



Tvorba aplikácií v 602SQL 8.0 jednoducho a rýchlo

Copyright © 2003 [Roman Riedl](#)
Copyright © 2003 [Software602 a.s.](#)

Verzia 1.06, 02/2003

OBSAH

ÚVOD	4
1. VLASTNOSTI DATABÁZOVÉHO SYSTÉMU 602SQL	5
1.1 Z HISTÓRIE	5
1.2 VLASTNOSTI SQL SERVERA	5
1.3 VLASTNOSTI KLIENTSKÉHO A VÝVOJOVÉHO PROSTREDIA	6
1.4 MOŽNOSTI PRIPOJENIA KLIENTOV NA SQL SERVER	6
2. INŠTALÁCIA A REGISTRÁCIA 602SQL	8
2.1 DOWNLOAD	8
2.2 INŠTALÁCIA VO WINDOWS	8
2.3 INŠTALÁCIA NA LINUXE	8
2.4 REGISTRÁCIA	9
2.5 INŠTALÁCIA APLIKÁCIÍ	10
3. ZÁKLADNÝ POPIS PROSTREDIA 602SQL	11
3.1 RIADIACA KONZOLA	11
3.2 PRVÉ SPUSTENIE KLIENTA	11
3.3 VYTVORENIE APLIKÁCIE	13
4. DATABÁZOVÉ TABUĽKY	14
4.1 VYTVÁRANIE TABUĽIEK	14
4.2 PODPORA ČEŠTINY A SLOVENČINY	17
4.3 RELAČNÉ VZŤAHY	18
4.4 IMPORT DÁT DO TABUĽIEK	19
5. NÁVRH DOTAZOV	20
5.1 DEFINÍCIA DOTAZU	20
5.2 VYTVORENIE DOTAZOV	20
6. NÁVRH FORMULÁROV	22
6.1 DEFINÍCIA FORMULÁRA	22
6.2 VYTVORENIE FORMULÁROV	22
6.3 FORMULÁRE A VYHĽADÁVANIE	24
6.4 POUŽITIE DOTAZOV Z VIACERÝCH TABUĽIEK	25
6.5 RELÁCIE VO FORMULÁROCH	26
7. MENU	28
7.1 MOŽNOSTI MENU A JEHO NÁVRH	28
8. PROGRAMY V 602SQL	29
8.1 PROGRAM V INTERNOM JAZYKU	29
8.2 ŠTRUKTÚRA PROGRAMU	29
8.3 PROCEDÚRY V INTERNOM JAZYKU	30
9. PROGRAMY ULOŽENÉ NA SERVERI	33
9.1 TEÓRIA	33

9.2 PROCEDÚRY ULOŽENÉ NA SERVERI	33
9.3 TRIGGERY	34
9.4 PROCEDÚRA VERSUS TRIGGER	34
9.5 PRÍKLAD PROCEDÚRY	35
9.6 SEKVENCIE	35
10. INTERNET KLIENTI	37
10.1 INŠTALÁCIA APACHE	37
10.2 DYNAMICKÁ WWW STRÁNKA	38
10.3 VYTVORENIE WWW OBJEKTU VO FORMWIZARDE	39
10.4 ÚPRAVA AUTOMATICKY VYTVORENÉHO OBJEKTU	40
10.5 SPOLUPRÁCA S PHP	42
11. ZDIEĽANIE DÁT MEDZI 602SQL A INÝMI SYSTÉMAMI	43
11.1 ODBC	43
11.2 ĎALŠIE MOŽNOSTI PRÍSTUPU NA SQL SERVER	43
12. FULLTEXTOVÉ VYHĽADÁVANIE	44
12.1 PODMIENKY POUŽITIA	44
12.2 APLIKOVANIE FULLTEXTU	44
13. BEZPEČNOSŤ	47
13.1 SYSTÉM PRIDEĽOVANIA PRÁV	47
13.2 ZABEZPEČENIE SERVERA	48
14. POROVNANIE 602SQL S INÝMI SQL SERVERAMI	50
ZÁVER	51

ÚVOD

Spomínate si ešte na editor Text602 od Software602? Presne tá istá firma rozšírila približne v roku 1992 portfólio svojich produktov o WinBase602. Kedysi jednoduchá databáza sa do svojej aktuálnej verzie 8.0 (keď už nesie názov 602SQL) rozrástla na výkonný databázový systém vybudovaný na jazyku SQL, i keď v jej začiatkoch by ste v manuáloch slovo SQL dlho hľadali.

602SQL v sebe obsahuje samotný SQL server a tiež nástroje na vytváranie aplikácií a to všetko v jednom balíku o veľkosti menej ako 10MB. To je vlastnosť, ktorá je v obdobných databázových systémoch ojedinelá. Jednoduchšie aplikácie je možné vyvíjať bez znalosti programovania – „len klikaním myšou“. Pre zložitejšie aplikácie sa používa integrovaný jazyk podobný Pascalu a ďalšie nástroje.

Ešte v úvode treba spomenúť jeden dôležitý fakt: za 602SQL sa platí, je to komerčný produkt. Licenčná politika je ale pre používateľov pozitívna – platí sa len za licencie, jednopoužívateľská verzia je zdarma – takže nič vám nebráni vytvoriť si akúkoľvek aplikáciu a licencie dokúpiť len ak ju budete potrebovať nasadiť do sieťovej prevádzky. Počet inštalácií klientov pripájajúcich sa na server nie je nijak obmedzený, počítajú sa len súčasne pripojení klienti. Za zmienku stojí aj fakt, že 5-používateľská licencia vrátane všetkých rozširujúcich funkcií je k dispozícii na dobu 30 dní od registrácie, fulltextové funkcie na dobu 90 dní.

Cieľom tejto brožúry je poskytnúť všetkým začínajúcim resp. mierne pokročilým tvorcom databázových aplikácií jednoduchý návod na vytvorenie konkrétnej vzorovej aplikácie. Pri čítaní textu so súčasným vytváraním jednotlivých položiek aplikácie podľa návodu získate prehľad v databázovom systéme 602SQL a zručnosť, ktorá vám umožní pustiť sa do zložitejších projektov. V texte budete upozorňovaní na podobnosti alebo odlišnosti v porovnaní s inými databázami, najčastejšie to bude s MS Access a MySQL.

Brožúra je členená do niekoľkých častí. Na začiatku budete oboznámení so základnými vlastnosťami 602SQL a s jej inštaláciou. Nasledovať bude podrobný návod na vytváranie jednotlivých typov objektov aplikácie aj s vysvetlením princípov. Návod bude rozčlenený tak, aby boli rozobrané všetky najčastejšie sa opakujúce činnosti tvorcov aplikácií a viedol vás čitateľov ku kompletnému vytvoreniu vzorovej aplikácie. Priebežne budete upozorňovaní na ďalšie nadväzujúce funkcie. Na záver si prečítate porovnanie vlastností 602SQL s niektorými podobnými SQL serverami.

Aby ste mohli konfrontovať priebeh tvorby vzorovej aplikácie a ľahšie odstrániť prípadné preklepy, nájdete vytvárané objekty podľa postupu ich tvorby v priložených súboroch, komprimovaných vo formáte ZIP. Na existenciu príloh budete upozorňovaní v texte. V prílohe sa zároveň nachádza aj celá aplikácia.

1. VLASTNOSTI DATABÁZOVÉHO SYSTÉMU 602SQL

1.1 Z HISTÓRIE

Systém WinBase602 bol predstavený vo verzii 1.0 v roku 1992. Designom vývojového prostredia pripomína MS Access, i keď na trh prišiel o niečo skôr. Na rozdiel od Accessu bol od začiatku vyvíjaný na princípe klient-server. Jazyk servera sa síce v prvých verziách podobal na SQL, ale nebol s ním plne kompatibilný. Autori si postupne uvedomovali potrebu kompatibility s SQL, preto bol každou verziou s týmto štandardom zblížovaný. Tento proces trval až do verzie 5.1.

Od verzie 1.0 až do verzie 4.0 bol určený pre 16-bitové Windows 3.x. Verzie 4.32 bola funkčne identická so 4.0, ale skompilovaná už len pre 32 bitové Windows. Vo verzii 8.0 došlo k premenovaniu produktu na 602SQL. História verzií a ich uvádzania na trh bola takáto:

1.0	1992	5.0	1997
2.0	1992	5.1	1998
2.1	1993	6.0	1999
3.0	1994	6.1	2000
3.1	1995	7.0	2000
4.0 a 4.32	1996	8.0	2002

Spolu s rozširovaním funkcií sa rozširoval aj počet platforiem servera. Kým klientské a vývojové prostredie bolo vždy určené pre platformu Windows, server bol od verzie 4.0 až do verzie 6.1 dodávaný aj pre Novell Netware. Verziou 6.0 sa začala písať história Linuxového servera.

1.2 VLASTNOSTI SQL SERVERA

- SQL server podľa normy SQL2 (92) s niektorými prvkami podľa normy SQL3/PSM (99), napr. triggery a procedúry uložené na serveri. Zahrnutá je aj podpora transakcií.
- Systém pridelovania práv na 3 úrovniach (používateľom, skupinám a roliam). Práva k dátam je možné pridelovať po stĺpcoch aj po riadkoch tabuľky. Tri skupiny správcov – prevádzkový, bezpečnostný a správca databázy.
- Digitálne podpisy, kompatibilné s X.509.
- Šifrovanie sieťovej komunikácie klientov so serverom.
- Overenie identity servera v sieti.
- Možnosť prevziať mená používateľov z domény pre účely automatického prihlasovania – pre server na platforme Windows.
- Server môže pracovať ako aplikácia alebo ako služba (resp. démon v Linuxe). Na jednom počítači môže bežať viac SQL serverov, každý nad inou databázou.
- Plná podpora česko-slovenského, nemeckého, anglického, francúzskeho a poľského triedenia a vyhľadávania.
- Podpora UNICODE.

- Spojenie klientov so serverom pomocou protokolov TCP/IP, NetBEUI a IPX/SPX.
- Replikácie medzi serverami v prípade ich priameho spojenia aj v prípade nepriameho spojenia cez poštové protokoly SMTP/POP3.
- Pravidelné zálohovanie databázy, žurnál zmien a istenie transakcií.
- Podpora pre odosielanie a prijímanie e-mailov.
- Fulltextové vyhľadávanie v textoch použiteľné pre všetky podporované jazyky, vrátane vyhľadávania výskytov hľadaného slova bez ohľadu na jeho zadaný gramatický tvar.

1.3 VLASTNOSTI KLIENTSKÉHO A VÝVOJOVÉHO PROSTREDIA

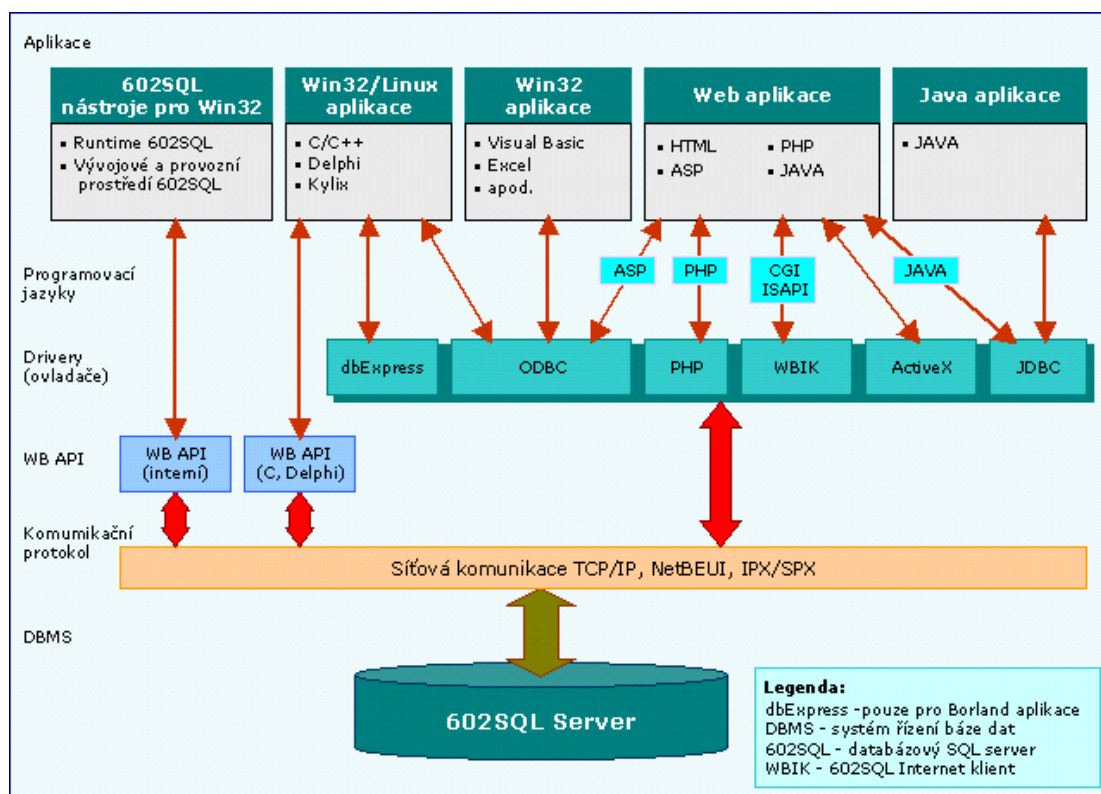
- Univerzálne prostredie klienta slúžiace na vývoj aplikácií pre Windows aj na ich prevádzkovanie.
- Možnosť vytvorenia runtime pre distribúciu hotových aplikácií.
- Riadiaci panel ako základný ovládací prvok slúžiaci na vytváranie objektov databázy.
- Interaktívne návrháre objektov tvoriacich dátový základ aplikácií (napr. tabuľky, dotazy a procedúry na serveri) a aj používateľského rozhrania (formuláre a zostavy, programy, menu, obrázky do pozadia atď).
- Vývojové a ladiace nástroje pre programy interného jazyka aj jazyka procedúr na serveri.
- Možnosť syntaktickej kontroly všetkých objektov aplikácií.
- Integrovaný textový editor so syntaktickým farebným odlišovaním a kontextovou nápovedou.

1.4 MOŽNOSTI PRIPOJENIA KLIENTOV NA SQL SERVER

- Aplikácie s priamym prístupom vytvorené vo vývojovom prostredí 602SQL. Len pre systém Windows.
- ODBC driver pre nepriame napojenie, existuje pre aplikácie vo Windows a Linux.
- DbExpress driver – umožňuje priame napojenie pre aplikácie vyvinuté v Borland Delphi (Windows) alebo Kylix (Linux).
- Internet klienti pre prístup do databázy z webového prehliadača, napríklad cez formuláre alebo dynamicky generované HTML stránky. Ich spojenie s webserverom je možné pomocou CGI, ISAPI alebo FastCGI. Na klientskej strane je to riešenie úplne nezávislé od typu operačného systému, potrebný je len webový prehliadač.
- PHP modul – umožňuje klientom systémov Windows a Linux pristupovať k dátam z jazyka PHP vloženého do HTML stránok.
- ActiveX – prostredníctvom pohľadu ActiveX umiestnenom v HTML stránke môžu klienti Windows pristupovať k dátam úplne rovnakým spôsobom, ako vo formulároch vytvorených vývojovým prostredím 602SQL.
- JDBC driver – na operačnom systéme nezávislý klient pre aplikácie napísané v jazyku Java.

- ODMA driver – server 602SQL môže fungovať ako sklad dokumentov a klientské programy Windows si môžu ukladať dokumenty do tohto skladu. Vyhľadávanie dokumentov na základe ich obsahu sa potom dá riešiť vyhľadávacími funkciami 602SQL.

Nasledujúca tabuľka je prevzatá z webu výrobcu a prehľadne ukazuje všetky možnosti pripojenia klientov na server 602SQL:



2. INŠTALÁCIA A REGISTRÁCIA 602SQL

2.1 DOWNLOAD

Aktuálnu verziu 602SQL si môžete stiahnuť na adrese <http://www.software602.cz/download/602sql8.htm>. Na uvedenej stránke je k dispozícii:

- Základný balík (server, klient, vývojové prostredie pre Windows)
- Extension Pack – doplnkové nástroje, príklady aplikácií a rôzne ovládače
- ODBC driver manažér
- SQL server, riadiaca konzola 602SQL-Xconsole pre Linux

Ďalšie súbory, aktualizácie a pod. nájdete na stránke určenej pre vývojárov <http://www.software602.cz/602sql>.

2.2 INŠTALÁCIA VO WINDOWS

602SQL môžete inštalovať na 32-bitových operačných systémoch Windows rady 9x (95, 98, Me) alebo rady NT (NT4, 2000, XP). Pre Windows95 a NT4 je potrebné mať nainštalované ďalšie rozšírenia, obsiahnuté napr. v inštalácii MS Internet Explorera verzie 4 a vyššej. Základnú inštaláciu zahájite spustením stiahnutého súboru 602sql8.exe. Počas inštalácie budete vyzvaní na vyplnenie niekoľkých údajov:

- meno
- firma
- cieľový adresár
- typ inštalácie (kompletná inštalácia, klient pre prístup na sieťový server, iba ODBC driver). Vyberte kompletnú inštaláciu
- cieľový adresár pre inštaláciu a pre databázu
- meno databázy, ak vytvárate novú (zadajte napr. „abc“)
- názov pre zástupcu v ponuke Štart - Programy
- možnosť spustenia riadiacej konzoly a databázy po ukončení inštalácie

Súbor 602sql8.exe je k dispozícii aj lokalizovaný do anglického jazyka s výnimkou súboru nápovedy, ktorý zatiaľ nie je preložený kompletne.

2.3 INŠTALÁCIA NA LINUXE

Pre Linux je k dispozícii server s ODBC driverom, riadiaca konzola „602SQL-Xconsole“ pre administráciu v grafickom prostredí Xfree86, doplnkové knižnice „602sql-libraries“ pre niektoré staršie distribúcie a zdrojový text jednoduchého klienta, v ktorom je možné zadávať SQL príkazy. Príklad inštalácie z .rpm balíčku v distribúcii RedHat:

```
rpm -i 602sql-8.0-0211121426.i386.rpm
rpm -i 602sql-Xconsole-8.0-0211121426.i386.rpm
ldconfig /usr/lib/602sql8
```


Aby X-console fungovala, je niekedy nutné použiť po inštalácii ešte „ldconfig“. Inštalovať je možné aj z grafického prostredia (napríklad KDE alebo GNOME) dvojklikom na .rpm balíček. Po inštalácii je možné spustiť „602inst8“ na vytvorenie databázy. Server sa spúšťa priamo cez „602sql8“ alebo ako démon príslušným nástrojom. Pri prvom spustení vypíše distribučné číslo a návod, ako ho zaregistrovať. Riadiaca konzola sa spúšťa príkazom „c602conf8“. Súčasťou linuxovej inštalácie je jednoduchý klient „602cli8“ pre vykonávanie SQL príkazov na otestovanie základných funkcií.

Tí, ktorí nepoužívajú RedHat alebo iné distribúcie podporujúce správu balíčkov .rpm (Mandrake, SUSE...), môžu použiť .tar.gz inštalačné balíčky.

2.4 REGISTRÁCIA

602SQL je v základnej jedno-používateľskej verzii zdarma. Je však nutné ju registrovať. Pre registráciu je potrebný prístup k internetu. Registrácia je prístupná na adrese <http://www.software602.cz/registrace/602sql.htm>. Na tejto stránke môžete zdarma zaregistrovať SQL server pre súčasný prístup jedného používateľa alebo uplatniť zakúpené distribučné číslo na Add-On licencie. Zaregistrovaním SQL servera dostanete možnosť používať 5-používateľské licencie po dobu 30 dní a fulltextové funkcie po dobu 90 dní.

Vlastnosti registračného procesu:

- Registruje sa iba server. Klienta ani vývojové prostredie nie je potrebné registrovať.
- Bez registrácie sa server odmietne spustiť.
- Registruje sa cez internet. Na základe inštalácie sa vygeneruje inštalačný kľúč. Ten sa vyplní vo formulári registračného portálu spolu s ďalšími údajmi. Na základe inštalačného kľúča bude vygenerované licenčné číslo, ktoré bude zaslané na Vašu e-mailovú adresu.
- Na identickej inštalácii Windows alebo Linux možno server registrovať iba raz. Po kompletnom preinštalovaní operačného systému je nutné registrovať znova.
- Add-On licencie je možné zdieľať v prípade, že na jednom počítači beží viac serverov

Pre aplikácie, ktoré budú prevádzkované na jedinom počítači teda stačí registrovať iba server zdarma. Tento stav vám bude stačiť aj pre vývoj sieťových aplikácií, server totiž nerozlišuje typ klienta, ktorý sa k nemu pripojuje (lokálny, sieťový, internet klient...), pre neho je rozhodujúci len počet súčasne pripojených klientov. Klienti pre priame pripojenie (integrovateľný klient, ODBC klient, dbExpress klient...) sú pripojení k serveru trvalo až do odhlásenia, internet klienti sa pripájajú len na čas vybavenia požiadavky od web servera (rádovo zlomky sekundy). S istými obmedzeniami (riziko vypísania hlásky o nedostatku licencií) je teda možné 602SQL zdarma používať aj v malých intranetových sieťach. Až keď pri postupnom zvyšovaní frekvencie používania presiahne počet takýchto výpadkov únosnú mieru, dokúpíte Add-On licencie (obvykle stačí 5). Druhé obmedzenie pre prevádzku zdarma je dané spomínanou 90 dňovou lehotou funkčnosti fulltextového vyhľadávania, ktoré sa natrvalo aktivuje až po zadaní akejkoľvek Add-On licencie.

2.5 INŠTALÁCIA APLIKÁCIÍ

Ešte pred začiatkom vývoja akejkoľvek aplikácie si musíte uvedomiť, akým spôsobom ju po dokončení budete šíriť. Najjednoduchší spôsob je taký, že ju ako autor vyexportujete do súborového tvaru. Takto vyexportovaná aplikácia sa potom priloží k inštalačnému súboru 602sql8.exe. Po inštalácii prostredia a vytvorení databázového súboru si používateľ aplikáciu nainportuje. Súčasťou aplikácie budú aj nástroje na správu databázy (napr. vytváranie používateľov, záloha, export/import dát atď).

Inštaláciu je možné aj zjednodušiť. Inštalačný balíček, t.j. súbor 602sql8.exe otvoríte napr. v programe WinZIP a extrahujete jeho obsah do vopred vytvoreného adresára. Úpravami obsahu súboru setup.ini, ktorý je jedným z týchto súborov je možné zautomatizovať proces inštalácie (obmedziť množstvo otázok inštalačného programu na minimum) a docieľiť, že sa bude súčasne s prostredím 602SQL importovať hotová aplikácia. Popis parametrov súboru setup.ini nájdete v elektronickej nápovede.

Importovanie aplikácie v niektorých prípadoch nemusí vyhovovať. Preto je k dispozícii aj ďalšia možnosť spočívajúca v tom, že priložíte k inštalačným súborom 602SQL celý databázový súbor wb8.fil. Tento spôsob má niekoľko výhod - v databázovom súbore môže byť viac aplikácií a taktiež dokážete jednoduchším spôsobom nastaviť vlastnosti databázy (zabezpečenie, preddefinovaných používateľov atď). Takto dokážete aj lepšie vyriešiť otázku ochrany vášho know-how pri distribúcii aplikácie.

Iná situácia vznikne, keď budete aplikácie šíriť pre sieťovú prevádzku. Spomínané inštalácie urobíte len na počítači, na ktorom bude bežať server 602SQL. Server môže byť spustený ako riadna aplikácia alebo ako služba (Windows na báze NT) resp. démon (Linux). Na ostatných počítačoch, na ktorých budú pracovať používatelia nainštalujete prevádzkové prostredie a zaregistrujete na nich meno počítača, kde beží 602SQL server. Aj inštaláciu prevádzkového prostredia (runtime) je možné zautomatizovať tak, aby nebolo potrebné odpovedať na žiadne otázky.

Ak šírite aplikácie vytvorené v jazyku internet klientov alebo PHP, inštalácia prevádzkových prostredí odpadá. Títo klienti potrebujú len prehliadač internetu, ktorý je už dlhú dobu štandardnou súčasťou moderných operačných systémov. V tomto prípade bude potrebné nainštalovať a nakonfigurovať web server a PHP na spoluprácu so 602SQL.

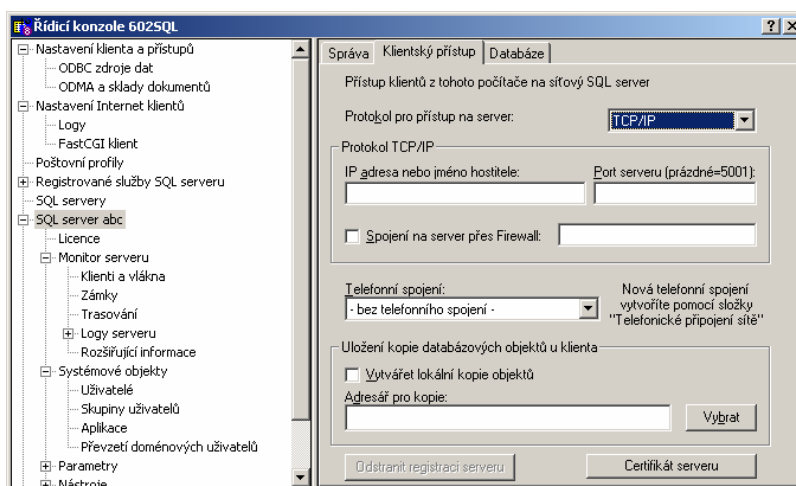
Špeciálnou možnosťou je inštalácia pre prácu s databázou umiestnenou na CD resp. DVD. 602SQL vie rozdeliť databázový súbor na dve časti – jedna časť bude umiestnená na CD-ROM, bude obsahovať základ databázy a nebude editovateľná. Druhá sa umiestni na lokálnom pevnom disku, do ktorej bude možné zapisovať zmeny. Typickým príkladom takéhoto využitia je distribúcia katalógov alebo cenníkov na médiách CD.

3. ZÁKLADNÝ POPIS PROSTREDIA 602SQL

3.1 RIADIACA KONZOLA

Riadiaca konzola je samostatná aplikácia. Má rovnaký vzhľad a funkcie pre Windows ako aj Linux (tam pracuje pod X). Jej pomocou možno kompletne administrovať databázový systém, vlastnosti vývojového prostredia a klienta pre Windows a vlastnosti servera.

Pomocou riadiacej konzoly je možné spravovať všetky činnosti správy servera, ktoré u starších verzií boli súčasťou klienta vrátane tých nastavení servera, ktoré bolo potrebné realizovať editáciou registrov Windows resp. konfiguračného súboru v /etc/602sql u Linuxu. Administrácia servera sa tým uskutočňuje v jedinej jednoduchej a prehľadnej aplikácii. Po inštalácii 602SQL už k počítaču so serverom netreba vôbec chodiť, resp. nie je potrebné používať ďalší software (napr. VNC, telnet a pod.), čo určite ocenia správcovia rozsiahlych sietí.



V ľavej časti je zoznam položiek, v pravej ich vlastnosti. Prvé položky v zozname sa týkajú lokálneho počítača:

- Nastavenie klienta a prístupov
- Nastavenie internet klientov
- Poštové profily
- Registrované služby SQL servera

Položka „SQL servery“ slúži na vytváranie lokálnych databáz resp. na registrovanie serverov v sieti, aby sa na ne mohol klient odkazovať. Pod ňou uvedené položky sa týkajú konkrétnych vlastností zaregistrovaných serverov, lokálnych aj sieťových a pri ich nastavovaní je potrebné sa prihlásiť.

3.2 PRVÉ SPUSTENIE KLIENTA

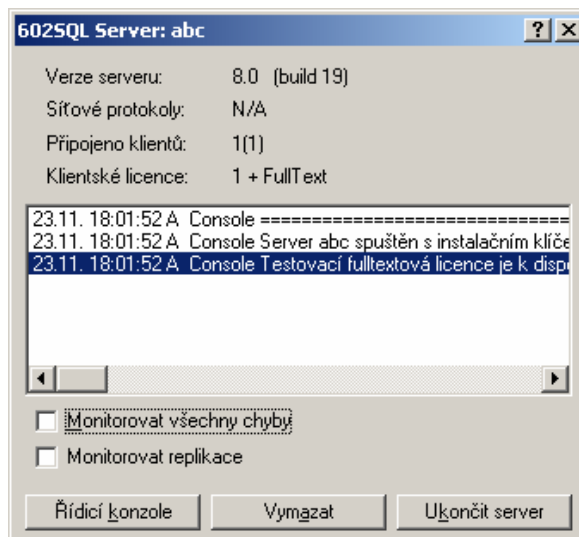
V menu „Štart – Programy“ sa pri inštalácii vytvorila skupina 602SQL. V nej sú nasledujúce odkazy:

- 602SQL
- 602SQL server
- Konfigurace 602SQL
- Náповěda 602SQL

Prvým odkazom sa spúšťa klientské a vývojové prostredie, druhým SQL server (nad databázou, ktorú ste vytvorili pri inštalácii). Ďalšie odkazy spúšajú riadiacu konzolu a súbor náповedy vo formáte Windows98 (.chm).

Čo si po spustení iste všimnete, je výzva na zadanie mena a hesla. Pri prvom spustení je v databáze definovaný len jeden používateľ „Anonymous“, ktorý má vždy prázdne heslo. Anonymous má práva administrátora databázy. Keď to bude potrebné, vytvoríte ďalších používateľov, pridáte im práva a používateľovi Anonymous ich zoberiete.

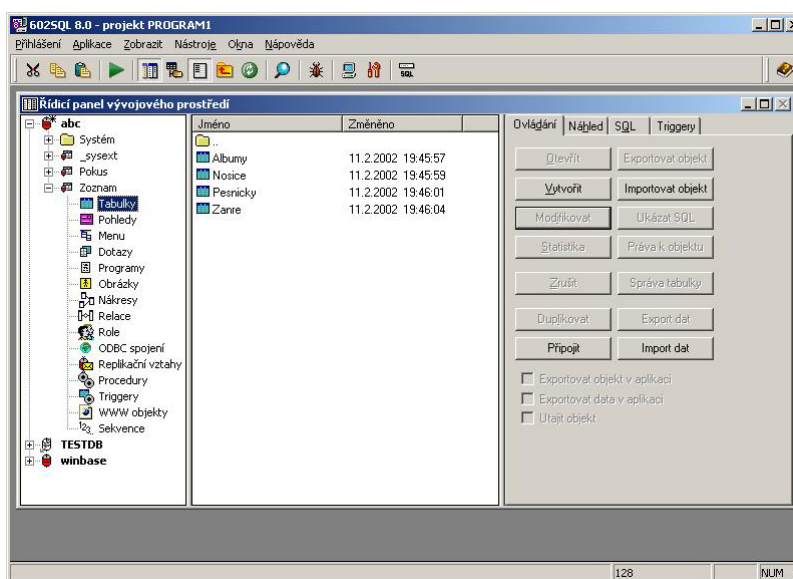
Ďalšia vec, ktorú si všimnete je, že na paneli úloh Windows vidíte spustené až dve aplikácie: 602SQL a 602SQL server. To je úplne v poriadku, veď klient aj server teraz bežia na rovnakom počítači. Do okna SQL servera sa vypisuje štandardný log.



Nastal čas popísať klienta a jeho vývojové prostredie. Celému prostrediu dominuje „Riadiaci panel vývojového prostredia“. V jeho ľavej časti vidíte zoznam registrovaných databáz. Nad každou databázou pracuje práve jeden 602SQL server. V každej databáze môže byť viac aplikácií. Vo Vašej databáze jednu aplikáciu už máte, volá sa „_sysext“, bola vytvorená automaticky pri prvom štarte a sú v nej umiestnené systémové procedúry. V strednej časti sa zobrazuje zoznam objektov podľa výberu z ľavej časti (tak, ako to poznáte z prieskumníka Windows) a v pravej časti je súbor funkčných tlačidiel pre označený objekt.

V každej databáze sa nachádza položka „Systém“, kde sa napríklad definujú používatelia a ich skupiny. Tiež sú tam prístupné všetky systémové tabuľky. 602SQL ako databázový systém sa na všetko pozerá ako na tabuľky. Preto tu napríklad nájdete tabuľku tabuliek „tabtab“, tabuľku všetkých objektov „objtab“, alebo aj tabuľku používateľov „usertab“.

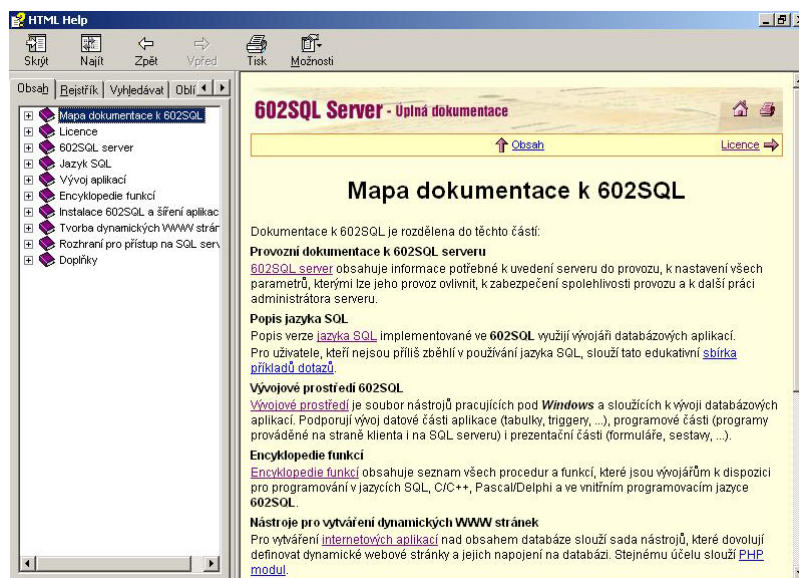
V každej aplikácii je možné vytvárať rôzne skupiny objektov. Sú to tabuľky, pohľady, menu, dotazy, programy, obrázky, nákresy, relácie, role, ODBC spojenia, replikačné vzťahy, procedúry, triggery, www



objekty a sekvencie. Všetko sa zobrazuje v riadiacom paneli. Riadiaci panel je týmto jednoduchý a prehľadný. Podobným spôsobom sa ovládajú aj iné databázy, napr. MS Access.

Databáza aj so všetkými jej objektmi je umiestnená v jedinom súbore wb8.fil, ktorý nájdete v adresári, kde ste 602SQL nainštalovali, pokiaľ ste pri inštalácii nezvolili iné umiestnenie. Aj tu nájdete podobu s Accessom, kde sa všetky údaje nachádzajú v súbore s príponou MDB.

Pokiaľ si v ktoromkoľvek štádiu práce neviete rady, stlačením klávesy F1 vyvoláte otvorenie súboru nápovedy. Nápoveda je kontextová, t.j. automaticky sa otvorí práve tá jej časť, ktorú potrebujete.



3.3 VYTVORENIE APLIKÁCIE

Ešte pred vytvorením aplikácie malá poznámka: vytvárať ale aj používať našu aplikáciu budeme pod kontom „Anonymous“, iné konto zatiaľ vytvárať nebudeme. Aké možnosti existujú pre aplikácie určené viacerým používateľom, si popíšeme neskôr.

Aplikácia v 602SQL sa vytvára stlačením tlačítka „Vytvoriť aplikáciu“ na riadiacom paneli vývojového prostredia. Aplikáciu pomenujte „Zoznam“. V mene aplikácie (a jej všetkých ďalších objektov) je možné použiť aj medzery a znaky s diakritikou, čo ale v niektorých prípadoch v spojení s inými programami (napr. web server) môže robiť problémy. Preto je vhodnejšie diakritiku v názvoch objektov nepoužívať.

4. DATABÁZOVÉ TABUĽKY

4.1 VYTVÁRANIE TABULIEK

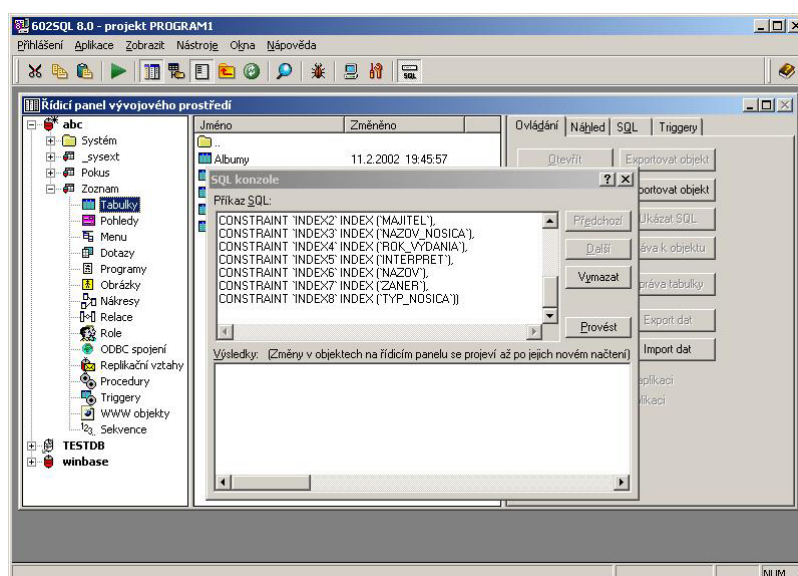
Toto je najdôležitejšia časť v návrhu každej relačnej databázy. Ak návrh tabuliek nezvládnete na prvý krát, budete mať neskôr veľmi veľa zbytočnej práce. Teória učí o piatich stupňoch tzv. normálnych foriem tabuliek. V jednoduchosti sa to dá vysvetliť tým, aby sa v navzájom súvisiacich relačne zviazaných tabuľkách údaje nevyskytovali viackrát a aby zaberali čo najmenej miesta. Niekedy - u zložitých databáz - sa toto nedá úplne dodržať a v záujme zrýchlenia dotazov sme nútení robiť aj kompromisy.

V našej aplikácii si budeme zapisovať všetky údaje o albume (názov, meno interpreta...) vrátane názvov pesničiek. Ďalej budeme evidovať o aký hudobný žáner ide a na akom nosiči máme album uložený (MG kazeta, CD MP3...). Tieto dáta by sme mohli dať do jednej jedinej tabuľky avšak poučení teóriou ich rozdelíme do viacerých. Po zrelej úvahe teda vytvoríme tieto 4 tabuľky:

- albumy - bude obsahovať zoznamy albumov s ich základnými údajmi (majiteľ albumu, názov, interpret, žáner, rok vydania, typ nosiča, názov nosiča)
- zanzre - bude obsahovať zoznam žánrov hudby
- nosice - bude obsahovať zoznam typov nosičov
- pesnicky - bude obsahovať názvy a poradové čísla pesničiek

V tabuľke albumov budeme evidovať aj názov nosiča - to pre prípad, že máme napr. CD so zbierkou MP3, kde sa dá uložiť viac albumov. U audio CD a DVD táto položka nemá význam a nebude sa vyplňať. Všetky tabuľky majú pre prípad možného neskoršieho použitia v sieti tzv. sledovacie atribúty - na začiatku každej tabuľky sú atribúty „zmenil“ a „datum_zmeny“. Sledovacie atribúty v 602SQL umožňujú u každého atribútu (stĺpca) tabuľky sledovať autora zmeny, čas zmeny a históriu zmien. Pokiaľ sa sledovacie atribúty umiestnia na začiatok definície, sleduje sa celá tabuľka, čo je náš prípad.

Tabuľky môžeme vytvárať jednoduchým spôsobom v návrhovom zobrazení - to otvoríte tlačítkom „Vytvoriť“ alebo neskôr „Modifikovať“. My si skúsime druhý spôsob pomocou SQL konzoly. Nebudem tu rozoberať večný boj zástancov príkazového riadku a grafického rozhrania, my rozumnejší sa takéto hádky neberieme vážne a používame to, čo je v danom prípade najjednoduchšie. Samozrejme pokiaľ máme takú možnosť; 602SQL ponúka obidva spôsoby. Konzola dokáže vykonať aj viac SQL



príkazov naraz, jednotlivé príkazy musia byť oddelené bodkočiarkou. Vložte teda nasledujúci text do okna SQL konzoly a stlačte tlačítko „Provésť“:

```
CREATE TABLE `ALBUMY` (
ZMENIL AUTOR ,
DATUM_ZMENY DATIM ,
CISLO INTEGER DEFAULT UNIQUE ,
MAJITEL CHAR(30) COLLATE CSISTRING ,
NAZOV CHAR(50) COLLATE CSISTRING ,
INTERPRET CHAR(50) COLLATE CSISTRING ,
ZANER INTEGER ,
ROK_VYDANIA SMALLINT ,
TYP_NOSICA INTEGER ,
NAZOV_NOSICA CHAR(50) COLLATE CSISTRING ,
CONSTRAINT `INDEX1` UNIQUE (`CISLO`)BY `PESNICKY`,
CONSTRAINT `INDEX2` INDEX (`MAJITEL`),
CONSTRAINT `INDEX3` INDEX (`NAZOV_NOSICA`),
CONSTRAINT `INDEX4` INDEX (`ROK_VYDANIA`),
CONSTRAINT `INDEX5` INDEX (`INTERPRET`),
CONSTRAINT `INDEX6` INDEX (`NAZOV`),
CONSTRAINT `INDEX7` INDEX (`ZANER`),
CONSTRAINT `INDEX8` INDEX (`TYP_NOSICA`));

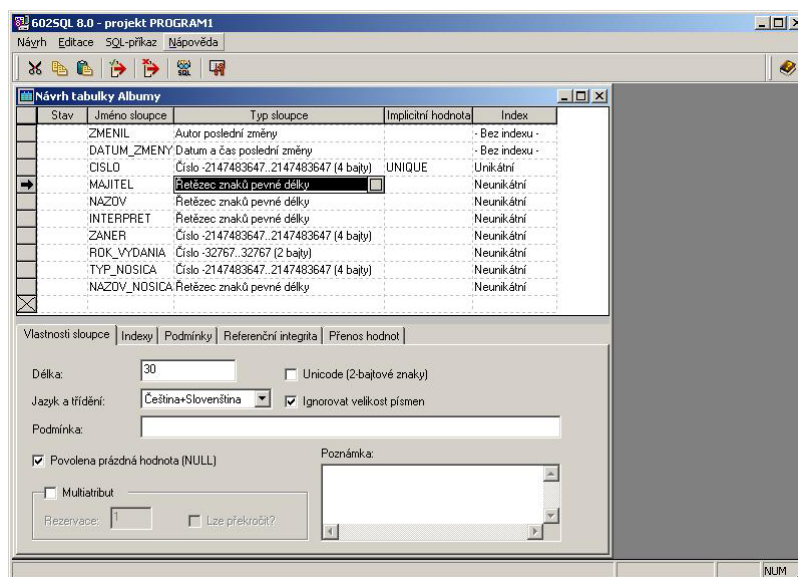
CREATE TABLE `NOSICE` (
ZMENIL AUTOR ,
DATUM_ZMENY DATIM ,
CISLO INTEGER DEFAULT UNIQUE ,
NOSIC CHAR(20) COLLATE CSISTRING ,
CONSTRAINT `INDEX1` UNIQUE (`CISLO`),
CONSTRAINT `INDEX2` INDEX (`NOSIC`));

CREATE TABLE `PESNICKY` (
ZMENIL AUTOR ,
DATUM_ZMENY DATIM ,
ALBUM INTEGER ,
C_PESNICKY SMALLINT ,
PESNICKA CHAR(20) COLLATE CSISTRING ,
CONSTRAINT `INDEX1` INDEX (`album`),
CONSTRAINT `INDEX2` INDEX (`C_PESNICKY`),
CONSTRAINT `INDEX3` INDEX (`PESNICKA`),
CONSTRAINT `REF1` FOREIGN KEY (`album`) REFERENCES `ALBUMY` (`CISLO`) ON
DELETE CASCADE);

CREATE TABLE `ZANRE` (
ZMENIL AUTOR ,
DATUM_ZMENY DATIM ,
CISLO INTEGER DEFAULT UNIQUE ,
ZANER CHAR(20) COLLATE CSISTRING ,
OK BIT DEFAULT false ,
CDDB BIT DEFAULT false ,
CONSTRAINT `INDEX1` UNIQUE (`CISLO`),
CONSTRAINT `INDEX2` INDEX (`ZANER`),
CONSTRAINT `INDEX3` INDEX (`OK`),
CONSTRAINT `INDEX4` INDEX (`CDDB`));
```

Text príkazov nemusíte ručne prepisovať, ale môžete použiť schránku Windows a súbor sql.txt, ktorý je súčasťou priloženého súboru [tabulky.zip](#).

Teraz si môžete vytvorené tabuľky pozrieť aj v návrhári tabuliek. Na to však musíte kliknúť na tlačítko aktualizácie zoznamov objektov v lište (druhé sprava). Inak sa klient nedozvie, že na serveri vznikli nové objekty a vy v zozname tabuliek nebudete nič vidieť. Tento prípad sa stáva keď sa vytvárajú objekty priamo na serveri alebo keď objekt vytvorí iný klient. V návrhári tabuliek je možné tabuľky aj vytvárať aj modifikovať interaktívnym spôsobom.



Každá z tabuliek má svoj kľúč - to je atribút v našom prípade typu INTEGER, ktorý sa používa pre vzájomné vzťahy tabuliek (relácie). Tabuľky nadriadené, t.j. tabuľky, na ktorých záznamy sa budú odkazovať iné tabuľky, majú na kľúči (vždy jednotne nazvanom „cislo“) definovaný unikátny index, pretože vytvorením indexu sa zrýchli vyhľadávanie. Hodnota atribútu „cislo“ sa v tabuľke nesmie opakovať, ak by sme sa pokúsili zaznamenať duplicitu, systém nám to nedovolí. Hodnoty do unikátneho kľúča môže zabezpečiť 602SQL sama a pri vytváraní záznamov sa o to nemusíme starať – ak do položky implicitná hodnota dáme text „UNIQUE“. Tabuľky podriadené, konkrétne „pesnický“, majú pochopiteľne kľúč s neunikátnym indexom a jeho implicitné hodnoty zabezpečíme vo formulároch – v tomto prípade do kľúča s názvom „album“ sa vloží hodnota kľúča „cislo“ z nadriadenej tabuľky. Pretože položiek v tabuľkách nemáme veľa, zaindexujeme neunikátnymi indexami všetky atribúty. Zrýchli sa tým vyhľadávanie. Maximálny počet indexov v tabuľke je 24. Indexovať sa oplatí všetky stĺpce, u ktorých sa v aplikácii predpokladá časté vyhľadávanie. Nie je však vhodné to s indexami preháňať. Čím viac indexov vytvoríme, tým budú dlhšie trvať operácie pridávania, vymazávania a zmeny záznamov.

Okrem premenných typu INTEGER - to je 32 bitové celé číslo (v návrhári sa pomenúvajú ako „číslo - 4 bajty“) používame ešte SMALLINT - čo je 16 bitové celé číslo (v návrhári „číslo - 2 bajty“). Systém podporuje aj 1 bajtové číslo (TINYINT) a 8 bajtové (BIGINT). K dispozícii je aj množstvo iných typov jednak podľa štandardu SQL a niekoľko typov špecifických len pre 602SQL.

Ešte upozorním na tabuľku „pesnický“ - v nej máme definovanú referenčnú integritu na tabuľku „albumy“. Zmyslom referenčnej integrity je (v našom prípade) zabezpečiť automatické zmazanie záznamov z podriadenej tabuľky pri zmazení záznamu v nadriadenej tabuľke. Konkrétne keď zmažem záznam v tabuľke „albumy“, zmažú sa automaticky aj zodpovedajúce záznamy v tabuľke „pesnický“. Referenčná integrita sa definuje v návrhovom zobrazení tabuliek, nie je však prístupná priamo v návrhovom okne, ale treba ju vyvolať v menu pod položkou „Vlastnosti“.

Ak by sa niečo nevydarilo, k dispozícii je súbor [tabulky.zip](#), v ktorom sú potrebné návrhy tabuliek. Import uskutočnite stlačením tlačítka „Importovať objekt“ v pravej časti riadiaceho panelu, keď v jeho ľavej časti je vybratý typ objektu „Tabulky“.

4.2 PODPORA ČEŠTINY A SLOVENČINY

602SQL podľa očakávania - veď ide o český produkt - patrí k tým málo systémom, ktoré majú podporu Č+S triedenia a vyhľadávania vyriešenú na 100%, dokonca svojimi ďalšími schopnosťami (napr. podporou národného fulltextového vyhľadávania, o ktorom bude reč neskôr) ostatných predstihuje. Podpora češtiny a slovenčiny totiž nebýva v databázových systémoch štandardne zahrnutá. Problémy sú najčastejšie v abecednom radení písmena „ch“ a písmen s diakritikou (napr. u MySQL, PostgreSQL alebo Interbase/Firebirdu). Tiež sa môže vyskytnúť problém so zobrazovaním a písaním znakových sád. V minulosti si autori aplikácií s diakritikou hlavy nelámali, snáď aj z dôvodu, že zodpovedajúce nástroje neboli k dispozícii. Jednoducho sa do databáz písalo bez mäkčeňov a dĺžňov. S pozostatkami takéhoto riešenia sa ešte občas stretávame. V dnešnej dobe sa u novo zavádzaných databázových riešení niečo také považuje za závažný nedostatok. Prečo by sa mali ľudia prispôbovať strojom, keď sa môžu stroje prispôbiť ľuďom?

Textové reťazce CHAR, v terminológii 602SQL pomenované ako STRING (v návrhári „řetazec znaku pevné délky“) môžu byť v rôznych jazykoch (ASCII t.j. anglicky, nemecky, francúzsky, poľsky, česky a slovensky). České a slovenské reťazce nesú pomenovanie CSSTRING alebo CSISTRING podľa spôsobu triedenia a vyhľadávania. CSSTRING vyhľadáva a zoraďuje s rozlíšením veľkosti písmen (CASE SENSITIVE), CSISTRING bez rozlíšenia veľkosti. V našich tabuľkách budeme používať CSISTRING. Podotýkam, že uvedené označenia budete potrebovať len pri návrhu tabuliek v textovej forme, v návrhári sa definuje jazyk a triedenie interaktívne výberom z ponuky.

Vyhľadávanie reťazcov je v praxi obvyklé troma spôsobmi:

- vyhľadanie na rovnosť, vyhľadávací dotaz *premenna* = „*hladany text*“, v SQL jazyku *premenna LIKE* „*hladany text*“
- vyhľadanie na prefix (t.j. začína na...), vyhľadávací dotaz *premenna* = „*hladany text*“, v SQL jazyku *premenna LIKE* „*hladany text%*“
- vyhľadanie podreťazca hocikde v hľadanom texte, vyhľadávací dotaz *premenna* = „*hladany text*“, v SQL jazyku *premenna LIKE* „*%hladany text%*“

V súvislosti s vyhľadávaním reťazcov je nutné upozorniť na špecifiká tvorby indexov reťazcových atribútov v 602SQL. Aj tu je možné definovať indexy pre zrýchlenie vyhľadávania, samozrejme uplatnia sa len pri vyhľadávaní v prvých dvoch uvedených prípadoch. Pre vyhľadanie na prefix u jazyka českého a slovenského je kvôli zložitosti vyhľadávacieho algoritmu vhodné vytvoriť okrem normálneho indexu, ktorý sa použije pri vyhľadávaní na rovnosť, ešte jeden špecificky definovaný index preto, aby sa zrýchlilo vyhľadanie. Je to daň za 100% národnú podporu. Vytvorenie takéhoto indexu doporučujem pre tabuľky s veľkým počtom záznamov (>100 000), kde by už čas vyhľadania mohol pôsobiť rušivým dojmom, pre menšie tabuľky aké sú aj v našej aplikácii to nie je potrebné. Aby však bolo jasné, ako sa takýto index vytvára, uvediem to na príklade stĺpca „nazov“. Do SQL príkazu na vytvorenie tabuľky „albumy“ by sme pridali ešte riadok:

```
CONSTRAINT `INDEX9` INDEX (CAST(`NAZOV` AS CHAR(50) COLLATE IGNORE_CASE)),
```

Ako parameter príkazu CHAR je uvedená dĺžka indexovaného stĺpca. V prípade, že potrebujeme rozlišovať veľkosť písmen, vypustí sa časť „COLLATE IGNORE_CASE“.

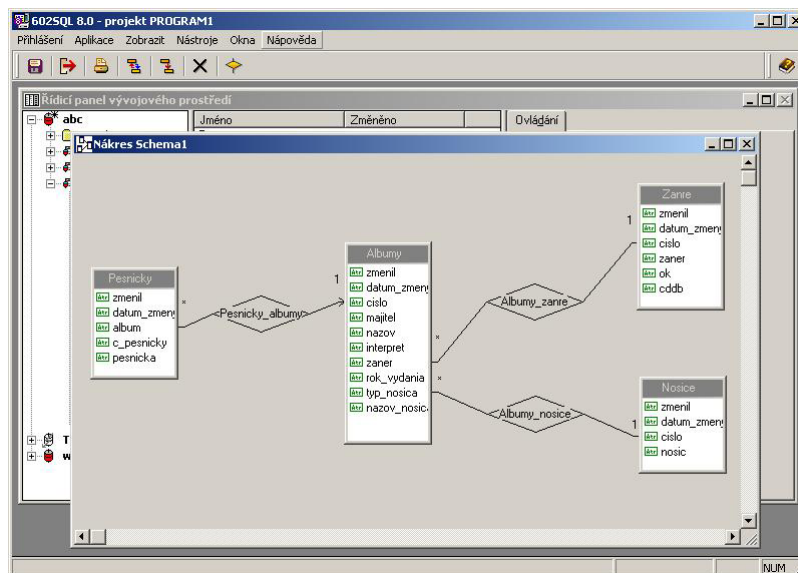
4.3 RELAČNÉ VZŤAHY

Relácie môžete vytvoriť v riadiacom paneli dvoma spôsobmi - vytvorením objektov typu relácia alebo objektu typu nákres. Použijeme druhú možnosť, lebo táto je jednoduchšia a hlavne prehľadnejšia. Vytvorte nový nákres a na jeho plochu vložte postupne všetky tabuľky (pravou myšou „Vložiť lokálny objekt“ - „Vložiť tabuľku“). Prispôbte veľkosť tabuliek, aby boli vidieť všetky atribúty a vhodne ich na ploche rozmiestnite - naľavo pesničky, v strede albumy a vpravo žánre a nosiče. Metódou drag&drop vytvoríme relácie:

- atribút „album“ z tabuľky „pesničky“ posuňte na atribút „cislo“ v tabuľke „albumy“ a pomenujte reláciu na „pesnickovy_albumy“.
- atribút „zaner“ z tabuľky „albumy“ posuňte na atribút „cislo“ v tabuľke „zanre“ a pomenujte reláciu na „albumy_zanre“.
- atribút „typ_nosica“ z tabuľky „albumy“ posuňte na atribút „cislo“ v tabuľke „nosice“ a pomenujte reláciu na „albumy_nosice“.

Vytvorené relácie budete vidieť aj v zozname objektov typu „relace“. Nákras uložte pod názvom „schema1“. Takýto spôsob vytvárania vzťahov medzi tabuľkami nájdete aj u MS Access. Do nákrasov môžete vkladať aj iné objekty ako tabuľky. Prispieva to k prehľadnosti aplikácie. Túto vlastnosť už Access nemá.

Ako som už spomínal, dalo by sa vytvoriť našu databázu aj tak, že všetky údaje budú iba v jednej tabuľke „albumy“. Takýto postup je pre programátorov databázy jednoduchší, ale používateľom by skomplikoval život. Vytvorením viacerých tabuliek a relácií medzi nimi sa totiž sleduje niekoľko cieľov. Ako prvá vás určite napadne optimalizácia miesta na disku. Ukážeme si to príklade, keď by sme vytvorili len jednu tabuľku



namiesto tabuliek „albumy“ a „pesničky“. Každá tabuľka je definovaná staticky. Atribúty tabuľky (stĺpce) zaberajú miesto bez ohľadu na to, či sú v nich zapísané údaje. Riadok tabuľky bude takto zberať zbytočne veľa miesta, pretože by sme museli vytvoriť tabuľku obsahujúcu veľa atribútov: okrem základných údajov o albume aj niekoľko stĺpcov pre názvy pesničiek. Aby sme nestratili žiadne údaje, musíme vytvoriť toľko stĺpcov, koľko predpokladáme ako maximum počtu pesničiek na albume. Čo ale urobíme, ak dostaneme do rúk album s väčším množstvom pesničiek, než sme predpokladali? Tomuto problému sa síce

môžeme vyhnúť inou konštrukciou tabuľky - tak, že každý riadok bude predstavovať jednu pesničku, v tomto prípade budú v tabuľke zbytočne duplicitné údaje o albume. Rozdelenie jednej tabuľky do dvoch je preto logickým výsledkom riešenia problému.

Ďalším dôvodom pre vytváranie relácií je to, že sa ľahšie aktualizujú údaje v databáze. Ak si predstavíme prípad, že chceme zmeniť názov typu hudobného žánru, znamená to opraviť údaje vo všetkých riadkoch, ktorých atribút „zaneer“ sa zhoduje so zmeneným. V relačnej databáze stačí opraviť len jeden riadok v nadriadenej tabuľke. V prípade zrušenia záznamu z nadriadenej tabuľky nám automatické vymazanie príslušných záznamov v podriadenej tabuľke zabezpečí definovanie referenčnej integrity.

Vráťme sa ešte k nákresu našej aplikácie. Vidíme na nej 3 relácie typu 1:N (t.j. k jednému záznamu z nadriadenej tabuľky je možné priradiť N záznamov z podriadenej tabuľky). Podriadená tabuľka má na grafe vo svojej blízkosti symbol *, nadriadená tabuľka symbol 1. Tabuľka „pesnička“ je podriadená vo vzťahu k tabuľke „albumy“. Tabuľka „albumy“ je podriadená vo vzťahu k tabuľkám „zanre“ a „nosice“. Referenčnú integritu typu „akcia pri zrušení“ sme definovali len v dvojici tabuliek „pesnicka“ -- „albumy“. V nadriadených tabuľkách „zanre“ a „nosice“ nepredpokladáme mazanie záznamov, čo zakážeme aj v príslušných formulároch.

602SQL umožňuje definovať aj referenčnú integritu typu „akcia pri zmene“. Keďže sme vytváranie indexov v nadriadenej tabuľke „albumy“ ponechali na systém (implicitná hodnota UNIQUE), takúto referenčnú integritu definovať nepotrebujeme.

4.4 IMPORT DÁT DO TABULIEK

Ďalším krokom bude naplnenie tabuliek hodnotami. Nebojte sa, údaje nebudete zadávať ručne, ale nájdete ich v priloženom súbore [tabulky2.zip](#). Obsah súboru rozbaľte na Váš disk. V riadiacom paneli označte tabuľku „nosice“ a kliknite na tlačítko „Import dat“. Vyberte príslušný súbor a preveďte import tlačítkom „Prenést data“. Podobný postup urobte aj s tabuľkou „zanre“.

V tabuľke „nosice“ som uviedol všetky typy nosičov (CD, DVD...) a typy komprimovaných audio/video súborov (MP3, AVI...), na ktoré som si spomenul. Tabuľka „zanre“ obsahuje zoznam žánrov hudby, ktorý som prevzal z WinAmpu. Tabuľka obsahuje aj atribút „cddb“ typu BOOLEAN (v SQL typ BIT) vypovedajúci o existencii zodpovedajúcich žánrov v CDDb (tých je menej). Naproti tomu WinAmp ich pozná podľa môjho názoru až príliš veľa. Aby som urobil kompromis, v tabuľke som vytvoril atribút „ok“ typu BOOLEAN, ktorý neskôr využijeme na určenie, aké údaje sa majú ponúkať.

Ešte poznámka: v súbore [tabulky2.zip](#) nájdete pre prípad problémov aj nákres „schema1“. V ňom musíte vytvoriť relácie vyššie uvedeným postupom. Relácie vám nemôžem dať k dispozícii, lebo tie sa na rozdiel od iných typov objektov nedajú importovať ani exportovať jednotlivo ale iba v rámci importu/exportu celej aplikácie. V priložených súboroch sa samozrejme nachádza aj celá výsledná aplikácia, ale počas čítania ďalšieho textu budeme niektoré objekty meniť – preto nepredbiehajte.

5. NÁVRH DOTAZOV

5.1 DEFINÍCIA DOTAZU

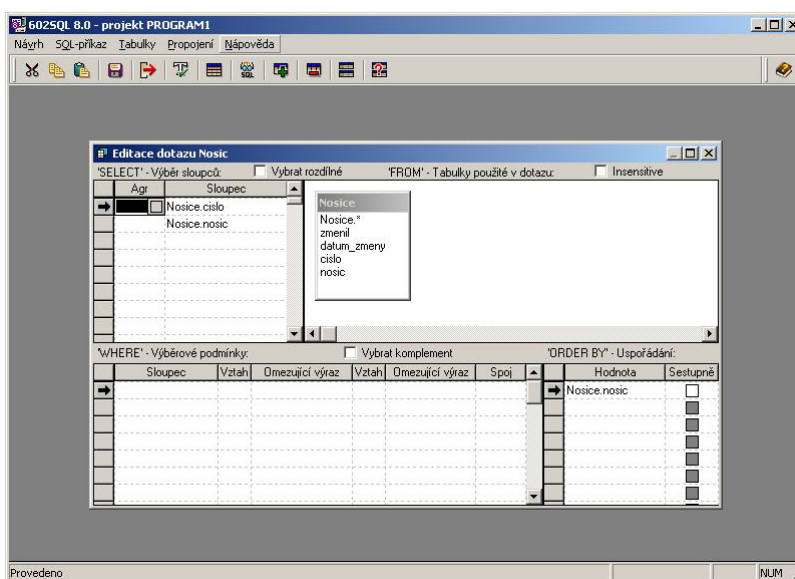
Dotaz je vlastne popis, ako sa majú získať dáta z tabuliek. Jeho základom je vždy SQL príkaz „SELECT“. V terminológii jazyka DDL (Data Definition Language), ktorá je súčasťou SQL sa na jeho označenie používa pomenovanie VIEW. Na dotaz sa môžu odkazovať formuláre ale aj iné objekty 602SQL – napríklad ak potrebujeme vytvoriť formulár, kde budú záznamy zoradené v inom poradí, než sú zapísané v tabuľke, použijeme ako jeho zdroj dát dotaz, ktorý si podľa potreby vopred vytvoríme. Preto si dotazy popíšeme ešte pred návrhom formulárov.

Poznáme 2 druhy dotazov: statické a dynamické. Statické dotazy sú pevne určené a sú jedným z druhov objektov, ktoré 602SQL pozná. Dynamické dotazy nie sú uložené v databáze, ale klienti si ich môžu vytvárať v programe a používať ich podobne ako statické. Nimi sa zaoberať nebudeme. Na vytváranie jednoduchších dotazov slúži návrhár dotazov, na zložitejšie budeme musieť použiť integrovaný textový editor. Mimochodom obdobný návrhár dotazov nájdete aj v aplikácii MS Access.

5.2 VYTVORENIE DOTAZOV

V predchádzajúcej časti sme vytvorili 4 tabuľky a teraz k nim vytvoríme dotazy. V riadiacom paneli 602SQL sa nastavte na „Dotazy“ a stlačte tlačítko „Vytvoriť“. Budete vyzvaní k zadaniu tabuľky, z ktorej dotaz vytvárate. Vyberte napríklad „nosice“. Ako vidno v návrhári, v tabuľke „nosice“ máme 4 atribúty. Z nich budeme potrebovať len „cislo“ a „nositic“, zvyšné dva sú pre našu prácu nepotrebné.

Touto optimalizáciou zmenšíme objem prenášaných dát medzi serverom a klientom a tým zrýchlime prácu - samozrejme v takej malej aplikácii ako je naša to má iba formálny význam. Do časti SELECT presunieme myšou z grafického zobrazenia tabuľky metódou „drag&drop“ (ťahaj a pust') postupne oba potrebné atribúty. Podmienkovú časť „WHERE“ vyplniť nebudeme. Vyplníme ale spôsob usporiadania „ORDER BY“. Zoradovať budeme podľa atribútu „nositic“, aj to sa v návrhári definuje pomocou „drag&drop“. Návrhár uzavrieme a dotaz uložíme pod menom „nositic“. Týmto sme vytvorili jednoduchý dotaz „SELECT nosice.cislo, nosice.nositic FROM nosice ORDER BY nosic“,



o čom sa môžete presvedčiť zobrazením dotazu v SQL - vybraním záložky SQL v pravej časti riadiaceho panela. Dvojklikom myšou na dotaz v riadiacom paneli sa zobrazí štandardný formulár s dátami.

Podobným spôsobom vytvoríme aj dotazy „zaneer“ a „pesnick“. Do dotazu „album“ vyberieme všetky atribúty okrem „zmenil“ a „datum_zmeny“. Zoraďovať budeme najprv podľa atribútu „interpret“ a potom podľa „nazov“. Všetky vytvárané dotazy nájdete v priloženom súbore [dotazy.zip](#).

6. NÁVRH FORMULÁROV

6.1 DEFINÍCIA FORMULÁRA

Formulár je jeden z objektov typu „Pohledy“. Môže mať tieto formy:

- Formulár - pre prácu s dátami
- Zostava – formuláre pre tlač dát
- Štítok - pre špeciálne tlačové zostavy štítkov
- Graf - formulár s grafickým zobrazením dát

Pre vytváranie a zobrazovanie grafov treba mať u klienta nainštalovaný grafový server MagicGraf602, ktorý je súčasťou produktu 602 PC Suite, ale dá sa získať aj samostatne. Pomocou MagicGrafu môžete podobne ako v aplikáciách MS Office vytvárať jednoduché grafy pre kancelárske použitie (koláčový, stĺpcový, 3D atď) na základe reálnych dát vo vašich tabuľkách. Je vhodný pre použitie vtedy, ak počty zobrazovaných údajov v grafe dosahujú rádovo stovky, pri vyššom počte sa grafy nezobrazujú správne. To nie je nedostatok, veď pre predpokladané použitie v ekonomických a štatistických zobrazeniach to úplne stačí. Navyše do úvahy treba zobrať aj rozlišovacie schopnosti obrazovky alebo tlačiarne.

6.2 VYTVORENIE FORMULÁROV

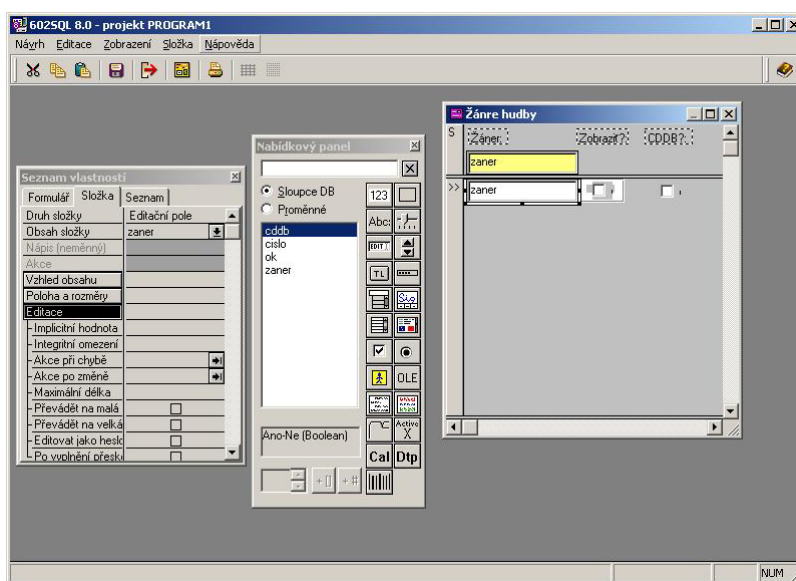
Podme teda vytvoriť ako prvý formulár na zobrazovanie typov nosičov. Na riadiacom paneli zvolíte vytvorenie nového pohľadu, vyberte typ „Formulár“. Ako zdroj dát použijeme dotaz „nosit“. V ďalšom kroku vyplníme titulok okna, napr. „Typy nosičov“ a vyberieme, že v okne sa budú zobrazovať viaceré záznamy. Spôsob, ako sa majú dáta v okne zobrazovať, je znázorňovaný graficky. Nám bude najviac vyhovovať prvý ponúknutý – formulár s dotazovými poliami. V ďalšom kroku sa dostaneme k zoznamu položiek vo formulári. Je zrejmé, že nepotrebujeme zobrazovať index „cislo“ – je to záležitosť systému a nás to vôbec nezaujíma. Prvú položku teda zmažeme spôsobom obvyklým vo Windows – označíme riadok kliknutím na čiernu šípku smerujúcu doprava a stlačíme klávesu Delete. Vo vlastnostiach poľa „nosit“ zaškrtneme „dotazové pole“.

V ďalšom kroku určíme vlastnosti formulára. Tu nám stačí zmeniť vlastnosť „Zrušení vybraného záznamu“ – v minulých dieloch sme povedali, že toto neumožníme. To bol zároveň posledný krok v návrhu formulára. Po stlačení tlačítka „Ukončit“ sa nám otvorí formulár v návrhovom zobrazení spolu s dvoma ďalšími oknami. Prvé z nich sa volá „Nabídkový panel“ a slúži na pridávanie položiek do formulára (drag&drop). Druhé okno „Seznam vlastností“ slúži na zmenu vlastností celého formulára alebo vybraných položiek. My už nebudeme robiť žiadne zmeny, preto formulár uzavrieme a uložíme pod názvom „nos“.

Dvojklikom na ikonu formulára vyvoláme jeho otvorenie. Prechádzanie medzi záznamami môžete vykonávať jednak z klávesnice (klávesy PageUp, PageDown) alebo pomocou myši klikaním na ikony na lište tlačítok prípadne ovládaním tiahla v pravej časti okna.

Ďalej vytvoríme formulár žánrov „zan“. Na začiatku návrhu môžete zvoliť ako šablónu pohľad „nos“, tým sa v novom formulári použijú vlastnosti pohľadu „nos“. Potom vytvoríme formulár „pesn“ – tu použijeme typ zobrazenia: viaceré záznamy, štandardný formulár (posledný ponúkaný typ). V zozname položiek ponechajte aj „album“ a v zozname vlastností zrušte zobrazovanie fiktívneho záznamu. Tento formulár použijeme ako subformulár do pohľadu „Album_podrobne“. Formulár „Album_podrobne“ vytvoríme so zdrojom z dotazu „album“. Bude zobrazovať iba jeden záznam, typ si vyberieme prvý z ponúkaných – zoznam pod sebou. Vo vlastnostiach nezakážeme rušenie záznamov. Do jeho voľnej pravej časti vložíme nový objekt – subformulár. V zozname vlastností subformulára vyberieme jeho názov. Spôsob synchronizácie zvolíme „Záznamy propojené reláci“, kde následne vyberieme meno relácie „Pesnicky_albumy“. Upravíme šírku subformulára tak, aby bolo vidieť všetky položky.

Vo formulári urobíme ešte malé zmeny. Návrhár nevie, že položky „zane“ a „typ_nosica“ nechceme zobrazovať ako čísla z číselníkov, ale tak, aby sa ponúkali texty. Preto položky, ktoré návrhár vytvoril, zmažte. Vytvoríme si novú položku typu combo: V okne zoznamu vlastností zvolíte „obsah složky“ atribút „zane“ a v časti pre combo vyplňte druh „překládaný“ a ako zdroj napíšete text „select zane,cislo from zane order by zane“. Podobne postupujte pri vytváraní comba aj u položky „typ_nosica“. Na záver v menu návrhára vyvolajte „Návrh“ - „Pořadí složek“ a upravte poradie zložiek, aby pri zadávaní dát (po stlačení Enter-u) resp. pri pohybe vo formulári (tabulátorom) neboli problémy.



Pri zadávaní nových záznamov je treba stlačiť na klávesnici „Insert“. Tým sa v okne presunieme na posledný, tzv. fiktívny záznam, do ktorého možno písať nové údaje. „Insert“ použijeme aj na zadávanie pesničiek. Aby sa do tabuľky „pesnicky“ vložila správna hodnota z nadriadeného formulára do atribútu „album“ v podriadenom subformulári, musí subformulár obsahovať položku „album“ (aj keď my ju nepotrebujeme vidieť) a nesmie umožniť zobrazovanie fiktívnych záznamov. V opačnom prípade by sme správne nadobúdanie hodnôt položky „album“ museli zabezpečiť sami programom, ale o tom bude reč až neskôr. Preto je vhodné u hodnoty položky „album“ v návrhovom zobrazení formulára „pesn“ vložiť podmienku aktivity napr. $1=2$, (t.j. takú, ktorá sa nikdy nesplní), čím zabránime používateľom nechcené prepísanie tejto hodnoty.

Ovládanie množstva akcií ako výber, vkladanie, rušenie záznamov atď sa u okien formulárov uskutočňuje pomocou ikón na lište tlačítok. Je možné nadefinovať si aj vlastnú lištu. Takúto lištu s minimálnym množstvom základných ikón je vhodné doplniť ďalšími tlačítkami v hlavovej časti formulárov so špecifickými funkciami. Tu sa na chvíľu pozastavíme. Každý formulár má 3 časti: hlavu, dátovú časť a päť. Normálne v okne vidíme

len dátovú časť, ostatné majú nulové rozmery a musíme ich pre prípad potreby odkryť – tesne na okraji okna formulára v návrhovom zobrazení sa kurzor myši zmení na šípku a vtedy rozmer hlavy resp. päty môžeme zväčšiť a do vytvoreného priestoru vkladať položky.

Teraz si zrejme uvedomujete, prečo som spomínal, že je návrh formulárov časovo najnáročnejší. Popisovať ďalšie možnosti návrhára je pre účely tohto textu zbytočné. Všetky možnosti máte vysvetlené v kontextovej elektronickej nápovede. Každý z vás určite má iný názor na vzhľad formulárov aplikácie a pokúsi sa ich vytvoriť podľa svojho vkusu.

6.3 FORMULÁRE A VYHLADÁVANIE

„Nos“ alebo „zan“ sú jednoduché číselníky a nie je dôvod vytvárať k nim viac formulárov. U rozsiahlejších tabuliek sa obvykle vytvára v aplikácii niekoľko rôznych formulárov. Formulár „Album_podrobne“ zobrazuje naraz len jeden záznam. Bude potrebné vytvoriť ešte jeden formulár – taký, ako sú „nos“ a „zan“, t.j. aby sa v ňom zobrazovalo viac záznamov naraz. Keďže všetky položky nepomestíme na obrazovku, budú tam len tie najzaujímavejšie. Ďalšia požiadavka bude ľahké vyhľadávanie – po vyhľadaní potrebných údajov umožníme skok na podrobný formulár.

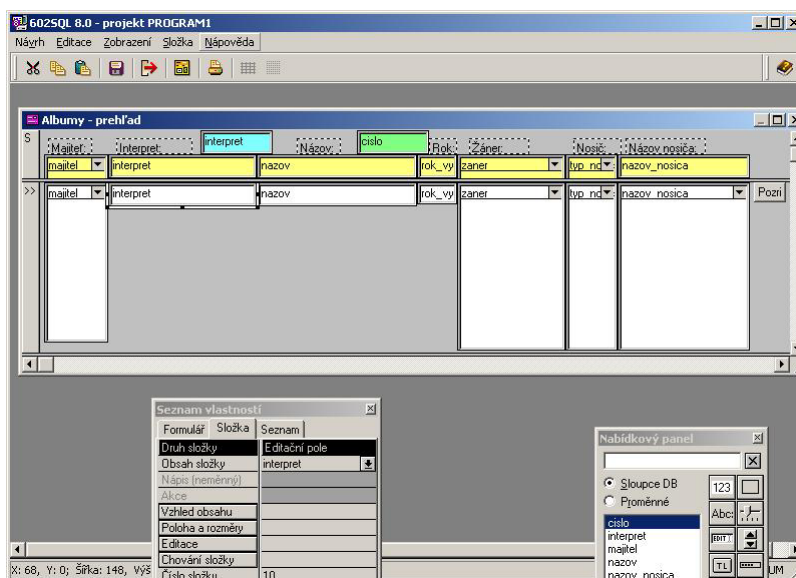
Vytvoríme teda formulár, ktorý nazveme „Album_prehľadovo“ - na viaceré záznamy, typ s dotazovými poliami. Vyberte na zobrazovanie len tie položky, ktoré považujete za potrebné. Pri troške snahy sa zmestia všetky okrem názvov pesničiek. Mne sa to podarilo aj pre grafické rozlíšenie 800x600. Horizontálny rozmer okna som zväčšil na maximálnu možnú mieru, šírku položiek formulára som zmenšil; v prehľadovom okne nie je potrebné, aby bolo vidno celý text. Ešte doplňujem, že rozlíšenie minimálne 800x600 pixelov budete potrebovať u všetkých formulárov v našej aplikácii. Pri návrhu formulárov je nutné stanoviť si nejaké minimálne rozlíšenie. Pri nižšom rozlíšení sa síce hlavné okno 602SQL prispôsobí tomuto rozlíšeniu, ale nebude vidno celá časť formulárov, čo zníži úroveň databázy ako z hľadiska estetického, tak aj úžitkového.

Vráťme sa k formuláru „Album_prehľadovo“. Podobne ako u formulára „Album_podrobne“ treba niektoré položky nahradiť combom. Môžete si zjednodušiť prácu a pomocou schránky Windows preniesť položky, ktoré sme už raz robili. Combo použijeme aj u položiek „majiteľ“ a „nazov_nosica“. Combo bude editačné a ako jeho zdroj uvedieme „SELECT DISTINCT majiteľ FROM albumy ORDER BY majiteľ“ resp. „SELECT DISTINCT nazov_nosica FROM albumy ORDER BY nazov_nosica“. Týmto umožníme u týchto položiek zjednodušenie zadávania nových údajov: môžeme písať text ručne alebo vybrať zo zoznamu už zadaných. Slovo DISTINCT v SQL príkaze zabezpečí, aby sa rovnaké hodnoty ponúkali iba raz. Aby sme zbytočne nezaviedli ďalšie tabuľky, SQL dotaz smeruje do rovnakého zdroja ako aj celý formulár.

Do úplne pravej časti okna vložte nový objekt „tlačítko“. V okne zoznamu vlastností mu definujte zobrazovaný text (napr. „Pozri“) a akciu: „vložit standardní akci“ – „otevřání a zavřání formuláře“ – „na stejný záznam otevřít formulář“ a vyberte meno otváraného formulára „Album_podrobne“. Pokiaľ recipročne vložíte do podrobného formulára tlačítko na otvorenie prehľadového, budete sa môcť ľahko prechádzať medzi oboma formami zobrazenia dát. V tom prípade ale nezabudnite dať na tlačítko ako druhú akciu zavretie prvého formulára. V opačnom prípade by sa pri prechádzaní pootváralo neprehľadné množstvo formulárov.

Ďalej upravíme vlastnosti vyhľadávacích polí. Pre ich odlišenie od dátových je vhodné im zmeniť farbu pozadia – napr. na žltú. To však nie je všetko: každé vyhľadávacie pole štandardne vyhľadá presne ten text, ktorý mu zadáme (pozn.: hovorím teraz len o textových poliach). My budeme chcieť viac – aby vyhľadávacie polia „interpret“, „nazov“ a „nazov_nosica“ vyhľadávali dynamicky počas zadávania záznamy obsahujúce hľadaný reťazec bez ohľadu na jeho postavenie (či už je na začiatku, v strede alebo na konci). Na to slúži vlastnosť „Dotazové pole“, kde treba zvoliť typ „dynamické“ a ako dotazovú podmienku zadať napr. u poľa „interpret“ výraz „interpret.=.%“. Ak by sme chceli vyhľadávanie záznamov, začínajúcich na hľadaný reťazec, použili by sme výraz „interpret.=.%“. A na záver skontrolujeme ešte poradie zložiek.

Formulár „Album_prehľadovo“ je na tom z pohľadu možností vyhľadávania celkom dobre. Avšak predsa len nevie úplne všetko. Povedzme, že potrebujem vyhľadať nejaký album, presne neviem aký (dosť častý prípad...), viem len, že žáner hudby je to Rock alebo Metal. V našom formulári by to znamenalo zvoliť vyhľadávacie kritérium najprv na žáner Rock a potom na Metal. A to nehovorím o tom, že žánrov obsahujúcich slovo „Rock“ je viac (práve toto nie je najvhodnejší príklad, my totiž nezobrazujeme všetky záznamy z tabuľky žánrov – vid' atribút „ok“, ale na pochopenie to stačí).



Problém je v tom, že položka typu combo neumožňuje zadať viacero hodnôt naraz. Samozrejme, skúsenejší vedia, že stačí kliknúť na ikonu „Zadať podmienky hľadania“ na lište a v otvorenom okne pre zadávanie dotazov QBE (Query By Example) postupne zadať kritériá do polí pod sebou – tie sa v oknách zobrazujúcich viacej záznamov akceptujú ako podmienka „alebo“ (po anglicky „OR“). Po zadaní kritérií ich treba potvrdiť kliknutím na ikonu „Uplatnenie zadaných obmedzení“. Takéto vymoženosti sú však pre bežných používateľov zložité, preto uprednostníme jednoduchší spôsob.

6.4 POUŽITIE DOTAZOV Z VIACERÝCH TABULIEK

Aby sa nové formuláre nepletli so starými, vyrobíme ich ďalšie verzie „Album_podrobne2“ a „Album_prehľadovo2“ a ako prvé opravíme akcie u tlačítok (iné názvy formulárov). Potom vytvoríme nový dotaz „Album2“ (v textovom editore, návrhár toto už nezvládne):

```
SELECT Albumy.cislo, Albumy.majitel, Albumy.nazov, Albumy.interpret,
Albumy.zaner, Albumy.rok_vydania, Albumy.typ_nosice, Albumy.nazov_nosica,
Nosice.cislo, Nosice.nosic, Zandre.cislo, Zandre.zaner, Zandre.ok, Zandre.cddb
FROM Albumy
LEFT OUTER JOIN Nosice ON (Nosice.cislo = Albumy.typ_nosica)
```

```
LEFT OUTER JOIN Zanre ON (Zanre.cislo = Albumy.zaner)
ORDER BY Albumy.interpret, Albumy.nazov
```

Spojením tabuliek pomocou LEFT JOIN vznikne toľko záznamov, koľko ich má prvá tabuľka, tie sa spoja s aspoň jedným záznamom z druhej tabuľky prípadne s prázdny záznamom. Dotaz „Album2“ uvidíme ako zdroj dát formulárov „Album_prehľadovo2“ a „Album_podrobne2“. Teraz môžeme v prehľadovom formulári zrušiť vyhľadávacie polia typu combo a zameniť ich za klasické editačné polia s tým, že uvidíme dotazové podmienky s použitím „.=.“ alebo „.=“. V takomto vyhľadávacom formulári už nie je problém nájsť napríklad aj všetky žánre obsahujúce konkrétne slovo alebo jeho časť. Formulár sa správa ako keby viedol len do jednej tabuľky s jedinou výnimkou – nejdú vkladať záznamy klasickým spôsobom, t.j. stlačením klávesy INSERT alebo tlačítkom s definovanou akciou „Vložiť záznam do bežného formulára“. Preto je v jeho vlastnostiach vhodné nastaviť zákaz vkladania (a aj rušenia) záznamov – tieto činnosti by sa museli ošetriť programovo.

V tomto štádiu už máme všetky základné formuláre hotové. Tvorbou tlačových zostáv sa nebudeme zaoberať. Je podobná ako u formulárov a nič zásadne nové by ste sa nedozvedeli. Aj tak má každý formulár na lište ikonu pre tlač, ktorá sa bude v núdzi dať využiť.

6.5 RELÁCIE VO FORMULÁROCH

602SQL je relačná databáza. Relačné vzťahy tabuliek je samozrejme možné uplatniť aj vo formulároch. V predchádzajúcej časti som čo-to naznačil. Preto by bolo vhodné (aj pre budúcnosť) nakrátko odložiť tvorbu našej aplikácie a systematicky všetky tieto možnosti popísať.

V prípade, že máme dve tabuľky, ktoré sú relačne zviazané (ako príklad použijeme dvojicu „Albumy“ – „Zanre“), máme niekoľko možností, ako vo formulároch vkladať do nadriadenej tabuľky „Albumy“ údaje z podriadenej tabuľky „Zanre“.

Ako prvú možnosť uvediem vkladanie prostredníctvom typu poľa „combo“ – čo sme aj v našom prípade použili. Combo slúži na výber jedného záznamu, teda je vhodné na relácie 1:N. Ako zdroj dát comba sme použili špeciálne upravený dotaz do tabuľky „Zanre“. Vkladanie je jednoduché, ale nevýhoda je v neprehľadnom výbere pri väčšom množstve záznamov v podriadenej tabuľke, navyše ich maximálny možný limit je 4000. Pri väčšom množstve sa nadbytočné záznamy v combe nezobrazia. Okrem toho na rýchle vyhľadanie položky v combe pomocou klávesnice sa akceptuje len prvý znak.

Combo je možné použiť nielen pre číselné atribúty (je síce prekladané textom, ale do tabuľky sa ukladajú číselné údaje) – čo bol presne náš prípad. Takto je zaručené, že sa do nadriadenej tabuľky vyberie presne jeden záznam z podriadenej tabuľky. Ak použijeme combo na textové atribúty, (tzv. editačné combo), jeho rozvinutím sa umožní výber položiek z podriadenej tabuľky, ale vybraný text pôjde hocikedy ručne prepísať. Takto sa nedá zaručiť, že bola vybraná hodnota presne zodpovedajúca hodnote z podriadenej tabuľky. Samozrejme takýto vzťah tabuliek už nie je definovaný reláciou, ide len o zjednodušený výber textu do tabuľky.

Druhou možnosťou je vkladanie prostredníctvom tlačítka s akciou „Otvoriť relační formulár“. V takomto prípade sa nad nadriadeným formulárom otvorí podriadený. V jeho

ľavej časti budú (navyššie oproti klasickému zobrazeniu formulára) vidieť tlačítka slúžiace na výber záznamov (podľa typu relácie 1:N alebo M:N bude sa dať vybrať jeden alebo viaceré záznamy). Toto je najuniverzálnejšie riešenie, i keď niekomu môže vadieť, že sa musia otvárať dve okná.

Tretia možnosť spočíva vo vytvorení formulára so subformulárom (tú sme použili pri vzťahu tabuliek „Albumy“ – „Pesnický“). Je najprehľadnejšia, avšak nie všetky činnosti je možno riešiť pomocou štandardne definovaných akcií. Ako vidíte, možností je viac, výber najvhodnejšieho riešenia závisí od konkrétneho použitia.

V súvislosti s relačnými vzťahmi môžem okrajovo spomenúť aj historicky starší spôsob definovania vzťahu medzi tabuľkami, ktorý 602SQL ešte stále podporuje. Ide o tzv. ukazovatele, česky „ukazatele“. Pri relačných vzťahoch je do nadriadenej tabuľky vkladaná celočíselná hodnota indexu z podriadenej tabuľky. U ukazovateľov sa do nadriadenej tabuľky zapisuje poradové číslo záznamu z podriadenej tabuľky. Ukazovateľ môže byť jednosmerný alebo obojsmerný, jeho definovaním 602SQL zakáže uvoľňovanie záznamov z podriadenej tabuľky, lebo by to malo za následok porušenie odkazov.

Pozn.: záznam sa pri zrušení v 602SQL z tabuľky nezmaže naozaj, ale len označí ako zrušený. Jeho skutočné zrušenie sa prevedie až akciou nazvanou „uvoľnenie“. Na miesto uvoľnených záznamov sa môžu vkladáť nové záznamy. Uvoľnený záznam zaberá miesto v tabuľke až do prevedenia jej skompaktovania. O počtoch platných, zrušených či uvoľnených záznamov v tabuľkách sa môžete informovať tlačítkom „Statistika“ riadiaceho panela.

Aj keď je princíp ukazovateľov odlišný, práca s ním vo formulároch je podobná ako u relácií. V jednom ohľade je to dokonca výhodnejšie – keď sme potrebovali jednoducho vyhľadávať napr. údaje v položke typu combo popisujúcej žánr hudby zadávaním textu, museli sme vytvoriť formulár, ktorý viedol do dotazu pozostávajúceho z dvoch tabuliek. U ukazovateľov toto nie je treba robiť, vyhľadávacie pole sa používa tak, ako keby sa príslušná položka nachádzala v nadriadenej tabuľke. Ak teda netrpíte potrebou vytvárať „čisté“ aplikácie, môžete pre niektoré prípady použiť aj ukazovatele.

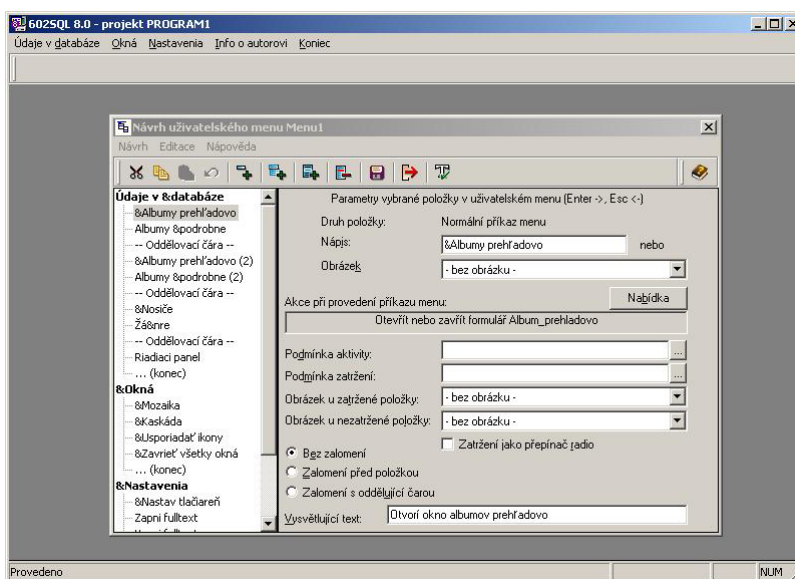
Formuláre vytvorené v tejto kapitole nájdete v priloženom súbore [objekty.zip](#).

7. MENU

7.1 MOŽNOSTI MENU A JEHO NÁVRH

V aplikácii je možné určiť jeden z objektov typu formulár, menu alebo program ako štartovací, ktorý sa bude otvárať automaticky pri štarte aplikácie namiesto riadiaceho panelu vývojového prostredia. V najjednoduchšom prípade môžete vytvoriť ešte jeden formulár bez zdroja dát, ktorý určíte ako štartovací. V ňom budú umiestnené tlačítka na otváranie doteraz vytvorených formulárov. Takýto spôsob sa používa v aplikáciách MS Accessu.

Oveľa efektívnejšiu možnosť dosiahnete vytvorením menu, čo už napr. Access tak jednoducho nedokáže. V riadiacom paneli vytvoríte nové menu tlačítkom „Vytvoriť“. Na výber dostanete dve možnosti – normálne alebo popup menu. Líšia sa spôsobom usporiadania: normálne je v horizontálnom a popup je vo vertikálnom smere. Postup pri vytváraní nových položiek v menu je nasledovný: Vytvoríte novú položku. V novej položke postupne vyplníte jej parametre (nápis, akciu po vybratí, podmienky aktivity atď.). Špeciálnymi prvkami menu sú submenu a oddelovacia čiara. Môžete tiež definovať vybranie položky menu pomocou klávesovej skratky (tzv. horúce klávesy), stačí pred požadovaný znak názvu uviesť „&“.



V našej aplikácii vytvoríme normálne menu, ktoré sa zvykne používať najčastejšie. Ako prvé navrhne submenu „Údaje v databáze“, v ktorom sa nachádzajú príkazy na otváranie všetkých našich formulárov. Ďalej nasleduje submenu „Okná“ na usporiadanie okien formulárov. Na záver sú príkazy „Nastav tlačiareň“, „Info o autorovi“ a „Koniec“. Nezabudnite ešte určiť menu ako štartovací objekt aplikácie. Vytvorením menu dáme ovládaniu aplikácie štandardný a profesionálny vzhľad. Menu vytvorené v tejto kapitole nájdete v priloženom súbore [objekty.zip](#).

8. PROGRAMY V 602SQL

8.1 PROGRAM V INTERNOM JAZYKU

Interný jazyk je odvodený z Pascalu a je obohatený o príkazy na manipuláciu s obsahom databázy, otváranie a prácu s formulármi, dotazmi, menu, tlačou zostáv atď. Pomenovanie „interný“ jazyk (myslené z pohľadu klienta) nosí preto, aby sa odlišil od programov vytvorených v externých jazykoch ako napr. C/C++, Delphi alebo aj SQL. Program v tomto jazyku nebeží na serveri, ale na počítači klienta. Každý program v internom jazyku pozostáva z týchto častí:

- deklarácie tabuliek, dotazov a globálnych premenných
- deklarácie procedúr a funkcií
- odkazy na ďalšie programy (INCLUDE)
- hlavný program

Deklarácie nie sú povinné, hlavný program áno. Takže najjednoduchší program vyzerá takto:

```
begin
end.
```

8.2 ŠTRUKTÚRA PROGRAMU

Na to, aby sa program spustil po štarte aplikácie, je nutné uložiť ho, preložiť a určiť ako štartovací objekt. Tým preberie riadenie aplikácie. Predchádzajúci mini-program samozrejme neurobí nič – okamžite po spustení sa ukončí. Na to, aby aj k niečomu slúžil, musíme urobiť dve veci – doplniť potrebné procedúry či funkcie a donútiť ho, aby ostal bežať až kým ho klient sám neukončí. Na to slúži konštrukcia s príkazom WHILE. Príklad je tu:

```
var koniec: boolean;           // na ukončenie (ano..nie)
    msg : integer;             // na ukončenie (msg)

begin                          // hlavny program

main_menu ("*menu1");          // menu

while not koniec and get_message(msg) do
begin
    if msg==1001 then
begin
        koniec:=Yesno_box("Otázka", "Ukončiť prácu?");
end
end;

end.
```

Na to, aby sme program odskúšali, musíme v pôvodnom menu zmeniť akciu ukončenia databázy – dajte tam „zaslať programu správu“ a do poľa správy vložte číslo 1001

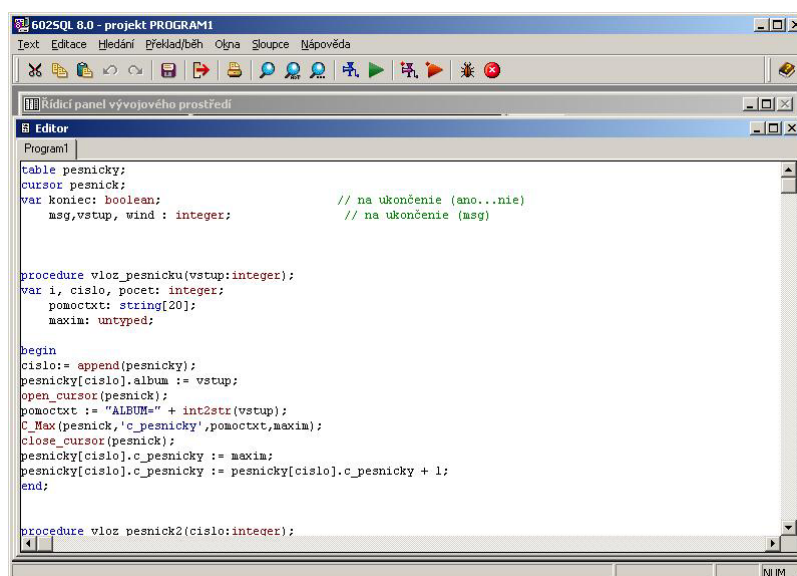
(čísla do 1000 sú rezervované systémom). A samozrejme náš program preložíme a uložíme pod názvom napr. „Program1“ a určíme ho ako štartovací objekt.

Teraz si vytvorený program popíšeme. Na jeho začiatku sa nachádzajú deklarácie globálnych premenných „koniec“ a „msg“. Vzápätí nasleduje príkaz pre načítanie menu a hlavný program s cyklom WHILE. Počas vykonávania cyklu používateľ môže pomocou príkazov menu otvárať formuláre, pracovať s dátami atď. Cyklus neskončí, dokedy nebudú splnené obidve podmienky spojené s AND. Funkcia GET_MESSAGE čaká na správu, ktorú predá programu používateľ. To môže byť stlačením tlačítka prevádzajúceho štandardnú akciu „zaslať programu správu“ vo formulári alebo v menu. Pokiaľ používateľ vyberie v menu položku „Koniec“, programu bude zaslaná správa 1001 a bude vyvolaný príkaz YESNO_BOX. Používateľ bude vyzvaný na odpoveď na otázku „Ukončiť prácu?“. V prípade kladnej odpovede bude do premennej „koniec“ vložená hodnota TRUE a program sa ukončí. Komentáre v programe sa v rámci jedného riadku oddeľujú dvoma lomítkami, pre viac riadkové sa používajú zložené zátvorky.

8.3 PROCEDÚRY V INTERNOM JAZYKU

Keď sme navrhovali formulár „Albumy_podrobne“, spomínal som, že pridávanie záznamov v subformulári „Pesn“ (názvy pesničiek) využitím štandardných funkcií nie je úplne ideálne. Na to, aby do tabuľky „Pesnicky“ bola automaticky vložená správna hodnota do atribútu „album“ ukazujúca do nadriadenej tabuľky „albumy“, musí subformulár obsahovať položku „albumy“. Táto hodnota ale používateľa nezaujíma, navyše blikajúci kurzor musí byť pri stlačení klávesy INSERT umiestnený v niektorom poli subformulára. Inak by systém vložil záznam nie do subformulára, ale do nadriadeného formulára, čo by tiež bežný používateľ neočakával. Tieto nedostatky môžeme odstrániť vytvorením procedúry, ktorej volanie zabezpečí správne chovanie systému.

Najprv upravíme subformulár „Pesn“. Zrušíme v ňom položku „albumy“ a vo vlastnostiach zakážeme vkladanie záznamov – to aby sa nedala použiť kláves INSERT. Vloženie novej pesničky budeme robiť pomocou tlačítka v nadriadenom formulári. To bude mať definované dve akcie: ako prvý príkaz volania procedúry „vloz_pesnicku(cislo)“ – procedúre sa predá parameter „cislo“ a potom štandardnú akciu „prekreslit formulár a synchronizovat subformuláre“ – aby sa informácia o vložení záznamu dostala ku klientovi a prekreslili sa obsahy formulárov.



```

602SQL 8.0 - projekt PROGRAM1
Text Editace Hledání Překlad/běh Ojra Sloupce Nápověda
Ridič panel vývojového prostředí
Editor
Program1
table pesnicky;
cursor pesnick;
var koniec: boolean; // na ukončenie (ano...nie)
msg, vstup, wind : integer; // na ukončenie (msg)

procedure vloz_pesnicku(vstup:integer);
var i, cislo, pocet: integer;
pomoctxt: string[20];
maxim: untyped;

begin
cislo:= append(pesnicky);
pesnicky[cislo].album := vstup;
open_cursor (pesnick);
pomoctxt := "ALBUM=" + int2str(vstup);
C_Max(pesnick, 'c_pesnicky', pomoctxt, maxim);
close_cursor(pesnick);
pesnicky[cislo].c_pesnicky := maxim;
pesnicky[cislo].c_pesnicky := pesnicky[cislo].c_pesnicky + 1;
end;

procedure vloz_pesnick2(cislo:integer);

```

Teraz si zhrňme, čo naša procedúra bude musieť vykonať:

- Vložiť záznam do tabuľky „pesnicky“
- Vložiť v novom zázname hodnotu predanú parametrom „cislo“ do atribútu „album“
- Zistiť, aké je najvyššie poradové číslo pesničky k danému albumu a do atribútu „c_pesnicky“ vložiť číslo o jedničku vyššie

Náš program (aj s procedúrou a potrebnými deklaráciami na začiatku) môže vyzeráť napríklad takto:

```
table pesnicky;
cursor pesnick;
var koniec: boolean;           // na ukončenie (ano...nie)
    msg : integer;           // na ukončenie (msg)

procedure vloz_pesnicku(vstup:integer);

var i, cislo, pocet: integer;
    pomoctxt: string[20];
    maxim: untyped;

begin
    cislo:= append(pesnicky);
    pesnicky[cislo].album := vstup;
    open_cursor(pesnick);
    pomoctxt := "ALBUM=" + int2str(vstup);
    C_Max(pesnick, 'c_pesnicky', pomoctxt, maxim);
    close_cursor(pesnick);
    pesnicky[cislo].c_pesnicky := maxim;
    pesnicky[cislo].c_pesnicky := pesnicky[cislo].c_pesnicky + 1;
end;

begin                               // hlavny program

main_menu("*menu1");                 // menu

while not koniec and get_message(msg) do
    begin
        if msg==1001 then
            begin
                koniec:=Yesno_box("Otázka", "Ukončiť prácu?");
            end
        end;
    end;

end.
```

Procedúru „vloz_pesnicku“ sme vložili medzi deklarácie a hlavný program. Na jej začiatku sú definície lokálnych premenných. Na prvú úlohu – vloženie záznamu – použijeme príkaz APPEND. Príkaz vracia poradové číslo záznamu, ktoré vložíme do premennej „cislo“. Príkazom:

```
pesnicky[cislo].album := vstup;
```

zabezpečíme druhú úlohu. Ako vidíte, na záznamy sa v internom jazyku odkazujeme konštrukciou „nazovtabuľky[cislozaznamu].menoatributu“. Posledná úloha je zložitejšia. Na jej splnenie musíme otvoriť pevný dotaz „pesnick“ a pomocou príkazu C_MAX zistiť maximálnu hodnotu atribútu „c_pesnicky“ za predpokladu splnenia ďalšej výberovej

podmienky (do dotazu sa vyberú len záznamy týkajúce sa konkrétneho albumu – pomocná premenná „pomoctxt“). C_MAX vráti výsledok do premennej „maxim“. Ďalej pre poriadok uzavrieme dotaz a do atribútu „c_pesnicky“ uložíme údaj o jedničku väčší.

Možnosti programovacieho jazyka sú pochopiteľne bohatšie. Podrobnejšie o programoch hovoriť nebudeme, pretože úmyslom tejto brožúry nie je naučiť vás programovať, ale ukázať základy na konkrétnom prípade. Ešte spomeniem, že v elektronickej nápovede nájdete podrobne popísané princípy programovacieho jazyka a encyklopédiu všetkých príkazov delenú jednak podľa abecedného poriadku alebo podľa významu. V prípade problémov je možné nastaviť sa kurzorom na problematický príkaz a klávesou F1 vyvolať kontextovú nápovedu. Okrem toho môžete za účelom odstránenia chýb ladiť programy v debuggeri.

Aj pre túto kapitolu si môžete nové resp. zmenené objekty nájsť v súbore [objekty2.zip](#).

9. PROGRAMY ULOŽENÉ NA SERVERI

9.1 TEÓRIA

Keďže táto časť SQL nebýva tak často popisovaná, doprajme si najprv trocha teórie. Jazyk SQL bol od svojich počiatkov navrhovaný ako neprocedurálny, t.j. zjednodušene jazyk, ktorým popisujeme, aký má byť výsledok a nezaujíma nás postup, ako sa k nemu dopracovať. V najčastejšom príkaze SQL jazyka SELECT popíšeme, že chceme z nejakej tabuľky vybrať určité stĺpce a tie môžu prípadne splniť nejakú podmienku. U procedurálnych jazykov popisujeme presne postupnosť krokov, ktoré sa majú urobiť. Procedurálne jazyky sú rozšírenejšie, delia sa na jazyky interpretované a prekladané, ich zástupcami sú napr. Basic, Pascal, C, Java a napríklad aj interný programovací jazyk 602SQL.

9.2 PROCEDÚRY ULOŽENÉ NA SERVERI

Pôvodný návrh autorov SQL však časom prestal postačovať. Existuje dosť prípadov aj v databázových systémoch, kedy je vhodné použiť procedurálny jazyk. Preto bol do SQL do návrhu normy SQL3 pridaný jazyk procedúr a triggerov. Schopnosť používať uložené procedúry a triggeru nie je v databázových systémoch úplnou samozrejmosťou, napr. MySQL ju v súčasnej stabilnej verzii 3.23 nemá. Uvedme si ako vyzerá text procedúry v 602SQL:

```
PROCEDURE CancelRes (IN lim INT);
BEGIN

  DECLARE PROCEDURE CheckResBook (code CHAR (8));
  BEGIN
    DECLARE pocet INT;
    SELECT COUNT (co) INTO pocet
    FROM Rezervace
    WHERE Rezervace.co = code;

    IF pocet = 1 THEN
      UPDATE Knihy
      SET Knihy.rezervovana = FALSE
      WHERE Knihy.kod = code;
    END IF;
  END;

  FOR radek AS pomc SENSITIVE CURSOR FOR
  SELECT *
  FROM Rezervace
  WHERE CURRENT_DATE - Rezervace.dat_rez > lim
  DO BEGIN
    CALL CheckResBook (radek.co);
    DELETE WHERE CURRENT OF pomc;
  END;
END FOR;
END
```

Táto procedúra v aplikácii Knihovňa (je to jeden z priložených príkladov aplikácií ku 602SQL) zabezpečuje zrušenie rezervácie kníh. Nechcem podrobne rozoberať túto procedúru (nepoznáte aplikáciu ani štruktúru jej tabuliek), len chcem upozorniť na niekoľko skutočností. Na začiatku procedúry je definícia mena a vstupných premenných. Telo procedúry sa začína a končí príkazmi BEGIN a END. V texte procedúry sa nachádzajú príkazy SQL. Štruktúrovanie procedúry sa zabezpečuje riadiacimi príkazmi ako LOOP, IF-THEN, WHILE, FOR atď. V rámci celej procedúry, ktorá je uložená ako samostatný objekt 602SQL je deklarácia ďalšej pomocnej pod-procedúry, ktorá je volaná pomocou príkazu CALL. Samozrejmosťou je aj volanie ďalších procedúr z aplikácie. Podobne ako programy aj procedúry je možné ladiť v debuggeri.

9.3 TRIGGERY

Trigger je ďalší z typov objektov 602SQL. Je to zvláštna procedúra, ktorá je viazaná k určitej tabuľke a začne sa vykonávať automaticky po vzniknutí určitej udalosti v tejto tabuľke. Trigger môže reagovať na zmenu hodnôt v riadku tabuľky a na pridanie alebo vymazanie riadku z tabuľky. Je možné zvoliť, či sa trigger bude spúšťať ešte pred vznikom udalosti (a teda bude pracovať so starými dátami) alebo až po nej.

Triggery zjednodušujú tvorbu aplikácií, lebo prenášajú časť práce databázovej aplikácie na server. Umožňujú napr. centralizované definovanie pravidiel platných pre informačný systém. Napríklad ak existuje v podnikovom informačnom systéme tabuľka zamestnancov, môžeme pomocou triggerov popísať, aké akcie musia byť urobené pri prijatí alebo prepustení zamestnanca, zmene platu atď. Tieto akcie sa naprogramujú na jednom mieste, ale budú slúžiť všetkým aplikáciám, ktoré manipulujú s tabuľkou zamestnancov. Dodržanie pravidiel pre údržbu evidencie zamestnancov tak bude zaisťovať server automaticky a neporušenosť dát bude zaistená bez ohľadu na možné chyby v meniacich sa aplikáciách. Napríklad nasledujúci trigger (opäť prevzatý z príkladov aplikácií) bude zabezpečovať po každom vložení nového riadku do tabuľky „firmy“ zmenu písmen atribútu „nazev“ na veľké:

```
TRIGGER VelkaPismena AFTER INSERT ON Firmy
REFERENCING NEW ROW AS novaradka

FOR EACH ROW
BEGIN
    SET novaradka.nazev=UPPER(novaradka.nazev);
END
```

9.4 PROCEDÚRA VERSUS TRIGGER

Na záver môžeme stručne zhrnúť podobnosti a rozdiely medzi procedúrami a triggerami. Rozdiel je v tom, že triggery sú zviazané s konkrétnou tabuľkou a spúšťajú sa automaticky na základe zmien v tejto tabuľke. Procedúry sa musia spustiť ručne – buď z ďalšej procedúry, z triggeru alebo z interného programovacieho jazyka 602SQL. Podobnosť je v tom, že obe používajú rovnaký jazyk. Ďalšia vlastnosť oboch týchto objektov je, že sa spúšťajú na počítači s 602SQL serverom. Keďže v tomto prípade odpadá zdržujúce

predávanie dát medzi klientom a serverom, vykonávanie tých istých úloh pomocou procedúr je značne rýchlejšie ako v internom programovacom jazyku.

9.5 PRÍKLAD PROCEDÚRY

Teraz si ukážeme ako by vyzerala procedúra volaná z interného jazyka pre našu aplikáciu. Na demonštráciu vytvoríme procedúru, ktorá bude vykonávať to isté, čo procedúra „vloz_pesnicku“ z programu „Program1“ v internom jazyku. Keďže procedúru uloženú na serveri nemôžeme volať priamo z formulára, musíme trochu komplikovane vytvoriť procedúru v internom jazyku „vloz_pesnick2“, odkiaľ už budeme volať procedúru „Proc1“ uloženú na serveri:

```
procedure vloz_pesnick2(cislo:integer);  
  
begin  
vstup:=cislo;  
#sql CALL Proc1(:<vstup);  
end;
```

Nezabudnite ešte v programe definovať globálnu premennú „vstup“ typu INTEGER (premenná prenášajúca sa do SQL musí byť globálna a nie lokálna). Z interného jazyka sa procedúra na serveri volá pomocou príkazu *#sql CALL*. Značka „:<“ znamená, že ide o prenos hodnoty od klienta na server – 602SQL umožňuje prenos oboma smermi, čím rozširuje normu SQL3, ktorá definuje len prenos od klienta k serveru.

Ďalej vytvoríme nový objekt – procedúru na serveri s obsahom:

```
PROCEDURE `PROC1`( IN vstup INT );  
  
BEGIN  
  DECLARE maxim INT;  
  SELECT MAX(c_pesnicky)+1 INTO maxim FROM pesnicky WHERE album = vstup;  
  INSERT INTO pesnicky (album, c_pesnicky) VALUES (vstup, maxim);  
END
```

Na začiatku deklarujeme pomocnú celočíselnú premennú „maxim“. V príkaze SELECT INTO zabezpečíme zistenie maximálneho poradového čísla pesničky k danému albumu do pomocnej premennej „maxim“. Príkazom INSERT vložíme správne hodnoty do atribútov tabuľky „album“ a „c_pesnicky“.

Príklad procedúry pre našu aplikáciu bol veľmi krátky a vôbec nedemonštruje možnosti jazyka procedúr. Preto som v úvode kapitoly použil na ukážku rozsiahlejšiu procedúru z inej aplikácie. Nové resp. zmenené objekty nájdete v súbore [procedury.zip](#).

9.6 SEKVENCIE

Sekvencia je zvláštny objekt – program uložený na serveri, ktorý má za účel generovať postupnosť celočíselných hodnôt. Sekvencie nie sú súčasťou štandardu SQL, prvýkrát boli implementované v systéme ORACLE. V našej aplikácii ich nepoužijeme.

Sekvencie môžu byť použité na generovanie kľúčov v tabuľkách (namiesto udania implicitnej hodnoty pomocou „UNIQUE“). Ich výhoda je v tom, že jedna sekvencia sa dá použiť aj pre viac tabuliek a môžeme ovplyvniť jej hodnoty – v prípade UNIQUE je totiž priradenie úplne v režii SQL servera. Sekvenciu je navyše možné v riadiacom paneli 602SQL reštartovať.

Vytvorenie sekvencie je jednoduché – v riadiacom paneli vytvoríme tlačítkom „Vytvoriť“ novú sekvenciu a zadáme jej meno a parametre – počiatočnú hodnotu, koncovú hodnotu, krok a určíme, či sa generované údaje majú cyklovať.

Na hodnoty sekvencií sa odkazujeme nasledovne (prvý príklad vráti naposledy generovanú hodnotu a druhý vygeneruje a vráti novú hodnotu):

```
meno_sekvencie.CURRVAL
```

```
meno_sekvencie.NEXTVAL
```

10. INTERNET KLIENTI

10.1 INŠTALÁCIA APACHE

Internet klienti 602SQL spolupracujú s webovým serverom, v našom prípade to bude spojenie s Apache prostredníctvom CGI klienta. Pre naše ukážky použijeme Windows verziu Apache bežiacu na tom istom počítači.

Web server môže byť aj inej výroby (výrobcom testované sú Apache, MS IIS a 602Messaging Server), môže bežať aj v inom operačnom systéme a na inom počítači a typ klienta môže byť ešte FastCGI alebo ISAPI. Na spoluprácu je potrebné iba funkčné spojenie protokolom TCP/IP.

Aktuálnu verziu Apache nájdete na stránke www.apache.org a je k dispozícii zdarma. V čase písania tejto brožúry sú k dispozícii stabilné verzie 1.3 a 2.0, použitie oboch je bez problémov. Inštalácia v prostredí Windows je jednoduchá. Počas nej zadáte názov domény (napr. mojadomena.sk), meno servera (napr. mojpcitac.mojadomena.sk) a e-mailovú adresu administrátora (napr. mojaadresa@mojadomena.sk). Tieto údaje pri použití na jednom počítači nie sú dôležité a môžete uviesť ľubovoľné údaje (ale formálne správne). V prípade, že pracujete na systéme založenom na NT, v ďalšom môžete zvoliť, či sa bude Apache spúšťať ako služba. Potom zvolíte typ inštalácie (najlepšie kompletnú) a adresár na inštaláciu – ponechajte štandardný. Vo vytvorenom adresári „C:\Program Files\Apache Group\Apache“ sa vytvorí niekoľko ďalších podadresárov, spomením hlavný (t.j. root) adresár pre dokumenty „htdocs“ a pre súbory konfigurácie „conf“.

Práve editácia konfiguračného súboru httpd.conf v adresári „conf“ je ďalším krokom na sprevádzkovanie Apache a jeho spolupráce s 602SQL. Do tohto súboru vložte v textovom editore (napr. Notepad) na zodpovedajúce miesta text:

```
ScriptAlias /cgi-bin/ "C:/Program Files/Apache Group/Apache/cgi-bin/"
```

a ešte text:

```
<Directory " C:/Program Files/Apache Group/Apache/cgi-bin">  
    AllowOverride None  
    Options None  
    Order allow,deny  
    Allow from all  
</Directory>
```

To platí za predpokladu, že Apache máte nainštalované v štandardne ponúkanom adresári. Posledným krokom je doplnenie cesty k 602SQL do systémovej premennej PATH. V systémoch Windows9x to urobíte vložением textu:

```
Path "C:/Program Files/Software602/602SQL"
```

do súboru autoexec.bat, v systémoch založených na NT (NT4, 2000, XP) to vykonáte vo vlastnostiach ikony Tento počítač, ktorú nájdete na ploche Windows. V systémoch Windows9x bude treba na záver previesť reštart počítača. Po ňom spustíte z menu „Štart -

Programy“ Apache server. Ak sa všetko podarilo ako sa malo, po spustení prehliadača internetu a zadaní adresy <http://localhost> by sa mala zobrazit' autormi Apache vytvorená skúšobná domovská stránka.

10.2 DYNAMICKÁ WWW STRÁNKA

Dynamické stránky sú ďalším typom objektov 602SQL, zobrazujú sa pod názvom „Www objekty“. Ich obsahom je text vo formáte jazyka HTML obohatený o špeciálne príkazy, na základe ktorých 602SQL server doplní do stránok údaje z databázy, pošle web serveru a ten ich pošle ďalej klientovi. Dynamická stránka obecné môže pozostávať z dvoch častí: z WBC konektora a HTW šablóny. Vo WBC konektore sa nachádza popis operácie s databázou (SQL príkazy a volania procedúr v internom jazyku). V HTW šablóne nájde 602SQL popis, ako zostaviť dynamickú stránku na základe údajov prečítaných z databázy pomocou WBC konektora. WBC konektor v rámci HTML je blok začínajúci tagom `<%wbc%>` a končiaci `<%/wbc%>`. HTW šablóna začína tagom `<%begindetail%>` a končí tagom `<%enddetail%>`. WBC konektor môže byť umiestnený hocikde v texte HTML kódu ale samozrejme ešte pred HTW šablónou, ktorá sa naň odkazuje. Napr. nasledujúca stránka zobrazí v tabuľke niekoľko stĺpcov z tabuľky „Albumy“:

```
<HTML>

<HEAD>
<TITLE>Albumy</TITLE>
</HEAD>

<BODY>

<H3>Albumy</H3>

<%wbc%>
SQLStatement: sel_1
+ SELECT Albumy.nazov, Albumy.interpret, Albumy.rok_vydania
+ FROM Albumy
+ ORDER BY Albumy.interpret,Albumy.nazov
<%/wbc%>

<TABLE>
<%begindetail sel_1%>
<TR>
<TD><%`nazov`%>&nbsp;</TD>
<TD><%`interpret`%>&nbsp;</TD>
<TD><%`rok_vydania`%>&nbsp;</TD>
</TR>
<%enddetail%>
</TABLE>

</BODY>

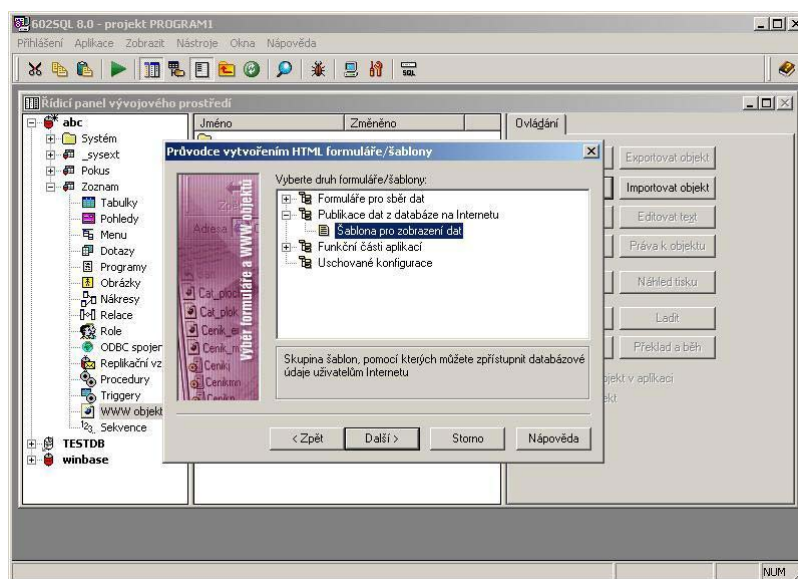
</HTML>
```

Vytvorte novú šablónu (zvoľte voľbu „v textovom editore“). Nahraďte automaticky vygenerovaný text vyššie uvedeným textom a šablónu uložte pod názvom „pokus“. Na vstup do dynamickej stránky vytvorte v Notepade HTML stránku s názvom „vstup.htm“, ktorá bude obsahovať takýto text:

```
<html>
<body>
<A HREF="http://localhost/cgi-
bin/602cgi8.exe/abc/zoznam/pokus.htw">vstúpte</A>
</body>
</html>
```

Názov databázy je „abc“, názov aplikácie v databáze je „zoznam“, vytvorená dynamická stránka má názov „pokus“. Odkaz na údaje zo stĺpcov tabuľky musí byť v bloku begindetail-enddetail a jeho syntax je `<% `nazov_stlpca` %>`.

Uvedený jednoduchý príklad zobrazenia údajov z databázy určite nepotrebuje ďalší komentár. Samozrejme jazyk internet klientov je oveľa bohatší, jeho súčasťou je aj možnosť vytvorenia formulárov na vkladanie údajov do databázy. Vytváranie dynamických stránok je možné pomocou integrovaného textového editora – samozrejme za predpokladu znalosti HTML jazyka a jazyka internet klientov 602SQL. Pre menej znalých je tu možnosť použitia návrhára dynamických stránok. Okno takéhoto návrhára obsahuje v ľavej časti zoznam komponentov a v pravej časti sú tlačítka, pomocou ktorých je možné manipulovať s komponentami.



Ďalšou možnosťou pripojenia na 602SQL je použitie modulu pre Apache `mod_602sql8.dll`. Túto knižnicu je potrebné nakopírovať z `ExtensionPacku` do adresára `C:\Program Files\Apache Group\Apache\modules` a do konfiguračného súboru `httpd.conf` doplniť riadky:

```
LoadModule module_602sql8 mod_602sql8.dll
AddModule mod_602sql8.c
```

```
<IfModule mod_602sql8.c>
Prefix602SQL8 /602/
</IfModule>
```

Klient sa v tomto prípade bude volať cez odkaz:

```
http://localhost/602/abc/zoznam/pokus.htw
```

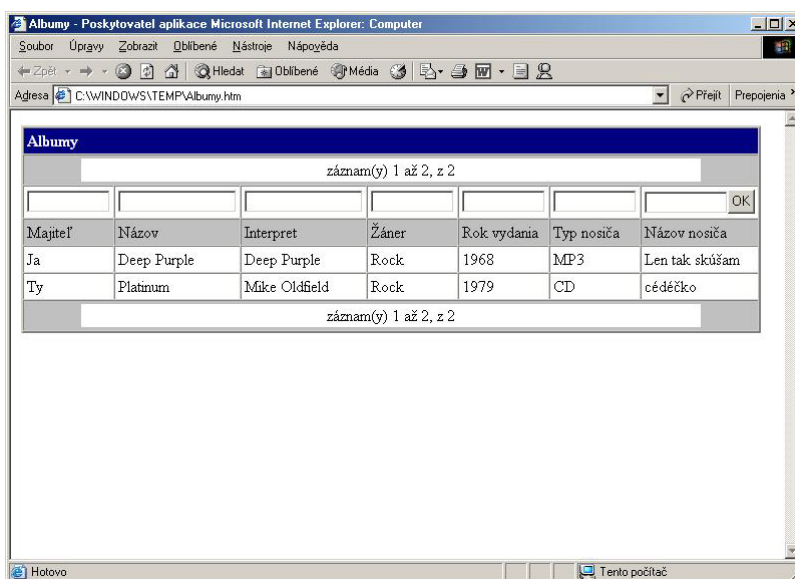
10.3 VYTVORENIE WWW OBJEKTU VO FORMWIZARDE

Pre tých najpohodľnejších je k dispozícii `FormWizard`, pomocou ktorého možno vytvoriť niekoľko základných typov dynamických stránok jednoducho, iba klikaním myšou. `FormWizard` dokáže vytvoriť dynamickú stránku pre niekoľko najčastejšie používaných

prípadov – pre vstup údajov do databázy a pre ich zobrazovanie. Vytvorenie stránky pozostáva z niekoľkých krokov:

- vytvoríme nový objekt typu „Www objekty“ v riadiacom paneli 602SQL
- vyberieme FormWizard ako nástroj pre vytváranie v okne voľby druhu nového objektu
- výber voľby „Šablona pro zobrazení dat“ v skupine „Publikace dat...“
- ako zdroj dát použijeme tabuľku „Albumy“ - nie dotaz „Album2“, pretože v takomto prípade by nám FormWizard neponúkol možnosť vytvárania dotazového poľa. Zdroj dát vo WBC konektore upravíme neskôr ručne
- zo zoznamu zobrazovaných stĺpcov vyberieme všetky okrem stĺpca „cislo“
- vo vlastnostiach u každého stĺpca zatrhneme vlastnosť „dotazové pole“
- v možnostiach zobrazovania môžeme zvoliť zobrazovanie všetkých záznamov alebo zobrazenie postupne po x-záznamoch. Zvolíme druhú možnosť – zobrazovanie po x-záznamoch, doplníme hodnotu, napr. 100
- pokiaľ je v poriadku komunikácia s web serverom, vygeneruje sa stránka a FormWizard ponúkne možnosť zobraziť HTML kód pre sprístupnenie vytvorenej dynamickej stránky – ponechajme možnosť otvorenia v textovom okne
- na výzvu „Uložit parametry pro příští použití“ odpovieme „nie“
- okno so zdrojovým kódom uložíme ako súbor s príponou vstup.htm do koreňového adresára „htdocs“ web servera Apache

Teraz spustite internetový prehliadač a zadajte adresu <http://localhost/vstup.htm>. Výsledok uvidíte na obrazovke. Je to HTML stránka, ktorá obsahuje požadované dáta vo forme tabuľky (tag <TABLE>). V jej záhlaví sú dotazové vstupné polia na zadávanie podmienok vyhľadávania. Zdrojový text www objektu vytvoreného FormWizardom neuvediem v texte celý (je príliš dlhý), ale je k dispozícii v priloženom súbore [www.zip](#).



10.4 ÚPRAVA AUTOMATICKY VYTVORENÉHO OBJEKTU

V ďalšom kroku sa pustíme do úprav www objektu (v textovom režime). Ako prvé upravíme zobrazovanie stĺpcov „zane“ a „typ_nosica“ tak, aby sa nezobrazovali čísla, ktoré reprezentujú odkazy do tabuliek „nosice“ a „zane“, ale príslušné texty. WBC konektor upravíme podľa vzoru dotazu „Album2“. Vygenerovaný WBC konektor vyzerá takto:


```

<%wbc%>
SQLStatement: sel_1[%start%+%count%]
+ SELECT
`majitel`,`nazov`,`interpret`,`rok_vydania`,`nazov_nosica`,`zaner`,`typ_nosica`
+ FROM Albumy
+ WHERE (("majitel_q%" IS NULL)OR(`majitel`.=."%majitel_q%"))
+   AND(("nazov_q%" IS NULL)OR(`nazov`.=."%nazov_q%"))
+   AND(("interpret_q%" IS NULL)OR(`interpret`.=."%interpret_q%"))
+   AND((str2int("%rok_vydania_q%") IS
NULL)OR(`rok_vydania`=str2int("%rok_vydania_q%")))
+   AND(("nazov_nosica_q%" IS
NULL)OR(`nazov_nosica`.=."%nazov_nosica_q%"))
+   AND((str2int("%zaner_q%") IS NULL)OR(`zaner`=str2int("%zaner_q%")))
+   AND((str2int("%typ_nosica_q%") IS
NULL)OR(`typ_nosica`=str2int("%typ_nosica_q%")))
</wbc%>

```

Upravený WBC konektor bude vyzerat' takto:

```

<%wbc%>
SQLStatement: sel_1[%start%+%count%]
+ SELECT
Albumy.majitel,Albumy.nazov,Albumy.interpret,Albumy.rok_vydania,Albumy.nazov_nosica,Zanre.zaner,Nosice.nosic
+ FROM Albumy LEFT OUTER JOIN Nosice ON (Nosice.cislo=Albumy.typ_nosica)
LEFT OUTER JOIN Zanre ON (Zanre.cislo=Albumy.zaner)
+ WHERE (("majitel_q%" IS NULL)OR(majitel.=."%majitel_q%"))
+   AND(("nazov_q%" IS NULL)OR(nazov.=."%nazov_q%"))
+   AND(("interpret_q%" IS NULL)OR(interpret.=."%interpret_q%"))
+   AND(("zaner_q%" IS NULL)OR(Zanre.zaner.=."%zaner_q%"))
+   AND(("rok_vydania_q%" IS
NULL)OR(int2str(rok_vydania).=">

```

Do pozornosti dávam, že okrem úpravy SQL príkazu v časti SELECT som pridal časť ORDER BY a urobil nutné zmeny v časti WHERE súvisiace so zmenou typu stĺpcov (žánre a typy nosičov) z číselných na textové.

Následne bude treba aj v HTW šablóne pozmeniť mená stĺpcov na „zaner“ a „nosic“. Všimnite si, že vo WBC konektore sa názvy stĺpcov zadávajú s názvom tabuľky (napr. „Albumy.nazov“), v HTW šablóne bez názvu tabuľky.

U niektorých stĺpcov bude ešte treba zmeniť spôsob vyhľadávania v dotazovom poli (všade je použité „.=.“, t.j. budú sa vyhľadávať záznamy obsahujúce zadaný (pod)reťazec znakov.

Pokiaľ by vás zaujímali iné aplikácie vytvorené v jazyku internet klientov, v príkladoch aplikácií pre 602SQL nájdete WebFórum. Je to diskusné fórum na internete s možnosťou lokálnej správy (klientský program 602SQL) aj diaľkovej správy (cez internet). Aj niektoré produkty, ktoré predáva Software602 (elektronický obchod, správa obchodných kontaktov atď, sklad dokumentov) sú vytvorené týmto spôsobom.

Taktiež celý elektronický obchod Software602 na internete vrátane nástrojov na generovanie licenčných čísel produktov firmy je tiež vytvorený na báze 602SQL. Ako ste mali možnosť spoznať, jazyk internet klientov je ľahko zvládnuteľný. Napriek tomu - určite z dôvodu značnej obľuby PHP - bol do posledných verzií pridaný modul na napojenie PHP do 602SQL (pre Windows aj Linux), čím sa ďalej rozšírili možnosti použitia 602SQL ako zdroja dát pre internetové aplikácie.

Oba www objekty (upravený aj neupravený), príklad vstupnej stránky ako aj výslednú HTML stránku nájdete v súbore [www.zip](#).

10.5 SPOLUPRÁCA S PHP

Okrem jazyka internet klientov, ktorý je špecifickou záležitosťou 602SQL je možné spolupracovať aj s PHP verzie 4. Pre spozajzdnenie spolupráce je najprv nutné skopírovať do adresára s PHP (C:\PHP\extensions) súbor php_602sql8.dll z ExtensionPacku a do súboru C:\Windows\php.ini doplniť riadok:

```
extension=php_602sql8
```

Potom budeme môcť vypísať údaje napríklad zo spomínanej tabuľky albumov skriptom albumy.php, ktorý umiestnime do koreňového adresára Apache pre dokumenty C:\Program Files\Apache Group\Apache2\htdocs. Skript bude obsahovať nasledujúci text:

```
<?php
$conn=wb_connect("abc", "zoznam", "Anonymous", "");
$res=wb_exec($conn, "SELECT Albumy.nazov, Albumy.interpret,
Albumy.rok_vydania FROM Albumy ORDER BY Albumy.interpret,Albumy.nazov");
wb_result_all($res);
wb_close($conn);
?>
```

Ide o jednoduchú demonštračnú ukážku výpisu dát z tabuľky albumov bez ďalšieho formátovania HTML stránky. Odkazovať sa na takýto PHP skript budeme cez URL adresu:

```
http://localhost/albumy.php
```

11. ZDIEĽANIE DÁT MEDZI 602SQL A INÝMI SYSTÉMAMI

11.1 ODBC

I keď to v našom príklade robiť nebudeme, považujem za užitočné uviesť, že vo vašich aplikáciách môžete použiť aj tabuľky, ktoré sa nachádzajú v iných databázových systémoch (Access, MySQL, Oracle...). Stačí, ak je pre tento systém k dispozícii funkčný ODBC ovládač. ODBC rozhranie bolo vytvorené firmou Microsoft pre Windows, ale v súčasnosti je k dispozícii aj verzia pre Linux. Na počítači, kde bude používaná Vaša aplikácia, musíte mať vytvorený zdroj dát ODBC (vytvára sa v Ovládacích paneloch Windows). Potom v riadiacom paneli 602SQL vytvoríte nové ODBC spojenie. Môžete zvoliť, či sa budú tabuľky iba pripájať alebo ich bude možné v cudzom zdroji aj vytvárať. Potom už stačí v riadiacom paneli 602SQL tlačítkom „Pripojiť“ vybrať potrebnú tabuľku z cudzieho zdroja dát.

Takto pripojenú cudziu tabuľku môžete použiť ako zdroj dát pre formuláre rovnakým spôsobom ako pre vlastné tabuľky. V internom programovacom jazyku sa ale s cudzími tabuľkami pracuje trochu odlišným spôsobom. Odlišnosti sú popísané v elektronickej nápovede.

602SQL vie dáta prostredníctvom ODBC aj poskytovať, t.j. nie je problém dáta z vašich tabuliek alebo dotazov dostať napríklad do Excelu. Ak budete mať záujem, môžete skúsiť aj také „lahôdky“, ako napríklad prostredníctvom prostredia 602SQL sa pripojiť do tabuľky v Accesse, vytvoriť duplikát v MySQL a preniesť do neho všetky dáta pôvodnej tabuľky.

11.2 ĎALŠIE MOŽNOSTI PRÍSTUPU NA SQL SERVER

Pokiaľ by vám v budúcnosti kvôli špeciálnym požiadavkám vývojové prostredie 602SQL nevyhovovalo, môžete vo svojich programoch volať funkcie naprogramované v externých knižniciach DLL. 602SQL tiež môžete použiť len ako dátový sklad a aplikáciu vybudovať v inom systéme. Dá sa použiť hocaký vývojový systém, ktorý sa vie napojiť do 602SQL prostredníctvom ODBC. K niektorým systémom Software602 dodáva špeciálne ovládače a rozhrania, napríklad môžete ešte použiť:

- dbExpress pre Delphi
- rozhranie pre C/C++, Visual Basic, Pascal
- ActiveX pre aplikácie Windows, napr. MS Internet Explorer
- JDBC rozhranie
- ODMA pre prístup k skladu dokumentov

Potrebné súbory, ovládače a príklady nájdete v ExtensionPacku, ich popis sa nachádza tiež v elektronickej nápovede.

12. FULLTEXTOVÉ VYHLÁDÁVANIE

12.1 PODMIENKY POUŽITIA

Fulltextové vyhľadávanie je rozšírením 602SQL, za ktoré sa platí. Na jeho odskúšanie môžete využiť aj dostatočne dlhú 90 dňovú lehotu na otestovanie, počítanú od registrácie licencie pre SQL server.

Fulltextové systémy obecné umožňujú rýchle indexované vyhľadávanie v dokumentoch tak, že systém vyhledá všetky výskyty hľadaného slova a to buď presne podľa jeho zadania alebo bez ohľadu na jeho zadaný gramatický tvar (pád, číslo, osoba...). T.j. napr. pri zadaní textu „jazykom“ nájde aj slovo „jazyk“, „jazyky“, „jazyka“ atď. 602SQL na tieto účely používa slovníky pre korekciu pravopisu z kancelárskeho balíku 602PCSuite. Priamo v inštalácii 602SQL sa nachádza podpora len pre český jazyk. Pre podporu ďalších jazykov je potrebné nainštalovať PCSuite a jeho rozšírenie PlusPack (resp. PCSuite Plus). Postačuje samotná inštalácia bez registrácie (rozšírenie nie je produkt z kategórie „zdarma“), ihneď po inštalácii môžete uvedené balíky odinštalovať – potrebné .dll knižnice v systéme Windows zostanú. Jedná sa o súbory s názvom spl__xx.dll (kde xx je dvojpísmenová skratka jazyka) v adresári „C:\Program Files\Common Files\Soft602\Dict“ alebo „/usr/share/602sql8“ v Linuxe. Druhá a jednoduchšia možnosť rozšírenia počtu jazykov pre fulltext je nakopírovanie týchto súborov do uvedeného adresára. Pri prevádzke 602SQL na sieti stačí inštalovať na počítač, kde beží 602SQL server. Fulltext je plne funkčný na oboch serveroch - pre Windows aj pre Linux.

Fulltext v 602SQL umožňuje indexovať obsah stĺpcov v tabuľkách, ktoré sú buď textových typov alebo typu OLE objekt (vložený v tabuľke alebo pripojený v externom súbore). Tiež umožňuje indexovať obsah dokumentov uložených v súboroch, ktorých mená sú v databáze; ich typ môže byť DOC, WPD, HTML, RTF alebo čistý text. Po vytvorení fulltextu a zaindexovaní dát je možné vyhľadávať slová a frázy v dokumentoch, prípadne aj ich kontext (t.j. slová nachádzajúce sa pred a za hľadanou frázou).

12.2 APLIKOVANIE FULLTEXTU

Podme porozmýšľať, ako by sme mohli obohatiť našu aplikáciu o takéto vyhľadávanie. Keď sa pozriete ešte raz na prehľadový formulár albumov „Album_prehľadovo“, zistíte, že by sa ešte zišla možnosť vyhľadať text, ktorý by sa vyskytoval vo viacerých stĺpcoch (napríklad môžeme chcieť vyhľadať nejaké slovo, ale nebudeme vedieť, či sa vyskytovalo v názve interpreta alebo názve albumu atď.). Samozrejme na toto nebudeme potrebovať fulltext. Stačí do formulára doplniť ďalšie vyhľadávacie pole zduplikovaním z niektorého z pôvodných polí a v jeho vlastnostiach (položka „chování zložky“ - „dotazové pole“) do dotazovej podmienky vložiť text „interpret.=.% OR nazov.=.% OR nazov_nosica.=.%“. Takto sa bude zadaním hľadaného reťazca vyhľadávať text naraz vo viacerých textových poliach formulára.

Teraz sa pokúsime urobiť to isté, ale pomocou fulltextu. Najprv musím zdôrazniť rozdiel medzi vyhľadávaním cez QBE dotazového poľa a fulltextom. V dotazovom poli QBE môžeme zadať aj časť slova, fulltext ale predpokladá, že vyhľadáваме celé slovo. Pri

zapnutej gramatike môžete fulltextu zadať hľadané slovo v ľubovoľnom gramatickom tvare – to samozrejme funguje len vtedy, keď sa dané slovo nachádza v slovníku vybraného jazyka. Z toho vyplýva aj ďalšia vlastnosť – v každom fulltexte je možno použiť súčasne iba jeden jazyk. V našom prípade teda nevyužijeme gramatické tvary z celkom jasného dôvodu – v zbierke máme určite albumy interpretov v rôznych jazykoch.

Ako sa vytvára fulltext? Funkciou „Fulltext_create“ volanou v procedúre. Vytvorte nový objekt - procedúru s názvom „Otvorfulltext“:

```
PROCEDURE `Otvorfulltext`() ;
BEGIN
  CALL Fulltext_create("ZOZNAM.FTX1", "basic_form=0;language=1;weighted=0");
END
```

„FTX1“ je názov fulltextu (v aplikácii ich môže byť viac), „basic_form=0“ znamená, že je gramatika vypnutá, „language=1“ – jazyk je slovenčina (v tomto prípade je to aj tak jedno, keďže sme gramatiku vypli), „weighted=0“ znamená, že sa nebudú rozlišovať váhy výskytu slov. Spustenie tejto procedúry spôsobí zapnutie fulltextu a vytvorenie dvoch pomocných tabuliek s indexami.

Ďalšiu procedúru „Zrusfulltext“, ktorú vytvoríme, budeme používať na zrušenie fulltextu:

```
PROCEDURE `ZRUSFULLTEXT`() ;
BEGIN
  CALL Fulltext_destroy("ZOZNAM.FTX1");
END
```

Na vytvorenie fulltextových indexov zo všetkých vytvorených záznamov použijeme procedúru „Make_ft“:

```
PROCEDURE `MAKE_FT`() ;
BEGIN
  FOR riadok AS cur INSENSITIVE CURSOR FOR
    SELECT * FROM Albumy
  DO
    CALL Fulltext_index_doc('ZOZNAM.FTX1', riadok.cislo, riadok.Nazov+
      " "+riadok.Interpret+" "+riadok.Nazov_nosica, '', 0);
  END FOR;
END
```

Táto procedúra prejde postupne všetky riadky tabuľky a zaindexuje funkciou „Fulltext_index_doc“ všetky texty. Keďže funkcia predpokladá ako vstup len jeden indexovaný reťazec, musíme všetky reťazce sčítať do jedného a medzi ne vložiť medzery, aby nedošlo k nežiadúcemu spojeniu slov.

Ostáva už len ošetriť prípad, ako indexovať pri zadávaní, zmene a rušení záznamov v tabuľke. Tiež použijeme funkciu „Fulltext_index_doc“, ale v triggeri „Zaindexuj“:

```
TRIGGER zaindexuj AFTER UPDATE ON Albumy
REFERENCING NEW AS nove FOR EACH ROW

BEGIN
  DECLARE pocet INT;
```

```
SELECT COUNT(tab_name) INTO pocet FROM tabtab WHERE tab_name =
"FTX_WORDTABFTX1";
IF pocet = 1 THEN CALL Fulltext_index_doc('ZOZNAM.FTX1', nove.cislo,
nove.Nazov+" "+nove.Interpret+" "+nove.Nazov_nosica, '', 0);
END IF;
END
```

Trigger sa musí spúšťať len keď je fulltext zapnutý. V opačnom prípade pri pokuse o zmenu v hociktorom riadku tabuľky obdržime chybovú hlášku a zmeny do tabuľky server nepovolí. Tento problém som vyriešil SELECT-om, v ktorom zisťujem existenciu jednej z pomocných tabuliek, ktoré vytvára fulltext v systémovej tabuľke „tabtab“ (zoznam všetkých tabuliek). Bude to fungovať za predpokladu, že náš fulltext bude mať meno „FTX1“ a nikto ho neskôr nepreimenuje a tiež, že v iných aplikáciách nikto nepoužije fulltext s rovnakým názvom.

Ako sa uskutočňuje vyhľadávanie vo formulároch? Podmienky kladieme do obyčajného dotazového poľa, kde do dotazovej podmienky uvedieme „Fulltext("ZOZNAM.FTX1", cislo, "%)“. Obsahom poľa musí byť stĺpec „cislo“.

Ovládanie fulltextu t.j. jeho zapínanie a vypínanie môžete uskutočňovať v riadiacom paneli vývojového prostredia 602SQL. Aby bol fulltext nastaviteľný aj priamo z aplikácie, bude treba upraviť menu a vložiť do neho patričné príkazy. Keďže procedúry na serveri sa nemôžu volať priamo z menu, budeme ich volať nepriamo cez procedúry v internom jazyku 602SQL.

Použitie fulltextového vyhľadávania môže vhodne doplniť štandardné možnosti vyhľadávania vo vašich aplikáciách. Keďže, žiaľ, nie je zadarmo, jeho využitie je skôr v podnikových informačných systémoch.

Okrem vytvorenia nových procedúr a trigrov budeme musieť zmeniť aj ďalšie objekty - vo formulároch doplniť dotazové pole pre fulltext, do programu príkazy na vytvorenie a indexovanie fulltextu a do menu príslušné odkazy. Všetky zmenami dotknuté objekty nájdete v súbore [fulltext.zip](#). Po importe nových objektov resp. nahradení starých novými verziami musíte fulltext zapnúť príslušným príkazom v menu.

Týmto sme sa dostali až k záveru vytvárania našej aplikácie. Celú ju nájdete v priloženom súbore [albumy.zip](#). Aplikácia má názov „Albumy XP“ a môžete ju voľne šíriť a upravovať.

13. BEZPEČNOSŤ

13.1 SYSTÉM PRIDEĽOVANIA PRÁV

Naša aplikácia „Albumy XP“ má predpoklad využitia skôr jedným používateľom na jednom počítači. V tomto prípade nie je potrebné sa ochranou dáť alebo aplikácie zaoberať. Dokonca sa parametrami príkazového riadku v odkaze spúšťajúcom aplikáciu dá prihlasovací dialóg úplne potlačiť. Vytvorte si zástupcu na súbor klienta 602cli8.exe a upravte jeho vlastnosti pomocou nasledujúceho vzoru:

```
"C:\Program Files\Software602\602SQL\602cli8.exe" &abc zoznam /S /L
```

Klient sa po štarte pokúsi spojiť s databázou zaregistrovanou pod názvom „Abc“. Ak nie je registrovaná ako sieťová, spustí zároveň SQL server „602sql8.exe“ v tzv. lokálnom režime. Vtedy server neprijíma požiadavky od sieťových klientov. „Zoznam“ je názov aplikácie, ktorá sa pri štarte vyberie. Parameter /S znamená, že sa má spustiť štartovací objekt aplikácie a /L, že sa uplatní pri prihlasovaní účtu používateľa „Anonymous“.

Iná situácia nastane pri nasadení zložitejších aplikácií určených pre viacerých používateľov. V takomto prípade je vysoko pravdepodobné, že používateľov bude nutné rozdeliť na skupiny, ktoré sa budú líšiť právomocami. Popíšme si, aké sú možnosti 602SQL v tejto oblasti a ukážme postup, ako Vašu prácu ochrániť pred zneužitím.

602SQL umožňuje pridelovať práva k objektom aplikácie (použitie, modifikácia, rušenie a poskytovanie práv) a k dátam tabuliek (čítanie, prepisovanie, vkladanie, rušenie záznamov a poskytovanie práv). Práva k dátam je možné pridelovať k celej tabuľke, ku stĺpcom tabuľky a dokonca aj k riadkom tabuľky. Práva je možné ľubovoľne kombinovať, výsledok bude daný ich logickým súčtom.

Zaradenie používateľa do skupiny je platné v rámci celej databázy. Existujú 4 skupiny, ktoré sú vytvorené automaticky:

- *Everybody* - skupina, do ktorej patria všetci používatelia
- *Db_admin* - skupina dátových správcov
- *Config_admin* - skupina prevádzkových správcov
- *Security_admin* - skupina bezpečnostných správcov

Ak je pridelené nejaké právo skupine *Everybody*, získajú toto právo automaticky všetci používatelia. Člen skupiny *Db_admin* má všetky práva k dátam a objektom databázy. Člen skupiny *Config_admin* má práva týkajúce sa prevádzky servera (napr. vytváranie používateľov, nastavovanie parametrov servera, kontrola a oprava databázy atď). Člen skupiny *Security_admin* má právo meniť nastavenia, týkajúce sa bezpečnosti (nastavovať šifrovanie, meniť heslá...).

Zaradenie používateľa do rolí je platné len v konkrétnej aplikácii. Podobne ako u skupín, aj u rolí existujú 4 štandardne definované:

- *Administrator* – má všetky práva k dátam aplikácie

- *Author* – má práva na modifikáciu objektov
- *Junior_user* – má práva na čítanie dát
- *Senior_user* – má práva na zápis dát

Ten používateľ, ktorý aplikáciu vytvoril je automaticky obsadený do rolí *Administrator* aj *Author*.

Práva možno prideliť používateľom, skupinám a roliam. Aby to nebolo také jednoduché, do rolí je možné obsadiť aj skupiny.

Pokiaľ Vám preddefinované skupiny a role nestačia, môžete si vytvoriť ďalšie podľa potreby. Podotýkam, že z hľadiska jednoduchosti správy používateľov je vždy potrebné uprednostňovať pridelovanie práv skupinám a roliam a nie priamo používateľom. Pri zmene pracovného zaradenia niektorého používateľa stačí ho preradiť do inej skupiny alebo role, pri zmenách oprávnení určitej množiny používateľov stačí upraviť právomoci dotknutých skupín alebo rolí. V opačnom prípade by správca musel používateľove práva meniť pre každý objekt v aplikácii zvlášť, čo je riešenie omnoho zdĺhavejšie a náchylnejšie k chybám.

Okrem uvedených práv k dátam môžete ochrániť svoju prácu (myšlienky a postupy) pred odcudzením tým, že aplikáciu exportujete v zašifrovanom tvare. Potom pôvodnú aplikáciu zmažete a importujete už ako zašifrovanú. V takomto stave nie je umožnené nikomu prečítať zdrojové texty všetkých objektov ani ich modifikácia, pričom 602SQL klienti pracujú normálne - automaticky ich dešifrujú. Pripomínam, že nesmiete zabudnúť urobiť zálohu aplikácie v nezašifrovanom stave.

13.2 ZABEZPEČENIE SERVERA

Ak plánujete vytvárať aplikácie určené pre sieťovú prevádzku, mali by ste na novom databázovom súbore vykonať túto postupnosť krokov:

- Prihláste sa ako *Anonymous* (inak sa prvýkrát ani nedá) a vytvorte nového používateľa.
- Vytvorte nového používateľa (budúceho administrátora databázy) a pridajte ho do skupiny *Db_admin*.
- Podobným spôsobom vytvorte aj prípadných členov skupín *Config_admin* a *Security_admin*.
- Používateľa *Anonymous* vyradíte zo všetkých štyroch preddefinovaných skupín.

Počas prevádzky servera je možné podľa potreby vykonávať tieto dodatočné opatrenia na zvýšenie bezpečnosti:

- Môžete rozdeliť administrátorské funkcie (*Db_admin*, *Security_admin* a *Config_admin*) medzi viaceré osoby.
- V každej nanovo vytvorenej aplikácii môžete vyradiť skupinu *Everybody* z role *junior_user* (toto priradenie 602SQL vykonáva automaticky po vytvorení novej aplikácie) a dôsledne skontrolovať obsadenie štandardných rolí používateľmi a skupinami.

- Pre prípad technickej poruchy (havária disku...) alebo omylu oprávneného používateľa je možnosť zapnúť pravidelné zálohovanie databázy, žurnál zmien a istenie transakcií. Pozor – istenie transakcií spomaľuje server.
- V prípade nemožnosti zabránenia prístupu nepovolaným osobám k databázovému súboru na úrovni operačného systému sa dá jeho obsah zašifrovať. 602SQL pozná viac úrovní, avšak čím kvalitnejšie šifrovanie, tým viac spomaľuje server.
- Na znemožnenie čítania alebo zmeny údajov prenášaných po nechránenej sieti (napr. internete) sa dá zapnúť šifrovanie sieťovej komunikácie .
- Je možnosť nastaviť overovanie identity servera.
- V aplikáciách môžete používať digitálne podpisy s certifikátmi X.509
- Dokážete zakázať prístup anonymných používateľov.
- Dá sa obmedziť rozsah IP adries, z ktorých sa možno na server prihlásiť.
- Môžete prikázať minimálnu dĺžku hesla, alebo nastaviť expiráciu hesla.

Vytvorenie nového správcu resp. viacerých správcov namiesto anonymného používateľa je pre zabezpečenie servera základný krok. Musíte si uvedomiť, že po vytvorení databázy táto nie je nijako zabezpečená, do databázy sa môže hocikto prihlásiť bez znalosti hesla, pričom disponuje všetkými právami.

Člen skupiny *Db_admin* má úplne všetky práva k dátam databázy. Pre dokonalejšie zabezpečenie je vhodné rozdeliť jeho činnosti medzi viacerých používateľov – správcov alebo autorov jednotlivých aplikácií a do skupiny *Db_admin* neobsadiť nikoho. Správcovské činnosti môžu vykonávať členovia štandardných skupín *Config_admin* a *Security_admin*, prípadne ďalších skupín a členovia rolí v konkrétnych aplikáciách.

Nastavovať bezpečnostné parametre je možné z klientskeho vývojového prostredia (vrátane príkazov programovacieho jazyka) a v nástroji „Řídící konzole 602SQL“, ktorá je k dispozícii pre Windows aj pre Linux.

Problematika bezpečnosti bola vysvetlená len v stručnosti a oprávnené si zaslúži viac Vašej pozornosti. Ďalšie podrobnosti sa dozviete v elektronickej nápovede 602SQL.

14. POROVNANIE 602SQL S INÝMI SQL SERVERAMI

Aby bolo možné poukázať na výhody resp. nevýhody použitia 602SQL ako riešenia pre tvorbu Vašich aplikácií uvádzam porovnávaciu tabuľku vlastností 602SQL a niektorých ďalších SQL serverov podobnej rozsiahlosti a výkonu, vypracovanú na základe zdrojov z internetu.

Ako vidno z obsahu tabuľky, vlastnosti vybraných serverov sú približne porovnateľné. Každý server je lepší v niečom inom, z komplexného hľadiska veľké rozdiely nie sú. 602SQL má ale jednoznačnú výhodu, spočívajúcu v integrovanom vývojovom prostredí pre vytváranie klientských aplikácií – nie je totiž len SQL serverom, ale kompletným databázovým systémom. Takto vytvorená aplikácia sa ovláda ako každý iný program pre Windows. U konkurenčných SQL databáz (a netýka sa to iba tu uvedených) je pre túto činnosť nutné použiť ďalšie a navyše platené produkty.

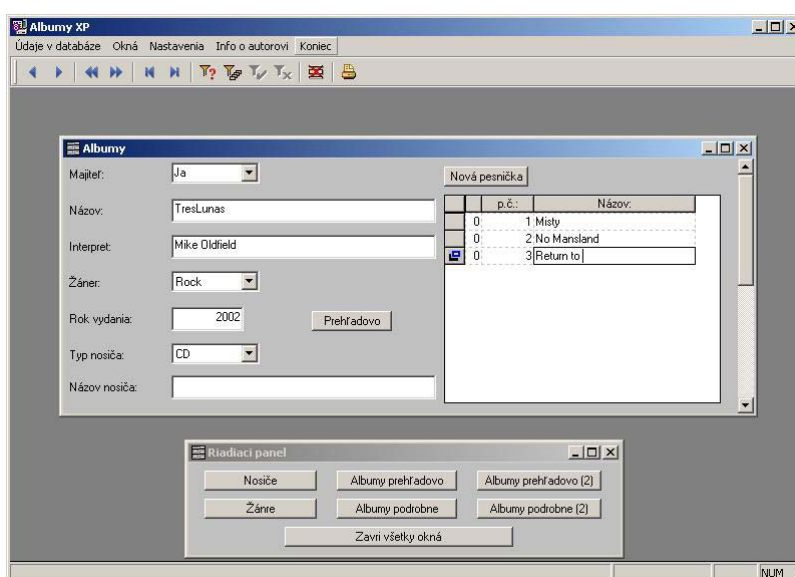
Vlastnosť\Meno	602SQL 8.0	MySQL 3.23	InterBase 6.5	PostgreSQL 7.2
Licencie	1 zdarma, >1 platené	GPL alebo komerčné platené	Platené	BSD
Platformy	Win32, Linux	Linux, BSD, OS/2, Win32	Linux, Solaris, Win32	Linux, BSD, BeOS, Mac, QNX, Win32
Programovanie	Integrované, ODBC, JDBC, C++, Delphi, Kylix, Pascal, PHP, ActiveX, dbExpress	ODBC, JDBC, C/C++, OLEDB, Delphi, Perl, Python, PHP	ODBC, JDBC, C/C++, Delphi, C++ Builder, Embedded SQL	ODBC, JDBC, C/C++, Embedded SQL (C), Tcl/Tk, Perl, Python, PHP
SQL štandard	SQL92 Intermediate level s niektorými časťami SQL99	SQL92 Entry level, niektoré časti SQL92 Intermediate level a SQL99	SQL92 Intermediate level s niektorými časťami SQL99	SQL92 Intermediate level s niektorými časťami SQL99
Práva k dátam	Aj k stĺpcom aj k riadkom, 3 úrovne	K stĺpcom	K stĺpcom, 2 úrovne	Atypické – podľa typu akcií a miesta prístupu
Č+S triedenie a vyhľadávanie	Áno	S problémami	S problémami	S problémami
UNICODE	Áno	Nie, bude v 4.0	Áno	Áno
Procedúry na serveri	Áno	Nie, bude v 4.0	Áno	Áno
Triggery	Áno	Nie	Áno	Áno
Zabezpečené spojenie	Áno	SSL	Nie, iba heslá	SSL, SSH, Kerberos
Zabezpečený databázový súbor	Áno	Čiastočne	Nie	Nie
Indexy	B-tree, Fulltext	B-tree	B-tree	B-tree, R-tree, Hash, Gist
Primárny kľúč v indexovaní	Áno	Čiastočne	Áno	Áno
Podpora transakcií	Áno, 2 úrovne	Čiastočne	Áno, 2 úrovne	Áno, 2 úrovne
Podpora SUB SELECTov	Áno	Nie, bude v 4.0	Áno	Áno
Software v dodávanom balíku	Windows klient, vývoj. prostredie, podpora pre e-mail	Administrátorská graf. konzola	Administrátorská graf. konzola	Administrátorská graf. a web konzola

ZÁVER

Táto brožúra mala za úlohu naučiť vás pracovať so systémom 602SQL a pripraviť na jeho intenzívnejšie využívanie. Pevne verím, že sa mi to podarilo.

Každý autor - programátor aplikácií sa potrebuje neustále vzdelávať. Preto na záver ešte uvediem zopár odkazov na zdroje, ktoré som pri tvorbe tejto brožúry použil a ktoré môžu poslúžiť aj Vám na ďalšie vzdelávanie:

- elektronická nápoveda 602SQL, je k dispozícii v CHM aj v HTML formáte
- príklady aplikácií pre 602SQL z Extension Pack-u
- informácie na internete na stránkach Software602, napr.:
 - stránky produktu 602SQL www.software602.cz/produkty/602sql8.htm
 - stránky určené pre vývojárov 602SQL www.software602.cz/602sql/ alebo WinBase602 www.software602.cz/winbase/
- konferencia o WinBase602/602SQL www.pandora.cz/conference/winbase
- seriál „WinBase602 – český SQL server“, www.zive.sk, 2001-2002, autor Roman Riedl
- seriál „SQL server po česku“, www.dbsvet.cz, 2002, autor František Skřivánek
- kniha „SQL kompletní kapesní průvodce“, autor Milan Šimůnek, GRADA 1999
- seriál „Databáze standardu SQL“, časopis Chip, autor Jaromír Kukul, 1998-1999
- seriál článkov o jazyku SQL, www.interval.cz, 2000-2001, autor Jaromír Skřivan



Poznámky: