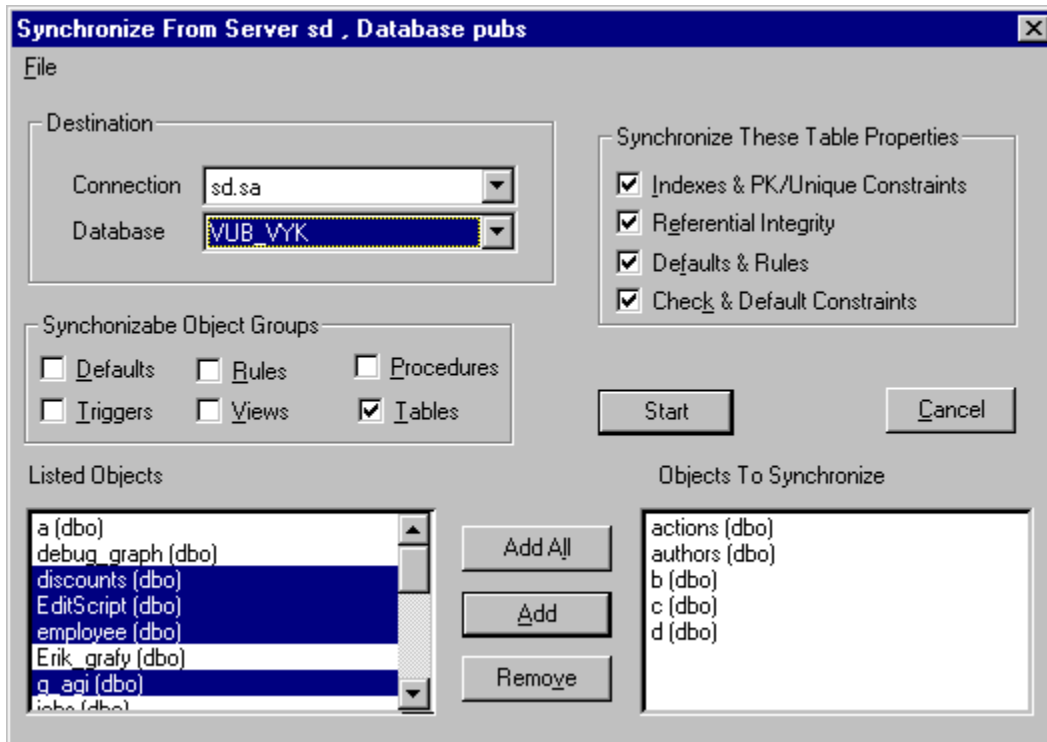
Ak sa užívateľ rozhodne pregenerovať všetky trigger nejakkej databázy, po tom, ako zmenil Option **Don't rollback (in trigger) in nested transaction** tak by mal postupovať nasledovne:

1. Zmeniť option.
2. Downsizenúť danú databázu
3. Funkciou **Current Database/Touch** nastaviť pregenerovanie všetkých triggerov.
4. Spustiť Upsize späť do pôvodnej databázy.

Synchronizácia štruktúry databáz.

Menu **Current Database** obsahuje funkciu **Synchronize**, ktorej úlohou je štruktúrovo zosynchronizovať databázy obsahujúce dáta (bez ich straty). Jedná sa o najširšiu synchronizáciu, ktorá sa dá vykonať bez nejakej nejednoznačnosti. Ďo sa týka tabuliek, tak synchronizované sú len tie z vybraných, ktoré buď neexistujú v synchronizovanej databáze a tie, ktoré majú v oboch databázach rovnaké atribúty, ich dátové typy a poradie. Ide teda o synchronizáciu **indexov**, **nabíndovaných defaultov** a **rules**, nadefinovaných **constraintov** (všetky typy), **relácií** a **triggrov**. Synchronizácia zdrojovo-kódových objektov ako **procedúr**, **defaultov** atď. prebieha iba u tých, ktoré buď neexistujú v **synchronizovanej databáze**, alebo sa ich výkonné časti zdrojového kódu líšia v **synchronizovanej databáze** tých, čo sa nachádzajú v **zdrojovej databáze**.

Funkcia **Synchronize** otvorí dialog box podobný **Transfer** dialog boxu.



Tu si má užívateľ možnosť vybrať, **Defaulty**, **Rules**, **Procedúry**, **Triggre**, **Viewy** a **Tabulky**, ktoré budú synchronizované. K tabuľkám si má možnosť vybrať, ktoré z ich **properties** majú byť synchronizované. Ďalej si má užívateľ možnosť vybrať synchronizovanú databázu, pre ktorú bude vygenerovaný script s názvom **synch_server_database.sql**, kde **server** je synch. server a **database** je synchronizovaná Databáza. Po stlačení **Start** sa vytvorí script. Ak tento script spustíte v synch. databáze, tak zosynchronizuje štruktúru objektov, ktoré boli k synchronizácii určené. Ak nejaká tabuľka mala rôznu štruktúru v oboch databázach, tak namiesto jej synchronizácie bude pre ňu script obsahovať riadok:

```
PRINT '----- > Table %s has different structure in each of selected databases, therefore no synchronization in this table is built
```

Ak boli synchronizované všetky objekty niektorej triedy, tak funkcia vygeneruje ešte jeden script s názvom **drop_server_database.sql** , kde **server** je synch. server a **database** je synchronizovaná databáza . Script obsahuje príkazy na zmazanie prebytočných objektov tých tried, pre ktoré boli synchronizované všetky objekty.

Scripty dodávané k programu.

V inštalovanom adresári Aplikácie sa nachádza séria T-SQL scriptov (.sql) ktoré majú za úlohu pomôcť pri nejakom špecifickom stave, ktorý mohol **výnimočne** nastať pri používaní Patik-a.

1. **CompactPatrik.sql** - je script, ktorý opraví databázu **patrik**, ktorá nie je kompaktná. Nekompaktnosť sa dá docieľiť premenovávaním a mazaním databáz inak ako v **Patrik**-ovi, častým prenosom databázy Patrik ale vlastne aj dlhodobým používaním aplikácie. Preto je rozumnejšie v nejakom scheduler-i nastaviť automatické spúšťanie scriptu povedzme každý mesiac.
2. **DropDatabase.sql** - program, ktorý slúži na dropnutie zle downsizenutej databázy, ktorá sa ešte nenachádza medzi databázami browse-ovateľnými Patrik-om. Po prvom spustení tohoto scriptu sa v databáze patrik vytvorí procedúra **DropDatabase**. Táto procedúra pri spustení s menom aké mala niesť downsizenutá databáza ako parametrom tento nepodarený downsize dropne.

Základné princípy aplikácie.

[Error Handling](#)

[Podrobný popis funkcií Upsize, Downsize](#)

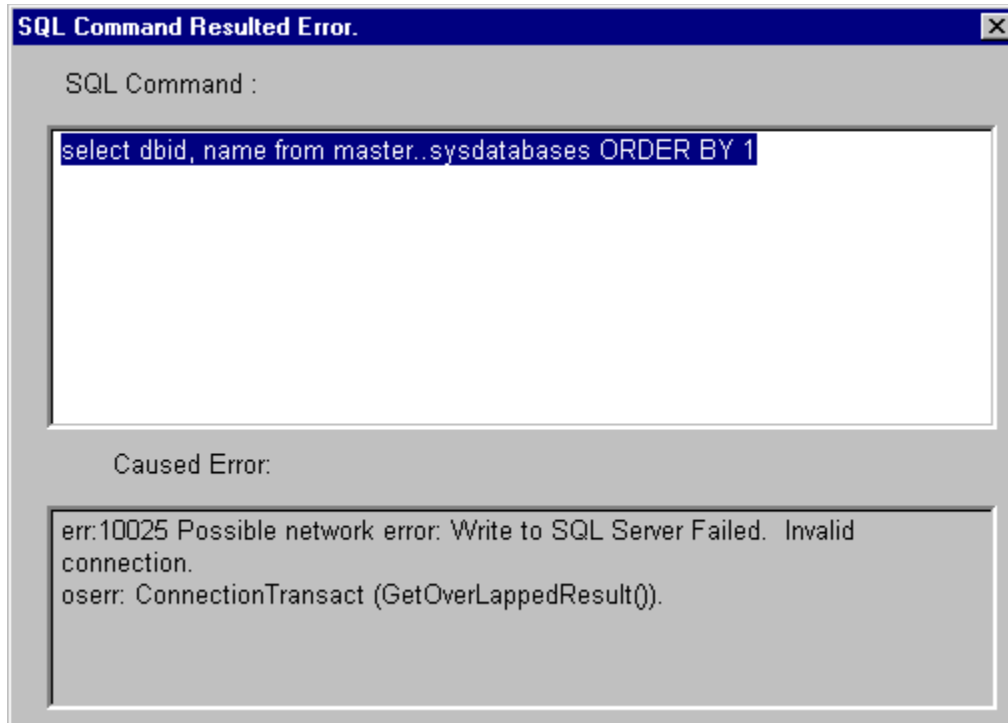
[Štruktúra databázy patrik](#)

[Patrik Server Options](#)

Error Handling

Neoèakávané chyby

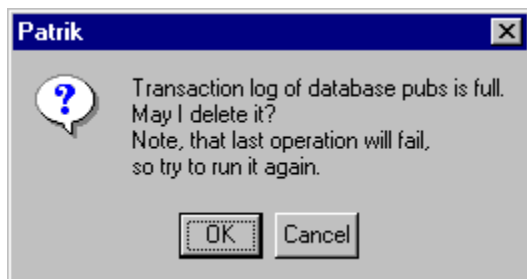
Jedná sa o Error handling na neoèakávanom mieste v aplikácii. Error mohol nastať nesprávnym používaním Patrik-a, chybou na SQL servi, Prerúšením komunikácie, alebo inou výnimkou. Pri takomto Error-e sa otvorí dialog box



V ktorého hornej èasti sa nachádza príkaz (alebo popis príkazu) ktorý mal byť spustený a v spodnej èasti sa nachádza chybová hláška. Dialog Box má slúžiť na lepšiu diagnózu chyby.

Preplnenie transakèného logu.

S èasu na èas môže nastať chyba preplnenia transakèného logu databázy, pri ktorom sa vypíše otázka



ktorá sa dotazuje na vôu užívatea ihneď dump-nú tento log èím sa síce aktuálny príkaz nevykoná, ale ten iaší už áno.

Padnutie procesu

Ľahšou výnimočnou situáciou je skončenie platnosti procesu (kill). Vtedy sa vypíše klasický SQL Error Dialog Box a po jeho zatvorení nebude proces (Connect) použiteľný. Bude teda potrebné napojiť sa server na ktorom padol proces ešte raz.

Podrobný popis funkcií Upsize, Downsize

Downsize

Proces downsize-u - vytvorenie databázy štruktúrovo identickej, len bez dát, k reálnej databáze pozostáva z vytvorenia tabuliek v databáze **patrik** podobných systémovým tabuľkám a prekopírovanie systémových tabuliek zdrojovej databázy do týchto novovytvorených. Ďalej je potrebné skopírovať vrstvu zdrojovej databázy pridanú Patrik-om. Jedná sa o komentáre, relačné view-y, pozície tabuliek v reláciách atď. Proces pozostáva iba s replikácie tabuliek databázy **patrik** samých do seba. Na konci sa pridá záznam do tabuľky **patrik..workdb**, ktorá obsahuje informácie o downsize-nutých databázach. Z tejto tabuľky, ako aj tabuľky sysdatabases sa pri napojení na server číta zoznam databáz browsovaných Patrik-om.

Downsize sa nespustí (a vypíše o tom chybovú hlášku) v týchto prípadoch:

1. Ak sa v zdrojovej databáze nachádzajú tabuľky s prefixom **patriktemp_** (viď. odstavec Upsize). V takomto prípade ich stačí z databázy zmazať a opätovne spustiť Downsize.
2. Ak už má zdrojovú databázu niekto Downsize-nutú.
3. Ak posledný Downsize neprebehol správne, alebo do konca. V takomto prípade je potrebné spustiť script **DropDatabase.sql** z inštalovaného adresára Patrik-a. A potom spustiť novovytvorenú procedúru **patrik..DropDatabase** s menom neukončene Downsize-nutej databázy ako parametrom. Toto meno sa dá zistiť s databázy **patrik** a zo znalosti procesu Downsize-u, ktorý je popísaný na konci odstavca *Štruktúra databázy patrik*.
4. Pri nekompaktnosti databázy **patrik**. V tomto prípade stačí spustiť script **CompactPatrik.sql** z inštalovaného adresára Patrik-a.

Drop pracovnej databázy.

Drop takto vytvorenej databázy pozostáva teda iba so zmazania záznamu s **patrik..workdb**, ktorý hovorí o existencii databázy, dropnutie kópií systémových tabuliek v databáze **patrik** a zmazanie záznamov, so štandardných tabuliek databázy **patrik**.

Upsize

Upsize - aplikovanie zmien vykonaných v pracovnej databáze na reálnu databázu prebieha optimálne čo sa týka voľby objektov, s ktorými bude pracovať. Pracuje sa totiž iba s tými objektami, ktoré boli v pracovnej databáze modifikované. Najskôr sa zmenia (popr. pridajú/zmažú) užívateľské dátové typy. Potom sa upsize venuje užívateľom. Neskôr sa začne upsize zmenených tabuliek. Patrik pozná niekedy typov zmien v tabuľke. Dali by sa rozdeliť do troch skupín:

1. Pridanie alebo Zmazanie tabuľky
2. Jednoduchá zmena tabuľky - zmena, ktorá sa dá docíliť príkazmi T-SQL-u a volaním systémových stored procedúr. Tento typ zmeny umožňuje aj väčšina, na trhu existujúcich, databázových managerov.
3. Zložitá zmena tabuľky - zmena, ktorú treba previesť trikmi, ktorý pozostáva z premenovania existujúcej tabuľky (používajú sa pomocné mená **patriktemp_xx**, kde xx je číslo) vytvorenia novej tabuľky so všetkými vlastnosťami starej tabuľky, ktorá je už ale taká, ako v pracovnej databáze, prenesenie dát zo starej tabuľky do novej, dropnutia pôvodnej tabuľky a vytvorenia relácií a triggrov nad novou tabuľkou. Vytvorenie triggrov a relácií neskôr, ako sa prenesú dáta je nutné kôli nevýhodnosti referenčnej integrity v takomto

špeciálnom prípade.

Po zmenách v tabuľkách sa ešte regenerujú triggre zmenených tabuliek a tiež ich relácie. Na koniec sa pracovná databáza zmaže (dropne) a vypíše sa množina výsledkov. Je dôležité vedieť výsledky Upsize-u. Netreba sa vystrážiť hlások o tom, že nejaký objekt bol úspešne premenovaný napriek tomu, že ste žiaden nepremenovávali. Jedná sa totiž isto o tabuľku vyššie opísanej skupiny è. 3. Ďalej sa dá riadiť pravidlom, že sa vo výsledkoch Upsize-u nevyskytne už žiadna chyba za prvou hláškou "Foreign key dropped". Pri Upsize-e by nemala nastať žiadna chyba vinou Patrik-a. Ak nejaká chyba nastane, stane sa to iba kôli užívateľovej nepozornosti, alebo nejakým Sw/Hw/Power zlyhaním. V žiadnom prípade by však nemalo dôjsť k strate dát. Jediný spôsob pri ktorom by sa mohli dáta s originálnej tabuľky (avšak nie celkom zo servra) stratiť je pri zmene tabuľky skupiny è. 3. Tam môže dôjsť k neúspešnému preneseniu dát z tabuľky **patriktemp_xx** do tabuľky jej pôvodného mena. V takomto prípade sa však príslušná tabuľka **patriktemp_xx** nedropne, čím dáta ostanú na serveri a bude na užívateľovi, aby si dáta nejakým vlastným `< INSERT ... SELECT ... FROM patriktemp_xx >` preniesol a tabuľku **patriktemp_xx** drop-ol. Nakoľko existencia tabuliek **patriktemp_xx** v databáze znemožňuje jej Downsize, je potrebné tieto tabuľky z databázy zmazať.

Upsize pozostáva s troch častí.

1. Vytvorenia Upsize-scriptu.
2. Spustenia Upsize-scriptu.
3. Drop pracovnej databázy.

Medzi časťami 1 a 2 sa môže vyskytnúť chybové hlásenie, ktoré informuje užívateľa o prípadných nežiadúcich následkoch spustenia Upsize-scriptu. V tomto stave sa dá ešte Upsize zastaviť.

Pre úplnosť viď ešte popis [štruktúry databázy patrik](#).

Štruktúra databázy patrik.

Znalos štruktúry databázy “patrik” nie je nutná k práci s programom Patrik. Táto kapitola je teda určená pre skúsených užívateľov SQL-Servra, ktorý môžu jej znalosť využiť na všemožné účely, z ktorých niektoré sú uvedené na konci tohoto článku. Databáza teda pozostáva z nasledovných tabuliek:

rel - tabuľka, v ktorej sú uložené pozície okien, v ktorých sú atribúty tabuliek zobrazované v Relational Mod

- **db** - meno databázy, ktorej tabuľka patrí
- **tbl** - meno tabuľky
- **vy** - poloha ľavého horného rohu okna s tabuľkou
- **cx** - šírka okna
- **cy** - výška okna

Z tejto tabuľky sa číta vždy pri Načítavaní a Refresh-ovaní relácií a ukladá sa do nej pri zvolení Save Layout v shortcut menu okna relačného modelu.

Tabuľky popisujúce vygenerované triggre.

Patrik pri upsize-e pregenerováva triggre, ktoré je nevyhnutné pregenerovať vzhľadom k zmenám vykonaným v Downsize-nutej databáze. Triggre, ktoré zabezpečujú referenčnú integritu alebo históriu v tabuľke obsahujú časti, ktoré sa modifikujú pri pregenerovaní. Každá takáto časť začína v triggrí komentárom /*PatrikB*/ a končí komentárom /*PatrikE*/. Zostávajúca časť triggra je určená na Business-Rules nakódované užívateľom. Táto sa pri upsize-e neprenerováva. Vráťme sa ale k tým, čo áno. Prenerovávajú sa práve podľa tabuliek **keysadd** a **patriklog**. Ak má tabuľka nejaké ne-constraint-ové relácie, o ktorých je zápis v tabuľke **keysadd** s nenulovým atribútom **flags** tak sa preneruje do triggra podprogram zabezpečujúci neporušovanie tejto referenčnej integrity. Tento podprogram sa nachádza na začiatku programu triggra, aby detekoval porušenie a zrušil transakciu ešte predtým ako zafungujú prípadné Business-rules. Ak má tabuľka zápis v tabuľke **patriklog** tak sa na koniec triggra vygeneruje podprogram (opäť ako Patrik-ovská časť triggra) ktorý zabezpečí zápis každej zmeny do historickej tabuľky.

Patrik sa neskôr po upsize-e riadi tým, čo sa nachádza v týchto tabuľkách a nie parsovaným zdrojovým kódom triggra. Keby teda došlo k zmazaniu nejakých záznamov v niektorej z nižšie uvedených tabuliek, tak by sa po nejakom upsize mohlo stať, že by sa triggre opäť zladili s tabuľkami, čo by malo za následok iný stav databázy, ako ho jej architekt naštudoval. Z toho istého dôvodu je dôležité dbať pri prenose databázy na súbežný prenos tabuliek **patriklog** a **keysadd** či už funkciou **Transfer/Patrik Database ...** alebo uskutočnením každého prenosu Patrik-om. Tieto tabuľky samozrejme nie je potrebné prenášať, ak v cieľovej sa cieľová databáza nebude modifikovať Patrik-om.

patriklog - tabuľka, v ktorej sú uložené informácie o tabuľkách nad ktorými je vytvorená história

- **db** - meno databázy, ktorej tabuľka patrí
- **tbl** - meno tabuľky v databáze db.

keysadd - tabuška, v ktorej sú uložené informácie o vlastnostiach relácií a ich popisy. Obsahuje záznamy aj pre constraint-ové relácie kôli komentáru. Vtedy je hodnota v atribúte flags tejto tabušky nedôležitá.

db - meno databázy, ktorej relácia patrí

id - meno tabušky na strane many relácie (detská tabuška)

depid - meno tabušky na strane one relácie (rodičovská tabuška)

flags - vlastnosti relácie určujú nasledovné bity:

1 - one to one vzťah

2 - existuje (alebo treba vytvoriť) delete trigger

4 - existuje update trigger

8 - delete trigger kaskádovo maže riadky detskej tabušky

16 - update trigger kaskádovo obnovuje riadky v detskej tabuške

str - 255 znakový komentár k relácii

Komentáre

Komentáre alebo poznámky k objektom sú udržiavané v dvoch tabuškách **conspects** a **notes**, ktorých štruktúra je myslím zrejmá.

Pohľady na Relačný model (view-y).

Každý pohľad (view) obsahuje zápis v tabuške **rel_view**. Tu sa nachádza iba informácia o jeho názve a tvorcovi. Štruktúra pohľadu sa nachádza v tabuške **view_item**, ktorá je spojená s **rel_view** reláciou One to Many. Tu sa nachádzajú informácie o polohe a viditeľnosti každej zúčastnenej tabušky daného pohľadu.

rel_view - tabuška s pohľadmi

- **id** - primárny kľúč tabušky
- **name** - meno pohľadu
- **db** - databáza, ktorej pohľad patrí
- **usr** - užívateľ, ktorý vytvoril pohľad

view_item - tabuška, v ktorej sú uložené pozície okien, v ktorých sú atribúty tabuliek zobrazované v *Relational Model* okne.

- **id** - príslušnosť tabušky do konkrétneho pohľadu.
- **tbl** - meno tabušky
- **vy** - poloha ľavého horného rohu okna s tabuškou

- **cx** - šírka a výška okna
- **flags** - viditeľnosť tabuľky
- **Workdb** - tu sú uložené údaje o fiktívnych (downsized) databázach
- **name** - meno fiktívnej databázy, bez prefixu “#”
- **sourcedb** - meno zdrojovej (downsized) databázy
- **status** - stav databázy. Nastaviteľný v menu “Current Database/Options” programu Patrik.

Ďalej sa v databáze patrik nachádza tabuľka [Tabuľka Options](#) a tabuľky v ktorých sa nachádzajú definície zostáv, ktorých popis neuvádzam.

Toto boli tabuľky, ktoré su vytvorené spolu s databázou patrik. Spolu s nimi je vytvorený aj kaskádový delete trigger nad tabuľkou **rel_view**, ktorý vymaže splou s mazaným pohľadom aj pozície okien vo **view_item**.

Tabuľky vzniknivé funkciou Upsize.

Ďalšie tabuľky sa objavia v databáze “patrik” po downsize. Povedzme, že niekdo downsize-ol

databázu “pubs”, pod fiktívne meno “work”. Pri Upsize-e pribudli postupne do databázy patrik nasledovné tabuľky:

sysalternates_work, syscolumns_work, syscomments_work, sysconstraints_work, sysidentities_work, sysobjects_work, syskeys_work, sysprotects_work, sysreferences_work, systypes_work a sysusers_work.

Tieto tabuľky majú rovnakú štruktúru, ako ich ekvivalenty (bez prípony samozrejme) v databáze “pubs”. Niektoré nepoužité stĺpce sú však použité programom Patrik. Tu je ich popis:

sysobjects.sysstat - popisuje zmeny, ktoré sa udiali nad objektom. Toto pole je využívané pri upsize databázy. Nasledujú bity, ktoré môžu byť v tomto poli nastavené:

- 1 - index tabuľky bol zmenený
- 2 - primárny kľúč tabuľky bol zmenený
- 4 - dátový typ nejakého stĺpca tabuľky bol zmenený
- 8 - meno nejakého stĺpca tabuľky bolo zmenené
- 16 - rule, alebo default nejakého stĺpca tabuľky bol zmenený
- 32 - tabuľka bola “veľmi” zmenená
- 64 - do tabuľky bol pridaný nový stĺpec
- 128 - meno tabuľky bolo zmenené
- 256 - tabuľka bola pridaná do databázy
- 512 - komentár pre nejaký stĺpec tabuľky bol zmenený
- 1024 - IDENTITY pre daná tabuľku bola zmenená
- 2048 - check, alebo default constraint bol zmenený pre nejaký stĺpec tabuľky
- 4096 - relácia pre tabuľku bola zmenená

sysypes.printfmt - popisuje zmeny, ktoré sa udiali nad dátovým typom.

- 1 - grupa užívateľov sa zmenila
- 2 - práva užívateľov sa zmenili
- 4 - užívateľ bol veľmi zmenený (ostatné zmeny)
- 8 - užívateľ bol pridaný do databázy

sysusers.environ - popisuje zmeny, ktoré sa udiali nad užívateľom databázy

Do databázy "patrik" sa skopíruje pri downsize ešte jedna tabuľka, ktorá však má inú štruktúru ako jej vzorová tabuľka, vo downsize-ovanej databáze (v našom prípade pubs)

- **sysindexes_work** vytvorená tabuľka
- **name** meno indexu
- **id** id tabuľky, ktorej index patrí
- **indid** id indexu
- **segment** číslo segmentu na ktorom sa vytvorí index
- **status** typ indexu
- **fill** reserved
- **keys** je typu char(65) a obsahuje čísla stĺpcov na ktoré sa vzťahuje index, utriedené podľa veľkosti a oddelené čiarkou (stĺpce sa v tomto prípade číslujú od jednotky)

Spolu s vyššie uvedenými tabuľkami sa vytvorí aj kaskádový delete trigger nad tabuľkou **sysobjects_work**, ktorí maže riadky z detských tabuliek **syscolumns_work** a **sysindexes_work**.

No a načo je znalosť týchto tabuliek? Uvediem niekoľko aplikácií:

1. MS Access Upsizing Tool pri upsize databázy upsizuje polia typu text v Accesse do polí typu varchar v SQL-i. Nakoľko je práca s typom varchar zdáhavejšia, je výhodné zmeniť v Patrikovi dátové typy varchar na char a upsiznúť databázu z Patrika. To by bolo jednoduché, keby však databáza obsahovala málo tabuliek, ale komu by sa chcelo pri veľkých tabuľkách robiť toľko zmien. Pri znalosti tabuliek v databáze "patrik" by bol však postup nasledujúci:

- Downsiznúť databázu pod meno "work"
- v Query editore spustiť takúto query:

```
UPDATE patrik..sysobjects_work SET sysstat=sysstat/4  
FROM patrik..sysobjects_work a, patrik..syscolumns_work b  
WHERE a.type='U' AND a.id=b.id AND b.usertype=2
```

```
UPDATE patrik..syscolumns_work SET type=47, usertype=1
```


WHERE usertype=2 AND name NOT LIKE('@%')

· Upsiznú databázu work.

2. Keď sa databáza z nejakých príèin (napr. power failure) nedownsizne úplne, užívateľ aspoň vie, ktoré tabuľky môže v databáze patrik dropnúť. Na ich komfortné dropnutie slúži aj, spolu s Patrikom dodávaný, script **DropDatabase.sql**.

Ví [Podrobný popis funkcií Upsize, Downsize](#) .

