DUM 13 téma: Příkazy pro výběr

ze sady: 3 **tematický okruh sady:** III. Databáze
ze šablony: 7 – Kancelářský software **určeno pro:** 4. ročník
vzdělávací obor: 18-20-M/01 Informační technologie
vzdělávací oblast: odborné vzdělávání
metodický list/anotace: viz VY_32_INOVACE_07313ml.pdf

Čtení obsahu databáze

Základní operací s databází a jejími tabulkami je čtení jejího obsahu. U relačních databází, vedených v některém systému řízení báze dat (SŘBD) nemá uživatel databáze přímý přístup k samotným datovým tabulkám, ale vhodně formátovaným dotazem si může jejich část vyžádat. Výstup pak systém poskytuje také ve formě tabulky, ovšem pouze dočasné a dostupné pouze autorovi dotazu.

Při dotazování do databází pak chceme získat z celé tabulky (či skupiny tabulek) vždy pouze jejich část – několik sloupců a/nebo řádků. Hovoříme pak o databázové selekci a databázové projekci.

Databázová projekce

Je výběrem jednoho či více sloupců (!) z tabulky či skupiny tabulek v databázi. Jazyk SQL pro tuto činnost používá příkaz SELECT – který je patrně nejznámějším a nejpoužívanějším příkazem tohoto jazyka.

Základní syntaxe příkazu je:

SELECT *sloupec,sloupec…* FROM *tabulka*

Kde seznam vybíraných sloupců musí korespondovat s existujícími sloupci ve zdrojové tabulce, ovšem nikoliv pořadím. Ve výstupu pak lze určit pořadí sloupců dle vlastní potřeby. Pokud chceme získat do výstupu všechny sloupce, bez uvádění jejich názvů, můžeme místo jejich seznamu zapsat zástupný znak „*“. Dotaz pak vypadá takto:

SELECT * FROM *tabulka*

Ačkoliv vypadá takový zápis jednoduše a elegantně, není příliš vhodné v praktických aplikacích takové zjednodušení používat, pokud k tomu není zvláštní důvod. Je nutné si uvědomit, že databázové sloupce mohou obsahovat velké množství dat (klidně i fotografie, celé filmy atd. – viz datové typy ve sloupcích) a nechávat si zaslat veškerá data z tabulky, která ale ani nepotřebujeme, zbytečně zatěžuje jak samotný databázový server – tak i komunikační linky a stroj na kterém provozujeme dotazující se aplikaci.

Databázová selekce

Je naproti tomu výběrem určitých řádků (datových vět) z tabulky, bez ohledu na související výběr řádků. Jazyk SQL nicméně obě operace spojuje a, logicky, pro selekci

vyžaduje provedení projekce. Samotný výběr řádků by totiž neměl smysl, pokud by uživatel neurčil, že chce vůbec nějaké datové sloupce získat.

Pro selekci (!) používáme příkaz WHERE, označovanou častěji jako klausule – právě proto, že se nedá použít samostatně, ale pouze v kombinaci s jiným příkazem. V následujících příkladech ho budeme kombinovat právě se SELECT:

SELECT *sloupce* FROM *tabulka* WHERE *podmínka*

Důležité je zmínit, že nejdříve databázový server vyhodnotí právě podmínkovou část (WHERE) a teprve poté provede projekci (SELECT). Podmínka samotná tak může obsahovat sloupce, které se ve výsledku vůbec neobjeví (SELECT je nepožaduje) – typicky různé pomocné identifikátory, které koncový uživatel vidět nemusí.

Konstrukce podmínek je odlišná v závislosti na datovém typu sloupce, podle kterého selekci provádíme. Nejjednodušší je varianta s číselnými sloupci (typy INT a podobně), kde přímo použijeme matematické operátory, známe například z jazyků C či PHP:

SELECT *jmeno, prijmeni* FROM *zamestnanci* WHERE *vek=50*

Zde ovšem používáme pro přímou shodu znak jednoduchého „=“. Nepíšeme totiž program a samotné „=“ nemá význam přiřazení. Obdobně lze využít i <,>,=>,=< a podobně. Neshoda se vyjadřuje většinou „! =„ některé implementace SQL přijmou i „<>“.

Kromě těchto operátorů máme možnost využít i zvláštních operátorů:

IN … za kterým následuje seznam vyhovujících hodnot, příklad:

SELECT *jmeno, prijmeni* FROM *zamestnanci* WHERE *vek IN (25,50,75)*

BETWEEN … obsahující hraniční hodnoty (které samotné jsou také považovány za vyhovující), podobně jako kombinace >= a <=:

SELECT *jmeno, prijmeni* FROM *zamestnanci* WHERE *vek BETWEEN 45 AND 50*

Při použití podmínek s alfanumerickými hodnotami používáme pro porovnávání nejčastěji direktivu LIKE, která dokáže porovnávat řetězce i s využitím zástupných znaků. Na rozdíl od obvyklých zvyklostí, v jazyce SQL používáme jako náhradu libovolného počtu znaků (tedy i žádného) znak „%“ a jako zástupný právě jednoho znaku pal „_“. Sestavený dotaz pak vypadá například:

SELECT *jmeno, prijmeni* FROM *zamestnanci* WHERE *jmeno* LIKE „Da%“

Vyhoví jakékoliv jméno začínající „Da…“ tedy David, Dana, Dan. Naopak úprava:

SELECT *jmeno, prijmeni* FROM *zamestnanci* WHERE *jmeno* LIKE „Dan%“

Omezí výběr nejen na „Dan“ ale i „Dana“ či „Daniela“ a „Danuše“… kdežto

SELECT *jmeno, prijmeni* FROM *zamestnanci* WHERE *jmeno* LIKE „Dan_“

Vyhoví už jen „Dana“

Jako zvláštní variantu podmínky můžeme testovat i hodnotu „NULL“ – tedy nevyplněnou hodnotu (prázdnou). Nelze ji zaměňovat za číselnou „0“ ani prázdný řetězec! Samozřejmě má taková podmínka smysl pouze u sloupců, kde je strukturou databáze dovoleno ho nevyplnit. Například:

SELECT *jmeno, prijmeni* FROM *zamestnanci* WHERE *pracoviste* IS NULL

Tuto zvláštní hodnotu naopak není možné testovat žádnou z podmínek >, < a podobně.

Podmínky je dále možné i negovat (IS NOT NULL) či spojovat logickými spojkami OR a AND (WHERE jmeno LIKE „Tom%“ OR jmeno LIKE „Jan%“) a podobně.

Řazení

Pořadí řádků ve výstupu můžeme ovlivnit direktivou ORDER BY na konci dotazu. Následuje jméno sloupce, či sloupců, podle kterých se má řazení provádět. Standardně dle abecedy či od nejmenšího k největšímu; po doplnění „DESC“ pak v opačném pořadí.

Typ národní abecedy, který se pro řazení použije, vybírá autor databáze při její tvorbě. Pozor – většina standardních nastavení databázových serverů umisťuje znaky s diakritikou až za poslední znak běžné abecedy (tedy „Á“ je až za „Z“).

Příklad:

SELECT *jmeno, prijmeni* FROM *zamestnanci* ORDER BY *prijmeni*

Úkoly pro samostatnou práci

Vytvořte a jakýmkoliv způsobem naplňte tabulku „auta“:

id_auta	znacka	barva	rok_vyroby	karoserie
100	Škoda	bílá	2013	combi
200	Ford	černá	2001	liftback
300	Škoda	žlutá	1993	sedan
400	Dacia	černá	2009	combi
500	Škoda	červená	1999	liftback
600	Audi	zelená	2009	supersport
700	Ford	bílá	2003	sedan

- sestavte příkaz, který zobrazí všechny vozy z roku 2000 a novější
- sestavte příkaz, který zobrazí pouze vozy Škoda a Audi
- sestavte příkaz, který zobrazí pouze vozy barvy začínající na „č“
- sestavte příkaz, který zobrazí tabulku seřazenou od nejnovějšího po nejstarší vozidlo
- sestavte příkaz, který zobrazí vozy typu sedan a supersport

Zdroje:

- Archiv autora