

# Vytváření aplikace

Tato kapitola zastřešuje poznatky obsažené v kapitolách o návrhu tabulek, formulářů, dotazů a menu. Popisuje, jak jednotlivé navržené komponenty poskládat do funkčního celku a jak dosáhnout jejich inteligentního chování, aniž by bylo nutno cokoli (nebo mnoho) programovat. Popisuje také, jak do formulářů přidat mnohé nové užitečné rysy.

Aplikace vytvořená pomocí metod popsanych v této kapitole má jako *startovní objekt* určité menu nebo formulář - to znamená, že se při spuštění aplikace na obrazovce zobrazí menu nebo otevře formulář. Pomocí příkazů v menu nebo ovládacích prvků ve formuláři pak uživatel může vyvolávat jednotlivé funkce aplikace.

Na tuto kapitolu navazuje (náročnější) kapitola o programování prezentační vrstvy.

## Terminologická poznámka

Rozlišujte pozorně dva významy slova **synchronizace** ve **WinBase602**:

- synchronizace mezi dvěma formuláři (např. mezi *superformulářem* a *subformulářem*) popisuje způsob, jak se při listování v jednom formuláři mění obsah nebo vybraný záznam v formuláři druhém;
- synchronizace mezi obsahem *databáze* a obsahem *formuláře* na obrazovce popisuje způsob, jak se změny provedené uživatelem na obrazovce přenášejí do databáze; používají se pojmy synchronizace *složková* a *záznamová*.

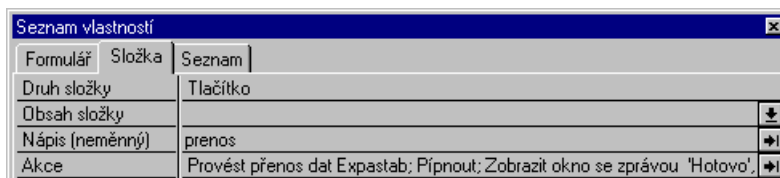
## Prostředí návrháře akcí

Návrhář akcí dovoluje snadno a obvykle i bez programování definovat akce, které se mají spustit například při stisku tlačítka ve formuláři nebo na liště, při provedení příkazu z menu a v mnoha jiných situacích.

Vstup do  
návrháře

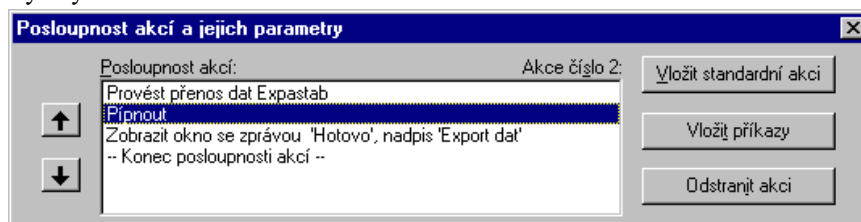
Do návrháře akcí vstoupíte z návrháře formulářů nebo menu z místa, kde se popisuje reakce na příslušnou událost. Například z panelu vlastností tlačítka vloženého do formuláře otevřete návrhář akcí stiskem malého tlačítka s šipkou vpravo. Při navrhování menu nebo ovládací lišty se otevírá stiskem tlačítka **Nabídka** nebo **Nabídka akcí**.

### Vyvolání ze seznamu vlastností složky



## Seznam akcí v návrháři

Definování akce nebo posloupnosti akcí se provádí ve speciálním okně nadepsaném **Akce a jejich parametry**. Okno je poměrně rozsáhlé, proto zde na obrázcích vidíte pouze jeho výřezy.



V horní části okna je seznam elementárních akcí v tom pořadí, v němž se mají provést. Novou akci vložíte do seznamu stiskem tlačítka **Vložit standardní akci**. Akce se vloží vždy před tu akci v seznamu, která je právě vybraná. Chcete-li tedy připojit další akci na konec seznamu, vyberte předem nápis - **Konec posloupnosti akcí** -.

Nově vložená akce je prázdná a je označena nápisem **Bez akce** do doby, dokud nedefinujete její obsah.

Tlačítka s šipkami můžete měnit pořadí akcí v posloupnosti, tlačítkem **Odstranit akci** lze vybranou akci z posloupnosti vymazat.

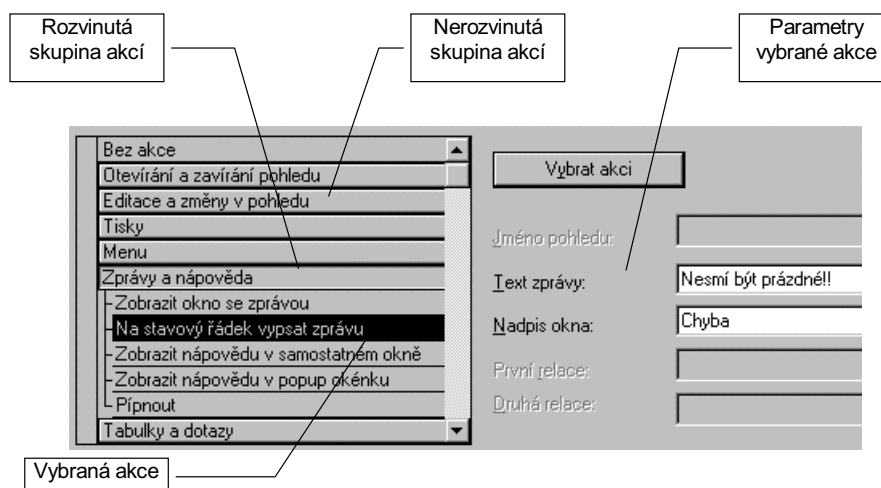
## Volba akce a zadání jejích parametrů

Ve spodní části okna návrháře akcí se volí akce odpovídající *vybrané* položce v seznamu akcí a zadávají se její parametry.

Vlevo je seznam možných akcí uspořádaný do skupin. Skupina akcí je znázorněna tlačítkem. Stisknuté tlačítko značí, že skupina je "rozbalená", tedy jsou vidět akce, které do ní patří.

Mezi akcemi je možné přecházet pomocí kurzorových šipek nebo pomocí myši. Chcete-li nějakou akci vybrat, musíte na ní buď poklepat nebo ji označit a stisknout tlačítko **Vybrat akci**.

## Nabídka akcí



Většina akcí vyžaduje doplnění informací - například pro otevření formuláře musíte zadat jeho jméno, pro vypsání zprávy její text apod. K zadání parametrů slouží comba a editační pole pod seznamem akcí. Jejich názvy a aktivita se mění podle druhu akce. Akci nelze uložit, dokud nevyplníte její parametry.

## Funkce jednotlivých akcí

Standardní akce jsou popsány v různých částech tohoto manuálu podle toho, k čemu se vztahují. Na tomto místě popíšeme pouze akce příliš nesouvisející s jinými kapitolami.

**Bez akce** - prázdná, dosud nedefinovaná akce.

**Zobrazit okno se zprávou** - otevře modální okno se zadanými informacemi.

**Na stavový řádek vypsát zprávu** - vypíše textovou informaci na stavový řádek **Win-Base602**.

**Exportovat/Importovat data z tabulky/dotazu** - umožní provést export/import dat ve vnitřním formátu a kódování **Windows**.

**Uvolnit záznamy v tabulce** - provede uvolnění smazaných záznamů.

**Uspořádat okna do mozaiky** - rozmístí okna na obrazovce tak, aby rovnoměrně vyplnila celý prostor.

**Uspořádat okna do kaskády** - rozmístí okna na obrazovce tak, aby bylo z každého vidět alespoň nadpis.

**Zaslat programu zprávu** - zpráva, kterou program přijme voláním funkce `Get_ext_message`;

**Spustit soubor dle registrace** - spustí aplikaci, která je ve **Windows** registrována pro soubor uvedený jako parametr akce, a předá ji tento soubor jako parametr. Aplikace je určena příponou souboru. Pokud je nad tímto type souboru definováno více operací, lze zadat jméno té, která se má provést. Není-li jméno vyplněno, provede se operace

*Open.* Například pro soubor s příponou BMP se standardně spouští program Paint a jsou k dispozici dvě operace: *Open* a *Print*.

## Zápis příkazů do návrháře

Do posloupnosti akcí lze zařadit také příkazy zapsané podle syntaxe vnitřního programovacího jazyka **WinBase602**. Nový příkaz se do posloupnosti vloží stiskem tlačítka **Vložit příkazy**. Tím se otevře editační pole, do něž lze příkaz zapsat.

V akci lze zadat libovolně složitý příkaz, ale nejčastěji se používají volání procedur, metod a jednoduchá přiřazení. Pokud potřebujete zadat složitější programové konstrukce, je výhodnější je napsat do programu jako samostatnou proceduru a v akci ji pouze zavolat. Příkazy z programu se totiž dají, na rozdíl od příkazů skrytých v akci, pohodlně ladit.

Při zápisu volání podprogramů a metod a odkazů na vlastnosti a proměnné lze využít návrhář faktoru. Ten se zavolá stiskem tlačítka se třemi tečkami a jím navržený faktor se vloží na pozici kurzoru do textu editovaného příkazu.

## Prostředí návrháře faktoru

Návrhář faktoru je jednoduchý nástroj, který dovoluje interaktivním způsobem vybrat a vložit do textu jméno proměnné nebo konstanty z projektu, volání procedury, funkce nebo metody případně přístup k vlastnosti formuláře nebo jeho složky.

### Omezení nabídky

Nabídka, která se objeví v návrhář faktoru, může být zúžena na základě kontextu, v němž byl návrhář otevřen:

- kontext může vyžadovat, aby bylo možno vybrat pouze proměnnou, například pokud se navrhuje parametr podprogramu předávaný referencí;
- kontext může vyžadovat hodnotu určitého typu.

Je-li návrhář zavolán s omezením typu, lze v něm toto omezení vypnout pomocí označovacího čtverce **Bez omezení typu**.

### Stránky

Na prvních dvou stránkách návrháře lze ze seznamu vybrat proměnné a konstanty deklarované v projektu.

Na stránce podprogramy se nabízí seznam podprogramů z projektu. Po volbě podprogramu lze zadat hodnoty jeho parametrů - buď v editačním poli nebo pomocí rekurzivně zvolaného návrháře.

## Stránka Metody a vlastnosti

Stránka **Metody a vlastnosti** vytváří přístup k vlastnosti nebo volání metody formuláře nebo jeho složky. Na stránce je třeba vyplnit několik údajů po řadě shora dolů:

### Metody a vlastnosti

The screenshot shows a dialog box titled "Nabídka proměnných, podprogramů, metod" with a tab labeled "Metody a vlastnosti". The dialog contains the following fields and options:

- 1. Formulář: - Lokální formulář THISFORM -
- 2. Subformulář: - Hlavní formulář -
- 3. Složka formuláře: TEXT\_POPIS
- 4. Metoda nebo vlastnost: WRITEFILE
- Typ hodnoty: Ano-Ne (Boolean)
- Parametry:
  - FileName: c:\vystup\popis.txt
  - Append: FALSE
- ☒ Bez omezení typu

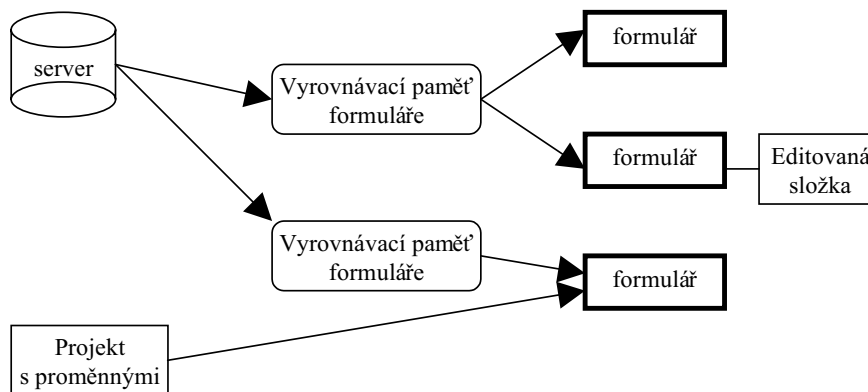
Buttons at the bottom: OK, Cancel, Apply, Help.

1. **Formulář:** lze zvolit buď některý formulář deklarovaný v projektu nebo některý formulář existující v aplikaci nebo **THISFORM** - ten formulář, v jehož kontextu faktor se bude vyhodnocovat.
2. **Subformulář:** dovoluje přejít od formuláře specifikovaného v předchozím bodu k jeho subformuláři nebo naopak k nadřazenému formuláři. Z nabídky lze vybrat **Hlavní formulář** (nepřechází se nikam), **Nadřazený formulář** (přechází se výše) nebo jméno některého subformuláře.
3. **Složka formuláře:** určuje, zda se má použít metoda či vlastnost formuláře nebo některého jeho složky. Lze vybrat **Formulář jako celek** nebo jméno složky.
4. **Metoda nebo vlastnost:** jméno metody nebo vlastnosti formuláře nebo jeho zvolené složky.
5. **Parametry:** Pokud jste zvolili metodu s parametry, doplníte hodnoty parametrů.

## Čtení dat do formuláře a ukládání změn

Tato sekce má poněkud techničtější charakter a při prvním čtení ji můžete s klidem přeskóčit. Její pochopení je žádoucí pro vývoj těch aplikací, v nichž bude na obrazovce současně otevřeno více formulářů do stejných dat.

Trasa dat se dá znázornit takto:



### Cesta dat z databáze do formuláře

Když otevřete formulář vedoucí do databáze, ten si buď vytvoří vyrovnávací paměť pro přístup k datům serveru, anebo (přejete-li si to) použije vyrovnávací paměť jiného formuláře. Do vyrovnávací paměti se data dostávají z databázového serveru prostřednictvím kurzoru (odpovědi na dotaz) nebo přímým čtením obsahu některé tabulky. Mimo to mohou být ve formuláři také data z proměnných právě otevřeného projektu (používáte-li vnitřní programovací jazyk).

### Cesta změn z formuláře do databáze

Při přepsání dat ve formuláři se nový obsah editované složky zapíše do vyrovnávací paměti po ukončení editace (ve standardním formuláři) nebo po opuštění složky (v obecném formuláři). Ve formuláři se překreslí ty složky, z nichž je nastavena vlastnost **Obnovit po změně v záznamu**. Zároveň se překreslí všechny formuláře sdílející stejnou vyrovnávací paměť.

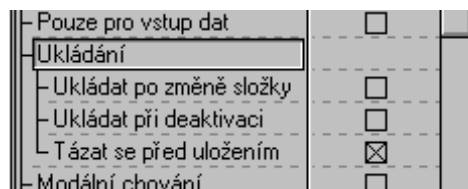
Změněná data se přenesou z vyrovnávací paměti do databáze při opuštění záznamu nebo při provedení akce **Uložit změny v formuláři** (nebo funkce **Commit\_view** nebo metody

`Commit_view`). Okamžitý přenos změněných dat do databáze lze zajistit nastavením příznaku **Ukládat po změně složky**.

Vlastnosti  
ovlivňující zápis  
změn

Je-li nastavena vlastnost **Ukládat po změně složky**, zápis do databáze proběhne po každém přepsání složky formuláře. Tento způsob zápisu nazýváme *složková synchronizace*. Není-li vlastnost nastavena, používá se tzv. *záznamová synchronizace*, kdy zápis do databáze proběhne až při opuštění editovaného záznamu.

Volba způsobu  
ukládání změn



Při záznamové synchronizaci je možno dále stanovit, má-li se editovaný záznam uložit při přechodu aktivity na jiné okno. Pokud ano, nastavte vlastnost **Ukládat při deaktivaci**.

Při zápisu změn v záznamu do databáze je možno vyžadovat dodatečné potvrzení - vlastnosti **Tázat se před uložením**. Je-li zapnuta, pak uživatel bude mít po přepsání každého záznamu možnost buď potvrdit nebo odvolat provedené změny, anebo pokračovat v editaci.

Odvolání změn

Změny ve vyrovnávací paměti lze odvolat bez zápisu do databáze buď stiskem kombinace kláves **Shift+Esc** nebo akci **Odvolat změny ve formuláři** (nebo metodou `Roll_back_view` nebo funkcí `Roll_back_view`).

Zavření  
formuláře  
bez uložení

Při zavření formuláře se změny provedené ve složkách přenesou do vyrovnávací paměti. Pokud s touto vyrovnávací pamětí nepracuje žádný jiný formulář, zaznamenají se do databáze, v opačném případě zůstanou ve vyrovnávací paměti a o jejich případné zaznamenání se musí postarat formulář, který tuto paměť také používá.

## Aktualizace formuláře

Potřebujete-li aktualizovat obsah formuláře na obrazovce, musíte uvážit rozsah akcí, které je nezbytné provést:

- pokud pouze chcete zobrazit nový obsah vyrovnávací paměti nebo proměnných projektu, pak provedete akci **Překreslit formulář** nebo zavoláte proceduru nebo metodu `Reset_view` s posledním parametrem `RESET_CONTROLS`;
- pokud chcete naplnit vyrovnávací paměť novými daty z databáze a překreslit formulář, pak provedete akci **Načíst data a překreslit formulář** nebo zavoláte proceduru nebo metodu `Reset_view` s posledním parametrem `RESET_CACHE` nebo `RESET_DELETIONS`;
- pokud chcete aktualizovat množinu záznamů vybraných do kurzoru, naplnit vyrovnávací paměť novými daty a překreslit formulář, pak provedete akci **Přebudovat**

**kurzor a překreslit formulář** nebo zavoláte proceduru nebo metodu **Reset\_view** s posledním parametrem `RESET_CURSOR`.

V druhém a třetím případě z vyrovnávací paměti zmizí změny, které dosud nebyly zaznamenány do databáze.

## Vkládání záznamů

### Jak umožnit vkládání?

Existují tři cesty, jak bez podpory programu umožnit uživateli vkládat záznamy do formuláře:

1. Vybavit formulář trvalým fiktivním záznamem.
2. Požadovat, aby uživatel, který chce vložit nový záznam, vytvořil fiktivní záznam stiskem klávesy **Insert**.
3. Vytvořit zvláštní formulář sloužící pouze pro vstup dat. Tento formulář obsahuje pouze fiktivní záznam.

### Alternativy

Alternativou ke stisku klávesy **Insert** (ve všech jejich níže popisovaných funkcích) je provedení metody **Insert** nebo provedení akce **Vložit záznam do běžného formuláře**. Díky tomu lze do formuláře, do menu nebo na lištu vložit tlačítko řídící vkládání záznamů nebo svěřit tuto úlohu programu.

### Formulář bez možnosti vkládat nové záznamy

Při vytváření nového formuláře pomocí průvodce lze znemožnit vkládání nových záznamů tím, že na čtvrté stránce průvodce zrušíte zatržení čtverce **Vkládání nových záznamů**.

V hotovém formuláři se vkládání záznamů zakáže tak, že se ve skupině **Chování formuláře** zatrhnou vlastnosti **Bez fiktivního** a **Zákaz vkládání Insert**.

### Formulář s trvale zobrazeným fiktivním záznamem

Toto je standardní chování formuláře. Poté, co do něj uživatel zapíše hodnoty do fiktivního záznamu, záznam se vloží a v formuláři se objeví nový fiktivní záznam.

Okamžik, v němž se nový záznam zapíše do databáze, závisí na nastavení příznaků ve skupině **Ukládání**. Záznam se může vložit buď hned po zapsání hodnoty první složky, nebo při deaktivaci formuláře, anebo až při opouštění fiktivního záznamu.

Chování prvních dvou způsobů závisí na tom, zda se změny v formuláři promítají do databáze už po změně složky nebo až po opuštění záznamu - nový záznam se do databáze skutečně vloží tehdy, když se do ní mají zapsat změny provedené na obrazovce. Formulář pro vstup dat zapisuje nový záznam do databáze až na explicitní žádost.



## Formulář s fiktivním záznamem vytvářeným na žádost

Tento případ se od předchozího liší pouze tím, že fiktivní záznam se vytvoří až na žádost uživatele a po vložení záznamu zanikne.

Takto se chovají formuláře, při jejichž vytváření na čtvrté stránce průvodce v **Detailech** odstraníte zatržení čtverce **Trvale zobrazovat fiktivní záznam**. V hotovém formuláři se tento režim vkládání zapne zatržením vlastnosti **Bez fiktivního** ve skupině **Chování formuláře**.

## Speciální formulář pouze pro vstup dat

Formulář pro vstup dat slouží pouze pro vkládání nových záznamů do tabulky a **nedovoluje** listovat v záznamech. Používá se tehdy, pokud je v aplikaci vhodné oddělit vkládání nových záznamů od manipulace s existujícími záznamy.

Při vytváření tohoto formuláře zvolte na čtvrté stránce průvodce variantu **Formulář sloužící pouze pro vložení nového záznamu**. Vytvořený formulář bude automaticky vybaven tlačítky pro řízení vkládání. Podle potřeby můžete některá z nich odstranit.

Hotový formulář změníte na formulář sloužící pouze pro vstup dat tak, že v seznamu vlastností zatrhnete čtverec **Pouze pro vstup dat**.

Formulář pro vstup dat má tyto vlastnosti:

- v formuláři je jediný záznam;
- po otevření formuláře budou jeho složky prázdné nebo naplněny svými implicitními hodnotami;
- zadáte-li hodnoty složek, přidáte je jako nový záznam do databáze stiskem klávesy **[Insert]**, voláním metody **Insert** popřípadě provedením akce **Vložit záznam do běžného formuláře**;
- před vložení záznamu se prověří integritní omezení všech složek a to i těch, do nichž jste nic nezapsali;
- po vložení záznamu se složky vyprázdní resp. naplní implicitními hodnotami a formulář je připraven pro vstup nového záznamu;
- pokud formulář zavřete před vložení záznamu do databáze, obsah databáze zůstane nezměněn;
- pokud stisknete **[Shift]+[F8]**, provedete akci **Zrušit běžný záznam** nebo zavoláte metodu **DelRec**, formulář se vyprázdní a naplní implicitními hodnotami. Do databáze se nevloží nic, ani se v ní nic nezruší.

Formulář pro vstup dat se v aplikacích typicky otevírá příkazem z menu nebo stiskem tlačítka v formuláři pro listování v datech. Použijete-li druhou variantu a chcete-li, aby se vložený záznam ihned objevil ve výchozím formuláři bez explicitního překreslování, musí oba formuláře sdílet kurzor a vyrovnávací paměť. Formulář pro vstup dat tedy mu-

síte otevřít akci **Na stejný záznam otevřít (modální) formulář** nebo volat funkci **Open\_view** s příznakem `PARENT_CURSOR`.

## Co přesně dělá stisk klávesy Insert?

Chování formuláře při stisku klávesy **Insert** (nebo při provedení alternativní akce či metody) závisí na druhu a stavu formuláře:

**Insert** na  
normálním  
záznamu

Je-li klávesa **Insert** stisknuta na normálním (existujícím) záznamu, pak:

1. pokud jsou ve vybraném záznamu přepsány některé hodnoty, změny jsou zapsány do databáze;
2. pokud ve formuláři není fiktivní záznam, vytvoří se dočasný fiktivní záznam;
3. do fiktivního záznamu se zapíší implicitní hodnoty definované v návrhu formuláře;
4. pokud jde o *subformulář relačně propojený* rovností hodnot dvou sloupců, pak se do spojovacího sloupce ve fiktivním záznamu zapíše hodnota ze spojovacího sloupce v nadřazeném formuláři (záznam je tím propojen s nadřazeným záznamem);
5. fiktivní záznam se stane vybraným záznamem a je možno do něj zapisovat hodnoty.

**Insert** na  
fiktivním  
záznamu

Je-li klávesa **Insert** stisknuta na fiktivním záznamu, pak:

1. pokud do fiktivního záznamu dosud nebylo nic zapsáno, pak se naplní implicitními hodnotami, ale do databáze se zatím nezapíše (reakce je stejná, jako na normálních záznamech);
2. pokud do fiktivního záznamu již bylo cokoli zapsáno, pak se celý jeho obsah zapíše do databáze a fiktivní záznam se tím stane normálním záznamem. Pokud formulář má mít trvalý fiktivní záznam, vytvoří se nový.

**Insert**  
v formuláři pro  
vstup dat

Je-li klávesa **Insert** stisknuta v formuláři pro vstup dat, pak:

1. obsah záznamu se vloží do databáze;
2. formulář se naplní implicitními hodnotami sloužícími pro zadání dalšího záznamu.

## Rušení záznamů

Ke zrušení běžného záznamu ve formuláři slouží kombinace kláves **Shift + F8**, akce **Zrušit běžný záznam** metoda **DelRec**. Akce a metoda umožňují rušit záznam stiskem vhodně definovaného tlačítka ve formuláři nebo na liště, případně příkazem z menu.

Ve formuláři může být rušení záznamů povoleno nebo zakázáno, je-li povoleno, může se vyžadovat potvrzení tohoto úmyslu.

Při vytváření nového formuláře pomocí průvodce povolíte nebo zakážete rušení záznamů na čtvrté stránce pomocí čtverce **Zrušení vybraného záznamu**. V **Detailech** lze specifikovat zda **Vyžadovat potvrzení zrušení**.

V hotovém formuláři nastavíte možnost rušení pomocí vlastnosti **Rušení záznamů** ve skupině **Chování formuláře**. Programově lze nastavit pomocí vlastnosti formuláře **Deletable**.

## Listování v záznamech a výběr záznamu

Listování v záznamech formuláře lze ovládat z klávesnice, myší pomocí posuvníku (táhla), pomocí tlačítek na ovládací liště nebo pomocí tlačítek ve formuláři. Programově lze listovat pomocí metod **FirstRec**; **LastRec**; **NextRec**; **PrevRec**; **NextPage** a **PrevPage**;

### Listování versus pevný záznam

Formulář může být buď určen buď pro zpřístupnění určitého záznamu, anebo pro volné listování v množině záznamů. Při vytváření formuláře lze jednu nebo druhou alternativu zvolit pomocí čtverce **Listování v záznamech** na čtvrté stránce průvodce. V **Detailech** lze dále stanovit, zda pro listování má být ve formuláři zobrazen posuvník (táhlo) u pravého okraje a zda se do formuláře mají vložit tlačítka pro listování.

Ve vytvořeném formuláři lze tyto schopnosti nastavit pomocí vlastnosti **Zákaz přechodů** ve skupině **Chování formuláře** a **Svislé táhlo** ve skupině **Okno formuláře**. Programově lze nastavit pomocí vlastnosti formuláře **Movable**.

### Prohlížení a výběr záznamů versus přístup ke složkám

Formulář může být buď určen buď pro prohlížení záznamů a případně výběr jednoho z nich, anebo pro práci se složkami a editaci. Při vytváření formuláře lze jednu nebo druhou alternativu zvolit pomocí poloh hlavního přepínače na čtvrté stránce průvodce.

Ve vytvořeném formuláři lze tyto varianty přepínat pomocí vlastnosti **Pouze pro výběr záznamu** ve skupině **Chování formuláře**.

Formulář určený pouze pro prohlížení a výběr záznamů nedovoluje editaci. Je-li navržen jako standardní formulář, pak se v něm zvýrazňuje vždy celý vybraný záznam a nikoli pouze jedna složka. Takovýto formulář obvykle obsahuje tlačítko, které s vybraným záznamem provede vhodnou akci nebo se využije akce na poklepání na záznamu ve standardním formuli.

## Provádění akcí vztahujících se k formuláři

Při práci s formulářem lze spouštět řadu akcí, ať už explicitně nebo jako vedlejší efekt prováděných operací. Tyto akce se popisují v seznamu vlastností formuláře a jeho složek.

### Akce definované pro formulář

Akce lze definovat pro tyto situace vztahující se k formuláři jako celku:

- otevření formuláře
- zavření formuláře
- přechod na nový záznam ve formuláři
- poklepání na záznam ve standardním formuláři
- kliknutí pravým tlačítkem myši na pozadí formuláře

Otevření  
formuláře

**Akce při otevření** se automaticky provede pokaždé, když je formulář otevřen, a to nezávisle na způsobu otevření. Při provádění této akce jsou již vytvořeny a naplněny složky pohledu.

Zavření  
formuláře

**Akce při zavírání formuláře** se automaticky provede bezprostředně před zavřením formuláře. Provede se bez ohledu na to, jakým způsobem je pohled zavřen (akcí uživatele, programem, atd.). Akce se provádí před zrušením složek pohledu.

Pokud potřebujete rozlišit akce při zavření pohledu jedním z několika příkazů menu nebo jedním z několika tlačítek, pak specifické akce musíte předřadit akcím zavírajícím pohled a **Akci při zavírání pohledu** nepoužívat.

Akce při vstupu  
na záznam

**Akce při vstupu na záznam** dovoluje například nestandardním způsobem synchronizovat obsah dvou nebo více formulářů. Pokud ji využijete tím způsobem, doporučujeme vypnout vlastnost **Automatické synchronizování**.

Akce při  
poklepání

**Akce při poklepání** na záznam se používá zejména ve formulářích sloužících pro výběr jednoho záznamu ze seznamu a spouští akcí související s tímto záznamem.

Akce před  
zápisem do  
databáze

Ve formuláři se nedefinuje žádná explicitní akce, která by se provedla před zápisem změn do databáze. Před uložením změn se prověří všechna integritní omezení definovaná na složkách. Pokud potřebujete provést nějaké dodatečné akce, naprogramujte je v některé funkci, která prověřuje integritní omezení.

### Akce definované pro složky

Tyto akce lze vyvolávat manipulací se složkami formuláře:

- stisk tlačítka ve formuláři nebo na jeho ovládací liště

- změna hodnoty editační složky provedena uživatelem
- chyba při zapisování nové hodnoty složky do cache formuláře
- kliknutí pravým tlačítkem myši na složce

### Akce po změně ve složce

Akce po změně umožňuje reagovat za změnu hodnoty složky a například přepsat hodnoty jiných složek. Nedoporučujeme provádět v akci po změně příliš složité akce.

**Pozor!**

V akci po změně se vyhněte zejména takovým akcím, které vyvolají opět změnu ve složce, která vyvolá akci atd. Taková akce vede nutně zacyklení a možnému k pádu systému.

### Akce při chybě

Akce při chybě se provede v těchto případech:

- uživatel запиše do složky údaj nepřipustného typu (např. písmena do celočíselné složky);
- hodnota zapsaná uživatelem poruší **Integritní omezení** definované pro složku v návrhu pohledu.

Typickou akcí při chybě je vypsání upozornění a vhodného vysvětlení v okně na obrazovce. Není-li akce při chybě definována, chyba se oznámí standardním způsobem.

Akce při chybě se nevyvolá, pokud chybu rozpozná až databázový server, například při porušení integritního omezení definovaného v návrhu tabulky, při zápisu duplicitní hodnoty do unikátního indexu apod. Tyto chyby jsou vždy hlášeny standardními chybovými okny.

### Akce při kliknutí pravým tlačítkem

Dle konvencí ovládání *Windows* by tato akce (je-li definována) měla vyvolávat popup menu vztahující se ke kontextu.

## Přehled a detail při listování v datech

Při prohlížení a editaci dat často stojí proti sobě dva požadavky:

- chceme současně vidět mnoho záznamů a listovat v nich;
- chceme mít přístup ke každému z mnoha údajů v záznamu.

Problém spočívá v tom, že obrazovka má jen omezenou velikost. Popíšeme zde dvě řešení tohoto problému:

1. otevírání dalšího formuláře na zobrazení detailů vybraného záznamu;
2. použití subformuláře synchronizovaného na stejný záznam;

Pokud je i v jediném záznamu více údajů, než lze přehledně umístit do okna formuláře, lze rozčlenit obsah formuláře do více stránek přepínaných pomocí záložek

## Rozdělení složek formuláře mezi přepínatelné záložky

Chcete-li v jednom formuláři zobrazit více údajů, než kolik se jich vejde na plochu obrazovky, lze část formuláře *využít vícenásobně* pomocí přepínatelných záložek. Ve vymezené části formuláře lze pak přepínat několik variant obsahu. Přepínanými variantami mohou být jak subformuláře, tak i ostatní složky a skupiny složek.

Formulář s obsahem přepínatelným pomocí záložek navrhnete takto:

1. Do formuláře vložíte složku **Záložky**. Na panelu vlastností v kolonce **Nápis** zadejte nadpisy všech záložek oddělené čárkami.
2. Do oblasti formuláře pod záložky vložíte všechny složky, jejichž viditelnost ve formuláři má být podmíněna přepínáním záložek.
3. Do podmínky viditelnosti každé z přepínaných složek napíšete:

`Tab_page(!!, číslo_složky_záložky)=pořad_číslo_záložky`

kde *číslo\_složky\_záložky* je číslo opsané ze seznamu vlastností této složky a *pořad\_číslo\_záložky* je pořadové číslo té záložky, na níž má být složka viditelná, počítáno od nuly. Je-li v formuláři pouze jedna složka typu záložky, pak se parametr *číslo\_složky\_záložky* ignoruje.

Již během návrhu formuláře budou viditelné pouze ty složky, které patří na aktivní stránku.

## Detailní formulář na vybraný záznam

Tento model předpokládá spolupráci dvou formulářů vedoucích do stejných dat. V prvním (přehledovém) formuláři je řada záznamů, ale z každého jen několik údajů. V druhém (detailním) formuláři je jediný záznam se všemi údaji - ten záznam, který je právě vybraný v přehledovém formuláři.

Poté, co navrhnete detailní formulář, vložíte do přehledového formuláře tlačítko s *akcí* **Na stejný záznam otevřít formulář** nebo **Na stejný záznam otevřít modální formulář** a zadáte jméno detailního formuláře.

**Detailní formulář otevřený z přehledového**

**Přehled zaměstnanců**

Jméno	Příjmení	Viz
Karel	Novotný	Detail
Jarmila	Veselá	Detail
Petr	Dvořák	Detail
Jana	Němcová	Detail

**Detaily o zaměstnanci**

Os. číslo: 2 Jméno: Jarmila Příjmení: Veselá  
 Plat: 12700.00 Oddělení: 4500 Nástup: 1.3.1999  
 Děti: Petr, Alena, Klára

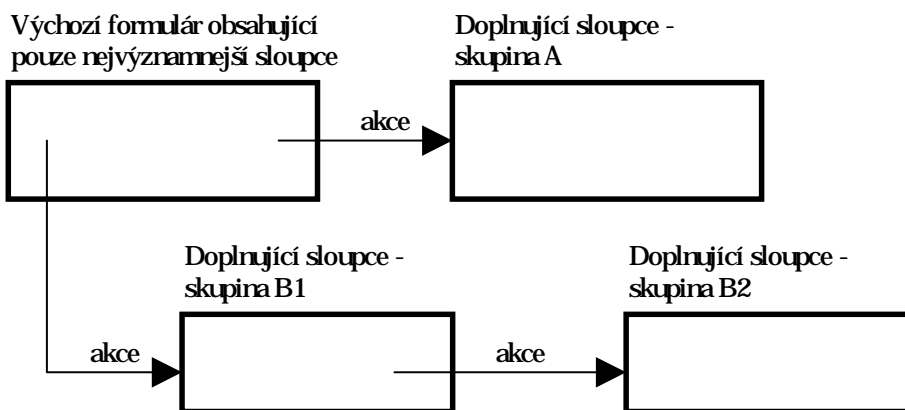
Otevřete-li detailní formulář jako modální, bude po dobu jeho otevření zbytek **Win-Base602** blokován. V opačném případě budete moci listovat v přehledovém formuláři a v detailním formuláři se vždy objeví data z právě vybraného záznamu. Obsah obou formulářů bude tedy synchronizován. Tuto synchronizaci lze vypnout pomocí vlastnosti **Automatické synchronizování** ve skupině **Chování formuláře**.

Obvykle přehledový formulář obsahuje pouze hodnotové složky a editovat data lze pouze v detailním formuláři. Nezbytné to však není. V detailním formuláři obvykle není možno přecházet mezi záznamy a není zobrazen svislý posuvník.

### Volitelné a postupné zobrazování sloupců

Modifikaci výše popsaného postupu lze použít také tehdy, pokud je v některém zdroji dat tolik sloupců, že se nevejdou na obrazovku, anebo z věcných důvodů není vhodné zobrazovat všechny najednou.

Pak navrhnete dva nebo více formulářů obsahujících různé sloupce z téhož zdroje dat. Formuláře seřadíte do posloupnosti nebo do stromové struktury a do nadřazených formulářů vložíte tlačítka otevírající *na stejný záznam* formuláře podřízené.



Ve výchozím formuláři je jeden nebo více záznamů, v ostatních formulářích jsou další sloupce jediného záznamu - toho, který je vybrán v předchozím formuláři.

### Sdílení kurzoru a vyrovnávacích pamětí v formulářích

Otevíráte-li další formuláře na stejný záznam výše popsanými *akcemi* nebo otevíráte-li další formuláře z programu s použitím příznaku `PARENT_CURSOR`, pak původní a nově otevíraný formulář budou sdílet stejný kurzor i vyrovnávací paměti pro komunikaci s databázovým serverem. Změny prováděné v jednom z formulářů se automaticky promítnou do všech ostatních. Kurzor se zavře až po uzavření všech těchto formulářů.

#### Pozor:

Podmínkou tohoto způsobu práce je, že kurzor výchozího formuláře obsahuje všechny sloupce použité ve všech dalších formulářích - jinak se formuláře nepodaří otevřít. Zrušené záznamy se mohou zobrazovat buď ve všech formulářích nebo v žádném z nich.

### Promítání změn z formuláře do formuláře bez sdílení kurzoru a vyrovnávací paměti

#### Proč nesdílet?

Otevřete-li dva formuláře vedoucí (zčásti nebo zcela) do stejných dat a nepoužijete-li přitom *akci Na stejný záznam otevřít (modální) formulář* ani volání funkce s parametrem `PARENT_CURSOR`, pak se **WinBase602** nedokáže automaticky postarat o přenášení změn z jednoho formuláře do druhého. Důvodem k volbě tohoto řešení je zpravidla situace, v níž se nedá v obou formulářích použít stejný kurzor. Například při pomalém internetovém nebo telefonním spojení klienta na server je efektivnější v přehledovém formuláři použít kurzor obsahující pouze minimum sloupců a v detailním formuláři na jeden záznam pak kurzor se všemi sloupci.

Pokud uživatel provede změny v formuláři nesdílejícím kurzor, ostatní formuláře se o nich nemohou dozvědět dříve, dokud nejsou zaznamenány do databáze. Pokud tedy dva formuláře vedou do stejných dat, může se stát, že změny provedené v jednom z nich nejsou v druhém vidět.

#### Jak přenést změny mezi formuláři?

Možných řešení promítnutí změn je několik. Nejprve je nutno zajistit přepsání změn z vyrovnávací paměti formuláře do databáze některou z těchto metod:

1. specifikovat složkovou synchronizaci změn ve formuláři, tedy zajistit, že hodnota každé složky se запиše do databáze ihned po dokončení její editace;
2. v okamžiku, kdy je nutno zajistit promítnutí změn, volat *akci Uložit změny ve formuláři* (nebo metodu či funkci `Commit_view`);
3. pokud stačí přenesení změn do databáze při opouštění formuláře, nastavit v návrhu formuláře příznak **Ukládat při deaktivaci**; deaktivace způsobená otevřením jiného formuláře však nastává až poté, co si tento formulář načte data;
4. pokud stačí přenesení změn do databáze při zavření formuláře, stane se tak automaticky.

Tato řešení jsou dostatečná, pokud druhý formulář bude teprve otevřen. Pokud již otevřen je, je třeba navíc znovu naplnit jeho vyrovnávací paměť provedením *akce Načíst data a překreslit formulář* (nebo procedurou či metodou `Reset_view` s posledním paramet-



rem `RESET_CACHE`). To lze provést například jako **Akci po změně**, **Akci při zavření** nebo akci vyvolanou tlačítkem ve formuláři, který provedl změnu dat.

## Přehledový (listovací) subformulář v formuláři

Dva formuláře na stejná data - přehledový s množstvím záznamů a detailní se všemi sloupci jednoho záznamu - se dají uspořádat do soustavy, v níž je přehledový formulář vložen do detailního. Detailní formulář se stává *superformulářem*, přehledový formulář jeho *subformulářem* sloužícím k pohodlnému listování. Postupujte takto:

1. Nejprve navrhnete přehledový formulář, nejlépe jako standardní formulář.
2. Pak vytvoříte detailní formulář - musí to být obecný formulář - a na volné místo do něj vložíte složku typu *subformulář*.
3. V seznamu vlastností této složky vyberte ve vlastnosti **Subpohled** jméno přehledového formuláře,
4. **Způsob synchronizace** subformuláře ve skupině **Subpohled** nastavte na **Stejný záznam**.

Formulář se subformulářem

Takto vytvoříte subformulář synchronizovaný na stejný záznam. To znamená, že při listování v jednom z formulářů se zároveň automaticky listuje v druhém formuláři, takže oba mají stále vybraný tentýž záznam. Přehlednost se zvýší, pokud ze *superformuláře* odstraníte svislé táhlo.

## Kurzor ze *superformuláře* v subformuláři

Ve vlastnosti subformuláře **Použít kurzor ze superpohledu** lze zvolit, zda subformulář má převzít kurzor (resp. tabulku) od *superformuláře* nebo zda má mít svůj vlastní zdroj dat, dle své definice. Doporučujeme používat kurzor ze *superformuláře*.

Stejný kurzor

Bude-li tento označovací čtverec zatržen, pak *superformulář* i subformulář pak budou trvale sdílet stejný kurzor i vyrovnávací paměti. Tento stav má několik důsledků:

- ušetří se čas i paměť při otevírání *superformuláře*, protože není třeba otevírat další kurzor pro subformulář a znovu načítat data;

- ušetří se čas při listování v formulářích, protože bude snadnější najít odpovídající záznam v druhém formuláři;
- záznamy budou v obou formulářích stejně seříděny;
- změny v datech, vložení i zrušení záznamů se okamžitě promítne do obou formulářů;
- při zadání dotazu v superformuláři se zadaná omezení a seřídění promítnou i do subformuláře;
- je-li superformuláři předán nový kurzor pomocí funkce `Set_fcursor`, předá se zároveň i do subformuláře.

Stejný kurzor lze použít pouze za předpokladu, že obsahuje všechny sloupce použité v obou formulářích.

#### Různé kurzory

V některých výjimečných případech, může mít smysl požadovat, aby superformulář i subformulář nevedly do téhož zdroje dat, např. jeden může být uspořádaný a druhý nikoliv apod. V praxi to ale není běžný případ. Pokud k němu dojde, je nutno použít metody promítání změn z formuláře do formuláře popsané v předchozí sekci.

### Tisk formuláře se subformulářem synchronizovaným na stejný záznam

Formulář se subformulářem synchronizovaným na stejný záznam není příliš vhodný pro tisk. Obsah sestavy se subformulářem se bude lišit od vzhledu formuláře na obrazovce. Ze subformuláře (tištěného jako součást superformuláře) se vytiskne v každém záznamu superformuláře pouze jeden záznam - ten samý, který je v superformuláři.

Toto chování může být nežádoucí v případech, kdy superformulář tvoří pouze jakýsi rámec pro data obsažená v subformuláři nebo subformulářích. V takovém případě zajistěte, aby superformulář vedl do proměnných projektu, nebo vypněte automatickou synchronizaci. V obou případech pak k synchronizaci nedojde a subformuláře se vytisknou celé.

## Vyhledávání a selekce dat, seřídění záznamů

Vyhledání záznamu nebo záznamů splňujících určité podmínky a jejich seřídění podle hodnot určitých sloupců patří mezi nejběžnější operace při práci s daty. Tyto akce mohou být implementovány v aplikaci pomocí dotazu na databázi a zobrazení kurzoru ve formuláři. Lze však také poskytnout uživateli možnost vybírat a třídit data podle potřeby.

Jako univerzální nástroj pro řešení tohoto úkolu má každý uživatel k dispozici možnost kdykoli otevřít dotazové okno QBE a v něm zadat omezující podmínky. Tento nástroj je velmi silný a obecný, ale v aplikacích může mít dvě nevýhody:

1. dotazové okno obsahující všechny složky původního formuláře může uživatelům připadat příliš složité v situaci, kdy chtějí hledat podle jednoho konkrétního sloupce;
2. uživatel může chtít vyhledávat bez otevírání dalších oken.

## Přehled způsobů výběru dat a třídění

Nyní popíšeme několik cest, jak připravit aplikaci, která uživateli nabídne dokonalejší způsoby výběru a třídění dat.

### Cesty k výběru dat

Data lze vybírat (filtrovat) těmito způsoby:

- zadáním omezení ve standardním dotazovém okně QBE, jehož obsah může být oproti výchozímu oknu mírně upraven;
- zadáním omezení v okně speciálně navrženém pro kladení určitých výběrových podmínek;
- vyplněním omezení do dotazových polí;
- sestrojením kurzoru a jeho vložením do formuláře (obvykle s podporou programu).

### Cesty k třídění dat

Data lze setřídít těmito způsoby:

- zadáním třídicího pořadí (čísla 1 až 8) ve standardním okně pro třídění, jehož obsah může být oproti výchozímu oknu mírně upraven;
- poklepáním na záhlaví sloupce ve standardním formuláři;
- provedením **akce Setřídít podle hodnot sloupce**;
- sestrojením kurzoru a jeho vložením do formuláře (obvykle s podporou programu).

Sestrojení kurzoru a jeho vložení do formuláře již leží mimo obsah této kapitoly. Dovoluje však nejobecnější výběr a třídění dat podle libovolných podmínek. Otevřený kurzor se buď předá otevíranému formuláři funkcí **Open\_view** nebo se vloží do již otevřeného formuláře zapsáním do vlastnosti **Cursnum** nebo funkcí **Set\_fcursor**.

### Skládání omezení

Výběr dat a setřídění, provedené všemi způsoby kromě předání kurzoru, se skládají. Lze proto například zadávat některá omezení ve speciálních dotazových oknech a jiná pomocí dotazových polí.

## Povolení a zákaz QBE

Využití standardního QBE k výběru a třídění záznamů lze při vytváření nového formuláře povolit nebo zakázat na čtvrté stránce průvodce pomocí čtverce **Kladení dotazů (výběr záznamů) a třídění**. V Detailch lze zadat, zda se příkazy pro QBE mají objevit i v systémovém menu formuláře.

V existujícím formuláři lze povolení QBE řídit nastavením vlastnosti **Zákaz QBE a třídění** ve skupině **Chování formuláře** a **Omezené menu** ve skupině **Okno formuláře**.

## Úprava dotazových formulářů QBE

Vzhled QBE formuláře pro kladení dotazu nebo omezení je přímo odvozen ze vzhledu výchozího formuláře. Snadno lze dosáhnout toho, aby některé složky formuláře byly viditelné resp. aktivní pouze ve výchozím formuláři nebo pouze ve zvoleném dotazovém formuláři.

K dosažení selektivní viditelnosti a aktivity slouží metoda nebo funkce `QBE_state`. Její hodnota je 0 pro normální formulář, 1 pro QBE formulář pro zadávání omezení a 2 pro formulář pro zadávání pořadí. Lze ji použít (mimo jiné) v podmínkách viditelnosti nebo aktivity složek.

Pokud některá složka formuláře nemá být viditelná v dotazovém formuláři, pak do její podmínky viditelnosti napíšete:

```
QBE_state(!) = 0    nebo    thisform.QBE_state=0
```

Pokud některé tlačítko má být neaktivní v dotazových formulářích, pak stejný výraz napíšete do jeho podmínky aktivity.

Pokud některé pole se má objevit pouze v formuláři pro zadávání třídění, napíšete do jeho podmínky viditelnosti výraz:

```
QBE_state(!) = 2    nebo    thisform.QBE_state=2
```

## Kladení dotazu v samostatném okně

### Navržení dotazového formuláře

Pro kladení dotazu lze navrhnout zvláštní formulář, který obsahuje pouze ty složky, na něž mají být kladena omezení. Navrhnete jej jako každý jiný formulář. Musí vést do stejného zdroje dat, jako formulář, v němž se bude používat.

Do formuláře, ve kterém chcete klást dotazy, vložte tlačítko s *akcí* **Zadat omezení ve speciálním okně** a v popisu *akce* zadejte jméno předem navrženého dotazového formuláře. Stiskem tohoto tlačítka otevřete dotazové okno v modálním režimu.

### Potvrzení dotazu

Zadávání dotazu v dotazovém okně ukončíte jako obvykle, tedy buď klávesou **[F3]** nebo příkazem **Dotaz/pořadí zadáno** ze systémového menu. Nejpráhlednější cestou je však vložit do dotazového okna tlačítko s *akcí* **Uplatnit omezení nebo uspořádání** nebo voláním metody **AcceptQuery** a kladení dotazu ukončovat tímto tlačítkem.

### Odvolání dotazu

Pokud dotazové okno zavřete jinak, např. příkazem **Zavřít** ze systémového menu, kombinací kláves **[Ctrl]+[F4]**, tlačítkem s *akcí* **Zavřít formulář** nebo voláním metody **Close**, pak se zadaný dotaz neuplatní.

Pokud by v dotazovém formuláři byla složka, která není v formuláři, z něhož dotaz otevřete, pak se její omezení neuplatní. Pokud potřebujete tuto vlastnost obejít, musíte do výchozího formuláře vložit neviditelnou složku.

Formulář sloužící k zadání dotazu lze také otevřít pomocí funkce **Open\_view** s příznakem `QUERY_VIEW`.

## Dotazová pole

Do obecného formuláře lze vložit speciální složky, které budou sloužit pouze k zadávání omezení na záznamy, které se v formuláři objevují. Tím se zjednodušuje proces kladení dotazu a vyhledávání záznamů, které obsahují určité hodnoty. Místo otevření dotazového okna (standardního QBE nebo uživatelem navrženého), vyplnění omezení a jeho potvrzení stačí pouze vyplnit omezení.

Vytvoření dotazových polí lze předepsat při vytváření nového obecného formuláře na třetí stránce průvodce zatržením u příslušného údaje čtverce **Dotazové pole**.

Do hotového formuláře se nové dotazové pole vloží jako každá jiná složka, ale mezi vlastnostmi složky vyberete v combu **Dotazové pole** buď **Statické** nebo **Dynamické dotazové pole** a případně vyplníte také **Dotazovou podmínku**.

Statické a  
dynamické  
dotazové pole

- **Dynamické dotazové pole** ovlivní obsah formuláře, jakmile je v něm změněn jakýkoli znak. Hodí se tudíž zejména pro vyhledávání pomocí prefixu řetězce. Postupným zapisováním např. jména se neustále zužuje množina vyhovujících záznamů v formuláři, až se objeví to hledané.
- **Statické dotazové pole** ovlivní obsah formuláře, jakmile je v něm změněna hodnota a stisknuta klávesa **Enter** nebo je složka opuštěna. Hodí se zejména pro vyhledávání pomocí kompletně zadané hodnoty (např. identifikačního čísla).

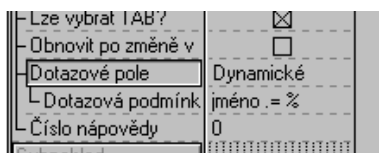
Dotazová  
podmínka

Pokud **Dotazová podmínka** vyplněna není, pak o funkci dotazového pole rozhoduje jméno sloupce zadané jako vlastnost **Obsah složky**. Přepokládejme, že dotazové pole má obsah `os_číslo`. Pak pokud při práci s formulářem uživatel napíše do dotazového pole hodnotu 200, pak se vyberou záznamy, v nichž je ve sloupci `os_číslo` hodnota 200. Pokud uživatel napíše >200, pak se vyberou záznamy, v nichž je hodnota sloupce `os_číslo` větší než 200.

Vyplněním **Dotazové podmínky** lze uživateli ulehčit zadávání složitějších podmínek. Uvedete v ní booleovský výraz, v němž jsou místa, do nichž se má vložit hodnota zadaná uživatelem, označena znakem %. Jakmile uživatel zadá omezující hodnotu, zapíše se na místa znaků % a do formuláře se vyberou záznamy splňující takto vzniklou podmínku. Například pokud **Dotazová podmínka** bude `os_číslo>%-10 AND os_číslo<%+10`, pak se do formuláře vyberou ty záznamy, v nich se hodnota sloupce `os_číslo` liší od uživatelem zadané o méně než 10.

Pokud **Dotazová podmínka** bude `jméno . = %`, pak se v formuláři objeví záznamy, v nichž hodnota sloupce **jméno** začíná zadaným řetězcem.

Vlastnosti  
složky týkající  
se dotazového  
pole



Omezení zadaná v dotazových polích se skládají s omezeními zadanými všemi ostatními způsoby. Lze je odstranit buď smazáním obsahu dotazového pole, anebo stejně, jak se odstraňuje položený dotaz (např. tlačítkem na ovládací liště).

## Kladení dotazu ve formuláři se subformulářem

Položíte-li ve formuláři dotaz, pak chování jeho subformulářů závisí na způsobu, kterým je subformulář propojen se superformulářem. Přitom nehraje roli, zda dotaz byl položen pomocí standardního QBE, v samostatném okně nebo pomocí dotazového pole.

Pokud je subformulář synchronizován relačně, pak položení dotazu změní množinu záznamů zobrazenou v superformuláři, ale obsah subformuláře závisí pouze na právě vybraném záznamu v superformuláři.

Pokud je subformulář synchronizován na stejný záznam, pak pokud nesdílí kurzor se superformulářem, nebude položením dotazu dotčen. Pokud je však pro subformulář nastaven příznak, že má používat kurzor ze superformuláře, pak se položení dotazu projeví v superformuláři i v subformuláři stejným způsobem.

Těchto vlastností lze v aplikacích vhodně využít. Zadáte-li v superformuláři výběrové podmínky (např. vyplněním dotazového pole), pak pokud formuláře nesdílejí kurzor, budete v subformuláři nadále vidět všechny záznamy, ale při listování v superformuláři budete přecházet pouze mezi záznamy splňujícími zadané podmínky. Naopak pokud se kurzor sdílí, budou v subformuláři jen ty záznamy, které podmínkám vyhovují.

Pozor:

Pokud zadáte podmínku, které nevyhovuje žádný záznam, a formulář nemá fiktivní záznam, pak všechny jeho složky z datových oblastí zmizí. Zmizí tedy i případné subformuláře a dotazová pole. Doporučujeme proto buď povolit trvalý fiktivní záznam v superformuláři, anebo umístit dotazová pole a další složky do záhlaví nebo paty okna.

## Třídění záznamů

Zatímco standardní okno pro zadávání třídění a také programem sestrojený kurzor dovolují třídít záznamy podle hodnot více sloupců, poklepáním na záhlaví sloupce nebo provedením akce **Setřídít podle hodnot sloupce** se záznamy setřídí podle jediného sloupce.

Použití dotazových polí v superformuláři lze kombinovat s tříděním záznamů pomocí záhlaví sloupců v subformuláři.

Třídění poklepáním na záhlaví sloupce lze při návrhu formuláře zakázat pomocí podmínky aktivity složky v záhlaví sloupce.

## Relačně propojené formuláře

V této sekci si připravíme nástroj, který využijeme v sekci následující. Jde o takový způsob vazby mezi dvěma formuláři, že každému záznamu v prvním formuláři je přiřazena množina záznamů zobrazena v druhém formuláři. Mechanismus synchronizace obou formulářů zajišťuje, že při listování v prvním formuláři se v druhém formuláři vždy zobrazují záznamy svázané s právě vybraným záznamem.

RELAČNÍ SYNCHRONIZACE FORMULÁŘŮ, o níž zde budeme hovořit, funguje tedy jedním směrem a také tím se liší od synchronizace na stejný záznam.

Záznamy v obou formulářích se propojují na základě vztahu mezi hodnotami sloupců. Nejjednodušší (a nejběžnější) případ je propojení záznamů při rovnosti hodnot ve dvou sloupcích, ale mechanismus relační synchronizace zvládne i složitější případy. Obecně je třeba, aby na základě hodnot v superformuláři bylo možno zadat *podmínku* pro výběr záznamů v subformuláři.

Dvojice relačně synchronizovaných formulářů může být vytvořena jako superformulář se subformulářem nebo jako dva samostatné formuláře, z nichž první otevírá v novém okně druhý.

### Formulář s relačním subformulářem

Při vytváření formuláře s relačně synchronizovaným subformulářem postupujte takto:

1. navrhněte subformulář - obvykle jako standardní formulář;
2. navrhněte superformulář jako obecný formulář a vložte do něj (dostatečně velkou) složku Subpohled;
3. nastavte ve vlastnosti **Subpohled** této složky jméno subformuláře;
4. zadejte způsob synchronizace obou formulářů jedním ze dvou níže popsaných způsobů.

### Propojení záznamů pomocí relace

Propojení záznamů ve formulářích relací znamená využití dříve navržené relace mezi tabulkami použitými ve zdrojích dat formulářů. Relace obsahuje jednu nebo více dvojic sloupců z obou tabulek. Do relace budou uvedeny ty záznamy, které mají v každé dvojici sloupců stejné hodnoty.

Propojení pomocí relace zadáte ve skupině vlastností **Subpohled**. **Způsob synchronizace** nastavíte na **Záznamy propojené relací** a **Jméno relace** vyberte z nabídky.

**Relační  
subformulář**

Chování složky	
Subpohled	Fakt_sub
- Způsob synchronizace	Záznamy propojené relací
- Použít kurzor ze s	
- Jméno relace	Fakt_firmy
- Propojovací hodnota	

**Propojení záznamů výběrovou podmínkou**

Tento způsob propojení dovoluje vybrat záznamy v subformuláři pomocí libovolné podmínky využívající hodnotu jednoho sloupce ze superformuláře.

Ve skupině vlastností **Subpohled** nastavte **Způsob synchronizace** na **Záznamy propojené podmínkou**. Jako **Propojovací hodnotu v superpohledu** zadejte jméno propojujícího sloupce superformuláře, jako **Výběrovou podmínku na subpohled** uveďte podmínku, na jejímž základě se budou vybírat záznamy do subformuláře. Tato podmínka musí mít tyto vlastnosti:

- použijí se v ní pouze jména sloupců ze subformuláře a znak %, který symbolizuje hodnotu sloupce superformuláře uvedeného jako **Propojovací hodnota**;
- po zápisu hodnoty propojovacího sloupce na místo znaku % musí vzniknout podmínka v jazyce SQL vyhodnotitelná na serveru;
- do subformuláře se vyberou ty záznamy, pro něž je tato podmínka splněná.

**Relační  
subformulář**

Chování složky	
Subpohled	Fakt_sub
- Způsob synchronizace	Záznamy propojené podmínkou
- Použít kurzor ze s	
- Jméno relace	
- Propojovací hodnota	cislo
- Výběrová podmínka	firma=%
Číslo složky	7

Jednoduchá výběrová podmínka, která propojuje záznamy na základě rovnosti hodnot dvou sloupců, má tvar:

sloupec\_subformuláře = %

Lze však použít i složitější podmínky, např.

ABS(sloupec\_subformuláře - %) <= 20

**Vkládání záznamů do subformuláře**

Stisknete-li v relačně synchronizovaném subformuláři klávesu **Insert** a jsou-li oba formuláře propojeny rovností sloupců, pak se do nově vkládaného záznamu v subformuláři automaticky запиše propojovací hodnota.

Při návrhu subformuláře je proto vhodné zajistit, aby se v něm fiktivní záznam nezobrazoval trvale, a objevil se až po stisku **Insert**, s již zapsanou propojovací hodnotou. Pokud by uživatel vložil záznam zapsáním hodnot do trvale zobrazeného fiktivního zá-



znamu, nebyl by nový záznam automaticky propojen s žádným záznamem nadřazeného formuláře.

## Relační synchronizace samostatných formulářů

Množinu záznamů relačně propojených s vybraným záznamem ve formuláři lze zobrazit v samostatném formuláři, který se z prvního formuláře otevře.

Pro otevření závislého formuláře použijete akci **Otevřít relačně synchronizovaný formulář**. V ní zadáte jméno formuláře a jméno relace, která uvádí obsah obou formulářů do souvislosti.

Pokud druhý formulář není modální, pak přechody mezi záznamy v prvním formuláři budou vyvolávat nové naplnění druhého formuláře.

## Relačně propojené tabulky

Téměř každá netriviální databázová aplikace pracuje s více tabulkami, které jsou v určitém vzájemném vztahu. **WinBase602** vztahy mezi tabulkami řeší dvěma způsoby: propojením relačním a propojením ukazatelovým. V této části si podrobně popíšeme relační propojení dat a uspořádání formulářů, které s nimi dovoluje efektivně pracovat.

Zatímco propojování dat pomocí ukazatelů obvykle přináší rychlejší výsledky v jednoduchých aplikacích, relačnímu propojení dáme přednost ve složitějších případech.

V dalším budeme předpokládat, že relace mezi tabulkami jsou již definovány a pojmenovány v aplikaci. Na jména relací se budete odvolávat v *akcích* otevírajících relační formuláře.

Jaké vztahy  
rozlišujeme...

Uvažujme dvě tabulky TA a TB, jejichž záznamy chceme propojovat. V první řadě je nutno rozlišit druh vazby mezi jejich záznamy. Může nastat jeden ze tří případů:

- **Vztah 1:1:** Každému záznamu v tabulce TA odpovídá *právě jeden* záznam v tabulce TB a naopak.
- **Vztah 1:N:** Každému záznamu v tabulce TA odpovídá množina záznamů z tabulky TB, ale každý záznam z tabulky TB přísluší k *nejvýše jednomu* záznamu z tabulky TA (v "silnější" variantě každý záznam z tabulky TB přísluší k *právě jednomu* záznamu z tabulky TA, v tabulce TB tedy nejsou záznamy, které by nepatřily k žádnému záznamu z TA).
- **Vztah M:N:** Každému záznamu v tabulce TA odpovídá množina záznamů z tabulky TB a naopak každému záznamu v tabulce TB odpovídá množina záznamů z tabulky TA.

Vztahem **1:1** se nebudeme podrobněji zabývat. Tabulky TA a TB lze totiž spojit do jedné tabulky, buď fyzicky anebo pomocí SQL. Pokud by platilo, že každému záznamu v tabulce TA odpovídá *nejvýše jeden záznam* v tabulce TB, pak jde o speciální případ vztahu **1:N** a lze s ním takto pracovat.

V ostatních případech přepokládejme, že pro vyjádření vztahu mezi záznamy slouží sloupec A v tabulce TA a sloupec B v tabulce TB. Obecně v relaci může být definováno více dvojic sloupců, ale na podstatně věci to nic nemění.

Pozor:

Pro efektivní práci s relačně propojenými záznamy je třeba, aby v tabulce TA byl index ke sloupci A a v tabulce TB index ke sloupci B.

## Vztah 1:N

Propojovací sloupce A a B jsou stejného typu a propojují dva záznamy právě tehdy, pokud v nich mají stejnou hodnotu. Na propojení záznamů v tomto vztahu lze hledět třemi způsoby a všechny tři způsoby lze libovolně kombinovat.

### Spojení tabulek

Pomocí dotazu **SELECT \* FROM TA, TB WHERE TA.A=TB.B** lze získat celkový přehled o všech vazbách mezi záznamy. Ve formuláři do odpovědi na tento dotaz:

- lze klást dodatečné omezující podmínky, které vyberou z jedné nebo druhé tabulky pouze určité záznamy;
- lze editovat hodnoty, avšak pokud editací změníte propojení záznamů, projeví se to až při novém položení dotazu;
- nelze vkládat ani rušit záznamy.

### Formulář na propojení od TA k TB

V formuláři vedoucím do tabulky TA (nebo do dotazu, který z ní bere záznamy) lze vytvořit relačně propojený subformulář se záznamy z tabulky TB. V každém záznamu z TA budou v subformuláři ty záznamy z TB, které jsou k němu připojené.

Subformulář navrhnete do tabulky TB. Při navrhování hlavního formuláře do TA v popisu složky Subpohled zadáte kromě jména subformuláře:

- buď **Druh propojení** jako **Záznamy propojené relací** a vyberete **Jméno relace**;
- anebo **Druh propojení** jako **Záznamy propojené podmínkou**, **Propojovací hodnota** bude sloupec A a **Výběrová podmínka** bude sloupec B resp. výraz **B=%**.

Vkládání  
záznamů  
s propojováním

V hlavním formuláři lze vkládat záznamy do TA nebo je rušit. V subformuláři lze vkládat záznamy do TB nebo je rušit. Pokud vložíte záznam do TB stiskem klávesy **Insert**, pak se v něm do sloupce B запиše hodnota sloupce A z vybraného záznamu v hlavním formuláři. Tím automaticky vznikne propojení mezi záznamem vloženým do subformuláře a

záznamem, v němž se subformulář nachází. (Tento mechanismus však funguje pouze tehdy, je-li výběrová podmínka subformuláře ve výše uvedeném tvaru.)

Tento způsob práce je velmi názorný. Hodí se zejména pro "silnější" verzi vztahu 1:N, v níž v TB neexistují záznamy, které by nepatřily k žádnému záznamu v TA. Pomocí subformuláře nelze totiž záznam v TB "odpojit" od záznamu v TA (leda přepsáním hodnoty sloupce B), ani "připojit" již existující, ale dosud nepřipojený záznam z TB.

Sloupec A musí mít v záznamech v TA unikátní hodnoty. Toho lze dosáhnout například automatickým přidělováním unikátních hodnot zajištěným v návrhu tabulky TA.

Samostatné  
formuláře

Alternativně je možno zobrazovat záznamy z TB v samostatném formuláři, nevloženém do hlavního formuláře, nýbrž otevíraném tlačítkem s akcí **Otevřít relačně synchronizovaný formulář**. Parametry této akce jsou jméno formuláře do TB a jméno propojující relace.

## Formulář na propojení od TB k TA

V formuláři vedoucím do tabulky TB (nebo do dotazu, který z ní bere záznamy), lze vytvářet propojení na záznamy z TA. Lze tak činit dvěma cestami:

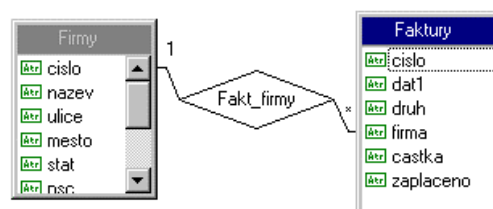
- pomocí složky *combo plněné dotazem*, která obsahuje texty identifikující záznamy z TA a po vybrání některé položky provádí přepis hodnoty sloupce A do sloupce B;
- pomocí tlačítka, které otevírá *relační formulář* do TA (a případně subformuláře, který obsahuje právě připojený záznam z TA). Jméno relace mezi TA a TB se uvede v parametrech akce otevírající relační formulář.

Hlavní formulář do TB umožňuje vkládání a rušení záznamů v TB. Pomocí relačního formuláře lze také vkládat a rušit záznamy v TA. Lze pracovat i se záznamy z TB, které nejsou spojeny s žádným záznamem z TA. Propojovat lze i fiktivní záznam v TB, propojení se pak zapíše do databáze při opuštění záznamu.

Relační formulář se z programu dá otevřít funkcí **Relate\_records**.

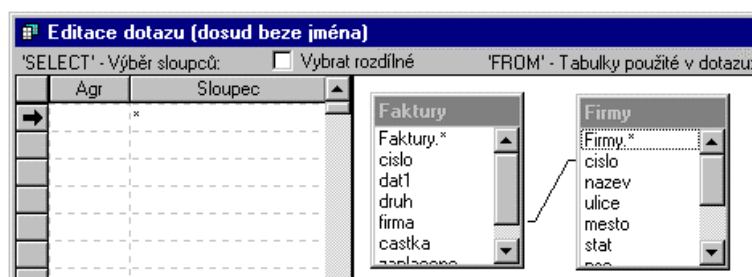
## Příklad vztahu 1 : N

Mějme tabulku FIRMY, v níž jsou zapsána plná jména a adresy našich obchodních partnerů a tabulku FAKTURY, kde zaznamenáváme naše obchodní případy. Do tabulky FAKTURY nebudeme samozřejmě zapisovat pro každý záznam plné jméno firmy, které se obchod týká, ale uvedeme v ní pouze číslo, které jednoznačně odpovídá firmě z tabulky FIRMY. Tabulka FIRMY tedy slouží jako číselník pro tabulku FAKTURY. Vazba mezi těmito dvěma tabulkami je tvořena relačním vztahem 1:N. V tabulce FIRMY je sloupec CISLO označující číslo firmy, totéž číslo firmy se zapisuje do sloupce FIRMA v tabulce FAKTURY. Vztah mezi tabulkami je znázorněn na obrázku. Tabulky jsou použity z aplikace PŘÍKLADY.

**Vztah mezi  
tabulkami****Spojení tabulek**

Při práci se záznamy faktur je však potřeba vidět i jméno a další údaje o firmě. Pro prohlížení a editaci údajů vyhoví vhodně navržený dotaz, který potřebné údaje z obou tabulek umí spojit a zobrazit v jediném formuláři.

V návrháři dotazů vložte obě tabulky, propojení obou tabulek podle relace se vytvoří automaticky. Pokud relace není definována, vytvořte propojení ručně.

**Propojení  
tabulek**

Formulář vedoucí do obdobného dotazu obsahuje tedy nejen údaje o faktuře z tabulky FAKTURY, ale i údaje o firmě z tabulky FIRMY.

Jiná situace je však u vkládání nových záznamů. V formuláři, vedoucím do odpovědi na dotaz vytvořený ze dvou tabulek, nelze vkládat záznamy.

**Formulář do  
dotazu**

Druh	Datum	Dodavatel nebo odběratel	Částka Kč	Zaplaceno dne
Došlé	1.1.1998	Brněnská drátovna a.s.	1000,-	15.1.1998
Odeslané	1.1.1998	Kancelářské potřeby Praha	2500,-	15.1.1998
Došlé	2.1.1998	Brněnská drátovna a.s.	2656,-	25.1.1998
Došlé	14.1.1998	Spojené šroubárny Kolín	100,50	24.1.1998
Odeslané	15.1.1998	Kancelářské potřeby Praha	5050,-	10.2.1998
Odeslané	16.2.1998	AFFEX spol. s r.o.	1500,-	16.2.1998

Nyní je jasné, že nové záznamy se musí vkládat do původních tabulek. V případě tabulky FIRMY žádný problém nenastane - nová firma bude tvořit nový záznam. Jak to bude ale se záznamy v tabulce FAKTURY?

### Formulář na propojení od FIRMA k FAKTURA

Ukázku tohoto přístupu vytvoříme na stejných datech. Standardní formulář do tabulky FAKTURY bude tvořit subformulář v superformuláři do tabulky FIRMA. V parametrech složky Subpohled je nutno definovat jméno relace nebo explicitní relační vazbu:

propojovací hodnota : CISLO

výběrová podmínka : FIRMA

Při otevření superformuláře budou obsahem subformuláře vždy ty záznamy z tabulky FAKTURY, které patří k vybrané firmě.

#### Relační subformulář

Číslo:	Datum:	Částka:	Zaplaceno:
1	1.1.1998	1000.00	15.1.1998
3	2.1.1998	2656.00	25.1.1998
9	27.2.1998	125.70	3.3.1998
18	20.6.1998	1000.00	25.6.1998

Z tohoto prostředí je možné i přidávat faktury - pokud v subformuláři stisknete klávesu **Insert**, nový záznam po uložení automaticky získá ve sloupci FIRMA číslo vybrané firmy.

### Formulář na propojení od FAKTURY k FIRMA

Pro údaj o firmě má tabulka FAKTURY vyhrazeno místo pro jedno celé číslo (sloupec FIRMA typu Integer). Protože není možné, abyste si pamatovali pro každou firmu její číslo, musí existovat způsob, kdy výběrem jména firmy ze seznamu firem (nejlépe setříděného) se správná číselná hodnota zapíše na správné místo do tabulky FAKTURY.

Ve **WinBase602** existují dva způsoby, jak naznačeného záměru dosáhnout:

1. Pro sloupec FIRMA vytvořit ne editační složku, nýbrž složku combo. Je nutné zvolit combo překládané, které nabízí něco jiného (seznam firem) než co zapisuje do sloupce (číslo firmy). Složka combo lze naplnit i údaji z jiné tabulky, než do které vede formulář, v němž je použita. V našem příkladě bude combo naplněno hodnotami dotazu:

```
select nazev, cislo from firmy order by nazev .
```

Toto řešení je pro uživatele rychlé a pohodlné, je však omezeno na 1000 položek v combu. Má-li tabulka (číselník) s údaji více položek, musíte použít způsob druhý.

#### Vložení hodnoty pomocí comba

Číslo faktury:	Datum vystavení:	Firma	Částka Kč:	Zaplaceno dne:
21	28.7.1998	AFFEX, spol. s r.o.	1210.00	
22	19.11.1998	AFFEX, spol. s r.o.	5000.00	
23	20.12.1998	Kancelářské potřeby Praha	6541.00	
24	31.12.1998	Kancelářské potřeby Praha	6500.00	
		Brněnská drátovna a.s.		
		Hutě Svidník		
		Kancelářské potřeby Praha		
		Nerez a.s.		
		Papírnictví Horváth		
		Potřeby pro hospodyňky a.s.		
		Spojené šroubárny Kolín		
		SuperOcel s.r.o.		
		Velkoobchod Extaze		

2. Vytvořte nový formulář do tabulky FIRMY (nebo do setříděného dotazu). Ve formuláři do tabulky FAKTURY si vytvořte tlačítko, kterému přiřadíte akci **Otevřít relační formulář** a doplňte jméno formuláře do FIRMY a jméno relace **Fakt\_firm**.

Potřebujete-li záznamu v tabulce FAKTURY přiřadit firmu, které se záznam týká, stisknete v něm tlačítko a otevře se formulář od druhé tabulky (číselníku) s označovacími tlačítky na levé straně. Jeden záznam označíte a zavřete formulář. Hodnota z označeného záznamu se přenesou do záznamu v původním formuláři.

#### Vložení hodnoty pomocí relačního formuláře

ČÍSLO	NAZEV	ULICE	MESTO
3	AFFEX, spol. s r.o.	Lomená 4	Bratislava
1	Brněnská drátovna a.s.	Dlouhá 45	Brno 1
10	Hutě Svidník	nám. SNP 4	Svidník
2	Kancelářské potřeby Praha	Rainerova 4	Praha 9
5	Nerez a.s.	Křivá 8	Praha 4
7	Papírnictví Horváth	Štefánikovo nám. 5	Zvolen
9	Potřeby pro hospodyňky a.s.	K lomu 3	Praha 4
4	Spojené šroubárny Kolín	Hlavní 44	Kolín
6	SuperOcel s.r.o.	U hutí 2	Kladno

Tento způsob není omezen v počtu záznamů v druhé tabulce. Formulář do faktur lze ještě vylepšit vložím relačně propojeného subformuláře, který bude obsahovat jediný záznam s jedním údajem - jménem vybrané firmy.

## Vztah M:N

Pro vyjádření tohoto vztahu pomocí nástrojů relační algebry potřebujeme kromě tabulek TA a TB ještě třetí *spojovací* tabulku, kterou nazveme TSPOJ. TSPOJ musí obsahovat nejméně dva sloupce SPOJA a SPOJB. SPOJA je stejného typu jako sloupec A, SPOJB jako B. Dva záznamy v tabulkách TA a TB jsou propojeny právě tehdy, pokud existuje záznam v tabulce TSPOJ takový, že  $A=SPOJA$  a zároveň  $B=SPOJB$ .

### Vytváření a rušení vazeb

Vazby mezi záznamy v TA a TB se vytvářejí tak, že se do tabulky TSPOJ vkládají nové záznamy, a ruší se tak, že se v TSPOJ ruší záznamy. Na rozdíl od vztahu 1:N se tedy při vytváření a rušení vazeb *nepřepisují* údaje v TA nebo TB.

Na vazby mezi záznamy lze hledět buď globálně, anebo od jedné tabulky směrem k druhé. Formuláře od TA k TB a od TB k TA jsou zcela symetrické, proto je nebudeme popisovat zvlášť.

## Spojení tabulek

Pomocí dotazu `SELECT * FROM TA, TSPOJ, TB WHERE TA.A=TSPOJ.SPOJA and TSPOJ.SPOJB=TB.B` lze získat celkový přehled o všech vazbách mezi záznamy. V formuláři do odpovědi na tento dotaz:

- lze klást dodatečné omezující podmínky, které vyberou z tabulek pouze určité záznamy;
- lze editovat hodnoty, avšak pokud editací změníte propojení záznamů, projeví se to až při novém položení dotazu;
- nelze vkládat ani rušit záznamy.

## Formulář od jedné tabulky k druhé

### Subformulář s připojenými záznamy

Při pohledu od jedné tabulky k druhé vidíme v hlavním formuláři záznamy z jedné tabulky a každý z těchto záznamů má v subformuláři připojené záznamy z druhé tabulky. Situace vypadá podobně, jako při vazbě 1:N, avšak pro různé záznamy hlavního formuláře se v subformulářích mohou opakovat stejné záznamy.

Realizace subformuláře je poněkud složitější, než v případě 1:N. Subformulář totiž vede do spojení dvou tabulek. Předpokládejme, že hlavní formulář vede do TA. Pak vytvoříte dotaz `SELECT * FROM TSPOJ, TB WHERE TSPOJ.SPOJB=TB.B` a subformulář vytvoříte nad tímto dotazem. Při navrhování hlavního formuláře v popisu složky Subpohled zadáte jako propojovací hodnotu sloupec A a jako výběrovou podmínku sloupec `TSPOJ.SPOJA` resp. výraz `TSPOJ.SPOJA=%`.

### Relační formulář pro editaci vazeb

Do subformuláře nelze vkládat záznamy ani rušit je, lze ovšem přepisovat hodnoty v záznamech. Editace propojení mezi záznamy z TA a TB se děje pomocí speciálního relačního formuláře, který otevřete stiskem tlačítka. V tomto relačním formuláři jsou za-

tržené ty záznamy, které jsou připojeny k vybranému záznamu v hlavním formuláři (tj. ty, které vidíte v subformuláři). Kliknutím na zatrhávací tlačítka (nebo stiskem klávesy **F8**) můžete vytvářet a rušit vazby. Lze zde také vkládat záznamy do TB a rušit je. Na rozdíl od relačního formuláře při vazbě 1:N zde může být současně zatrženo více než jeden záznam.

Při zadávání parametrů akce, kterou se otevírá relační formulář, je nutno vyplnit dvě relace: jméno relace od TA k TSPOJ a jméno relace od TSPOJ k TB.

#### Další sloupce spojení

Tabulka TSPOJ může mít i další sloupce, které doplňují informace o vazbě mezi TA a TB. Tyto sloupce lze editovat v subformuláři. Například pokud TA je tabulka pacientů, TB tabulka léků a tabulka TSPOJ vyjadřuje vztah "pacientu byl předepsán lék", pak další sloupec tabulky TSPOJ může obsahovat datum předepsání a počet předepsaných balení.

#### Pozor:

Propojování relačním formulářem se nehodí pro případ, kdy mezi záznamy z TA a TB lze vytvářet multivazby (vícenásobné vazby - jeden pacient dostal tentýž lék vícekrát). Stiskem propojovacího tlačítka v relačním formuláři by se nedala vytvořit druhá a další vazba ani zrušit jedna z mnoha vazeb. Zobrazení připojených záznamů v subformuláři však funguje i v tomto případě: záznamy vícenásobně připojené se zobrazí vícekrát, po každé s jinými hodnotami dodatečných sloupců z TSPOJ.

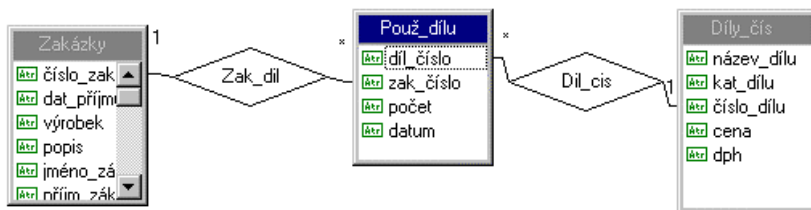
Relační formulář nelze otevřít na fiktivním záznamu v hlavním formuláři. Propojováním na fiktivním záznamu by totiž došlo k porušení referenční integrity databáze.

## Příklad vztahu M : N

Relační vztah N : M budeme demonstrovat na vztahu mezi zakázkou z tabulky ZAKÁZKY a náhradními díly z tabulky DÍLY\_ČÍS. Pro každou zakázku může být použito různých dílů a každý díl může být použit v různých zakázkách. Propojovací tabulka se nazývá POUŽ\_DÍLU. V tabulce ZAKÁZKY je sloupec ČÍSLO\_ZAK, v tabulce DÍLY\_ČÍS je sloupec ČÍSLO\_DÍLU, v tabulce POUŽ\_DÍLU jsou sloupce ZAK\_ČÍSLO a DÍL\_ČÍSLO.

Vztahy mezi tabulkami jsou znázorněny na obrázku.

#### Vztahy mezi tabulkami



#### Pozor!

Relační propojení definovaná ve schématu aplikace jsou v případě vazeb N:M **nezbytná** a nelze je obejít, mají hlubší význam než pouze grafické znázornění. Jména relací



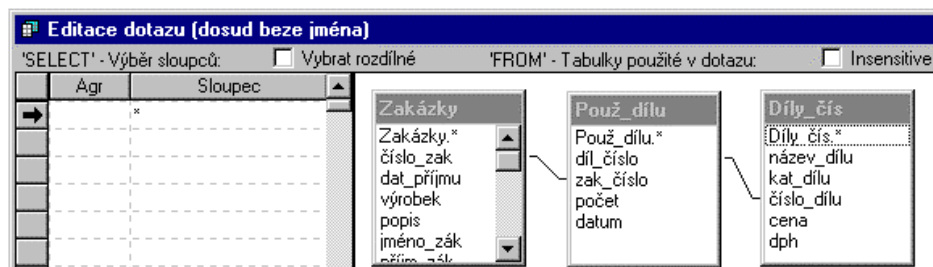
(ZAK\_DIL a DIL\_CIS) použijte později k definování relačního formuláře pro vkládání vazeb mezi tabulkami ZAKÁZKY a DIL\_ČÍSLO.

Tabulky jsou použity z aplikace SERVIS.

## Spojení tabulek

Pro zobrazení a prohlížení údajů o zakázce i použitých dílech je nutné vytvořit dotaz ze všech tří propojených tabulek. Počátek návrhu bude vypadat jako na obrázku.

### Propojení tabulek



V formuláři otevřeném do tohoto dotazu je vidět použití každého dílu v každé zakázce, ale není v něm možno vkládat ani rušit záznamy, natož pak přidávat nebo odstraňovat některé vazby.

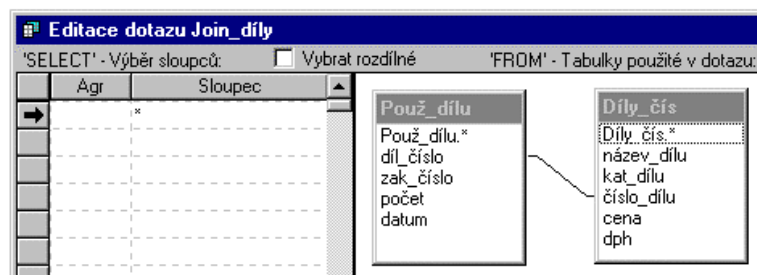
## Formulář od jedné tabulky k druhé

V příkladu použijeme pouze jednu polovinu problému - bude nás zajímat, které díly jsou použity v jedné zakázce. Opačný postup (jaké zakázky využívají jeden konkrétní díl) je zcela symetrický.

### Zobrazení

Abychom si mohli prohlížet zakázky a na ně použité díly, vytvoříme formulář se subformulářem. Hlavní formulář (superformulář) povede do tabulky zakázek, subformulář, v němž budou vidět použité díly musí vést do dotazu vytvořeného z obou zbývajících tabulek. Dotaz nazveme JOIN\_DÍLY a jeho definice bude podobná té na obrázku.

### Dotaz pro subformulář



Dále je nutno vytvořit standardní formulář do tohoto dotazu a obecný formulář do tabulky zakázek, jehož jednou složkou bude právě vytvořený subformulář. V parametrech složky subformulář uvedete definici relačního propojení:

propojovací hodnota : ČÍSLO\_ZAK

výběrová podmínka : ZAK\_ČÍSLO

Tím je splněna první část úlohy: ve formuláři vidíte vždy jednu zakázku a v jeho subformuláři seznam použitých náhradních dílů.

#### Superformulář se subformulářem

Název_dílu	Počet	Datum
pérko do hodinek	1	26.2.1999
ciferník	1	26.2.1999
baterie 1.2 V VARETA	1	26.2.1999

#### Vkládání vazeb

Druhou částí úlohy je umožnit vkládání nových a rušení starých vazeb mezi zakázkou a použitými díly.

Na rozdíl od prohlížení vazeb, kde mají smysl oba směry nazírání na problém (jaké díly používá jedna zakázka a jaké zakázky využívají jeden díl), při vkládání vystačíme se směrem jediným. Z logiky úlohy totiž vyplývá, že stačí, když budeme k jedné označené zakázce připojovat náhradní díly.

Řešení tohoto problému nabízí opět **relační formulář**. Vytvořte tedy formulář do tabulky DÍLY\_ČÍS (se jménem např. FDÍLY). Tento formulář se později použije jako formulář relační a pomocí něho se budou vazby vkládat i rušit.

Relační formulář do tabulky dílů je vhodné otevřít např. z formuláře z předchozího odstavce - přehledu dílů jedné zakázky. V tomto formuláři je nutno vytvořit tlačítko s definovanou akcí **Otevřít relační formulář** a vyplnit správně parametry - jméno otevíraného formuláře (FDÍLY) jména obou relací. Jako první se uvede jméno relace mezi tabulkami ZAKÁZKY a POUŽ\_DÍLU (ZAK\_DÍL) a jako druhá relace mezi tabulkami POUŽ\_DÍLU a DÍLY\_ČÍS (DIL\_CIS).

## Akce na tlačítku

**Posloupnost akcí a jejich parametry**

Posloupnost akcí: Akce číslo 1: Vložit standardní akci

Otevřít relační pohled Fdíly, relace Zak\_díl, druhá relace Díl\_cis

-- Konec posloupnosti akcí --

Up Dn

Vložit příkazy

Odstranit akci

Bez akce

Otvírání a zavírání pohledu

- Otevřít pohled

- Zavřít pohled

- Otevřít nebo zavřít pohled

- Na stejný záznam otevřít pohled

- Otevřít modální pohled

- Na stejný záznam otevřít modální pohled

- Otevřít relační pohled

- Otevřít relačně synchronizovaný pohled

- Otevřít vázací pohled

- Vložit záznam a otevřít na něm pohled

Vybrat akci

Jméno pohledu: Fdíly

Text zprávy:

Sloupec:

První relace: Zak\_díl

Druhá relace: Díl\_cis

OK Storno nápověda

Takto definovaná akce na tlačítku otevře formulář FDÍLY tak, že v jeho levé části se objeví sloupec přepínacích tlačítek, která známým způsobem signalizují vytvořené vazby: modrá "fajka" - vazba existuje, červený křížek - vazba neexistuje. Stiskem tlačítka (myší nebo klávesou **F8**) se vytvoří nová vazba, tzn. vloží se záznam do spojovací tabulky POUŽ\_DÍLU s příslušnými hodnotami spojovacích sloupců (číslem zakázky a číslem použitého dílu).

## Relační formulář

**Výběr dílů použitých při práci na zakázce**

Hotovo

✓	1	pérko do hodinek	budík
✗	2	klička náhradní	budík
✓	3	sklo	budík
✗	4	cierník	budík
✗	5	baterie 1.2 V VARETA	baterky
✗	6	baterie 1.5 V SUPERHEAT	baterky

## Ukazatele a přivazování záznamů

Mezi záznamy v databázových tabulkách mohou existovat vazby vytvořené pomocí ukazatelů. Sloupec typu ukazatel v jednom záznamu může ukazovat na jiný záznam ve stejné nebo v jiné tabulce. Multiatribut typu ukazatel může ukazovat na množinu záznamů.

Vytváření těchto vazeb nazýváme PŘIVAZOVÁNÍM ZÁZNAMŮ, jejich rušení ODVAZOVÁNÍM ZÁZNAMŮ. Přivazování a odvazování není ničím jiným než zapisováním vhodných hodnot do sloupců typu ukazatel.

Případy, řešené ukazatelovými vazbami, lze ve většině případů nahradit obecnějším a snáze přenositelným propojením relačním. V nových verzích **WinBase602** je práce s relačními vazbami stejně uživatelsky pohodlná jako práce s ukazateli.

Tabulky s ukazateli nejde exportovat v jiném než interním formátu, tyto tabulky nelze replikovat, nelze s nimi pracovat pomocí SQL, nelze se připojit na ně přes ODBC, nelze je zkomprimovat.

### Jak ukazatele fungují

Ve **WinBase602** jsou k dispozici jednosměrné a obousměrné ukazatele. Je-li v tabulce TA sloupec typu obousměrný ukazatel do tabulky TB, pak v tabulce TB musí být sloupec typu obousměrný ukazatel do tabulky TA. Každá změna jednoho z těchto ukazatelů vyvolá automatickou změnu protisměrného ukazatele. **WinBase602** sama zajistí, že kdykoli ukazatel v záznamu X z tabulky TA ukazuje na záznam Y v tabulce TB, pak také ukazatel v záznamu Y v tabulce TB ukazuje na záznam X z tabulky TA. U obousměrných ukazatelů stačí tedy vkládat do databáze údaje o propojení záznamů jedním směrem a automaticky získáte obousměrné propojení.

Pozor:

Při použití obousměrných ukazatelů je nebytné, aby buď *oba* ukazatele byly multiatributy, anebo *žádný* z nich. To platí i tehdy, když je vztah mezi tabulkami typu 1:N.

### Vázací formuláře

Vytváření ukazatelových vazeb neboli editace hodnot ukazatelů se děje v formulářích prostřednictvím tzv. VÁZACÍCH FORMULÁŘŮ. Předpokládejme, že máme formulář obsahující záznamy z tabulky TA, a že TA má mezi svými sloupci ukazatel do tabulky TB. Součástí formuláře do TA je tlačítko, kterým se dá otevřít formulář do tabulky TB. Předpokládejme, že tlačítkem v záznamu X z tabulky TA takový formulář otevřete. Zjistíte, že

má jednu zvláštnost: v levé části každého záznamu je přepínač, kterým se dá kterýkoli záznam vybrat, nebo vybrání zrušit. Zapnutím tohoto přepínače v záznamu Y vytvoříte ukazatelovou vazbu ze záznamu X do záznamu Y. Pokud je ukazatel obousměrný, současně vznikne vazba z Y do X.

**Jsou-li ukazatele nebo obousměrné ukazatele multiatributy, pak ve vázacím formuláři lze současně vybrat víc záznamů (zapnout více než jeden přepínač). V opačném případě lze ke každému záznamu přivázat jediný záznam.**

Je-li ve vázacím formuláři mnoho záznamů, použijte pro vybrání toho, který chcete přivázat, možnost položit dotaz nebo záznamy uspořádat.

Vázací formulář uzavřete stejně jako každý jiný formulář. Dokud je vázací formulář otevřen, nelze pracovat s obsahem formuláře, z něhož byl vázací formulář otevřen.

V průběhu práce s vázacím formulářem je na stavové řádce údaj o počtu přivázaných záznamů.

## Odkazy na zrušené záznamy

**Rušením záznamů nezanikají ukazatelové vazby mezi nimi. Proto se snadno může stát, že záznam ukazuje na zrušený záznam. Takové ukazatele mohou být (a zpravidla jsou) nežádoucí. Například při použití multiukazatelů odkazy na zrušené záznamy zbytečně zabírají místo a překážejí při práci.**

Pokud je v formuláři složka, jejíž hodnota se získává prostřednictvím ukazatele na zrušený záznam, pak se v této složce objeví text **Odkaz na zrušený záznam** (resp. jeho začátek, pokud se tam nevejde celý).

Odkazy na zrušené záznamy se nadají odstranit pomocí běžných vázacích formulářů, protože v nich zpravidla zrušené záznamy nejsou vidět, a tudíž se nedají odvázat.



Pro odvazování zrušených záznamů existuje zvláštní funkce, kterou spustíte pomocí ovládacího tlačítka **Odvázat zrušené** nebo klávesovým повеlem **Ctrl+F8**. Akce je přístupná pouze ve vázacím formuláři. Na obrazovce se objeví okno, které vám nabídne odvázaní zrušených záznamů buď pouze od toho záznamu z prvního formuláře, na němž jste otevřeli vázací formulář, anebo od všech záznamů z prvního formuláře; vyberte si a stiskněte některé tlačítko v tomto okně. Odvázání zrušených záznamů od všech záznamů rozsáhlé tabulky může chvíli trvat.

