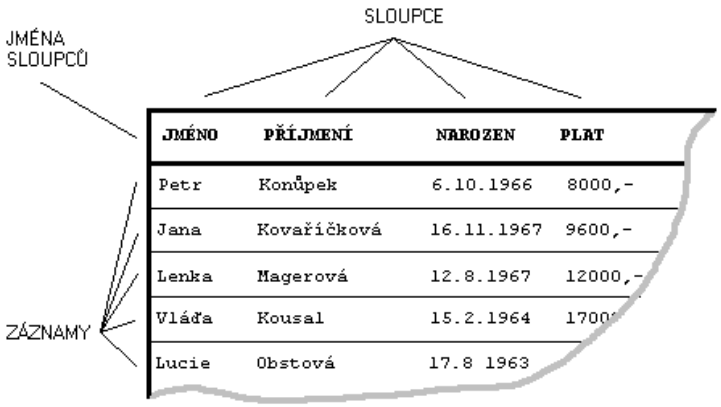
Návrh databázových tabulek

Data jsou v databázi uložena v tzv. TABULKÁCH. Každé využití databáze proto začíná vytvořením vhodných databázových tabulek.

Co je to databázová tabulka?

Databázová tabulka slouží k uchování dat, která mají stejnou vnitřní strukturu a logicky patří k sobě. Představu databázové tabulky dobře vystihuje tento obrázek:

Databázová tabulka



Záznamy a sloupce

Tabulka z obrázku obsahuje údaje o zaměstnancích nějaké firmy. Skládá se z jistého počtu ZÁZNAMŮ – každý záznam popisuje jednoho zaměstnance a je zde zobrazen jako jedna řádka. Tabulka – a s ní každý záznam – je rozdělena do SLOUPCŮ obsahujících elementární data.

Struktura a obsah

Na tabulku lze pohlížet ze dvou stran. Můžeme popsat její strukturu, tedy jména všech sloupců a co se do těch sloupců smí zapisovat. Můžeme se také zajímat o obsah tabulky, tedy o data, která se zrovna v tabulce nacházejí. Za života databázové tabulky se struktura mění jen zřídka, zatímco obsah, data, se doplňují a upravují neustále.

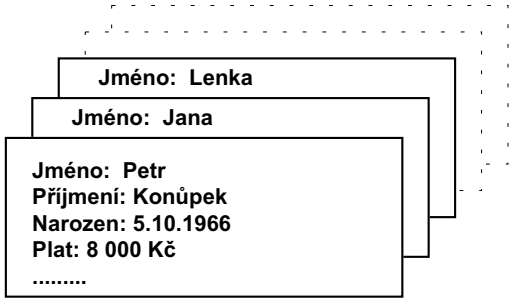
Typ sloupce

Každý sloupec tabulky má určený TYP. Typ říká, co smí být do sloupce zapsáno, jako například číslo, libovolné znaky, datum, obrázek a podobně. Typy mají dvojí význam: jednak dovolují okamžitě odhalit některé chyby při zapisování údajů (např. písmena v čísle), jednak sdělují databázovému serveru, jak má s údajem zacházet. Na základě typu server pozná například to, že při seřazení záznamů podle příjmení má třídit lexikografic-

ky, při seřazení podle platu třídit na základě číselné hodnoty a při seřazení podle data narození třídit podle roku, měsíce a dne.

Data obsažená v tabulce mohou být uživateli databázové aplikace předkládána v libovolné jiné formě. Typickými příklady jsou formulář zobrazující v jednom okamžiku vždy jen jeden záznam, anebo sestava, v níž jsou záznamy uspořádány, rozděleny do skupin a doplněny například o skupinové součty.

Stejná tabulka v podobě kartotéčních lístků



Prázdné údaje

V databázovém záznamu mohou některé sloupce zůstat prázdné. Prázdné hodnotě sloupce říkáme NULL nebo NONE. Hodnota NULL je něco zcela jiného než hodnota 0 nebo kterákoli jiná. Ukazuje to tento příklad: Předpokládejme, že ve škole sledujeme známky z nepovinného předmětu "španělský jazyk". Jeden žák má známku 1, druhý známku 2, třetí na tento předmět nechodil a tak nemá žádnou známku, tedy NULL. Jaký je známkový průměr? Databáze správně odpoví, že 1.5, zatímco kdybychom u třetího žáka uvedli cokoli jiného (např. nulu) výsledek by byl chybný (1.0).

Na co myslet při navrhování tabulek

Vytváření nové databázové aplikace vždy začíná návrhem struktury tabulky nebo tabulek. Rozvržení dat do tabulek je klíčovým momentem, které ovlivní všechny další kroky. Proto se vyplatí věnovat čas na podrobnou analýzu struktury dat a vztahů mezi nimi.

Na začátku máte představu o tom, co chcete v databázi zachytit. Základní otázka, kterou si opakovaně položíte, zní "Vystačím pro určitá data s jednou tabulkou, nebo budu potřebovat soustavu tabulek (provázaných nebo nezávislých)?". Tatáž otázka v jiné formulaci: "Kdy vystačím s jednotnou strukturou záznamu a kdy potřebuji záznamy různého druhu?".

Příklad:

Uvažujme evidenci účastníků a příspěvků na nějaké konferenci. Budete chtít evidovat jména účastníků a názvy jejich příspěvků. Pokud každý účastník bude mít nanejvýš jeden příspěvek a příspěvky nebudou mít více než jednoho autora, snadno vystačíte s jednou tabulkou. Bude v ní v každém záznamu jak jméno účastníka, tak i název jeho příspěvku. Pokud má pouze malá část účastníků vlastní příspěvek, pak místo pro název příspěvku bude v řadě záznamů nevyužito, ale to je vcelku přijatelná daň za jednoduchost návrhu.

Více tabulek

Pokud mají někteří účastníci vícero příspěvků a pokud chcete v databázi zachytit i fakt, že více účastníků může být spoluautory stejného příspěvku, pak s jedinou tabulkou stěží vystačíte. Vyplatí se navrhnout tabulky dvě, jednu pro účastníky, druhou pro příspěvky. Pak ale budete muset řešit problém, jak zjistit, jaký příspěvek patří ke kterému účastníkovi.

Jedno možné řešení, specifické pro **WinBase602,** k tomu využívá ukazatele: například každý účastník může odkazovat na všechny svoje příspěvky a každý příspěvek na všechny svoje autory.

Klasický RELAČNÍ PŘÍSTUP naproti tomu uvádí data obsažená v různých tabulkách do souvislostí na základě stejných hodnot v určitých sloupcích. Ve většině případů je relační propojení záznamů pružnější než použití ukazatelů.

Relace nebo ukazatele?

Na základě zkušeností uživatelů lze doporučit tento přístup: Ukazatele mohou podstatně urychlit a zjednodušit návrh nepříliš složité aplikace, zahrnující několik tabulek. Navrhujete-li však rozsáhlé databázové schéma s více tabulkami (nebo očekáváte-li, že se aplikace do těchto rozměrů postupně rozroste) *je vhodnější dávat přednost relačnímu přístupu*. Ukazatele můžete uplatnit v okrajových oblastech velké aplikace, ale není dobré na nich stavět klíčové vztahy mezi údaji.

Rozsáhlé aplikace totiž vyžadují větší míru přizpůsobivosti postupnému doplňování nebo pozměňování návrhu. Tuto flexibilitu poskytuje relační model dat. Do jisté míry to platí i o multiatributech. Multiatributy s výhodou využijete tam, kde potřebujete prostě ukládat do databáze množiny údajů. Propojování záznamů na základě hodnot multiatributů nebo počítání agregovaných funkcí nad multiatributy však nebude efektivní.

Vztahy mezi tabulkami

Vyšetřujete-li vztahy mezi tabulkami, vyplatí se je rozdělit do tří druhů:

- **vztah 1:1** - záznam v jedné tabulce je vždy svázán s (nejvýše) jedním záznamem v druhé tabulce a naopak;
- **vztah 1:N** - záznam v jedné tabulce je svázán s množinou záznamů z druhé tabulky, každý záznam z druhé tabulky však přináleží nejvýše jednomu záznamu z první tabulky;
- **vztah M:N** - záznam v jedné tabulce je svázán s množinou záznamů z druhé tabulky a naopak (např. výše uvedený vztah mezi autory a příspěvky).

Pro vyjádření prvních dvou vztahů lze stejně snadno použít ukazatelů i relačního propojení, které doporučujeme. Vztah **M:N** se v relačním modelu reprezentuje obvykle pomocí třetí, pomocné tabulky. Při použití multiukazatelů pomocnou tabulku nepotřebujete.

Podrobný popis a příklady na relační vztahy naleznete v kapitole *Vytváření aplikací*.

Které údaje lze spojit ?

V rámci návrhu každé tabulky bývá nutno uvážit, které údaje lze spojit do jediného sloupce a které je lepší mít ve zvláštních sloupcích. Např. v evidenci osob lze spojit jméno, příjmení a případný titul osoby do jediného sloupce. Pak ale, pokud příjmení nebude na začátku, nebude snadné podle něj osoby alfabeticky setřídit.

Návrh tabulek a analýza vztahů mezi nimi je velmi rozsáhlé téma a je předmětem vědeckého výzkumu. Nebudeme se jim zde podrobněji zabývat, je ale dobré si odnést toto ponaučení: Vyplatí se věnovat čas pečlivému návrhu struktury tabulek na začátku vytváření aplikace a tím ušetřit mnohem víc času na pozdějších úpravách.

Návrh tabulky lze změnit

Návrh tabulky lze kdykoli později změnit. Data obsažená v tabulce lze přenést do nové tabulky nebo rozdělit do několika tabulek, lze také data z různých tabulek spojit do jedné. Toto téma je popsáno v kapitole o hromadných operacích s daty. Pamatujte však na to, že přenos velkých objemů dat může chvíli trvat.

Každý sloupec, který uvedete v definici tabulky a který není multiatributem, bude zabírat v každém záznamu určité místo. Sloupce pevné velikosti budou toto místo zabírat bez ohledu na to, zda do nich zapíšete jakoukoli hodnotu. Proto pokud v návrhu tabulky uvedete množství málo využívaných sloupců, pak v databázi vznikne nevyužité místo.

Použití specifických rysů WinBase602

WinBase602, oproti jiným relačním databázovým systémům, přináší některé další rysy a způsoby uchovávání dat a vztahů mezi nimi. Jedná se o sloupce typu ukazatel, sledovací atributy a použití multiatributů. Všechny tyto rysy se dají s výhodou využívat v konkrétních případech, se kterými se seznámíte v manuálech, zatímco jinde mohou návrh aplikace zkomplikovat. Doporučujeme rozšířené vlastnosti **WinBase602** začít používat tam, kde postupy obvyklé z relačních databází a jejich struktury přestávají vyhovovat nebo jsou pracnější apod. Použití těchto rysů má totiž kromě výhod i svá omezení. Zvláště upozorňujeme, že není rozumné mít v aplikaci tabulky propojované mnoha obousměrnými multiukazateli, které nahrazují relační vztahy.

Vytvoření databázové tabulky

Novou tabulku lze vytvořit buď ručně ve vývojovém prostředí **WinBase602** pomocí vizuálního návrháře tabulek, nebo z programu pomocí SQL příkazu **`CREATE TABLE`**. Popis SQL příkazů a jejich volání z jazyka naleznete v kapitole o programování a v elektronické nápovědě. Zde popíšeme pouze ruční, interaktivní postup návrhu.

Předpokládáme, že již existuje aplikace, do níž má být nová tabulka zahrnuta. Na řídicím panelu vyberte v této aplikaci **Tabulky** a proveďte akci **Vytvořit**. Tím otevřete návrhář tabulek.

Poznámka

Termínem „proveďte akci **XYZ**“ *se v celém manuálu myslí: stiskněte tlačítko* **XYZ** *v pravé části řídicího panelu nebo vyberte příkaz* **XYZ** *z popup menu označeného objektu.*

Poté, co v návrháři zadáte všechny vlastnosti tabulky, můžete ji vytvořit stiskem tlačítka na liště nebo příkazem **Vytvořit** z menu ***Návrh***. Přitom budete vyzváni, abyste pro tabulku zadali jméno. Vytvořená tabulka se objeví na řídicím panelu.

Použijete-li příkaz **Nevytvářet** nebo zavřete-li okno návrháře, tabulka se nevytvoří a zadaný popis se zahodí.

Během navrhování tabulky lze pomocí tlačítka na liště nebo příkazu **Ukázat** z menu ***SQL-příkaz*** zobrazit příkaz `CREATE TABLE`, který vytvoří specifikovanou tabulky.

Definování sloupců tabulky

Definice sloupců tvoří základ každého návrhu tabulky. Sloupce se popisují v levé části vizuálního návrháře, každý na jedné řádce. Pro každý sloupec zadáte nejprve jeho jméno a pak typ.

Popis sloupce tabulky

Pro některé typy je třeba v pravé části okna zadat doplňující údaje, například pro řetězec znaků zadáte jeho maximální délku. Mezi levou a pravou částí se přechází stiskem klávesy F6.

Jméno sloupce

Jméno sloupce smí mít nejvýše 31 znaků a nesmí obsahovat znak ` (obrácený apostrof). Pokud ve jménu použijete mezery nebo nealfanumerické znaky, pak při uvádění jména sloupce v popisu indexu, v integritních omezeních, v programech apod. budete muset jej uzavřít do obrácených apostrofů, např. `` `Adresa: číslo popisné` ``.

Jméno sloupce nesmí být totožné s klíčovým slovem nebo předdefinovaným identifikátorem vnitřního programovacího jazyka – **WinBase602** to hlídá a na případnou shodu vás upozorní. V tabulce nesmějí být dva sloupce stejného jména. V různých tabulkách lze používat stejně pojmenované sloupce.

Typy sloupců

Typ sloupce vyberte z nabídky, kterou otevřete buď kliknutím na tlačítku s šipkou dolů nebo klávesovým povelem Alt+↓. Na vybraný typ v nabídce pak buď kliknete myší nebo stisknete klávesu Enter.

Níže uvedený přehled typů **WinBase602** obsahuje kromě názvu povolený rozsah hodnot a také počet bajtů, který v databázi zabere hodnota příslušného typu.

Typ	česky	Rozsah	Délka
BOOLEAN	ANO-NE	TRUE (Ano) a FALSE (Ne)	1
CHAR	ZNAK	Libovolný znak (pouze jeden)	1
SHORT	MALÉ CELÉ ČÍSLO	-32767 až 32767	2
INTEGER	VELKÉ CELÉ ČÍSLO	-2147483647 až 2147483647	4
MONEY	PENÍZE	$-1.4*10^{12}$ až $1.4*10^{12}$, 2 desetinná místa	6
REAL	REÁLNÉ ČÍSLO	$1.7*10^{-308}$ až $1.7*10^{308}$, kladné i záporné	8
STRING	ŘETĚZEC	Libovolné znaky	zvol.
CSSTRING	ČESKÝ ŘETĚZEC	Libovolné znaky	zvol.
CSISTRING	ČESKÝ ŘETĚZEC	Libovolné znaky	zvol.
BINARY	BINÁRNÍ DATA	Libovolné bajty včetně bajtu 0	zvol.
DATE	DATUM	Datum od počátku letopočtu	4
TIME	ČAS	0:0:0.000 až 23:59:59.999	4
TIMESTAMP	DATUM A ČAS	Datum a čas s přesností na vteřiny Datum od r. 1900 do 2033	4
POINTER	UKAZATEL	Odkaz na libovolný záznam určené tabulky	4
BIPTR	OBOUSMĚRNÝ UKAZATEL	Odkaz na libovolný záznam určené tabulky	4
TEXT	TEXT	Libovolný text	prom.
RASTER	OBRÁZEK	Obrázek ve formátu BMP, PCX, TIFF, GIF nebo WPG	prom.
OLE, NOSPEC	OLE	OLE objekt, libovolná data	prom.
AUTOR	AUTORIZACE	Identifikace uživatele **WinBase602**	2
DATUMOVKA	DATUMOVKA	Datum a čas změny s přesností na sekundy	4
HISTORIE	HISTORIE	Historie sledovaného sloupce	prom
PODPIS	PODPIS	Digitální podpis záznamu nebo jeho části	prom.

Ve sloupci **délka** zkratka **zvol.** označuje pevně zvolenou délku udanou v návrhu tabulky, zkratka **prom.** označuje proměnnou délku závislou na množství zapsaných dat.

Číselné typy

Hodnoty čísel, uložených v databázi, jsou (na rozdíl od např. formátu Xbase) ve vnitřním (binárním) tvaru. Například všechna reálná čísla zabírají v databázi 8 bajtů bez ohledu na to, jak velké číslo do sloupce zapíšete. Proto při definování tabulky nemusíte specifikovat počty číslic před a za desetinnou tečkou.

Doporučujeme pro každý číselný údaj pečlivě volit přiměřený datový typ – nemá smysl například pro celá čísla použít sloupec typu **Real**, protože zabere víc místa a veškeré operace jsou s ním složitější.

Řetězce znaků

Je-li sloupec typu ŘETĚZEC ZNAKŮ, pak se vpravo od okna s popisem sloupců do pole **Délka řetězce** uvádí maximální délka řetězce. Tato délka smí být v rozsahu 1 až 255. Pokud se při práci s tabulkou do sloupce určité délky pokusíte zapsat delší řetězec, přijdete o přebývající znaky na jeho konci.

Typy STRING, CSSTRING a CSISTRING se neliší v tom, co mohou mít za hodnotu. Rozdíl je pouze v tom, jak se jejich hodnoty uspořádávají:

- STRING - uspořádání odvozené z interního kódu znaků, tzn. CH bude mezi C a D, písmena s diakritikou budou na konci abecedy;
- CSSTRING - uspořádání dle pravidel českého (a slovenského) jazyka, přitom velká písmena jsou před malými;
- CSISTRING - uspořádání dle pravidel českého (a slovenského) jazyka, přitom se nerozlišuje mezi velkými a malými písmeny.

Obecně platí, že je rozumné vybírat typ, který co nejpřesněji vystihuje povahu informace, která má být hodnotou sloupce. Je-li tedy hodnotou číslo (PLAT, POČET_KUSŮ), použijte některý z číselných typů, a nikoli typ řetězec znaků. Do řetězce sice také můžete zapisovat čísla, ale jednak s nimi pak nelze přímo provádět aritmetické operace, jednak obvykle zaberou víc místa. Typ řetězec použijte například pro údaj TELEFON nebo RODNÉ ČÍSLO, kde se mohou vyskytovat kromě číslic i mezery či pomlčky a nehrozí, že by někdo s těmito čísly chtěl provádět aritmetické operace.

Binární data

Typ BINARY slouží k zapsání libovolné posloupnosti bajtů, včetně znaků s kódy nižšími než 32, které normálně nelze z klávesnice zadat. Lze zapsat i znak s kódem 0.

Typy proměnné velikosti

Typy TEXT, RASTER, OLE, HISTORIE a PODPIS nazýváme typy **proměnné velikosti**. Velikost místa, které zabírají v databázi, závisí na objemu dat, které do nich vložíte.

Hodnota typu proměnné velikosti zabírá 10 bajtů, pokud je prázdná. Prostor se těmto hodnotám přiděluje po jednotkách, z nichž nejmenší má v současné verzi **WinBase602** 128 bajtů. Maximální délka hodnoty je v současnosti cca 8MB. Tento limit je třeba brát v úvahu například při ukládání video-clipů do databáze a raději využít technologie připojených objektů z OLE.

Ukazatele

Vztah mezi dvěma záznamy ve **WinBase602** může být vyjádřen pomocí ukazatelů. Existují přitom dvě možnosti: buď ukazatel vede z jednoho záznamu určité tabulky na nějaký jiný záznam, anebo je propojení obousměrné a oba záznamy ukazují na sebe navzájem.

Pro každý sloupec typu UKAZATEL (jednosměrný i obousměrný) je nutno uvést, do které tabulky bude ukazatel směřovat. Jméno této tabulky vyplníte vpravo od okna se seznamem sloupců (do pole označeného **Cílová tabulka**). V okamžiku zadávání tohoto jména cílová tabulka ještě nemusí existovat.

Pokud do definice nějaké tabulky A zahrnete obousměrný ukazatel do tabulky B, pak také v definici tabulky B musí být obousměrný ukazatel do tabulky A. V opačném případě ukazatelová vazba mezi tabulkami nebude fungovat.

Pokud v tabulkách A a B je více **vzájemných** obousměrných ukazatelů, pak se spárují v pořadí svého výskytu v definicích těchto tabulek.

Povolení prázdné hodnoty

Zaškrtávací čtverec nadepsaný **Smí mít prázdnou hodnotu** určuje, zda je přípustné, aby sloupec měl v některém záznamu tabulky prázdnou, tedy nevyplněnou hodnotu (NULL). Pokud není zaškrtnut, pak do tabulky nelze vložit záznam bez zadání hodnoty tohoto sloupce.

Doporučujeme ponechávat tento čtverec zaškrtnutý, zvláště pokud chcete vkládat záznamy programem nebo formulářem se složkovou synchronizací.

Implicitní hodnota

Do pole **Implicitní hodnota** se zadává hodnota, která se má objevit v každém novém záznamu. Hodnota musí patřit do typu zvoleného pro sloupec. Implicitní hodnotu lze zadat i jako výraz nebo volání funkce, která vrací potřebnou hodnotu. Například pokud pro sloupec typu **Datum** zadáte implicitní hodnotu **`CURRENT_DATE`**, pak se v každém novém záznamu automaticky objeví datum jeho vložení.

USER

Speciální implicitní hodnota **`USER`** označuje uživatele, který záznam vložil. Dá se použít pro sloupce typu řetězec znaků nebo **Binary** s délkou alespoň 12. Jde-li o znakový sloupec, zapíše se do něj přihlašovací jméno uživatele, do binárního sloupce se zapisuje interní identifikace uživatele na serveru.

UNIQUE

Pro sloupce typu **Integer** lze uvést jako implicitní hodnotu klíčové slovo **`UNIQUE`**. Do sloupce se pak zapíše hodnota, která se liší od hodnot tohoto sloupce ve všech ostatních záznamech stejné tabulky. Sloupec s touto implicitní hodnotou můžete pak využívat jako

unikátní klíč k tabulce. Tento rys však funguje pouze tehdy, pokud se implicitní hodnota zadá, dokud je tabulka prázdná, nikoli až při modifikaci naplněné tabulky. Modifikace totiž přenese původní hodnoty sloupce, které nemusí být unikátní. Sloupec s touto implicitní hodnotou není editovatelný. Pro generování unikátních klíčů lze použít také sekvencí, které poskytují vyšší míru kontroly nad přidělovanými hodnotami. O sekvencích více v kapitole *Jazyk SQL* v *Příručce vývojáře*.

Implicitní hodnoty je možné definovat zároveň v tabulce i ve složkách formuláře. Při vložení nového záznamu pomocí formuláře se uplatní obě definice. Sejdou-li se v jednom sloupci dvě implicitní hodnoty, prosadí se hodnota určená v definici formuláře.

Multiatributy

MULTIATRIBUTY jsou zvláštní druh sloupců v tabulce, specifický pro **WinBas602**. Vyznačují se tím, že mohou v sobě *současně obsahovat více než jednu hodnotu*. Jsou dvojího druhu: buď mají pevný předem daný počet hodnot, nebo proměnný počet.

Sloupec se stane multiatributem, když pro něj zaškrtnete čtverec **Multiatribut** v pravé části okna. Tím se zpřístupní další dvě pole návrháře, která slouží k určení počtu hodnot multiatributu.

V poli **Rezervace** uvedete, pro kolik hodnot multiatributu se má pevně rezervovat místo. Rezervovat lze místo pro 0 až 127 hodnot. Zaškrtnutím čtverce **Lze překročit** určíte, zda se rezervovaný počet smí překročit. Pokud čtverec není zatržen, nelze zapsat více hodnot, než pro kolik je rezervováno místo.

Multiatributy nepatří do standardů SQL. Nelze je použít v indexech a proto vyhledávání nebo třídění podle některé z hodnot multiatributu bude pomalejší. Na práci s multiatributy existuje řada přirozených omezení i na klientské straně aplikace, například v oblasti zobrazování, editace, exportu a importu dat.

Místo na disku…

Multiatributy zaberou vždy součin délky jedné hodnoty a počtu rezervovaných hodnot plus 2 bajty. Pokud mají proměnný počet složek, zaberou navíc 6 bajtů plus dynamicky alokovaný prostor, který se přiděluje podle stejných pravidel, jako prostor pro sloupce proměnné velikosti.

Sledovací atributy

WinBase602 **umožňuje sledovat datum poslední změny, autora poslední změny a historii změn hodnot ve sloupci. Chcete-li něco z toho využít, musíte za sledovaným sloupcem definovat další sloupce, tzv.** **SLEDOVACÍ ATRIBUTY.**

Sledovací atributy mají jediný účel - vypovídat o tom, co se dělo s určitým (sledovaným) sloupcem tabulky. Hodnotu sledovacích atributů nelze odstranit ani změnit. Mění ji pou-

ze databázový systém v závislosti na akcích prováděných s hodnotou sledovaného sloupce .

Autorizace

Ke sloupci může být připojen sledovací atribut typu AUTORIZACE. V tom případě se sleduje, kdo provedl poslední změnu hodnoty.

Pozor: Exportem tabulky se sledovacím atributem **Autorizace** a jejím přenesením do jiné databáze se autorizační informace neztratí, ale může být přečtena pouze tehdy, pokud stejný uživatel (téhož ID) existuje i v cílové databázi. Uživatele lze přenést exportem a importem.

Datumovka

Ke sloupci může být připojen sledovací atribut typu DATUMOVKA. V takovém případě se sleduje, kdy byla provedena poslední změna hodnoty.

Historie

Ke sloupci může být připojen sledovací atribut typu HISTORIE. V takovém případě si databáze pamatuje nejen současnou hodnotu sloupce, ale také hodnoty předešlé. Do historie mohou být zaznamenávány kromě minulých hodnot také autoři a čas prováděných změn.

Při každé změně hodnoty se do historie automaticky zaznamenává její dosavadní hodnota. Uživatel nemůže odstranit ani změnit záznamy o historii. Pouze správce databáze může odstranit z historie záznamy starší než určité datum nebo ponechat pouze stanovený počet nejnovějších záznamů.

Umístění sledovacího atributu

To, ke kterému sloupci sledovací atributy patří, je dáno jejich vzájemnou polohou v definici tabulky. Všechny sledovací atributy musí být vždy umístěny **bezprostředně za** tím sloupcem, který mají sledovat.

Funkci sledovacího atributu určuje jeho typ. Typ **Autorizace** znamená sledování autora změn, typ **Datumovka** znamená sledování času a data změn, typ **Historie** znamená sledování historie hodnot.

Pořadí sledovacích atributů a historie

Ze sledovacích atributů lze použít libovolnou podmnožinu. Pokud se použije **Historie** a ještě některý další sledovací atribut, pak je významné pořadí, v němž jsou za sledovaným sloupcem uvedeny. Do historie se totiž zahrnují kromě hodnot sledovaného také hodnoty těch sledovacích atributů, které jsou *mezi* sledovaným sloupcem a atributem **Historie**.

Předpokládejme, že sledujeme sloupec **X** pomocí sledovacích atributů **A** typu **Autorizace**, **D** typu **Datumovka** a **H** typu **Historie**. Uvažujme tři případy pořadí těchto sloupců.

- **X H A D** - v historii se sledují pouze hodnoty **X**;
- **X A H D** - v historii budou u hodnot **X** také autoři změn, ale nikoli čas změn;
- **X A D H** - v historii jsou hodnoty **X**, čas i autoři změn.

Sledování změn v záznamu

Výše popsané umístění sledovacích atributů dovoluje sledovat zásahy do vybraného (sledovaného) sloupce. V případě autora a času změn má ovšem také smysl sledovat zásahy do **záznamu** jako do celku, bez rozlišení, kterého sloupce se zásah týkal. K tomu účelu můžete definovat sledovací atributy typu **Autorizace** a **Datumovka** před všemi ostatním sloupci tabulky. Pak se do nich zaznamená každá změna provedená s kterýmkoli sloupcem záznamu.

Na rozdíl od digitálního podpisu nejsou při implementaci sledovacích atributů použity šifrovací algoritmy, proto existuje teoretická možnost jejich pozměnění přímým přepisem vhodného místa v databázovém souboru. Tomu lze bránit přístupovými právy k databázovému souboru nebo jeho šifrováním.

Modifikování návrhu tabulky

Před zahájením změn ve struktuře tabulky je třeba si uvědomit možné nebezpečí ztráty dat. Pokud například zrušíte sloupec v definici tabulky, budou zničeny všechny hodnoty z tohoto sloupce ve všech záznamech tabulky.

Změny v definici tabulky může provádět pouze majitel tabulky (ten, kdo ji vytvořil), správce databáze nebo aplikace a případně další uživatelé, kterým toto právo bylo explicitně poskytnuto.

Stav sloupce

Informaci o změnách ve sloupci lze nalézt v poli **Stav sloupce**:

Nový - stav po přidání nového sloupce do návrhu. Upozorňuje, že sloupec ještě není součástí tabulky.

Nezměněný - se sloupcem nebyla při modifikování tabulky dosud provedena žádná změna.

Změněný - uživatel provedl změnu v typu nebo jiné vlastnosti sloupce. Do pole **Hodnota** lze zapsat, jaká data se mají do změněného sloupce přenést.

Zrušený - uživatel tento sloupec zrušil. Jméno sloupce je v závorkách, řádek je neaktivní.

Modifikaci návrhu tabulky ukončíte z menu ***Návrh*** příkazem **Provést změny** nebo **Odvolat změny**, případně tlačítkem na liště. Zvolíte-li provedení zadaných změn, **WinBase602** provede syntaktickou kontrolu popisu. Najde-li chybu, oznámí ji a umožní opravu.

Pozor!

Před provedením změn v tabulce obsahující data pečlivě zkontrolujte, zda je u každého změněného sloupce v poli **Hodnota** to, co se má ve sloupci objevit.

Během modifikování tabulky lze pomocí tlačítka na liště nebo příkazu **Ukázat** z menu ***SQL-příkaz*** zobrazit příkaz `ALTER TABLE`, který provede změny v tabulce.

Vkládání a rušení sloupců

Nové sloupce se obvykle vkládají na konec definice tabulky. Nový sloupec je nutno vložit mezi existující sloupce pouze tehdy, když k některému sloupci přidáváte sledovací atribut nebo když vkládáte digitální podpis sledující pouze část sloupců.

Chcete-li vložit nový sloupec na určité místo mezi jiné sloupce v tabulce, pak nejprve vyberte ten sloupec, **před** nějž chcete vkládat. Pak proveďte příkaz **Vložit řádek** z menu ***Editace*** nebo stiskněte klávesu Insert. Tím vznikne v popisu sloupců volný řádek, na kterém popíšete nový sloupec.

Vybraný sloupec můžete z návrhu tabulky zrušit příkazem **Zrušit řádek** z menu ***Editace***. Klávesový povel je Ctrl+F8.

Pokud rušený sloupec byl v návrhu tabulky již před otevřením návrháře, pak se jeho řádek zneaktivní a jméno sloupce se objeví v závorkách. Zrušíte-li nově přidaný sloupec, pak z návrhu prostě zmizí.

Změna jména sloupce

Chcete-li změnit jméno sloupce, který již v tabulce existuje, přepište jméno v příslušném sloupci návrháře a stiskněte Enter nebo Tab. Do pole **Hodnota** se nastaví původní jméno sloupce, aby se zajistil přenos hodnoty. Původní řádek se přesune na konec seznamu a bude označen jako **Zrušený**.

Změna typu sloupce

WinBase602 umožňuje, aby při změně typu sloupce uživatel nepřišel o svá data. Typ sloupce změňte pomocí comba s nabídkou typů. Poté do pole **Hodnota** zapište výraz, popisující typovou konverzi z typu starého na typ nový. Při první změně typu **WinBase602** sama navrhne vhodnou konverzi, při dalších změnách ve stejném sloupci již nechá vyplnění na vás.

Příklad:

Měníte-li sloupec CISLO typu **Real** na sloupec typu **Integer**, do pole hodnota patří:

```
Trunc(CISLO)
```

Měníte-li sloupec RETEZ z řetězce na číslo typu **Short**, použijte konverzi:

```
Str2int(RETEZ)
```

Přenesou se pouze ta data, která lze převést na číslo.

Pole Hodnota a přenos hodnot

Obsah pole **Hodnota** určuje, jaká hodnota se při modifikaci tabulky naplněné daty má přenést ze staré tabulky do vybraného sloupce v její nové verzi.

Výraz v poli **Hodnota** musí být stejného typu jako sloupec, kterého se týká. Pokud při modifikaci tabulky dochází ke změně typu sloupce, obsahuje vhodnou konverzní funkci. Potřebné konverze jsou stručně shrnuty v této tabulce:

Z typu \ na typ	Boolean	Char	Short	Integer	Money	Real	String	Date	Time	Ptr	Text
Boolean	OK	1	1	1	1	1	1	1	1	-	-
Char	2	OK	3	3	-	-	OK	-	-	-	-
Short	2	4	OK	OK	OK	OK	5	6	7	OK	-
Integer	2	4	OK	OK	OK	OK	5	6	7	OK	-
Money	2	-	OK	OK	OK	OK	8	-	-	-	-
Real	2	-	9	9	OK	OK	10	-	-	-	-
String	2	OK	11	11	12	13	OK	14	15	-	OK
Date	2	-	16	16	16	16	17	OK	-	-	-
Time	2	-	18	18	18	18	19	-	OK	-	-
Ptr	-	-	OK	OK	-	-	-	-	-	OK	-
Text	-	-	-	-	-	-	OK	-	-	-	OK

Tabulka popisuje možnosti konverze typu uvedeného v záhlaví řádku na typy uvedené v záhlavích sloupců. Obsah tabulky není symetrický.

Typ **Ptr** je v této tabulce zkratkou za typy **Pointer** a **Biptr**, tedy za jednosměrné a obousměrné ukazatele. Údaje o typu **String** se v plné míře vztahují i na typy **Csstring** a **Csistring**, navíc hodnoty těchto tří typů se dají navzájem přiřazovat bez konverzí.

Značky v tabulce znamenají toto:

značka	Význam
OK	lze přímo přiřadit, konverze není potřebná nebo proběhne automaticky;
pomlčka	nelze jednoduše konvertovat;
1	použijte podmíněný výraz ve tvaru *sloupec_typu_Boolean* ? *výraz1* : *výraz2*
2	použijte relaci ve tvaru: *sloupec* = *hodnota*
3	konvertujte funkcí Ord;
4	konvertujte funkcí Chr;
5	konvertujte funkcí Int2str;
6	využijte funkci Make_date;
7	využijte funkcí Make_time;
8	konvertujte funkcí Money2str;
9	konvertujte funkcí Trunc nebo Round;
10	konvertujte funkcí Real2Str;
11	konvertujte funkcí Str2Int;
12	konvertujte funkcí Str2Money;

13	konvertujte funkcí Str2Real;
14	konvertujte funkcí Str2Date;
15	konvertujte funkcí Str2Time;
16	využijte funkce Day, Month, Year;
17	konvertujte funkcí Date2str;
18	využijte funkce Hour, Minute, Second, Sec1000;
19	konvertujte funkcí Time2str.

Přesný popis funkcí naleznete v elektronické *Encyklopedii funkcí* ***WinBase602***.

Některé složitější konverze se dají provádět i ve dvou krocích, např. textový zápis roku ve sloupci ROK typu **String** převedete na datum prvního dne v roku voláním dvou funkcí:

```
Make_date(1, 1, Str2int(ROK) )
```

Častý případ konverze sloupce ZN typu **Char** obsahujícího hodnoty 'A' a 'N' na sloupec typu Boolean (značka 2) lze provést následovně:

```
Ord(ZN) = Ord('A')
```

Přiřazovat lze i sloupce s hodnotou proměnné délky a celé multiatributy najednou. Dokonce lze přiřadit i dva multiatributy proměnné velikosti.

Do pole **Hodnota** lze psát výrazy obsahující i ty sloupce, které rušíte.

Při modifikaci tabulky nemusí být vztah mezi sloupci jeden k jednomu. Pokud sloupce PLAT a PŘÍPLATEK nahrazujete sloupcem CELKOVÝ_PLAT, v jeho poli **Hodnota** bude výraz `PLAT+PŘÍPLATEK`. Pokud chcete rozdělit obsah sloupce DATUM NAROZ do tří sloupců DEN_NAROZ, MES_NAROZ a ROK_NAROZ, pak v jejich poli **Hodnota** budou po řadě výrazy `Day(`DATUM NAROZ`)`, `Month(`DATUM NAROZ`)`, `Year(`DATUM NAROZ`)`.

Přenos zrušených záznamů a ukazatelové vazby

Modifikováním struktury tabulky nejsou narušeny ukazatelové vazby ve vztahu k jiným tabulkám. Všechny záznamy si podrží svá absolutní čísla.

Pokud je do tabulky přidáván nebo z tabulky odstraňován nějaký sloupec nebo pokud se mění typ či délka některého sloupce, pak se při modifikování tabulky automaticky provede *uvolnění zrušených záznamů*. To znamená, že zrušené záznamy v tabulce nebude poté možno obnovit.

Indexy k tabulce a klíče

INDEX je pomocná struktura svázaná s tabulkou umožňující rychle vyhledávat záznamy podle hodnot v určitých sloupcích a rychle uspořádat záznamy podle těchto hodnot. Návrhář tabulky *popíše*, jaké indexy si pro ní přeje. O další údržbu a využití indexu se již pak stará pouze databázový server. Existence indexu ovlivňuje rychlost provádění operací s tabulkou: nepatrně zpomaluje provádění změn v tabulce a podstatně zrychluje vyhledávání a třídění dat.

Klíč

Index se specifikuje pomocí svého KLÍČE. Klíč je hodnota, podle níž index uspořádává záznamy. Klíč je v nejjednodušším případě sloupec tabulky, v obecnosti je zadán jedním až osmi výrazy, v nichž se vyskytují sloupce tabulky.

Druhy indexů

WinBase602 rozlišuje mezi třemi druhy indexů:

- INDEX UNIKÁTNÍ (jedinečný)- způsobí, že do tabulky nebude možno vložit dva záznamy se stejnou hodnotou klíčem, pokud hodnota klíče je jiná než NULL. Při pokusu vložit do tabulky záznam se stejným klíčem, jaký má záznam v tabulce již obsažený, dojde k chybě.
- INDEX NEUNIKÁTNÍ - hodnota klíče se smí v záznamech libovolně opakovat;
- PRIMÁRNÍ KLÍČ - speciální druh unikátního indexu, smí existovat pouze jeden pro tabulku.

Unikání indexy a hodnota NULL

Záznamy, které mají hodnotu klíče rovnou NULL, se do unikátních indexů nezaznamenávají. Proto pokud necháte obsah formuláře nebo kurzor setřídit podle hodnot sloupce, který je klíčem v unikátním indexu, záznamy s hodnotou NULL zmizí.

Omezení na indexy

V indexech nelze použít sloupce s typem proměnné velikosti, sloupce typu ukazatel, multiatributy a sledovací atributy. Lze tedy použít pouze sloupce typu **Boolean**, **Char**, **Short**, **Integer**, **Money**, **Real**, **Date**, **Time**, **Timestamp**, **String**, **Csstring, Csistring** a **Binary**.

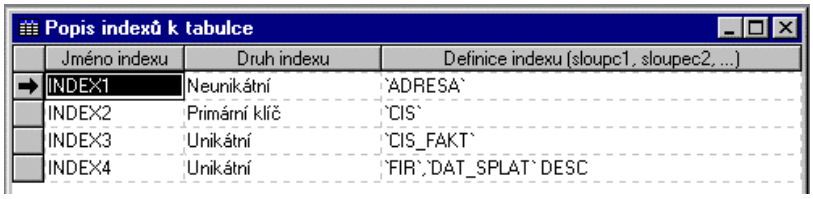
K tabulce je možno definovat nejvýše 24 indexů.

Jak definovat indexy

Indexy se definují při návrhu tabulky, a to jedním z dvou způsobů:

- Index, jehož klíč tvoří jediný sloupec, lze specifikovat v popisu tohoto sloupce. V pravé dolní části návrháře vyberete z comba nazvaného **Index** odpovídající druh indexu. Index lze zrušit, když na combu stisknete klávesu **Delete**.
- Složitější indexy se popisují ve zvláštním okně, které v návrháři tabulek otevřete (nebo zavřete) pomocí tlačítka na liště nebo příkazem **Indexy** z menu ***Vlastnosti***.

Okno s indexy



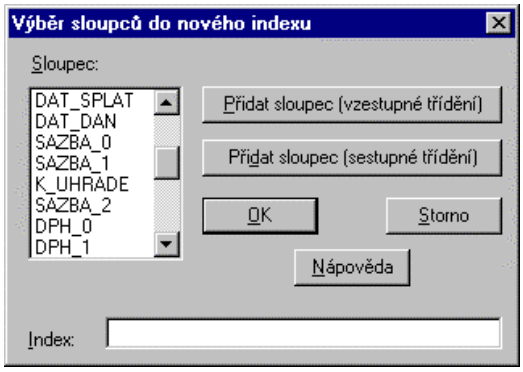
Každý řádek v indexovém okně popisuje jeden index. V editačním poli **Jméno indexu** určujete nepovinné jméno indexu. Pomocí comba **Druh indexu** zvolíte odpovídající druh. Do editačního pole **Definice indexu** zadejte klíč, tedy výraz nebo posloupnost výrazů oddělených čárkami. Za každým výrazem smí být slovo `DESC`, které vyjadřuje požadavek na *sestupné* uspořádání podle jeho hodnot, tedy od největších hodnot k nejmenším.

Jména sloupců obsahující nealfanumerické znaky musí být uzavřena v obrácených apostrofech, ostatní mohou.

Přidání nového indexu

Chcete-li přidat další index, stiskněte klávesu **Insert**. Objeví se dialog, který Vám pomůže s jeho sestavením.

Sestavování indexu



V seznamu vyberte sloupec, který má být v indexu. Podle směru třídění stiskněte jedno z dvou tlačítek **Přidat sloupec**. Má-li být v klíči indexu více sloupců, opakujte tuto akci. Stiskem tlačítka **OK** pak přenesete sloupce do okna s s popisem indexů, kde můžete definici indexu dále editovat, například zvolit druh indexu nebo pojmenovat jej.

Zrušení indexu

Index zrušíte tak, že označíte celý řádek jako ve standardním pohledu (myší kliknutím na políčko před prvním sloupcem, z klávesnice kombinací kláves **Shift**+**Mezerník**) a stisknete klávesu **Delete**.

Příklady:

Použitá tabulka FIRMY:

Index zadaný:

```
STAT,PSC,NAZEV
```

uspořádává záznamy v tabulce podle hodnot sloupce `STAT` a v případě jejich rovnosti podle hodnot sloupce `PSC`. Jsou-li oba sloupce ve dvou záznamech stejné, záznamy budou uspořádány podle hodnot sloupce `NAZEV`.

Index zadaný:

```
DAT_ZAPISU DESC,NAZEV
```

uspořádá záznamy nejprve **sestupně** podle hodnot ve sloupci `DATUM` (od nejnovějších k nejstarším) a v rámci stejných dat vzestupně podle sloupce `NAZEV`.

Index zadaný:

```
NAZEV+MESTO
```

uspořádá záznamy podle řetězcového spojení obou sloupců.

Sloupce typu řetězec znaků v indexu

Definujete-li index, v jehož klíči je sloupec typu String, Csstring nebo Csistring, můžete zadat, že součástí klíče má být pouze *prefix* hodnoty sloupce určité délky. To má smysl, pokud je hodnota sloupce poměrně dlouhá, chcete šetřit místem na disku nebo časem při indexování a nevadí vám, že při uspořádání se bude brát v úvahu pouze určitý prefix hodnoty.

Například v evidenci osob může být nutné počítat s příjmením dlouhým např. 35 písmen, ale pro alfabetické setřídění lze mnohdy v praxi vystačit s prvními deseti až patnácti písmeny.

Za jménem sloupce typu řetězec znaků v definici indexu uveďte v hranatých závorkách, kolik počátečních znaků má být zahrnuto do klíče.

Příklad:

Je-li sloupec NAZEV z tabulky FIRMY dlouhý 40 znaků, do popisu indexu můžete napsat:

```
NAZEV[20]
```

Integritní omezení v tabulce

INTEGRITNÍ OMEZENÍ slouží ke stanovení omezujících *podmínek* na hodnoty sloupců v záznamu. Databázový server nepovolí zapsat do tabulky záznam, který by tyto podmínky nesplňoval, nebo upravit obsah záznamu tak, že by podmínky byly porušeny.

Integritní omezení specifikovaná ve formulářích nebo implementovaná v programech hlídají na aplikační úrovni *určitý způsob* práce s daty. Integritní omezení specifikovaná

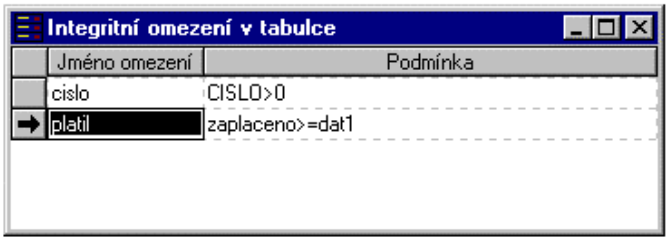
v tabulce prověřují *všechny způsoby* práce s touto tabulkou, aniž by programátor aplikace na to musel stále myslet.

Jak definovat integritní omezení

Integritní omezení se definují při návrhu tabulky, a to jedním ze dvou způsobů:

- jednoduchá omezení týkající se jednoho sloupce můžete uvést v popisu tohoto sloupce. V pravé dolní části návrháře zapíšete do pole **Integritní omezení** podmínku pro daný sloupec.
- složitější integritní omezení se popisují ve zvláštním okně, které v návrháři tabulek otevřete (nebo zavřete) tlačítkem na liště nebo příkazem **Integritní omezení** z menu ***Vlastnosti***.

Okno s integritními omezeními



Jména sloupců obsahující nealfanumerické znaky musí být uzavřena v obrácených apostrofech.

Přidání omezení

Chcete-li přidat další integritní omezení, stiskněte klávesu **Insert**. Do seznamu se přidá další řádek. Vyplníte **Jméno omezení** (nepovinné) a **Podmínku**. Pro podmínku můžete použít operátory, názvy sloupců a standardní funkce jazyka.

Zrušení omezení

Integritní omezení zrušíte tak, že označíte celý řádek jako ve standardním pohledu (myší kliknutím na políčko před prvním sloupcem, z klávesnice kombinací kláves **Shift** + **Mezerník**) a stisknete klávesu **Delete**.

Příklady:

V tabulce FAKTURY musí být datum zaplacení (ZAPLACENO) větší nebo rovno datu vystavení (DAT1). Do podmínky omezení zapište výraz:

```
ZAPLACENO >= DAT1
```

Referenční integrita mezi tabulkami

REFERENČNÍ INTEGRITA je nástroj, který pomáhá udržovat konzistenci dat v relačně propojených tabulkách. Relační propojení tabulek se obvykle definuje pomocí tzv. *relací* mezi tabulkami – viz zvláštní kapitolu tohoto manuálu.

Motivační příklad

Mějme tabulky FIRMY a FAKTURY. Tabulky jsou navrženy tak, že do seznamu faktur se nezapisují jména a sídla firem, ale pouze jejich identifikační čísla. Tato čísla spolu s dalšími podrobnými informacemi o firmě jsou uloženy v tabulce FIRMY. Tabulka FIRMY je v případě vkládání záznamu do tabulky FAKTURY pouze jakýmsi číselníkem firem, jehož jedna hodnota musí být dosazena do příslušného sloupce tabulky FAKTURY. Zajistit, aby mohla být vložena pouze vyhovující hodnota, je první úkol referenční integrity.

Na druhé straně při rušení některých záznamů v tabulce FIRMY můžete zjistit, že jste omylem zrušili údaj o firmě, která figuruje v tabulce faktur. Zajistit, že k tomu nedojde, je druhý úkol referenční integrity.

Pravidla referenční integrity

Referenční integrita dat se vždy definuje pro dvě tabulky. Tabulka, v níž definujete referenční integritu, se nazývá PODŘÍZENÁ tabulka. Tabulce, na níž se odkazuje, říkáme NADŘÍZENÁ. Referenční integrita propojuje hodnoty určitého sloupce v podřízené tabulce s hodnotami jiného sloupce v nadřízené tabulce a říká toto:

Hodnota sloupce v každém záznamu z podřízené tabulky má být *stejná* jako hodnota sloupce v některém záznamu z nadřízené tabulky (číselníku).

Definujete-li referenční integritu, server bude hlídat dodržení tohoto pravidla takto:

- Přidáváte-li záznam do podřízené tabulky, hodnotě sloupce definovaného v referenční integritě musí odpovídat hodnota sloupce v nadřízené tabulce.
- Přepis hodnoty nebo smazání záznamu v nadřízené tabulce bude povoleno jen tehdy, není-li v podřízené tabulce ani jeden záznam se stejnou hodnotou sloupce, pro nějž je definována referenční integrita.

Pravidla referenční integrity platí pro všechny operace s daty v daných tabulkách. Budete-li např. do podřízené tabulky importovat záznamy, import skončí, narazí-li na záznam nevyhovující integritě. Podobně, když se budete pokoušet zrušit tabulku, která pro jinou tabulku slouží jako nadřízená, referenční integrita to nepovolí, protože by to odporovalo druhému pravidlu.

Sloupec nadřízené tabulky zapojený do referenční integrity musí být v unikátním indexu. Sloupec podřízené tabulky musí být v libovolném indexu (unikátním nebo neunikátním).

Referenční integritu lze definovat i mezi enticemi (dvojicemi, trojicemi atd.) sloupců ve dvojici tabulek. Pak se vyžaduje rovnost hodnot odpovídajících si sloupců v entici a přítomnost indexů pro tyto entice.

Referenční integrita a hodnota NULL

Referenční integritní omezení, která říká, že ke každému záznamu v tabulce T musí existovat záznam v tabulce S takový, že hodnota sloupce A z tabulky T je stejná jako hodnota sloupce B z tabulky S, nemusí být dodrženo, pokud sloupec A má hodnotu

NULL. Do tabulky T lze tedy např. vložit záznam s hodnotami NULL ve všech sloupcích, aniž by se vyžadovala přítomnost záznamu v tabulce S, který by měl ve sloupci B také hodnotu NULL.

Totéž platí pro případ, že se referenční integrita týká entice sloupců A1,...,An a všechny mají hodnotu NULL.

Jak definovat pravidla referenční integrity

Stejně jako pro celý návrh tabulky i zde platí, že včasné a důkladné rozmyšlení problému ulehčí práci autorům aplikace i uživatelům, naopak ale pravidla referenční integrity vytvořené pokusně a "zapomenutá", stejně jako nepromyšlená struktura tabulek, přinesou více problémů než užitku.

Máte-li vhodně navržené tabulky a rozmyšlená relační propojení, můžete navrhnout pravidla referenční integrity dvou tabulek. **Referenční integrita se vždy definuje v tabulce podřízené**. V návrháři *podřízené* tabulky stisknete tlačítko na liště nebo provedete příkaz **Referenční integrita** z menu ***Vlastnosti***. Otevře se okno s dříve definovanými pravidly referenční integrity v této tabulce.

Okno s referenční integritou

Do pole **Místní sloupec(ce)** patří jméno sloupce z právě modifikované tabulky, jehož hodnota se má rovnat hodnotě určitého sloupce z tabulky druhé. Jméno této tabulky je v poli **Do tabulky**. Pole **Vzdálený(é) sloupec(ce)** obsahuje jméno sloupce z tabulky v předchozím poli. Jedná se o ten sloupec, jehož hodnoty tvoří číselník hodnot pro místní sloupec.

Je-li referenční integrita popsána enticemi sloupců, pak do polí **Místní sloupec(ce)** a **Vzdálený(é) sloupec(ce)** uvedete více jmen sloupců oddělených čárkami.

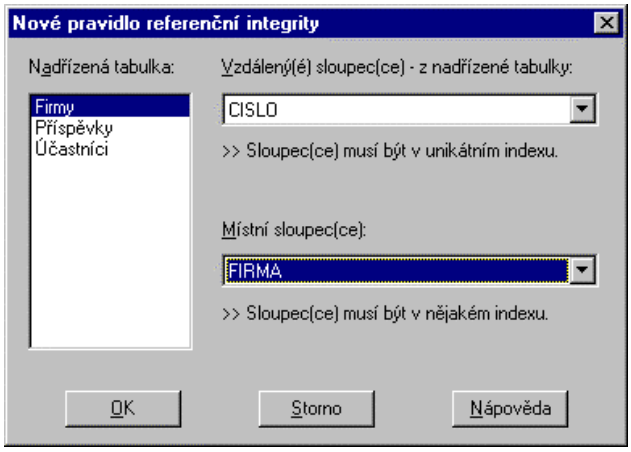
Jména sloupců obsahující nealfanumerické znaky musí být uzavřena v obrácených apostrofech. Jméno nadřazené tabulky se uvádí bez obrácených apostrofů.

Přidání pravidla

Chcete-li přidat pravidlo referenční integrity, stiskněte klávesu **Insert**. Otevře se dialogové okno umožňující výběr nadřízené tabulky a sloupců na obou stranách.

V combech se nabídnou pouze ty sloupce nebo jejich entice, které se vyskytují v indexech potřebného druhu.

Definování referenční integrity



Zrušení pravidla

Pravidlo referenční integrity zrušíte tak, že označíte celý řádek jako ve standardním pohledu (myší kliknutím na políčko před prvním sloupcem, z klávesnice kombinací kláves **Shift** + **Mezerník**) a stisknete klávesu **Delete**.

Akce při změně a akce při zrušení: aktivní referenční integrita

V popisu pravidla referenční integrity lze specifikovat, co se má stát, pokud se někdo pokusí v nadřazené tabulce provést operaci, která by referenční integritu porušila. Akce se předepisuje zvlášť pro pokus zrušit záznam, na nějž odkazují záznamu z podřízené tabulky, a zvlášť pro pokus změnit hodnoty v takovém záznamu. Lze si vybrat z těchto možností:

Volba	***Co se stane***
Odvolat akci	Akce se neprovede, požadavek skončí chybou a odvoláním změn
Dosadit NULL	V připojených záznamech podřízené tabulky se do sloupců, které podřízený záznam propojují s nadřízeným záznamem, zapíše hodnota NULL
Dosadit implicitní	V připojených záznamech podřízené tabulky se do sloupců, které podřízený záznam propojují s nadřízeným záznamem, zapíše implicitní hodnota určena v definici tabulky
Kaskádní změna	Změna provedena v nadřízené tabulce se provede i na připojených záznamech podřízené tabulky: při zrušení nadřazeného záznamu se zruší všechny k němu připojené podřízené záznamy, při změně hodnot propojovacího klíče se stejná změna provede i na klíčích v podřízených záznamech

Při pokusu narušit referenční integritu v podřízené tabulce dojde vždy k chybě a odvolání akce.

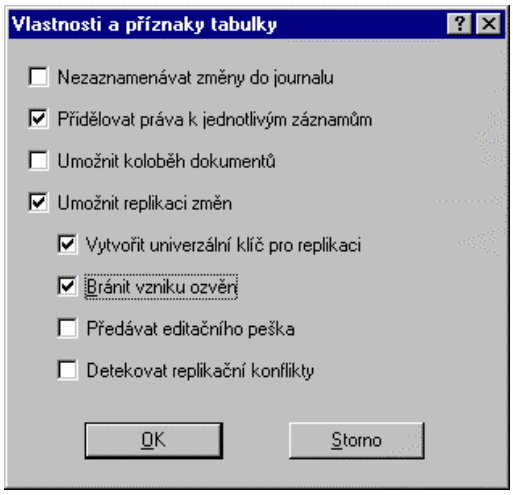
Referenční integrita v nákresu schématu aplikace

Alternativně lze referenční integritu definovat (nebo rušit) pomocí schématu aplikace obsahujícího obě tabulky a relaci mezi nimi. Ve schématech je názorněji než v návrháři vidět vztah mezi tabulkami.

Příznaky tabulky

Ke každé tabulce lze podle potřeby uvést další příznaky, kterými se ovlivní její chování. Příznaky ze zadávají v dialogovém okně, které se otevře tlačítkem na liště nebo příkazem **Příznaky** menu ***Vlastnosti***.

Příznaky tabulky



Nezaznamenávat změny do journalu - změny provedené v tabulce s tímto příznakem se nebudou zapisovat do journalu aktualizací (o journalu více v kapitole o správě systému v manuálu k SQL serveru). Příznak se používá zejména pro pracovní a pomocné tabulky, jejichž obsah nemusí být pomocí journalu obnovován.

Přidělovat práva k jednotlivým záznamům - příznak se použije, je-li potřebné, aby *jednotlivé záznamy* byly samostatně chráněny systémem práv. V takovéto tabulce bude smět konkrétní záznam přepsat pouze jeho autor, správce a ti, jimž byla k němu přidělena práva. Podrobnosti o přidělování práv najdete v kapitole *Správa uživatelů a jejich práv* v manuálu k serveru.

Umožnit koloběh dokumentů - je-li v aplikaci počítáno s koloběhem dokumentů této tabulky, zatrhněte tento příznak. Více v kapitole *Navrhování koloběhu dokumentů* v tomto manuálu.

Umožnit replikaci změn - příznak povolující, aby se obsah tabulky replikoval mezi databázovými servery. Zatržením čtverce se zpřístupní řada dalších dílčích nastavení, které jsou podrobně rozebrány v kapitole *Návrh replikujících aplikací* v *Příručce vývojáře*..