

Návrh dotazů

V databázových aplikacích zpravidla nestačí pracovat s daty tak, jak jsou uložena v jednotlivých tabulkách. Potřebujeme data třídit, vybírat podle různých kritérií a propojovat záznamy z různých tabulek. Síla databázových serverů a jazyka SQL spočívá v tom, že tyto akce nemusíme programovat, stačí, když vhodně popíšeme, jaká data potřebujeme.

Ve formulářích lze data vyhledávat a třídit pomocí nástroje nazvaného **QBE** (*Query By Example*). Je to rychlý a názorný způsob, nicméně má několik nevýhod: formulář si "nepamatuje" vyhledávací dotaz a je nutné ho pokaždé opakovat, nelze položit složité dotazy s použitím seskupování a výpočtů a zdroj dat je omezen na původní obsah formuláře.

Všechny tyto požadavky nesplněné v jednoduchém QBE můžete uplatnit jinde. Máte možnost navrhnout si plnohodnotný **DOTAZ** a vložit ho jako jeden z objektů do své aplikace.

Definování a použití dotazů

DOTAZ na databázi položíte tehdy, budete-li potřebovat vybrat z jedné nebo více tabulek množinu dat zadanou určitými podmínkami a vhodně ji uspořádat.

Jazyk SQL

Všeobecně používaným nástrojem pro zápis dotazů je jazyk **SQL** (*Structured Query Language*), konkrétně jeho příkaz **SELECT**. Dotazy jsou ve **WinBase602** vnitřně uloženy v tomto jazyce. Ve **WinBase602** lze pokládat dotazy převážně odpovídající úrovni **Intermediate** standardu **SQL 92**.

Návrhář dotazů

Pro vytvoření vlastních dotazů se však nebudete muset seznamovat s jazykem SQL, protože můžete využívat interaktivní prostředí *vizuálního návrháře dotazů*. Vizualní práce s dotazy je podstatně názornější než zápis dotazu v jazyce SQL.

Využití dotazu

Zápis dotazu se jako objekt uloží pod svým jménem do databáze. Použijete-li dotaz uložený v databázi jako zdroj dat pro formulář nebo sestavu, zobrazí se nebo vytisknou data tvořící odpověď na dotaz.

Pevný a proměnný kurzor

Pokud chcete položit dotaz v programu, pak můžete buď použít některý dotaz uložený v databázi a otevřít tzv. **PEVNÝ KURZOR** nebo vytvořit text dotazu přímo v programu a otevřít pro něj tzv. **PROMĚNNÝ KURZOR**.

Ovládání vizuálního návrháře dotazů

Nyní se seznámíme se způsobem práce s návrhářem dotazů a jednotlivými jeho částmi. Pokud nejste zbláhli v používání databázových dotazů, budete možná postrádat motivaci k popisovaným činnostem. Tu najdete ve sbírce příkladů na použití dotazů na konci této kapitoly.

Během práce s návrhářem se průběžně provádí syntaktická kontrola dotazu a signalizují se chyby. Návrhář má menší vyjadřovací schopnost než jazyk SQL, a proto existují (složitě) dotazy, které se dají zformulovat v SQL, ale nikoli v návrhář. Ty můžete vytvářet pomocí textového editoru – viz dále v této kapitole.

Vytvoření, testování a modifikace dotazu

Chcete-li vytvořit nový dotaz, který bude uložen v databázi, pak postupujte takto:

1. Na řídicím panelu vyberte v aplikaci kategorii **Dotazy**;
2. Proveďte akci **Vytvořit**, čímž otevřete návrhář dotazů;
3. Na obrazovce se objeví seznam tabulek a dotazů z aplikace. Vyberte první tabulku nebo dotaz, z nichž budete čerpat data, a stiskněte **OK**.

Chcete-li modifikovat dotaz pomocí vizuálního návrháře, postupujte takto:

1. Na řídicím panelu vyberte konkrétní dotaz;
2. Proveďte akci **Modifikovat**.



Pro uložení vytvořeného dotazu a opuštění vizuálního návrháře slouží tlačítka na liště nebo příkazy **Uložit** a **Konec návrhu** v menu *Návrh*. Uložit lze i dotaz, který dosud není syntakticky správně.



Během práce na dotazu lze vyvolat kompletní kontrolu syntaxe tlačítkem na liště nebo příkazem **Test syntaxe** z menu *Návrh*. Pomocí tlačítka na liště nebo příkazu **Ukázat** z menu *SQL-příkaz* lze kdykoli zobrazit text editovaného dotazu v jazyce SQL.

Otevření odpovědi na dotaz

Kdykoli během editace dotazu můžete otevřít standardní formulář obsahující odpověď na dotaz, tedy záznamy vybrané podle kritérií stanovených v dotazu. Tuto akci vyvoláte buď tlačítkem na liště nebo příkazem **Odpověď** z menu *Návrh*.



Doba vytváření odpovědi záleží na dotazu, na počtu záznamů v tabulkách figurujících v dotazu a na indexech k těmto tabulkám. Vytváření odpovědi lze kdykoli přerušit pomocí příkazu **Přerušit** v systémovém menu nebo klávesovým povelom **[Ctrl]+[Break]**.

Pokud odpověď na dotaz neobsahuje žádný záznam (žádný záznam nevyhověl zadaným podmínkám), místo prázdného formuláře se zobrazí pouze zpráva o této skutečnosti.

Struktura návrháře dotazů

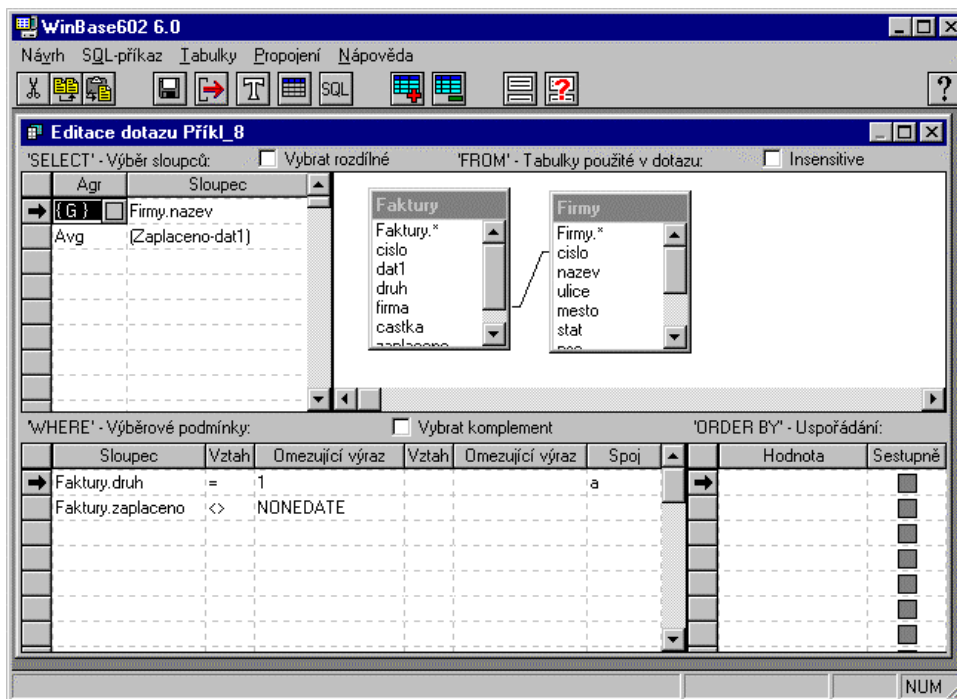
Okno návrháře dotazů se skládá ze čtyř hlavních oblastí:

- Vlevo nahoře je oblast, v níž se popíší jednotlivé sloupce, které se mají objevit v odpovědi na tento dotaz. Každý sloupec je popsán na jedné řádce. V levé části řádky lze pomocí comba vybrat agregační funkce. Tuto oblast budeme nazývat oblastí **SELECT**.
- Vpravo nahoře je oblast, v níž se nacházejí seznamy sloupců jednotlivých tabulek a dotazů použitých v navrhovaném dotazu. Mezi seznamy jsou čarami znázorněna propojení, která znamenají, že v odpovědi na dotaz musí oba propojené sloupce mít stejnou hodnotu. Oblast se dá posouvat ("rolovat") vpravo a vlevo pomocí táhla ležícího podél spodního okraje oblasti. Seznamy sloupců můžete v rámci této oblasti libovolně přemísťovat. Tuto oblast budeme nazývat oblastí **FROM**.
- Vlevo dole je oblast, do níž lze vyplnit *omezující podmínky* na hodnoty ve sloupcích. Do odpovědi na dotaz budou vybrány pouze záznamy, jejichž hodnoty splňují zde uvedené podmínky. Na každé řádce této oblasti lze zadat jednu nebo dvě omezující podmínky na jeden sloupec. Tuto oblast budeme nazývat oblastí **WHERE**.
- Vpravo dole je oblast, v níž se zadává uspořádání záznamů v odpovědi na dotaz. Na každé řádce této oblasti lze uvést sloupec z odpovědi na dotaz nebo výraz (příp. celé číslo). Vedle můžete vyznačit, zda uspořádání má být vzestupné nebo sestupné. Tuto oblast budeme nazývat oblastí **ORDER BY**.

Mimo tyto čtyři oblasti jsou v okně návrháře ještě tři přepínače:

Vyloučení duplicit	První přepínač s nápisem Vybrat rozdílné zabezpečí, že ve výsledné množině záznamů nebudou žádné dva záznamy stejné. Při porovnávání záznamů se ovšem nepřihlíží k hodnotám multiatributů a sloupců s hodnotou proměnné velikosti.
Uložení změn v datech	Druhý přepínač – Insensitive – zajistí, že případné změny provedené v odpovědi na dotaz (přepsání dat) se neprojeví v tabulkách (dotazech), z nichž dotaz vzniknul. Zápis dotazu v SQL se v tomto případě poněkud změní.
Negace podmínek	Poslední přepínač se jmenuje Vybrat komplement . Je-li tento přepínač zapnutý, pak platí <i>negace</i> zadaného omezení. To znamená, že do odpovědi na dotaz se vyberou ty záznamy, které <i>nesplňují</i> podmínky specifikované v oblasti WHERE . Negace se nevztahuje na propojení mezi výchozími tabulkami a dotazy.
Indikátory	V pravém horním rohu okna návrháře se objevují nápisy GROUP! a HAVING! indikující, že je zadáno vytváření skupin, resp. je položena podmínka na skupiny.

Okno návrháře dotazů



Princip ovládání návrháře

Pro přechody mezi oblastmi a mezi seznamy sloupců v oblasti **FROM** použijte myš nebo klávesový povel **F6**.

Drag & drop

Pro kopírování jmen sloupců ze seznamů v oblasti **FROM** do ostatních oblastí lze použít techniky *drag & drop* tedy "táhni a upust". Chcete-li přenést některé jméno, pak postupujte takto:

1. stiskněte levé tlačítko myši nad jménem, které chcete přenést;
2. držte tlačítko stále stisknuté a přesuňte myš na to místo, kam chcete jméno sloupce zapsat; přitom se mění tvaru kurzoru: na místech, kam lze jméno zapsat, má tvar obdélníku s nápisem, na ostatních místech vypadá jako zakazující dopravní značka;
3. uvolněte tlačítko myši; tím jméno zapíšete na zvolené místo.

Tabulky a dotazy použité v dotazu

Prvním krokem při editaci dotazu je obvykle výběr těch tabulek (případně dotazů), z nichž se budou vybírat data do odpovědi na vytvářený dotaz.

Zjednodušení:

V navrhovaném dotazu lze využít data z tabulek nebo i z jiných, již existujících dotazů. Pro jednoduchost budeme dále hovořit pouze o tabulkách. O dotazech, použitých v návrhu, platí přesně totéž, co o nich.

Tabulka a seznam sloupců

Každé tabulce použité v dotazu přísluší jeden seznam sloupců v oblasti **FROM**. Z tohoto seznamu budete vybírat jména a přenášet je do ostatních oblastí. První položka každého seznamu označuje všechny sloupce této tabulky a slouží pouze k případnému přenesení do oblasti **SELECT**.

Přidání a odstranění tabulky

Chcete-li do dotazu **přidat novou tabulku**, pak buď proveďte v menu **Tabulky** příkaz **Přidat** nebo stiskněte na ovládací liště příslušné tlačítko. Poté přepínačem zvolte nabídku tabulek nebo dotazů a vyberte ten objekt, který hodláte přidat.



Chcete-li **odstranit některou tabulku z dotazu**, pak její seznam sloupců označte myší nebo pomocí klávesy **F6** (okolo vybraného jména v seznamu se objeví nepřilíh zřetelný tečkovaný rám) a buď proveďte v menu **Tabulky** příkaz **Odebrat** nebo stiskněte příslušné tlačítko na ovládací liště.

Pozor!

Odstraní-li z dotazu některou tabulku, musíte ze všech oblastí návrhu dotazu odstranit všechny odkazy na její sloupce. Jinak bude dotaz signalizovat syntaktickou chybu.

Jména sloupců v dotazu

Při vytváření dotazu se využívají sloupce z tabulek zařazených do dotazu. V dotazu, který obsahuje více tabulek, však nemusí být jméno sloupce *jednoznačné*. Proto jsou v návrhářích sloupce tabulek označeny zápisy ve tvaru:

tabulka.sloupec

Alias

Pokud je v dotazu dvakrát (nebo vícekrát) použita *tatáž* tabulka, pak je nutno její výskyty rozlišit pomocí tzv. ALIASU. Alias je jméno, které jednoznačně určuje výskyt tabulky v dotazu. Sloupce tabulky, která má alias, se označují zápisem ve tvaru:

alias.sloupec

Přidáváte-li do dotazu novou tabulku, pak pokud v dotazu již stejná tabulka existuje, návrhář ji přidělí alias. Alias vytvoří tak, že ke jménu tabulky připojí rozlišující číslo. Pokud vám vyhovuje jiný alias, máte možnost jej změnit. Označte tabulku a pak z menu **Tabulky** proveďte příkaz **Alias**. Poté zadejte nové znění aliasu.

Změníte-li alias tabulky, musíte dodatečně zrušit nebo změnit ve všech oblastech návrhu výskyty aliasu původního.

Alias tabulky je vždy uveden v záhlaví seznamu sloupců tabulky.

Propojování tabulek

Pokud dotaz obsahuje více než jednu tabulku, pak jsou tyto tabulky zpravidla PROPOJENÉ. Propojením dvou sloupců ze dvou tabulek vyjádříte požadavek, aby v odpovědi na dotaz tyto dva sloupce měly stejnou hodnotu.

Znázornění propojení

Propojení sloupců je v dotazu znázorněno spojovací čarou mezi dvěma jmény v seznamech. Jelikož čáry vedou přímo, vyplatí se seznamy uspořádat tak, aby se čáry neprotínaly a nezatemňovaly tím návrh.

Jsou-li v dotazu *nepropojené* tabulky, pak se v odpovědi bude kombinovat každý záznam z první tabulky s každým záznamem z druhé tabulky. Výsledný počet záznamů může být značně vysoký. Nezádáte-li žádná omezení, je počet záznamů v odpovědi součinem počtů záznamů všech tabulek použitých v dotazu. Jsou-li tedy v dotazu 3 tabulky po 1000 záznamech, odpověď může obsahovat až 1 000 000 000 záznamů a doba jejího vytváření zcela jistě přesáhne meze Vaší trpělivosti. Navíc výsledek zřejmě není to, s čím jste počítali.

Jaké sloupce lze propojit

Propojit lze pouze takové dva sloupce, jejichž hodnoty lze porovnávat. Zpravidla se propojují sloupce stejných typů. Informace o typu sloupce se vypíše na stavový řádek, když kliknete na jméno sloupce v seznamu. Propojovat pochopitelně nelze první, speciální položky (*) v seznamech, které slouží k označení množiny všech sloupců tabulky.

Volba **Vybrat komplement** se na propojení nevztahuje. Ať je zatržena či nikoli, do odpovědi se budou vybírat pouze záznamy splňující rovnost hodnot sloupců specifikované propojením.

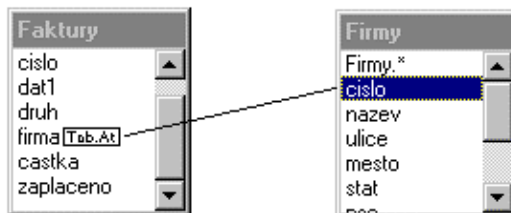
Automatické propojení

Je-li v aplikaci předem vytvořena *relace* mezi dvěma tabulkami, bude po vložení těchto dvou tabulek do návrháře propojení vytvořeno automaticky.

Vytvoření propojení

Chcete-li propojit dva sloupce, pak můžete použít myš nebo klávesnici. Při propojování pomocí myši jedno jméno uchopíte a upustíte jej nad druhým jménem.

Propojování tabulek myší



Při propojování pomocí klávesnice postupujte takto:

1. označte první jméno;
2. z menu **Propojení** proveďte příkaz **Propojit sloupce**;

3. označte druhé jméno;
4. opět z menu **Propojení** proved'te příkaz **Propojit sloupce**.

Zrušení propojení

Chcete-li propojení mezi sloupci zrušit, pak:

1. myší označte čáru, která je propojuje;
2. z menu **Propojení** proved'te příkaz **Zrušit propojení**.

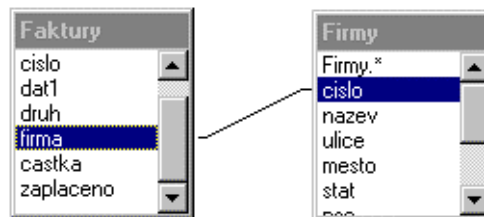
Pomocí klávesnice postupujte takto:

1. označte jedno z propojených jmen;
2. dvakrát proved'te v menu **Propojení** příkaz **Zrušit propojení**; po prvním provedení se propojovací čára zvýrazní, po druhém se zruší.

Vnitřní a vnější propojení tabulek

WinBase602 umí vytvořit dva typy propojení.

Vnitřní propojení

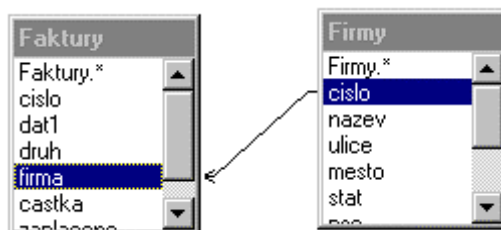


Vnitřní propojení (v SQL `INNER JOIN`) - není-li explicitně uveden jiný typ, jedná se vždy o běžné propojení dvou tabulek na rovnost. **WinBase602** vybere ty dvojice záznamů z první a druhé tabulky, pro něž se rovnají hodnoty propojených sloupců. Propojení je znázorněno jednoduchou čarou.

Vnější propojení (v SQL `LEFT (RIGHT) OUTER JOIN`) - použijete tehdy, chcete-li, aby v odpovědi byl zastoupen každý záznam jedné z tabulek nejméně jednou, a to i tehdy, když se v druhé tabulce nenajde žádný záznam, s nímž by se dal propojit. Scházející hodnoty sloupců z druhé tabulky budou v odpovědi nahrazeny hodnotami `NULL`.

Vnější plné propojení (v SQL `FULL OUTER JOIN`) - použijete tehdy, chcete-li, aby v odpovědi byl zastoupen každý záznam z každé tabulky nejméně jednou, a to i tehdy, když se v protější tabulce nenajde žádný záznam, s nímž by se dal propojit. Scházející hodnoty sloupců z protější tabulky budou v odpovědi nahrazeny hodnotami `NULL`.

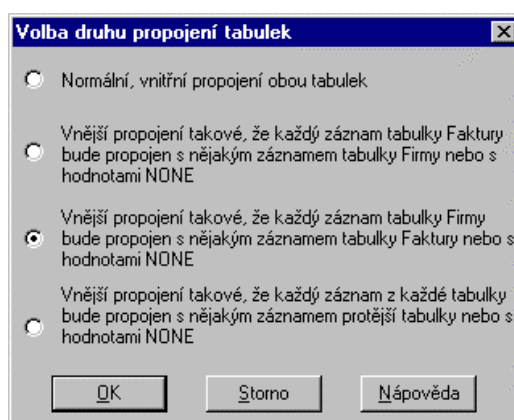
Propojení je v návrháři znázorněno šipkou, která vede z tabulky, z níž se berou všechny záznamy do tabulky, která doplní buď odpovídající hodnoty nebo `NULL`. Ve sbírce má příklad na vnější propojení číslo 14.

Vnější propojení

Pro vnější propojování více tabulek se vizuální návrhář dotazů příliš nehodí, protože v něm nelze znázornit pořadí propojování. Plnou kontrolu nad vnějším propojováním více tabulek poskytne formulace nebo následná kontrola dotazu přímo v jazyce SQL.

Vytvoření a zrušení vnějšího propojení

Vložte do návrháře obě tabulky a vytvořte normální propojení. Stiskem pravého tlačítka na spojovací čáře nebo příkazem **Druh propojení** z menu **Propojení** otevřete okno s výběrem druhů propojení.

Druhy propojení

Volíte-li vnější propojení, musíte určit jeho *směr*. Aby nedošlo k omylu, jsou v okně směry přímo vypsány. Vyberete jednu z možností a stisknete tlačítko **OK**.

Výběr sloupců a výrazů do odpovědi

V oblasti SELECT popíšete, jaké sloupce se mají objevit v odpovědi na dotaz. Můžete v ní uvést:

- jméno některého sloupce z některé tabulky z použité v dotazu;
- hvězdičku (*), která symbolizuje všechny sloupce všech použitých tabulek;
- jméno tabulky, tečku a hvězdičku, což symbolizuje všechny sloupce uvedené tabulky;

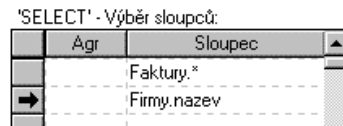
- agregační funkce, pro výpočet součtů, průměrů, maxim a podobně;
- výraz obsahující jména sloupců, konstanty, funkce SQL, agregační funkce nebo subdotazy.

Sloupce tabulek v odpovědi

Sloupce tabulek, jejichž hodnoty chcete mít v odpovědi, přeneste myší ze seznamu nebo zapište do oblasti SELECT.

Všechny sloupce z tabulky

Pokud chcete zahrnout do odpovědi *všechny* sloupce *některé* tabulky, pak do pravé části oblasti SELECT přeneste první položku seznamu tvořenou jménem resp. aliasem tabulky a hvězdičkou.



Chcete-li vybrat všechny sloupce *všech* tabulek, pak stačí do oblasti SELECT zapsat jednu hvězdičku. Tato hvězdička v této oblasti již je v každém nově vytvořeném dotazu. Hvězdičku nelze použít v dotazech, obsahujících seskupování.

Agregační funkce v dotazu

V dotazu můžete použít tzv. agregačních funkcí **COUNT** (počet záznamů s hodnotou různou od NULL), **SUM** (součet), **AVG** (průměr), **MAX** (maximum) a **MIN** (minimum). Tyto funkce se zapisují do levé části oblasti SELECT pomocí combo boxu. Všechny agregační funkce mají jako parametr jméno sloupce nebo výraz zapsaný na téže řádce s výjimkou funkce **COUNT**, která může mít jako parametr i hvězdičku – pak vrátí počet všech záznamů, bez ohledu na hodnoty.

Položka v combu označená **{G}** slouží k definování skupiny. Vytvářet skupiny můžete též podle hodnoty výrazu - viz příklad 13.



Nejsou-li definovány skupiny, pak v odpovědi na dotaz obsahující agregační funkce bude jediný záznam. V tomto záznamu budou hodnoty agregačních funkcí spočtené přes všechny záznamy vyhovující kritériím v dotazu. Jsou-li definovány skupiny, pak se záznamy vyhovující kritériím rozdělí do skupin podle hodnot sloupců označených **{G}**. V odpovědi se objeví jeden záznam pro každou skupinu a v něm budou hodnoty agregačních funkcí spočtené nad záznamy ve skupině.

Při definování dotazů s agregačními funkcemi přes skupiny platí důležité pravidlo: **jméno sloupce v oblasti SELECT musí být buď parametr agregační funkce nebo se přes něj seskupuje.**

Obecné výrazy v dotazu

V pravé části oblasti SELECT může stát i obecný výraz tvořený jmény sloupců, agregačními funkcemi a funkcemi jazyka SQL. V tomto případě výraz zapíšete do pravé části oblasti SELECT a levou část necháte nevyplněnou.

'SELECT' - Výběr sloupců:

	Ágr	Sloupec
		zaplacenodatl
		"Firma "+nazev

Při tvoření výrazu nezapomeňte, že všechny použité agregační funkce se vždy počítají přes **stejnou** množinu záznamů, **nelze** tedy mít výraz typu:

$$\text{SUM}(\text{za_skupinu}) / \text{SUM}(\text{za_vsechno}) .$$

Agregační funkce nelze do sebe vnořovat.

Jména sloupců v odpovědi na dotaz

Všechny sloupce odpovědi na dotaz jsou pojmenované. Je-li sloupec odpovědi sloupcem některé z výchozích tabulek, pak bude mít stejné jméno. Je-li sloupec odpovědi definován výrazem, bude označen `EXPRn`. První výskyt výrazu bude označen `EXPR1`, druhý `EXPR2` atd. Sloupce zadané jako hodnoty agregačních funkcí se označují `SUMn`, `COUNTn`, `MAXn`, `MINn` a `AVGn`.

Vlastní jména
sloupců

Autor dotazu má možnost předefinovat jména sloupců automaticky vytvořená dle předchozího odstavce, a přidělit sloupcům vlastní jména. V oblasti SELECT se označí sloupec odpovědi, který se má přejmenovat a provede se příkaz **Jméno sloupce** z menu *Návrh*. Do okna, které se otevře, zapíšete nové jméno sloupce. Skutečnost, že sloupec má zadané jméno, signalizuje slůvko **As** v záhlaví řádky.

'SELECT' - Výběr sloupců:

	Ágr	Sloupec
As { G }		Firmw.nazev
As Count		Faktury.cislo

Zadávání podmínek na výběr záznamů do odpovědi

Podmínky, podle nichž se vyberou záznamy do odpovědi na dotaz, se specifikují v oblasti **WHERE**.

Každá podmínka obsahuje v prvním sloupci jméno sloupce z některé tabulky, na jehož hodnotu je tato podmínka kladena. V druhém a třetím sloupci jsou pak relační znaménko a výraz.

Ve čtvrtém a pátém sloupci lze zadat další podmínku. Jsou-li takto na jedné řádce zadány dvě podmínky, musí být splněny obě. Dvě podmínky se typicky používají k vyjádření požadavku, aby hodnota ležela v určitém intervalu (mezi zadanými hodnotami).

Následující obrázek ilustruje situaci, kdy potřebujete zadat podmínku pro výběr cenového rozmezí; zde např. "faktury, za něž jsme zaplatili mezi 1000,- a 5000,- Kč".

Spojení dvou podmínek

'WHERE' - Výběrové podmínky: Vybrat komplement

	Sloupec	Vztah	Omezující výraz	Vztah	Omezující výraz	Spoj
<input checked="" type="checkbox"/>	Faktury.castka	>	1000	<	5000	

Relační operátory

Při definování podmínky můžete použít celou řadu relačních operátorů. Jejich nabídka se zpřístupní pomocí combo boxu ve sloupečku **Vztah** v oblasti WHERE. Relační znaménka a jejich význam jsou obsažena v následující tabulce:

Znaménko	Význam	Poznámka
=	rovná se	
>	větší než	
<	menší než	
>=	větší nebo rovno než	
<=	menší nebo rovno než	
<>	nerovná se	
.=	je prefixem	Řetězcový operátor
.=.	Obsahuje	řetězcový operátor, též pro typ TEXT
~	Je podobný	řetězcový operátor
..	obsahuje hodnotu	pouze pro multiatributy
IS	predikát používaný klauzulích jako IS NULL, IS TRUE, IS NOT UNKNOWN,...	
IN	náležení do seznamu nebo subdotazu	

Logické operátory

Podmínky zadané na jednotlivých řádkách se spojují logickými operátory v šestém sloupci. Lze zvolit spojku **A** nebo spojku **NEBO**. Podmínky spojené spojkou **A** musí být splněny všechny současně, z podmínek splněných spojkou **NEBO** stačí splnit jednu (kte-

roukoli). Spojka **A** má přednost před spojkou **NEBO**, proto se skupiny podmínek spojených pomocí **A** poté spojují spojkou **NEBO**.

Předešlý příklad na obrázku si rozšíříme o dovětek “a jsou to faktury došlé”.

Další podmínka

	Sloupec	Vztah	Omezující výraz	Vztah	Omezující výraz	Spoj
→	Faktury.castka	>	1000	<	5000	a
	Faktury.druh	=	1			

Definování skupin záznamů

Větší část problému byla vysvětlena v kapitole o naplnění oblasti SELECT návrháře.



Stiskem tlačítka na liště nebo provedením příkazu **Skupiny** z menu *Návrh* se otevře okno pro zápis definice skupin. Jsou-li již skupiny definovány z části SELECT, v editační řádce okna vidíte příslušnou definici. Praktické použití tohoto příkazu spočívá v tom, že je možné změnit automaticky zjištěné skupiny. V praxi se příliš nevyužívá.

Přítomnost definice skupiny v dotazu indikuje slovo **GROUP!** v pravé horní části návrháře.

Zadávání podmínek pro skupiny



Ve složitých dotazech se skupinami se někdy vyžaduje omezit výslednou množinu skupin. Takové omezení se zadává do okna otevřeného tlačítkem na liště nebo příkazem **Podmínka na skupiny** z menu *Návrh*.

Sloupce, přes které se neprovádí seskupování, se v podmínce mohou vyskytnout pouze v agregačních funkcích. Typickým použitím této podmínky je výběr pouze těch skupin, na nichž mají agregační funkce určité hodnoty.

Přítomnost definice podmínky na skupiny v dotazu indikuje slovo **HAVING!** v pravé horní části návrháře.

Uspořádání záznamů v odpovědi

Uspořádání záznamů v odpovědi zadáte tak, že do oblasti **ORDER BY** vyplníte výrazy, podle jejichž hodnot se bude uspořádávat. Je-li zatržen čtverec **Sestupně**, budou se záznamy uspořádávat sestupně.

**Uspořádání
odpovědi**

'ORDER BY' - Uspořádání:

	Hodnota	Sestupně
	Faktury.dat1	<input checked="" type="checkbox"/>
	Firmy.nazev	<input type="checkbox"/>
		<input type="checkbox"/>

Výrazy uvedené výše mají při uspořádání přednost před výrazy uvedenými níže. Hodnota výrazu uvedeného níže se uplatní pouze tehdy, pokud se v porovnávaných záznamech rovnají hodnoty všech výrazů uvedených výše.

V oblasti ORDER BY se smí použít pouze ty objekty, které jsou uvedeny v oblasti SELECT.

Chcete-li třídit podle hodnot výrazu použitých v části SELECT, je třeba místo výrazu zapsat jeho název. Název výrazu se přiděluje buď automaticky (EXPR_n, SUM_n, COUNT_n apod.) nebo se použije název zadaný pomocí příkazu **Jméno sloupce**.

Jiný ekvivalentní způsob definice třídění se zapisuje pomocí pořadových čísel položek z oblasti SELECT. Tento způsob je zde k dispozici pouze z důvodů kompatibility se staršími normami SQL.

Chcete-li třídit záznamy vzniklé sjednocením (UNION) dvou dotazů s nestejně se jmenujícími položkami v části SELECT, použijte jména z prvního SELECTu.

Vlastnosti dotazů a jejich optimalizace

Editovatelné a needitovatelné dotazy

V závislosti na formulaci dotazu vytvoříte dotaz jednoho ze dvou druhů: buď EDITOVATELNÝ nebo NEEDITOVATELNÝ. Liší se v tom, zda lze měnit obsah databáze přepisováním odpovědi na dotaz.

**Editovatelné
dotazy**

Editovatelné dotazy obsahují ve své odpovědi odkazy na data uložená v tabulkách. Přepíšete-li údaj v odpovědi na editovatelný dotaz, dojde ve skutečnosti k přepsání příslušného údaje v některé tabulce. Každý záznam v odpovědi na editovatelný dotaz odkazuje buď na záznam z některé tabulky, nebo na entici záznamů pocházejících po jednom z entice tabulek uvedených v klauzuli FROM v dotazu.

Editovatelný dotaz vytvoříte tehdy, když dodržíte tato pravidla:

- v části SELECT nepoužijete agregační funkce;
- v dotazu nepoužijete seskupování záznamů (klauzule GROUP BY);
- nepoužijete klauzule UNION, INTERSECT ani EXCEPT.

Ne všechny sloupce v editovatelné dotazu musejí být editovatelné. Ty sloupce, které odpovídají výrazu v klauzuli SELECT, se nedají editovat nikdy.

Needitovatelné dotazy

Needitovatelné dotazy mají svoji odpověď uloženou v soukromé tabulce. Tato pomocná dočasná tabulka vznikne při vytváření odpovědi vhodnými výpočty a kopírováním dat z databázových tabulek. Při uzavření dotazu bude dočasná tabulka zrušena. Data v odpovědi na needitovatelný dotaz nelze a nemá smysl přepisovat, protože taková změna by stejně neovlivnila stav databáze.

Needitovatelný dotaz vznikne, pokud použijete agregační funkci v klauzuli SELECT, použijete seskupování záznamů nebo klauzule UNION, INTERSECT nebo EXCEPT.

Insensitive

Lze také vytvořit dotazy, které jsou sice editovatelné, ale změny zapsané do odpovědi se neuloží do příslušných tabulek, ze kterých dotaz vzniknul. Takovýto dotaz se navrhne zatržením čtverce **Insensitive** v pravém horním rohu návrháře.

Optimalizace dotazů

OPTIMALIZACE DOTAZU je nalézání takového postupu vyhodnocení dotazu, který zabere co nejméně času a dalších zdrojů na serveru.

Optimalizace dotazu je úkolem serveru. Server disponuje řadou algoritmů pro úpravu dotazu tak, aby se dal rychleji vyhodnotit. V některých případech se však může stát, že se serveru nepodaří najít optimální postup - například proto, že nemá potřebné informace o míře selektivity různých omezujících podmínek.

Autor dotazu může v těchto případech serveru napovědět, jakým způsobem má dotaz vyhodnocovat. Pro kteroukoli tabulku použitou v dotazu může určit, který její index se má použít k výběru záznamů.

Tuto informaci lze zapsat pouze do textové podoby dotazu. Za jméno tabulky v klauzuli FROM se napíše klíčové slovo INDEX a jméno indexu. Optimalizátor dotazu pak bude ignorovat všechny ostatní indexy k tabulce, a použije-li nějaký, pak pouze ten, který je určen.

Informace o postupu vyhodnocení dotazu

Postup, který server použije při vyhodnocování dotazu, lze vypsát pomocí tlačítka **Optimalizace** na řídicím panelu. Detailní struktura tohoto zápisu vykračuje za rámec znalostí uživatele **WinBase602**. Pozornost je třeba věnovat zejména těmto údajům umístěným za jmény tabulek:

tabulka accessed by index *n* - k vyhledání vhodných záznamů v tabulce se využívá index *s* uvedeným pořadovým číslem (v pořadí dle návrhu tabulky, od nuly);

tabulka exhaustive satisfying conditions *podmínka* - k vyhledání vhodných záznamů v tabulce se projdou všechny její záznamy a vyberou se ty, které vyhovují zadané podmínce;

tabulka exhaustive - použijí se všechny záznamy z tabulky.

Ruční optimalizace dotazu směřuje k tomu, aby:

- co nejvíce tabulek bylo procházeno pomocí indexu;
- přitom se využíval index, který vybere co nejmenší množství záznamů;
- procházejí-li se všechny záznamy v některé tabulce, aby byly filtrovány podmínkou.

Práce s dotazy v jazyce SQL

Databázový server interně pracuje s dotazy zapsanými v jazyce SQL. Autor databázové aplikace může buď využívat vizuální návrhář, který jej od jazyka SQL odstíní, anebo může dát přednost přímému zápisu dotazů v SQL pomocí textového editoru.



Pro libovolný dotaz uložený v databázi je jeho zápis v jazyce SQL vidět na řídicím panelu v pravé oblasti na stránce **SQL**.

Textová editace dotazu

Chcete-li vytvořit nový dotaz přímo v SQL, pak:

1. Na řídicím panelu vyberte v aplikaci kategorii **Dotazy**;
2. Proveďte akci **Vytvořit**;
3. V nabídnutém seznamu tabulek a dotazů stiskněte tlačítko **Návrh dotazu v SQL**.

Chcete-li modifikovat existující dotaz pomocí textového editoru, pak:

1. Na řídicím panelu vyberte konkrétní dotaz;
2. Proveďte akci **Editovat SQL**.

Pokud chcete editovat dotaz v samostatném okně, pak při otevírání editoru z řídicího panelu podržte stisknutou klávesu **[Ctrl]**. Pro editaci lze také použít externí textový editor, který zvolíte v parametrech **WinBase602** v menu **Nástroje**. Externí editor se volá pomocí kontextového popup menu v seznamu objektů na řídicím panelu. Nedoporučujeme používat takové externí editory, které by do textu mohly vložit vlastní formátovací znaky.

Z prostředí vizuálního návrháře lze do textové editace přejít pomocí příkazu **Editovat** z menu **SQL-příkaz**.



Během editace textu dotazu lze kdykoli zkontrolovat jeho syntaxi pomocí tlačítka na liště nebo příkazu **Překlad**.

POZOR:

Dotaz pozměněný textovým editorem již nemusí být dále čitelný pro vizuální návrhář dotazů (někdy stačí, aby ubyly zdánlivě nepodstatné závorky). V editoru lze však zadat i dotazy, jejichž složitost přesahuje schopnosti vizuálního návrháře – například ty, které využívají operace UNION, INTERSECT nebo EXCEPT.

Syntaxe dotazu v SQL

Textový zápis dotazu tvoří příkaz SELECT z jazyka SQL. Jeho zjednodušená syntaxe vypadá takto:

```
SELECT [DISTINCT] seznam_položek
FROM výčet_tabulek_nebo_dotazů
[WHERE podmínka]
[HAVING podmínka]
[ORDER BY popis_uspořádání]
[GROUP BY seznam_výrazů]
```

přičemž části v hranatých závorkách mohou chybět.

S popisem jednotlivých částí příkazu se můžete seznámit ve sbírce příkladů dotazů, kde ke každému příkladu je uvedena i syntaxe příkazu SELECT v jazyce SQL. Podrobnější popis najdete v elektronické dokumentaci k jazyku SQL.

Sbírka příkladů dotazů

Příklady budou řazeny od nejjednodušších k obtížnějším. Po vysvětlení bude vždy následovat textový zápis dotazu v jazyce SQL.

Používat se budou tabulky FAKTURY a FIRMY z aplikace PŘÍKLADY. Použité dotazy naleznete v téže aplikaci nazvané DOTAZ_01 až _N. Tabulka FIRMY je jednoduchý adresář našich obchodních partnerů. Má tuto strukturu:

Sloupec	Typ	Popis
CISLO	INTEGER	číslo firmy; primární klíč
NAZEV	CSISTRING[40]	název firmy
MĚSTO	CSISTRING[20]	sídelní město
ULICE	CSISTRING[20]	ulice a číslo
STAT	CSISTRING[4]	zkratka státu (CR nebo SR)
PSC	CSISTRING[6]	směrovací číslo
STATUS	CHAR	O = odběratelé; D = dodavatelé
DAT_ZAPISU	DATE	datum zavedení do tabulky
PRACOVNÍK	CSISTRING[20]	Odpovědný pracovník

Tabulka FAKTURY je seznam všech faktur příšlých i odešlých. Má tuto strukturu:

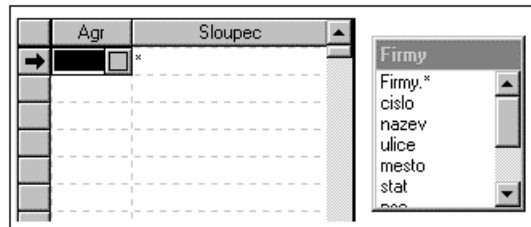
Sloupec	Typ	Popis
CISLO	SHORT	číslo faktury; primární klíč
DAT1	DATE	datum vystavení nebo příchodu
DRUH	SHORT	1 = došlé; 2 = odeslané
FIRMA	INTEGER	číslo firmy z tabulky firem
CASTKA	MONEY	kolik peněz
ZAPLACENO	DATE	datum zaplacení

Povšimněte si, že do tabulky FAKTURY se nezapisují celé názvy firem, ale pouze odkazy do adresáře firem tvořené číslem firmy ve sloupci FIRMA. Toto propojení tabulek nazýváme propojení relační.

Příklad 1

Zadání: Vyberte všechny platné záznamy z tabulky FIRMY.

Popis řešení: Vytvořte nový dotaz a vyberte tabulku FIRMY. V části SELECT se automaticky nastaví hvězdička, která znamená "všechny sloupce z tabulky". Nic víc není potřeba dělat.



SQL: SELECT *
FROM Firma

Poznámka: Čím se odpověď do tohoto dotazu liší od formuláře z tabulky FIRMA? Na první pohled ničím, ale rozdíl tu je. Zatímco v tabulce jsou uloženy i zrušené záznamy, do odpovědi se vyberou pouze platné záznamy. Tento rozdíl se projeví v absolutním číslování záznamů. Tato čísla záznamů hrají důležitou roli zvláště při programování.

Příklad 2

Zadání: Vypište názvy firem (z tabulky FIRMY) a seřadíte je podle názvu.

Popis řešení: Vložte do návrháře tabulku FIRMY. Pomocí myši přeneste sloupec NÁZEV ze seznamu do pravé části oblasti SELECT. Tentýž sloupec přeneste do oblasti ORDER BY.



```
SQL: SELECT Firma.nazev
      FROM Firma
      ORDER BY Firma.nazev
```

Poznámka: Povšimněte si, že návrhář zapisuje názvy sloupců ve tvaru *Jméno_tabulky.Jméno_sloupce*. Je to z toho důvodu, že při použití více tabulek se mohou vyskytnout dva sloupce stejného jména. V tomto případě je možné příkaz (např. v programu) zadat i jako:

```
SELECT nazev
FROM Firma
ORDER BY nazev
```

Příklad 3

Zadání: Vypište názvy firem z ČR a jméno jejich odpovědného pracovníka.

Popis řešení: Vložte do návrháře tabulky FIRMY. Do oblasti SELECT přeneste sloupce NAZEV a PRACOVNIK. Uchopte myší sloupec STAT a upusťte ho v oblasti WHERE nad prvním řádkem sloupce Sloupec (případně запиšte ručně). Ve vedlejším sloupečku vyberte z nabídky operátorů potřebnou rovnost (=) a v dalším sloupci запиšte hodnotu "CR" včetně uvozovek.

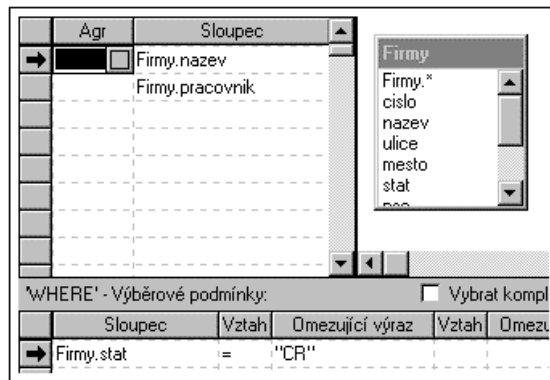
```
SQL: SELECT Firma.nazev, Firma.pracovnik
      FROM Firma
      WHERE (Firma.stat="CR")
```

Poznámky: a) Výčet sloupců za slovem SELECT se odděluje čárkami.

b) Podmínka v části WHERE se v textové podobě dává do kulatých závorek. V takto jednoduchém výrazu to není nutné, ve složitějších případech mohou chybějící závorky způsobit potíže.

c) V podmínce WHERE se mohou vyskytovat i sloupce nepoužité v části SELECT.

- d) Na tomto příkladu si můžeme ukázat i použití čtverce **Vybrat komplement**. Jeho zaškrtnutím dostaneme z podmínky 'Firma z ČR' podmínku 'Firma odevšad, pouze ne z ČR'. Jinými slovy dostaneme negaci původní podmínky.



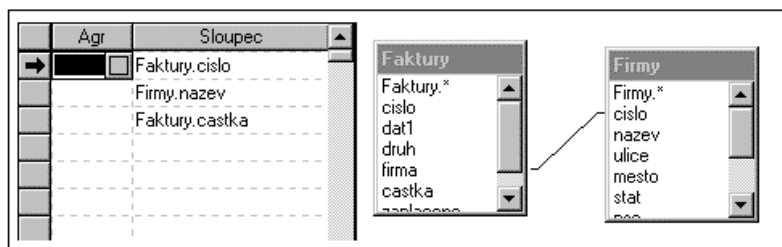
Příklad 4

Zadání: Připojte fakturám vždy název a adresu firmy.

Popis řešení: Do návrháře vložte obě tabulky: FIRMY i FAKTURY. V seznamech nalistujte jména sloupců, která tvoří relační propojení, tzn. v tabulce FIRMY sloupec CISLO a v tabulce FAKTURY sloupec FIRMA, tak, aby nebyla schována za okrajem seznamu. Uchopte myší jeden z této dvojice a upusťte ho nad druhým sloupcem. Mezi seznamy se vytvoří tenká čára spojující oba sloupce.

Pokud je v aplikaci navrženo schéma a vytvořena relace mezi těmito dvěma tabulkami, propojení se vytvoří samo.

Doplňte položky do seznamu v části SELECT. Tím je úkol splněn.



SQL: SELECT Faktury.cisko, Firmy.nazev, Faktury.castka
FROM Faktury, Firmy

```
WHERE Faktury.firma=Firmy.cislo
```

Poznámky: a) Všimněte si, že ačkoliv se propojení provedlo v části FROM návrháře, jeho popis je zapsán tam, kam patří - totiž do části WHERE.

b) Do formuláře do odpovědi na takto položený dotaz (propojení více tabulek) nelze vkládat nové záznamy.

c) Uvědomte si, co by se stalo, kdybyste zapomněli propojit obě tabulky a neměli další omezující podmínky. Návrhář sám od sebe nepozná, které dva sloupce spolu souvisí (vztah nemusí být pouze na rovnost!) a proto začne vytvářet odpověď tak, že spojí každý záznam z první tabulky s každým z druhé tabulky.

Upravte si tento dotaz ještě takto: vyberte pouze sloupce CISLO a CASTKA z tabulky FAKTURY a NAZEV z tabulky FIRMY a nechte si výsledek seřadit podle sloupce NAZEV. Výsledek pak může vypadat takto:

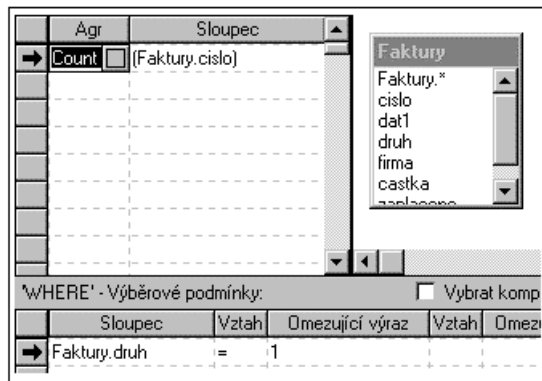
Cislo	Nazev	Castka
6	AFFEX, spol. s r.o.	1500.00
8	AFFEX, spol. s r.o.	2300.00
12	AFFEX, spol. s r.o.	544.00
21	AFFEX, spol. s r.o.	1210.00
22	AFFEX, spol. s r.o.	5000.00
1	Brněnská drátovna a.s.	1000.00
3	Brněnská drátovna a.s.	2656.00
9	Brněnská drátovna a.s.	125.70
18	Brněnská drátovna a.s.	1000.00
2	Kancelářské potřeby Praha	2500.00
5	Kancelářské potřeby Praha	5050.00
14	Kancelářské potřeby Praha	4000.00

Příklad 5

Zadání: Zjistěte, kolik faktur je došlých.

Popis řešení: Vložte tabulku FAKTURY. Do části SELECT vložte např. sloupec CISLO. Z combo boxu vlevo vyberte funkci COUNT. V části WHERE definujte omezující podmínku DRUH=1 (došlé faktury).

```
SQL: SELECT COUNT(Faktury.cislo)
      FROM Faktury
      WHERE (Faktury.druh=1)
```



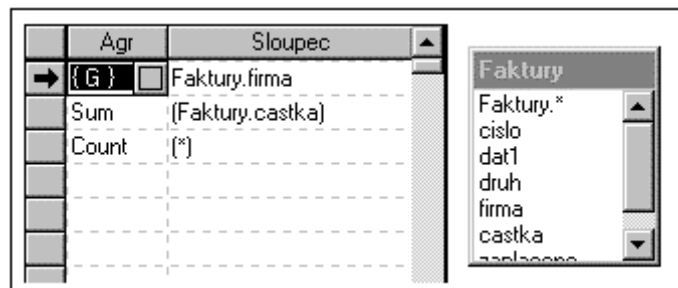
Poznámky: a) Agregáčnı́ funkce se počítá přes jedinou skupinu, a tou jsou všechny platné záznamy. Z toho plyne, že výsledkem musí být jediný řádek s tolika hodnotami, kolik je v části SELECT položek. Dotaz s agregáčními funkcemi není editovatelný.

b) Parametry funkce **COUNT**: Počítáte-li záznamy v dotazu, velice záleží na sloupci, který použijete jako parametr funkce **COUNT**. Výsledky se totiž mohou od sebe lišit. Funkce **COUNT** počítá pouze ty záznamy, které ve sloupci, jež je jejím parametrem, nemá hodnotu NULL. Chcete-li spočítat všechny záznamy, pak jako parametr funkce **COUNT** použijte hvězdičku nebo takový sloupec, u něhož máte zajištěno, že nenabývá hodnot NULL.

Příklad 6

Zadání: Zjistěte obraty jednotlivých firem a kolika fakturami toho dosáhli.

Popis řešení: Musíte vytvořit skupiny záznamů podle jednotlivých firem. Vložte tabulku FAKTURY. Skupinu vytvoříte tak, že sloupec FIRMA přenesete do části SELECT. Z comba vlevo vyberte {G}. Nyní doplňte funkci **SUM** pro sloupec CASTKA a funkci **COUNT** s parametrem hvězdička (viz poznámka k minulému příkladu). Výsledek seřídíte podle jmen firem.



```
SQL: SELECT Faktury.firma,SUM(Faktury.castka),COUNT(*)
      FROM Faktury
      GROUP BY Faktury.firma
      ORDER BY Faktury.firma
```

Poznámky: a) Odpověď bude obsahovat tolik záznamů, kolik je v tabulce rozdílných firem. Hodnoty odpovědi nebudou editovatelné.

b) Závorka u parametru agregační funkce (ve sloupci Sloupec se doplní při dalším otevření návrháře.

c) Kombinací tohoto příkladu s příkladem 4 (relační propojení) můžete do odpovědi dostat též název firmy místo jejího čísla.

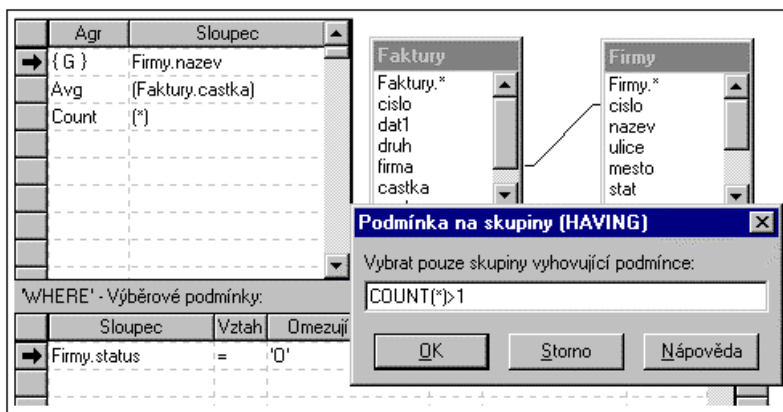
Příklad 7

Zadání: Vypište seznam odběratelů, průměrnou hodnotu jejich obratu, s vyloučením těch, kteří odebrali zboží pouze jednou.

Popis řešení: Nejdříve je třeba zajistit, aby v dotazu byly jak názvy firem, tak i obraty. Musíte tedy do návrháře vložit obě tabulky. Musíte je také propojit přes sloupce FIRMY.CISLO a FAKTURY.FIRMA. Dále je třeba vytvořit skupiny přes firmy. Aby se ve výsledku objevila jména firem, vložte do SELECTu sloupec NAZEV a skupiny definujte na něm. Doplněte funkci **AVG** na sloupec CASTKA a funkci **COUNT** s parametrem hvězdičkou. Abyste vyloučili dodavatele, v oblasti WHERE definujte podmínku STATUS='O'. Tím je splněna první část. Zbývá vyloučit ty odběratele, kteří odebrali zboží pouze jednou.



Pro zadání takovéto podmínky není již na ploše návrháře místo. Je nutné ji zapsat ručně do okna, které se otevře po stisku tlačítka na liště. Do editačního boxu napište podmínku ve tvaru `COUNT(*) > 1`.



```
SQL: SELECT Firmy.nazev,
        AVG(Faktury.castka), COUNT(*)
FROM Faktury, Firmy
WHERE Faktury.firma = Firmy.cislo
      AND (Firmy.status='O')
GROUP BY Firmy.nazev
HAVING COUNT(*)>1
```

Poznámky: a) Sloupec STATUS je typu Znak (Char) a proto se musí zapisovat v apostrofě ('') a nikoli v uvozovkách (") jako typ řetězec.

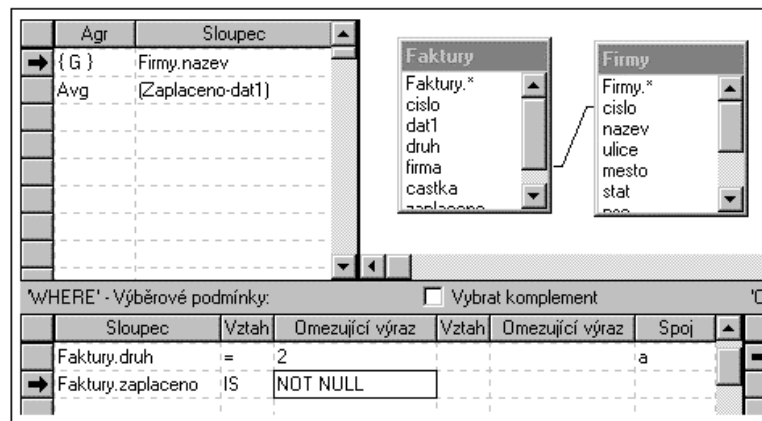
b) Všimněte si, čím se od sebe liší podmínky WHERE a HAVING. Podmínka WHERE se dá uplatnit ještě **před** sestavením skupin, zatímco podmínka HAVING se může vyhodnotit až **po** sestavení skupin.

Příklad 8

Zadání: Zjistěte průměrnou dobu zaplacení faktur odběrateli.

Popis řešení: Vložte si do návrháře obě tabulky. Abychom ve výsledku viděli název firmy, propojte obě tabulky jako v příkladu 4. Vytvořte skupiny podle sloupce NAZEV z tabulky FIRMY. Do pravé části dalšího řádku oblasti SELECT запиšte výraz, který chcete vypočítat: rozdíl dvou dat. Doplňte agregační funkci průměr (**AVG**).

V části WHERE definujte omezující podmínky: zajímají nás pouze faktury odeslané a jen ty, které už byly zaplacené. Že faktura již byla zaplacená se pozná z toho, že ve sloupci ZAPLACENO není prázdná hodnota (hodnota NULL).



```
SQL: SELECT nazev, Avg(zaplaceno-dat1)
      FROM Faktury, Firmy
      WHERE Faktury.firma= Firmy.cislo
            AND (druh=2 AND zaplaceno IS NOT NULL)
      GROUP BY Firmy.nazev
```

Poznámky: a) Všimněte si, že agregační funkce může mít za parametr i výraz.

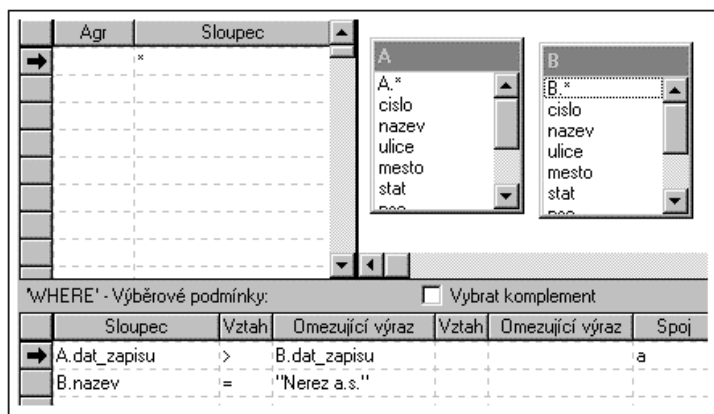
b) Výsledek agregační funkce bude vždy tentýž jako typ argumentu, proto je průměrný počet dnů uveden jako celé číslo.

Příklad 9

Zadání: Vypište ty firmy, které byly do databáze zavedeny až po firmě Nerez a.s.

Popis řešení: Vložte do návrháře **dvakrát** tabulku FIRMU. Návrhář automaticky pozmění jméno druhé tabulky na FIRMU0. Tato jména (pokud vám nevyhovují) je možné změnit pomocí příkazu **Alias** v menu **Tabulky**. V našem příkladu jsou nazvány A a B.

Nyní nelze použít relační propojení spojovací čarou mezi tabulkami, neboť vztah mezi nimi je nerovnost. Vytvořte proto podmínku pomocí drag & drop v oblasti WHERE. Ze seznamu A přeneste položku DAT_ZAPISU do pole **Sloupec** a ze seznamu B tentýž sloupec do pole **Omezující výraz**. Zvolte operátor "větší než" (>). Druhou podmínkou zvolte firmu z tabulky B.



```
SQL: SELECT *
      FROM Firmy A, Firmy B
      WHERE (A.dat_zapisu>B.dat_zapisu
            AND B.nazev="Nerez a.s.")
```

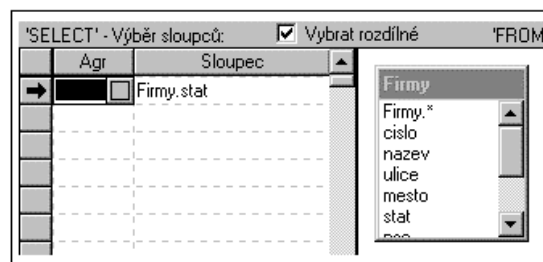

Poznámky: a) Návrhář musí rozlišovat mezi oběma tabulkami, proto si druhou nazval FIRMYY0 a uživatel přejmenoval obě tabulky na A a B.

b) Jak je vidět, vnitřní relační propojení (dokonce na nerovnost) můžete provádět i v rámci jedné tabulky. Tento způsob, anglicky nazývaný *Self-Join* je méně častý než propojení různých tabulek.

Příklad 10

Zadání: Zjistěte, z jakých států jsou vaši obchodní partneři. (Předpokládejme, že máme bohatší styky se světem než fiktivní firma v našem příkladu).

Popis řešení: Nejjednodušším řešením je vypsát si státy firem s vyloučením duplicit. Vložte tabulku FIRMYY a do SELECTu přeneste sloupec STAT. Nad oblastí SELECT se nalézá označovací čtverec **Vybrat rozdílné**, který zaškrtněte.



SQL: SELECT DISTINCT Firmy.stat
FROM Firmy

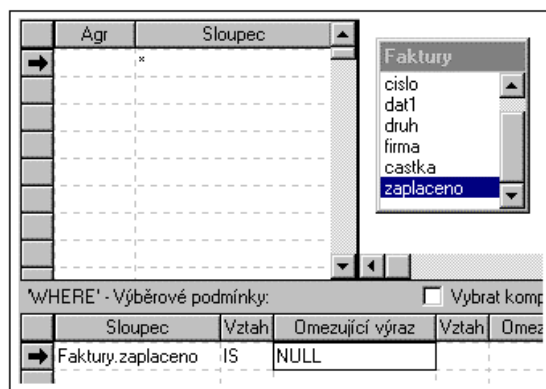
Poznámka: Klauzuli DISTINCT lze použít i na vybrané skupiny záznamů. Klauzule DISTINCT rozlišuje pouze mezi prvními osmi sloupci.

Příklad 11

Zadání: Vyberte ty faktury, které nejsou zaplacené

Popis řešení: Do části WHERE přeneste sloupec ZAPLACENO a vyberte z nabídky operátor IS. Do pole **Omezující výraz** doplňte konstantu NULL.

SQL: SELECT *
FROM Faktury
WHERE (Faktury.zaplaceno IS NULL)



Poznámka: a) Operátor IS NULL se používá právě v takovýchto příkladech: vybrat ty záznamy, v nichž některý sloupec je nebo není vyplněn.

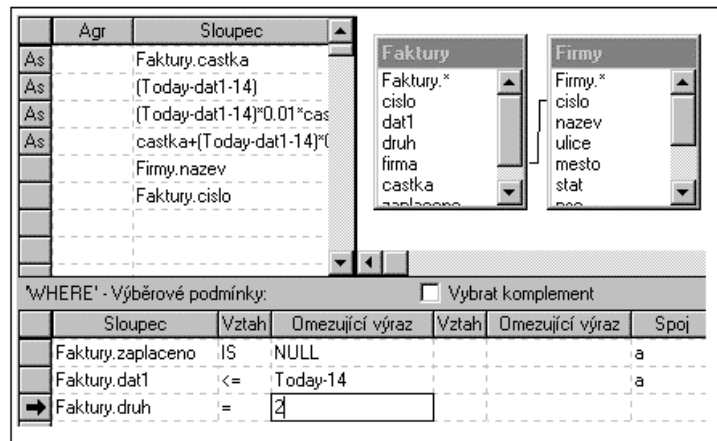
b) Pokud bychom chtěli tuto podmínku obrátit (negovat), lze použít dva ekvivalentní zápisy: WHERE (Faktury.zaplaceno IS NOT NULL) nebo WHERE NOT (Faktury.zaplaceno IS NULL).

Příklad 12

Zadání: Vytvořte dotaz, který zjistí nezaplacené faktury odběratelů a vypočítá penále k dnešnímu dni. Lhůta na zaplacení nechť je 14 dní, penále je 1% z ceny za každý den po lhůtě.

Popis řešení: Rozeberme si problém: nezaplacené faktury zjistíte podmínkou ZAPLACENO IS NULL; faktury po dni splatnosti vyberete podmínkou: dat1<=Today-14, kde **Today** je standardní funkce jazyka, která vrací dnešní datum; faktury odeslané podmínkou DRUH=2.

Pro odpověď budete potřebovat tyto výrazy: počet dní po splatnosti zjistíte výrazem Today-dat1-14; penále vypočítáte jako počet dní po splatnosti krát 1% z částky (výraz (Today-dat1-14)*0.01*castka) a konečně celkovou částku k zaplacení dostanete tak, že k původní ceně připočtete penále. Všechny tyto výrazy musíte zapsat do pravé části oblasti SELECT (pomožte si kopírováním stejných výrazů přes schránku). Jako v jiných příkladech vložte do návrháře obě tabulky a propojte je relačním vztahem. Přidejte ještě sloupec NAZEV a CISLO a původní částku (CASTKA) faktury, aby se šlo v odpovědi lépe orientovat.



SQL:

```

SELECT Faktury.castka,
       (Today-dat1-14),
       (Today-dat1-14)*0.01*castka,
       castka+(Today-dat1-14)*0.01*castka,
       Firmy.nazev,
       Faktury.cislo
FROM Faktury, Firmy
WHERE Faktury.firma=Firmy.cislo
      AND (Faktury.zaplaceno IS NULL AND
           Faktury.dat1<=Today-14 AND
           Faktury.druh=2)

```

Poznámky: a) Všimněte si, že odpověď na tento dotaz se liší den ode dne - jak narůstá penále.

b) Ve výrazu v části SELECT i v podmínce WHERE lze používat standardní funkce programovacího jazyka - popis viz Encyklopedie funkcí.

c) Do záhlaví sloupců v odpovědi **WinBase602** dosazuje nic moc neříkající `EXPRn`. Navíc se sloupce mohou zobrazovat v pořadí jiném, než bychom chtěli, případně v jiném stylu zobrazení. Proto běžný způsob pro prezentaci dat z odpovědi na dotaz není otevřít dotaz v jeho standardní podobě, nýbrž vytvořit si vlastní formulář do těchto dat. Velice jednoduchý, za několik sekund vytvořený standardní formulář může vypadat například takto:

Vlastní pohled do dotazu Příkl_12						
Fa. čísl	Firma	Původní částka	Počet_dní	Penále	Celkem	
21	AFFEX, spol. s r.o.	1210.00	152	1839,200	3049,200	
22	AFFEX, spol. s r.o.	5000.00	38	1900,000	6900,000	
23	Kancelářské potřeby Praha	6541.00	7	457,870	6998,870	

Někdy postačí si upravit dotaz. Pomocí příkazu **Jméno sloupce** je možné nahradit názvy EXPRn, SUMn apod. svými vlastními názvy výrazů - PENÁLE apod. Tyto názvy se objeví i v záhlaví standardního formuláře při otevření dotazu. Začátek textové definice by pak vypadal např. takto:

```
SELECT Faktury.castka AS ČÁSTKA,
      (Today-dat1-14) AS POČET_DNÍ,
      (Today-dat1-14)*0.01*castka AS PENÁLE,
      castka+(Today-dat1-14)*0.01*castka AS CELKEM,
      Firmy.nazev AS NÁZEV,
      Faktury.cislo AS FA_ČÍSLO
FROM Faktury, Firmy
...
```

d) I tak mocný prostředek jako SQL má svá omezení: pokud byste chtěli tento dotaz rozšířit o součet penále za jednotlivé firmy, zjistíte, že to nejde. Musíte si na to vytvořit dotaz další, jenž bude začínat:

```
SELECT SUM(castka+(Today-dat1-14)*0.01*castka), FIRMY.nazev
FROM Faktury, Firmy
WHERE ...
GROUP BY Firmy.nazev
```

Příklad 13

Zadání: Zjistěte obraty a počty faktur firem po měsících.

Popis řešení: V dotazu se musí vytvořit skupiny záznamů stejné firmy a v rámci této skupiny ještě podrobnější dělení na jednotlivé měsíce.

Vložte do návrháře obě tabulky a propojte je. Přeneste sloupec NAZEV do oblasti SELECT a definujte skupiny. Z hodnoty sloupce DAT1 nás zajímá pouze měsíc - vytvořte proto výraz `Month(dat1)` na dalším řádku v oblasti SELECT a definujte skupiny. Nakonec vytvořte agregační funkce `SUM(castka)` a `COUNT(cislo)`.

	Agr	Atribut
→	{ G }	Firmy.nazev
As	{ G }	Month(dat1)
As	Count	(Faktury.cislo)
As	Sum	(Faktury.castka)

Faktury.*		Firmy.*	
cislo		cislo	
dat1		nazev	
druh		ulice	
firma		mesto	
castka		stat	

```
SQL: SELECT Firmy.nazev, Month(dat1) AS Měsíc,
        COUNT(Faktury.cislo) AS Počet,
        SUM(Faktury.castka) AS Suma_Kč
FROM Faktury, Firmy
WHERE Faktury.firma= Firmy.cislo
GROUP BY Firmy.nazev,Month(dat1)
```

Poznámka: WinBase602 umožňuje, aby se skupiny vytvářely podle hodnoty výrazu, nejenom podle hodnoty sloupce. Je to rozšíření oproti standardu SQL, nicméně se domníváme, že to uživatelé ocení.

Odpověď na tento dotaz vypadá takto:

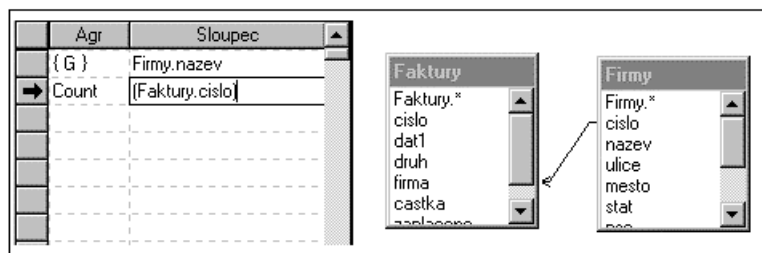
	Nazev	Měsíc	Počet	Suma_kč
→	Brněnská drátovna a.s.	1	2	3656.00
	Brněnská drátovna a.s.	2	1	125.70
	Brněnská drátovna a.s.	6	1	1000.00
	Kancelářské potřeby Praha	1	2	7550.00
	Kancelářské potřeby Praha	4	1	4000.00
	Kancelářské potřeby Praha	12	2	13041.00
	AFFEX, spol. s r.o.	2	2	3800.00
	AFFEX, spol. s r.o.	4	1	544.00
	AFFEX, spol. s r.o.	7	1	1210.00

Příklad 14

Zadání: Vypište seznam všech firem a počet jejich faktur, včetně těch firem, které nemají ještě ani jednu fakturu.

Popis řešení: Úloha musí být řešena pomocí vnějšího propojení tabulek. Kdybychom použili vnitřní propojení, do odpovědi by se nevybraly ty firmy, které nemají dosud ani jednu fakturu, neboli v tabulce FAKTURY není jejich číslo ani v jedné hodnotě sloupce FIRMA.

Vložte do návrháře obě tabulky. Vytvořte myší "normální" propojení. Nyní stiskněte pravé tlačítko myši na spojovací čáře. Objeví se dialog, z něhož vyberte druhou nebo třetí možnost podle toho, v jakém pořadí jste tabulky vkládali. Po uzavření okna musí šipka vést od tabulky FIRMA k tabulce FAKTURY. Už známým způsobem vytvořte skupiny podle názvu firmy a funkci pro zjištění počtu faktur (COUNT(Faktury.Cislo)).



```
SQL: SELECT Firmy.nazev, COUNT(Faktury.cislo)
FROM Firma LEFT OUTER JOIN Faktury
ON (Firma.cislo=Faktury.firma)
GROUP BY Firmy.nazev
```

Poznámka: Rozdíl mezi oběma druhy propojení si uvědomíte nejlépe tím, že propojení v tomto dotazu pozměníte na propojení vnitřní (pravým tlačítkem myši klikněte na čáře ...) a výsledky si porovnáte.

Pohled do dotazu Příkl_14		
Nazev	Count1	
Brněnská drátovna a.s.	4	
Kancelářské potřeby Praha	5	
AFFEX, spol. s r.o.	5	
Spojené šroubárny Kolín	2	
Nerez a.s.	3	
SuperOcel s.r.o.	1	
Papírnictví Horváth	3	
Velkoobchod Extaze	1	
Potřeby pro hospodyňky a.s.	0	
Hutě Svidník	0	

Pohled do dotazu Dupl_14		
Nazev	Count1	
Brněnská drátovna a.s.	4	
Kancelářské potřeby Praha	5	
AFFEX, spol. s r.o.	5	
Spojené šroubárny Kolín	2	
Nerez a.s.	3	
SuperOcel s.r.o.	1	
Papírnictví Horváth	3	
Velkoobchod Extaze	1	

Vnější spojení

Vnitřní spojení

Příklad 15

Zadání: Vyberte firmy, které byly do databáze zapsány v měsících leden až březen včetně.

Popis řešení: Nejjednodušší bude použit množinový operátor IN ve tvaru IN (1,2,3), kde v závorkách je výčet hodnot (čísla měsíců). Číslo měsíce z hodnoty sloupce DAT_ZAPISU lze získat funkcí **Month**.



SQL:

```
SELECT Firmy.cislo,Firmy.nazev,Firmy.pracovnik,
       Firmy.dat_zapisu
FROM Firmy
WHERE (Month(dat_zapisu) IN (1,2,3))
```

Poznámky: a) stejného cíle by šlo dosáhnout i pomocí jiných zápisů podmínky, např.

```
WHERE (Month(dat_zapisu) BETWEEN 1 AND 3)
```

nebo

```
WHERE (Month(dat_zapisu) >=1 AND Month(dat_zapisu) <=3)
```

b) operátor IN má i druhé použití – hodnota se pomocí něho porovnává s výsledkem subdotazu. Podrobnosti o subdotazech naleznete v elektronické nápovědě k SQL.

Příklad 16

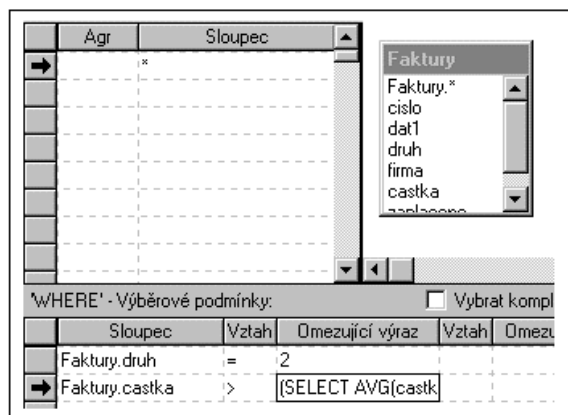
Zadání: Vyberte faktury jednoho typu za více než je průměrná hodnota faktury tohoto typu.

Popis řešení: Rozdělíme si úlohu na dva kroky. Nejprve spočteme průměrnou hodnotu faktury:

```
SELECT AVG(castka)
FROM Faktury
WHERE druh=2
```

Tento dotaz by bylo výhodné umístit jako součást podmínky v sekci WHERE. **Win-Base602** takovéto použití dotazu (neboli subdotazu) v dotazu umožňuje.

Dotazy se subdotazy je teoreticky možné vytvářet v návrháři (subdotaz se zapíše do vstupního pole pro výraz), ale pro většinu uživatelů bude rychlejší zapisovat text dotazu přímo v syntaxi SQL v editoru.



SQL:

```
SELECT *
FROM Faktury
WHERE (Faktury.druh=2 AND
      Faktury.castka > (SELECT AVG(castka)
                        FROM Faktury
                        WHERE druh=2))
```

Poznámky: a) Subdotaz se vždy ohraničen kulatými závorkami.

b) Dotaz je editovatelný, protože agregační funkce je pouze v subdotazu.

c) V podobném dotazu není možné použít klauzuli HAVING (jak to umožňovala **Win-Base602** do verze 5).

Příklad 17

Zadání: Vyberte faktury firem z Prahy

Popis řešení: Zvolíme řešení se subdotazem. Subdotaz vybere firmy z Prahy:

```
SELECT cislo
FROM Firmy
WHERE mesto LIKE "Praha%"
```

Dotaz na tabulku Faktury použije subdotaz k vybrání těch faktur, jejichž číslo firmy se nachází v subdotazu. Doslovný přepis pak vypadá takto:


```
SQL: SELECT cislo, firma
      FROM Faktury
      WHERE firma IN (SELECT cislo
                     FROM Firmy
                     WHERE mesto LIKE "Praha%")
```

Poznámka: a) Predikát IN je zde (na rozdíl od příkladu 15) použit ve významu zjištění výskytu v odpovědi na subdotaz. Subdotaz musí obsahovat pouze jeden sloupec.

b) Stejného efektu by se dosáhlo zápisem `firma=ANY (subdotaz)místo firma IN (subdotaz)`.

c) Predikát LIKE slouží k porovnávání hodnoty sloupce MESTO se vzorem zadaným výrazem "Praha%", což je SQL zápis ve **WinBase602** používaného operátoru .= (řetězec začíná ...). Predikát LIKE je mnohem obecnější – viz nápověda k SQL.

d) Již z dřívějších příkladů byste měli umět vytvořit tento dotaz i bez subdotazu:

```
SELECT Faktury.cislo, Faktury.firma
FROM Faktury, Firmy
WHERE Firmy.cislo=Faktury.firma
AND (Firmy.mesto.="Praha")
```

Příklad 18

Zadání: Vyberte firmy, které mají faktury za více než 10000,- Kč.

Popis řešení: Zvolíme opět řešení se subdotazem. Subdotaz pro každou firmu spočte její faktury. Pak už stačí subdotaz porovnat s konstantou 10000.

```
SQL: SELECT A.nazev
      FROM Firmy A
      WHERE 10000 < (SELECT SUM(castka)
                    FROM Faktury B
                    WHERE A.cislo=B.firma)
```

Poznámky: a) Zde použitý subdotaz se odlišuje od subdotazů v předešlých příkladech – pro každý záznam tabulky Firmy má subdotaz jinou hodnotu.

b) Pokud bychom chtěli vypsat tytéž firmy a jejich obraty, musíme použít variantu dotazu bez subdotazu:

```
SELECT A.nazev, SUM(B.castka)
FROM Firmy A, Faktury B
WHERE A.cislo=B.firma
GROUP BY B.firma
HAVING 10000 < SUM(B.castka)
```

c) všimněte si, že dotaz se subdotazem je editovatelný, zatímco dotaz s propojením nikoliv.

Příklad 19

Zadání: Vypište názvy firem, které mají faktury za více než 5000,- Kč.

Popis řešení: V tomto příkladu ukážeme použití subdotazu v části SELECT. Subdotaz pro každé číslo faktury vrátí jméno firmy. Toto jméno lze použít ve výčtu sloupců dotazu.

```
SQL:  SELECT DISTINCT (SELECT nazev
                        FROM Firmy
                        WHERE Firmy.cislo=Faktury.firma) AS Nazev
      FROM Faktury
      WHERE castka > 5000
```

Poznámky: a) Sloupec Nazev odpovědi není samozřejmě editovatelný.

b) Stejného výsledku (ale s možností editace odpovědi) lze dosáhnout takto:

```
SELECT Firmy.nazev
FROM Firmy
WHERE Firmy.cislo =ANY (SELECT Faktury.firma
                       FROM Faktury
                       WHERE castka > 5000
                       AND Faktury.firma=Firmy.cislo)
```

c) Stejného výsledku lze dosáhnout i bez subdotazů, pomocí JOINu, např. takto:

```
SELECT DISTINCT Firmy.nazev
FROM Faktury, Firmy
WHERE (castka > 5000)
      AND (Firmy.cislo=Faktury.firma)
```

Příklad 20

Zadání: Mějme tabulky FAKTURY a FAKT_2 a v nich data z let 1998 a 1997. Vypište faktury odběratelů z dvou měsíců na přelomu roku 1997.

Popis řešení: Úloha vyžaduje spojení dat ze dvou tabulek, nikoli však na základě relačního vztahu, nýbrž pouhým přidáním záznamů za sebe. Sjednocení více tabulek sice WinBase602 pomocí SQL umí vytvořit, nicméně tato operace není ani obvyklá ani častá. Není proto ani implementována do interaktivního návrháře. Pokud chcete tabulky sjednotit, musíte zapsat příkaz SQL přímo v jeho textové podobě. Obě části dotazu spojte klíčovým slovem UNION a výsledek můžete seřadit sestupně podle data.

```
SQL:  SELECT *
      FROM Fakt_2
      WHERE dat1 > 30.11.1997
      UNION
```

```
SELECT *
FROM Faktury
WHERE dat1 < 1.2.1998
ORDER BY dat1 DESC
```

Poznámky: a) S takto definovaným dotazem pracujete stejně jako s kterýmkoliv jiným vytvořeným v návrháři s jedinou výjimkou: při pokusu modifikovat dotaz v návrháři dostanete upozornění, že dotaz je příliš složitý pro úpravy v návrháři.

b) Obě části sjednocení musí mít v části SELECT položky stejných typů.

c) Seřídění se týká obou částí dotazu, třídít jednotlivé části před sjednocením nelze.

d) Pokud jsou v textové podobě definice na prvním řádku čísla v komentářových závorkách, značí rozmístění seznamů tabulek v návrháři a pro tento konkrétní příklad, kdy se návrhář již nebude dále používat, není důvod proč je nasmazat.

Příklad 21

Zadání: Vyberte nejvyšší fakturu každého typu.

Popis řešení: Vytvoříme dva dotazy pro nejvyšší fakturu obou typů (s využitím subdotazů). Pak výsledky sjednotíme pomocí klauzule UNION.

```
SQL: SELECT "došlé", castka
      FROM Faktury
      WHERE castka = (SELECT MAX(castka)
                    FROM Faktury
                    WHERE druh=1)

      UNION

      SELECT "odeslané", castka
      FROM Faktury
      WHERE castka = (SELECT MAX(castka)
                    FROM Faktury
                    WHERE druh=2)
```

Poznámky: a) Abychom ve výsledku odlišili oba dotazy, je první sloupec konstantní. Sloupce v obou dotazech musí mít stejnou strukturu, proto je nutné, aby řetězcové konstanty měly stejný počet znaků.

Příklad 22

Zadání: Vyberte ty firmy, které měly faktury za více než 10000,-Kč letos (tabulka FAKTURY) i vloni (tabulka FAKT_2).

Popis řešení: Vytvoříme dva dotazy podle příkladu 18. Pak provedeme množinový průnik odpovědí - pomocí klauzule INTERSECT.

```
SQL:  SELECT A.nazev
      FROM Firmy A, Faktury B
      WHERE A.cislo=B.firma
      GROUP BY B.firma
      HAVING 10000 < SUM(B.castka)

      INTERSECT CORRESPONDING BY (nazev)

      SELECT A.nazev
      FROM Firmy A, Fakt_2 B
      WHERE A.cislo=B.firma
      GROUP BY B.firma
      HAVING 10000 < SUM(B.castka)
```

Poznámky: a) Malou změnou lze vytvořit dotaz na firmy, které tohoto obrátu dosáhly letos ale ne vloni. Stačí použít množinový rozdíl EXCEPT.

```
SQL:  SELECT A.nazev
      FROM Firmy A, Faktury B
      WHERE A.cislo=B.firma
      GROUP BY B.firma
      HAVING 10000 < SUM(B.castka)

      EXCEPT CORRESPONDING BY (nazev)

      SELECT A.nazev
      FROM Firmy A, Fakt_2 B
      WHERE A.cislo=B.firma
      GROUP BY B.firma
      HAVING 10000 < SUM(B.castka)
```

Přenosy dat z odpovědi na dotaz

Konverze dotazu na tabulku

Z obsahu odpovědi na dotaz lze vytvořit novou tabulku.

Postupujte takto:

1. na řídicím panelu označte zvolený dotaz;
2. proveďte akci **Export dat**; tím otevřete návrhář přenosu se zvoleným dotazem přednastaveným jako zdroj dat pro přenos;
3. jako cíl přenosu zvolte **Tabulka ve WinBase602** a do editační složky vpravo napište jméno nové tabulky, která má přenosem vzniknout;

4. tlačítkem **Vytvořit popis** vytvořte a poté případně modifikujte popis přenosu;
5. tlačítkem **Přenést data** proveďte operaci.

Vytvoření tabulky z odpovědi na dotaz je účelné tehdy, pokud chcete zachovat obsah odpovědi a uchránit jej před vlivem změn, které v budoucnu nastanou v tabulkách, jichž se dotaz týká. Tabulka vytvořená z odpovědi zabere nové místo v databázi. Sama odpověď místo trvale nezabírala.

Export dat z odpovědi

Obsah odpovědi na dotaz můžete exportovat do souboru ve zvoleném formátu.

Postupujte takto:

1. na řídicím panelu označte zvolený dotaz;
2. proveďte akci **Export dat**; tím otevřete návrhář přenosu se zvoleným dotazem přednastaveným jako zdroj dat pro přenos;
3. vyberte formát a jméno cílového souboru;
4. tlačítkem **Vytvořit popis** vytvořte a poté případně modifikujte popis přenosu;
5. tlačítkem **Přenést data** proveďte operaci.

