

Historie SQL

- IBM vyvinula sadu příkazů pro ovládání databází **SEQUEL** (Structured English Query Language)
- Oracle Corporation přinesla svou platformu **Oracle Database**
- IBM uvedla **SQL/DS** a **DB2**
- Další systémy - **Progres**, **Informix** a **SyBase**
- Všechny používaly varianty jazyka **SEQUEL**, který se později přejmenoval na **SQL**

Datové typy v SQL

Základní kategorie

- Číselné typy
- Řetězcové typy
- Typy pro datum a čas

Číselné typy

- TINYINT
- SMALLINT
- MEDIUMINT
- INT
- BIG INT
- FLOAT
- DOUBLE
- DECIMAL

Číselné typy se liší možnou velikostí uchovávaného čísla nebo jeho desetinnými čísly.

Řetězcové typy

- CHAR
- VARCHAR
- TINYBLOB
- BLOB
- MEDIUMBLOB
- LONGBLOB
- TINYTEXT
- TEXT
- MEDIUMTEXT
- LONGTEXT

Řetězcové typy se liší možným počtem uchovaných znaků nebo porovnáváním jejich velikostí. (některé porovnávají některé ne).

Datумы a časy

- DATE
- TIME
- DATETIME
- TIMESTAMP
- YEAR

Liší se formátem zapisování. TIMESTAMP do sebe vloží aktuální čas při změně záznamu, ve kterém se nachází.

Modifikátory

- **NULL**
 - Sloupec může zůstat prázdný
- **NOT NULL**
 - Sloupec nemůže zůstat prázdný
- **UNSIGNED**
 - Sloupec nemůže mít záporné hodnoty a může tak obsahovat více čísel v plusových hodnotách
- **SIGNED**
 - Sloupec může obsahovat záporné hodnoty, sníží se jeho nejvyšší možná hodnota v kladných hodnotách
- **AUTO_INCREMENT**
 - Každý záznam přičítá k poslednímu 1 a automaticky se tak s každým přidaným záznamem zvedá hodnota v daném sloupci
- **DEFAULT**
 - Označuje jakou hodnotu bude mít sloupec, pokud do něj žádnou hodnotu nedosadíme

Integritní omezení

Udržují databázi ve stavu, kdy jsou ve všech polích pravdivé a validní informace. Integritní omezení způsobí například to, že když je do databáze vložen záznam s prázdným polem a daný sloupec má nastaveno omezení NOT NULL, tak databáze daný záznam odmítne.

- **DEFAULT**
- **NOT NULL**
- **PRIMARY KEY**
 - Hodnota primárního klíče musí být jedinečná

- **FOREIGN KEY**
 - Hodnota cizího klíče musí být jedinečná
 - Odkazuje na záznamy z jiných tabulek(relací)
- **UNIQUE**
 - Hodnota daného sloupce musí být jedinečná

Příkazy pro práci s databází, tabulkami a sloupci

- CREATE DATABASE db_name;
- USE db_name;
- DROP DATABASE db_name;
- CREATE TABLE table_name (column_name column_datatype column_modifiers);
- ALTER TABLE table_name (DROP/ALTER/MODIFY) column_name column_datatype;
- DROP TABLE table_name;

Příkazy pro manipulaci s daty

- INSERT INTO table_name (column_name, column_name) VALUES (value, value)
- UPDATE table_name SET column_name = value WHERE condition;
- DELETE FROM table_name WHERE condition;

Příkaz SELECT

- Slouží pro sběr dat z databáze a sesbírané data zobrazí příkaz v tabulce výsledků
- SELECT (column_name or * or SQL aggregate function) FROM table_name WHERE condition;
 - * označí všechny sloupce v dané tabulce
 - Column_name označí pouze daný sloupec v dané tabulce
 - Do příkazu se může dát také SQL funkce, například funkcí COUNT() můžeme spočítat počet záznamů (COUNT(*))

Spojování tabulek

- Slouží ke spojení záznamů z dvou nebo více tabulek, na základě pojení pomocí cizích klíčů
- Používá se k tomu příkaz JOIN

```
SELECT Orders.OrderID, Customers.CustomerName, Orders.OrderDate
FROM Orders
INNER JOIN Customers ON Orders.CustomerID=Customers.CustomerID;
```

Agregační funkce

Tyto funkce umožní provést akce na hodnotách a vrácení výsledku v jedné číselné hodnotě

- **AVG()**
 - spočítá průměr hodnot
- **COUNT()**
 - Spočítá počet záznamů v dané tabulce
- **MIN()**
 - Vráť minimální hodnotu
- **MAX()**
 - Vráť maximální hodnotu
- **SUM()**
 - Vráť součet všech hodnot

Skalární funkce

- Vrací jednu specifickou hodnotu
- Může přijmout více parametrů, provede kalkulaci a poté vrátí jednu hodnotu

Typy

- Funkce pro práci s řetězcí
- Funkce pro práci s čísly
- Funkce pro práci s datumem
- Funkce na konvertování
- Funkce na zpracovávání NULL hodnoty

Příklady

- **CONCAT**
 - Funkce vrátí jeden string spojený ze dvou nebo více řetězců
 - `SELECT CONCAT("Karel", "je", "nej") => Kareljenej`
- **LENGTH**
 - Funkce vrátí délku řetězce
 - `SELECT LENGTH("Ahoj ja jsem karel") => 18`
- **ROUND**
 - Zaokrouhlí hodnotu
 - `SELECT ROUND(59.9) => 60`