

Jazyk SQL (rozdělení SQL, příkaz SELECT, projekce, restrikce, agregace, druhá restrikce)

Jazyk SQL

Strukturovaný dotazovací jazyk (Structured Query Language). Vyvinula jej firma IBM. Jedná se o neprocedurální jazyk – sdělujeme, co chceme získat, ale ne jak. Dotaz i odpověď jsou textové řetězce. Nezáleží na velikosti písmen příkazů. Jazyk lze s menšími změnami použít ve většině databázových systémů (Oracle, SQL Server, MySQL, Postgre).

Dotazy mohou vést k vytváření nových databází, tabulek, indexů, pohledů, triggerů, procedur, uvedené struktury mohou dotazy modifikovat, rušit nebo aktualizovat hodnoty dat v tabulkách.

Rozdělení SQL

DDL

Data Definition Language. Definice struktur, vytváření a rušení objektů a změna jejich struktury. Patří sem příkazy **CREATE** DATABASE/TABLE/INDEX/VIEW/SEQUENCE, **ALTER** TABLE/VIEW/SEQUENCE, **DROP** TABLE/INDEX/VIEW, ...

DML

Data Manipulation Language. Umožňuje výběr, vkládání, mazání řádků a aktualizace. Patří sem příkazy **SELECT**, **INSERT**, **DELETE** a **UPDATE**.

DCL

Data Control Language. Speciální příkazy pro řízení provozu a údržbu databáze či např. změnu uživatelských privilegií. Příkazy **CREATE/ALTER/DROP** USER, **GRANT**, **REVOKE**.

Příkaz SELECT

Příkaz SELECT umožňuje z databáze vybrat/načíst nějaká data. Jeho základní syntaxe je:

```
SELECT popis projekce
FROM tabulka
WHERE podmínka restrikce řádků
GROUP BY klíč agregace
HAVING podmínka restrikce skupiny
ORDER BY klíč třídění;
```

Projekce

Projekce je ve své podstatě určení dat, která se mají vybrat – může např. obsahovat jednotlivé názvy sloupců v tabulce oddělené čárkami, jejichž hodnoty mají být načteny nebo znak * určující, že se mají vybrat hodnoty ze všech sloupců. Do projekce je možné přidat také návratovou hodnotu vykonané agregační funkce. Ve výsledku může projekční část SQL řetězce vypadat například následovně:

```
SELECT * FROM tabulka;  
SELECT id, username, password FROM tabulka;  
SELECT username, CONCAT(firstname, " ", surname) AS fullname FROM tabulka;
```

Restrikce

Restrikce umožňuje omezit výběr dat jen na podmnožinu všech dat. Jde o podmínky, které musí být při dotazu `SELECT` splněny, aby byl řádek zahrnut do návratového řetězce příkazu `SELECT`. Řádek je vybrán, když podmínka (logický výraz) nabývá hodnoty `TRUE`.

Podmínka se skládá z kombinace logických výrazů, logických spojek `AND`, `OR` a `NOT` a kulatých závorek. Logické výrazy lze sestavovat z názvů sloupců, z relačních operátorů či ze speciálních klíčových slov. V jednotlivých částech mohou vystupovat také základní matematické operátory.

Operátor `LIKE` slouží pro porovnávání řetězců. Navíc nabízí možnost jednoduchých výrazů (znak `%` nahrazuje libovolný počet znaků a znak `_` libovolný jeden znak. `NOT LIKE` = opak `LIKE`.

Hodnota `BETWEEN min AND max` zjistí, zda je hodnota větší nebo rovna `min` a zároveň menší nebo rovna `max`. `NOT BETWEEN` = opak `BETWEEN`.

`IN` (seznam hodnot) hledá hodnoty dle zadaného seznamu (např. `SELECT * FROM student WHERE vek IN(15, 16, 17)`; – najde studenty ve věku 15, 16 a 17 let. `NOT IN` = opak `IN`.

Agregace

Agragace = *shlukování*. Databáze nabízí spoustu tzv. agregačních funkcí. To jsou funkce, které zpracovávají několik hodnot a jako výsledek vrátí hodnotu jednu. Agregační funkce se používají přímo při použití příkazu `SELECT`. Funkce se nad řádky vykoná a výsledek je `SELECTEM` vrácen. Stejně jako v jiných programovacích jazycích mají tyto funkce závorky a případně parametry.

Druhá restrikce

Klauzule `HAVING`. Definuje podobnou filtrační podmínku na řádky jako klauzule `WHERE`. Rozdíl mezi nimi tkví v tom, že `WHERE` definuje podmínku, která se aplikuje na řádky před seskupením, zatímco `HAVING` definuje podmínku, která se aplikuje na skupiny řádků po dokončení jejich seskupení dle klauzule `GROUP BY`.

V klauzuli `HAVING` lze použít jen agregační funkce nebo konstanty.