

# **Structured Query Language**

Referát k předmětu Informatika II

Vypracoval: Štěpán Stříteský 24. 2. 2025

Obor: Podniková ekonomie

Specializace: Management výroby

# 1. Úvod

## 1.1. Co je Structured Query Language

Structured query language (SQL), neboli **strukturovaný dotazovací jazyk** je standardizovaný doménově specifický programovací jazyk, který slouží k manipulaci a správě dat v relačních databázích a správě databází samotných. Poskytuje uživateli možnost data vyhledávat, manipulovat s nimi a modifikovat je. Dále jej lze využít ke kontrole a udělování přístupových oprávnění nebo k vytváření a úpravě databázových struktur. SQL je nezávislý na platformách, které jej využívají, zároveň je mezi nimi do jisté míry přenositelný.

## 1.2. Historie jazyka SQL

SQL vznikl v 70. letech 20. století u společnosti IBM ve Spojených státech amerických jako Structured English Query Language (SEQUEL). Hlavní motivací byl vznik takzvaného relačního modelu, což je, zjednodušeně řečeno, třídění a ukládání dat do struktury sloupců a řádků, mezi nimiž existují jasně definované vztahy. K prvním komerčně využitelným aplikacím jazyka začalo docházet koncem 70. let prostřednictvím již zmíněné IBM a Relational Software (dnes Oracle). Od roku 1986 je SQL jazykem standardizovaným jako Databázový jazyk SQL, k aktualizaci tohoto standardu dochází v průměru každé čtyři roky, poslední takovou byla ta v roce 2023.

## 2. Základní pojmy

**Relační databáze** je způsob ukládání a prezentace dat v tabulkové formě.

**Table/Tabulka** je základní strukturou pro ukládání dat, ve které každý řádek reprezentuje jeden záznam a každý sloupec jeden atribut neboli vlastnost konkrétního záznamu. Relační databáze musí poskytovat možnost s daty v této formě manipulovat, tedy tabulky vytvářet, filtrovat, spojovat, agregovat i mazat. Mezi tabulkami mohou vznikat vztahy, relace, které je propojují.

**Queries/Dotazy** slouží k interakci uživatele s databází. S jejich pomocí lze získávat informace z dat uložených v databázi, vytvářet nové záznamy, upravovat ty stávající, nebo mazat ty nepotřebné, duplicitní a irelevantní. Dotaz může tvořen kombinací klauzulí, výrazů, predikátů, názvů tabulek, sloupců, či jednotlivých záznamů.

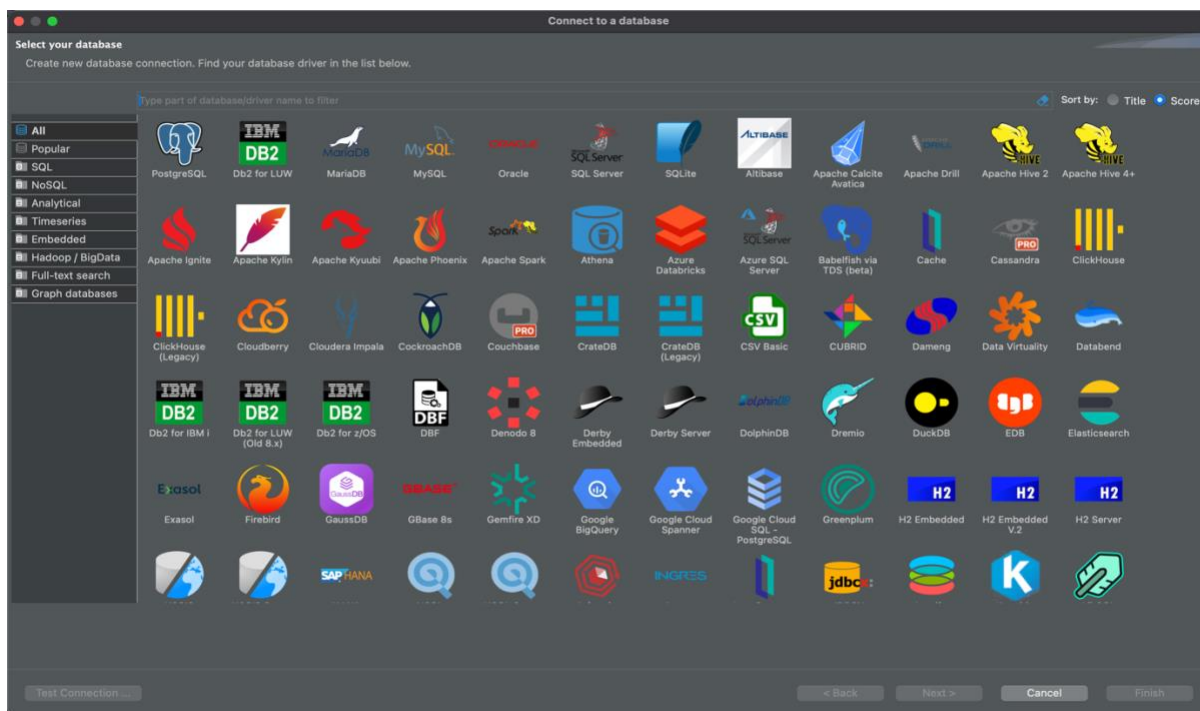
**Constraint/Omezení** definují podmínky, za kterých mohou být data uložena v databázi, čímž pomáhají udržet jejich přesnost, úplnost, neporušenost a konzistenci. Příkladem omezení může být podmínka, která stanoví, že konkrétní sloupec záznamu nesmí obsahovat prázdnou hodnotu, tedy NULL

**Transaction/Transakce** je skupina SQL pokynů a výrazů, které jsou databází vykonány jako celek, čímž umožňují zvrátit změny těmito pokyny provedené v případě, že některé další pokyny v rámci transakce nebylo možné provést. Transakcí může být třeba vypsání takových záznamů tabulky, u nichž se hodnota uživatelem zvoleného atributu rovná určité hodnotě.

**RDBMS/Systém pro správu relační databáze** aplikuje všechny již zmíněné pojmy a tvoří grafické uživatelské rozhraní pro jednodušší práci s databázemi. Prostřednictvím těchto systémů může uživatel vytvářet a mazat tabulky, vkládat do nich záznamy a posléze z přítomných dat získávat informace. Mezi tyto systémy patří například MySQL Workbench od společnosti Oracle, Microsoft SQL Server nebo DBeaver. S výjimkou poslední jmenované nabízejí tyto platformy i vlastní řešení databáze.

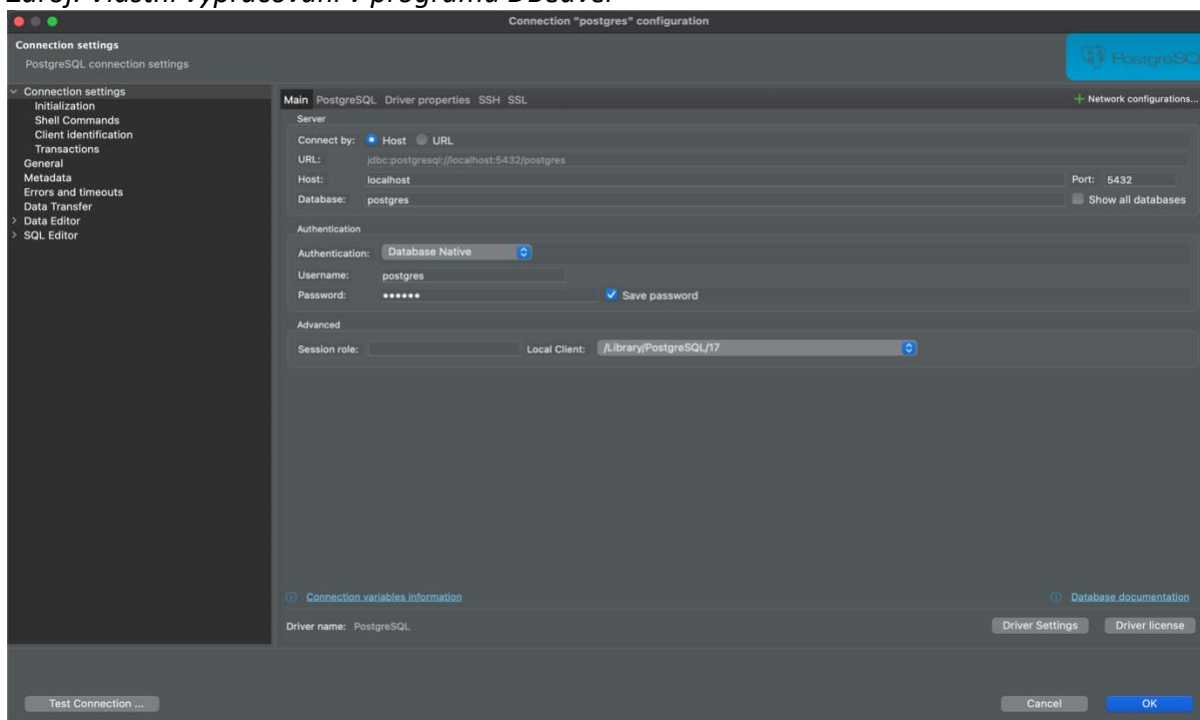
### 3. Použití SQL

K použití SQL je v první řadě potřeba mít k dispozici systém pro správu relační databáze, jehož součástí je i databáze samotná. Pro účely tohoto referátu byl použit systém DBeaver, který je dostupný v bezplatné i placené verzi. Po instalaci a spuštění DBeaveru je třeba jej propojit s databází. Postup lze vidět na obrázcích 1 a 2.



**Obr. 1** Propojení RDBMS s databází

*Zdroj: Vlastní vypracování v programu DBeaver*



**Obr. 2** Konfigurace propojení RDBMS s databází

*Zdroj: Vlastní vypracování v programu DBeaver*

Po připojení k databázi lze začít s databází komunikovat prostřednictvím dotazů. Struktura dotazu záleží na tom, jak chceme s daty manipulovat a jsou-li data vůbec přítomna. V závislosti na těchto faktorech se SQL dělí do pěti podúrovní.

### 3.1. Data Definition Language/jazyk definující data

Pokud databáze neobsahuje žádná data, můžeme začít tvorbou tabulek prostřednictvím dotazů generujících tabulky. K tomu slouží DDL.

**CREATE TABLE** – slouží k vytvoření tabulky

**ALTER TABLE** – upraví stávající tabulku

**DROP TABLE** – vymaže tabulku

### 3.2. Data Manipulation Language/jazyk k manipulaci dat

Jsou-li v databázi přítomné tabulky, jsou zapotřebí dotazy, kterými do tabulek vložíme data.

**INSERT INTO** – vloží zadané záznamy do tabulky.

**UPDATE** – upraví již existující záznamy.

**DELETE** – smaže vybrané záznamy.

### 3.3. Data Query Language/dotazovací jazyk

Pro následné získávání konkrétních záznamů, jejich filtraci, případně agregaci slouží DQL, jehož jádrem je výraz SELECT.

**SELECT** – sdělí databázi, jaká data má vybrat a zobrazit uživateli.

**FROM** – stanoví ze které tabulky mají být potřebná data vybrána.

**WHERE** – vyfiltruje vybraná data podle určitého atributu a jeho hodnoty.

**ORDER BY** – seřadí zobrazovaná data podle zvoleného atributu.

**GROUP BY** – seskupí vybraná data do skupin podle zvoleného společného atributu.

Konkrétní příklady použití DQL lze vidět na obr. 3, kde z databáze získáme počty osob nově nakažených virem SARS-CoV-2 v jednotlivých zemích světa dne 1. 5. 2020. Obr. 4 pak ukazuje součet všech nově nakažených osob ve všech zemích světa dne 1. 5. 2020.

### 3.4. Data Control Language/jazyk pro kontrolu dat

Slouží správci databáze k udělování a zamítání přístupu k datům, jejich manipulaci, případně změnám.

**GRANT** – přidělí uživateli oprávnění.

**REVOKE** – odebere uživateli oprávnění.

### 3.5. Data Transaction Language/jazyk pro kontrolu datových transakcí

Mají za úkol spravovat datové transakce a udržovat tak neporušenost a stálost dat v databázi.

**COMMIT** – uloží všechny změny, ke kterým došlo v průběhu transakce.

**ROLLBACK** – zvrátí změny, ke kterým došlo v průběhu transakce.

**SAVEPOINT** – stanoví bod, ke kterému se transakce vrátí v případě ROLLBACK pokynu.

Database Navigator: postgres, localhost:5432, data\_academy\_content

SQL Script: `select distinct date, country, confirmed from covid19_basic_differences cbd where date = '2020-05-01' order by country;`

	date	country	confirmed
1	2020-05-01	Afghanistan	164
2	2020-05-01	Albania	9
3	2020-05-01	Algeria	148
4	2020-05-01	Andorra	0
5	2020-05-01	Angola	3
6	2020-05-01	Antigua and Barbuda	1
7	2020-05-01	Argentina	104
8	2020-05-01	Armenia	82
9	2020-05-01	Austria	79
10	2020-05-01	Azerbaijan	50
11	2020-05-01	Bahamas	0
12	2020-05-01	Bahrain	130
13	2020-05-01	Bangladesh	571
14	2020-05-01	Barbados	0

189 row(s) fetched - 0.002s (0.001s fetch), on 2025-02-26 at 11:09:42

**Obr. 3** Počty nově nakažených v zemích světa dne 1. 5. 2020 seřazeny dle zemí abecedně  
Zdroj: Vlastní vypracování v programu DBeaver

Database Navigator: postgres, localhost:5432, data\_academy\_content

SQL Script: `select date, sum(confirmed) from covid19_basic_differences cbd where date = '2020-05-01' group by date;`

	date	sum
1	2020-05-01	172,928

1 row(s) fetched - 0.010s (0.001s fetch), on 2025-02-26 at 11:08:57

**Obr. 4** Součet nově nakažených v zemích světa dne 1. 5. 2020  
Zdroj: Vlastní vypracování v programu DBeaver

## 4. Závěr

SQL je hojně využíván ke správě a interakci s relačními databázemi například v rámci datové analýzy. Mezi jeho výhody patří efektivnost, se kterou byl navržen, aby umožňoval použití

komplexních dotazů nad velkými datovými sadami. ANSI a ISO standardizace napomáhá přenositelnosti SQL kódu mezi platformami, zároveň jej lze modifikovat a přizpůsobit tak potřebám dané aplikace.

Co se limitů jazyka týče, provádění pokročilých operací vyžaduje hlubokou technickou znalost a porozumění, na rozdíl od relativně intuitivní syntaxe jednoduchých dotazů. Dále může představovat výzvu využití SQL pro práci s nestrukturovanými daty, tedy daty, která nelze ukládat ve formě tabulek. Rovněž se lze setkat s odchylkami syntaxe mezi jednotlivými platformami a RDBMS, což klade překážky dokonalé přenositelnosti kódu mezi nimi.

I přes některé své nevýhody je SQL využíván v oblastech jako jsou e-commerce, zdravotnictví, bankovníctví nebo datová věda, kde s jeho využitím lze zpracovávat objednávky, inventáře nebo produktové katalogy, spravovat záznamy pacientů a programy úkonů, analyzovat peněžní transakce a generovat finanční reporty nebo v kombinaci s nástroji jako jsou programovací jazyky Python a R kombinovat SQL databáze s algoritmy strojového učení.

## 5. Zdroje

<https://en.wikipedia.org/wiki/SQL>

[https://www.geeksforgeeks.org/what-is-sql/?ref=gcse\\_outind](https://www.geeksforgeeks.org/what-is-sql/?ref=gcse_outind)

<https://aws.amazon.com/what-is/sql/>

<https://dbeaver.io/download/>

<https://engeto.cz/datova-akademie/>

Databáze je součástí kurzu Datová akademie ENGETO (viz odkaz na předchozím řádku)

Vlastní zkušenosti