

Jazyk SQL

Dialekt, standart, DML, DDL, DCL, TCL,
INSERT, UPDATE, INSERT

Dialekt

- Existují různé dialekty SQL:
 - Syntaxe
 - Case sensitive
 - Rozdíl ve využití uvozovek
 - Jiné příkazy

	SQL Server	MySQL	PostgreSQL	SQLite
SELECT ...	Select [col1], [col2]	SELECT col1, col2	SELECT col1, col2	SELECT col1, col2
Data from tables is case sensitive?	Yes WHERE name = 'John' Or WHERE name = 'john' are not the same	No WHERE name = 'John' Or WHERE name = 'john' are the same	Yes WHERE name = 'John' Or WHERE name = 'john' are not the same	Yes WHERE name = 'John' Or WHERE name = 'john' are not the same
Using quotation marks	name = 'John' only	name = 'John' or name = "John"	name = 'John' only	name = 'John' or name = "John"
Aliases for columns and tables	SELECT AVG(col1)=avg1	SELECT AVG(col1) AS avg1	SELECT AVG(col1) AS avg1	SELECT AVG(col1) AS avg1
Working with dates	GETDATE() DATEPART()	CURDATE() CURTIME() EXTRACT()	CURRENT_DATE() CURRENT_TIME() EXTRACT()	DATE('now') strftime()
Window functions i.e., OVER(), PARTITION BY()	Yes	Yes	Yes	No (need to use subqueries instead)

Standart

- Americký institut ANSI chtěl vydat jako nový standart RDL
- ANSI založil standart na SQL
- Standart SQL-86
- V roce 1992 byl přijat nový standart, z důvodu nedostatků SQL-92 (SQL2)
- Nejnovější je SQL3 (SQL-99), reaguje na aktuální potřeby databáze

DML (Data Manipulation Language)

- Manipulace s daty a databázi
 - INSERT – vložení záznamu (jednoho či více)
 - UPDATE – úprava existujících dat v tabulce
 - DELETE – smazání dat v tabulce
- INSERT INTO** nazev_tabulky (sloupec1, sloupec2) **VALUES** („hodnota“, „hodnota“)

- **UPDATE** nazev_tabulky **SET** sloupec1 = „hodnota“ **WHERE** sloupecX = „hodnota“
- **DELETE FROM** nazev_tabulky **WHERE** vek=„57“
- Příklad:
 - **INSERT INTO** sousedstvo (čp, jméno, věk) **VALUES** („179“, „Lojza od vedle“, „57“)
 - **UPDATE** sousedstvo **SET** jméno = „Ludvík“ **WHERE** čp = „179“
 - **DELETE FROM** sousedstvo **WHERE** věk = „57“ **AND** jméno = „Ludvík“

DDL (Data Definiton Language)

- Definování dat:
 - **CREATE** – vytvoření databáze, či nové tabulky
 - **DROP** – vymazání databáze či tabulky
 - **ALTER TABLE...RENAME TO** – přejmenování tabulky
 - **ALTER TABLE...ADD/DROP/MODIFY** – úprava sloupečku v tabulce
- **CREATE DATABASE** nazev_databaze
- **CREATE TABLE** nazev_tabulky (sloupec1 = typ_dat(velikost), sloupec2 ...)
- **ALTER TABLE** nazev_tabulky **RENAME TO** novy_nazev
- **ALTER TABLE** nazev_tabulky **ADD** sloupec1 = typ_dat
- Příklad:
 - **CREATE DATABASE** seznam_kytar
 - **CREATE TABLE** kytary (model = varchar(255), pocet_strun = int)
 - **ALTER TABLE** seznam_kytar **RENAME TO** databaze_kytary
 - **ALTER TABLE** nazev_tabulky **ADD** drevo = varchar(255)

DCL (Data Control Language)

- Nastavování přístupových práv
 - **GRANT** – přidělení oprávnění uživateli k určitým tabulkám
 - **REVOKE** – příkaz pro odebrání práv uživateli
- **GRANT** práva **ON** nazev_tabulky **TO** jmeno_uzivatele
- **REVOKE** prave **ON** nazev_tabulky **FROM** jmeno_uzivatele
- Příklad:
 - **GRANT** DELETE, UPDATE (Plat, Číslo) **ON** Zaměstnanec **TO** Petr
 - **REVOKE** DELETE, UPDATE (Plat, Číslo) **ON** Zaměstnanec **FROM** PUBLIC

TCL (Transactional Control Language)

- Příkazy pro správu databázových transakcí
 - **START TRANSACTION (BEGIN)** - zahájení transakce
 - Žádná změna od tohoto okamžiku nebude viditelná ostatním uživatelům
 - **COMMIT (GO)** - potvrzení transakce
 - Potvrdí transakci, ostatním uživatelům je přístupná změna
 - **ROLLBACK** – zrušení transakce, návrat do původního stavu
 - Stornuje modifikované objekty v databázi a vrátí do stavu před transakcí, důležité pro zachování databázové integrity
- Příklad:

- **BEGIN TRANSACTION** --definice transakce --DDL nebo DML operace
- **END TRANSACTION;**
- **COMMIT;** --potvrzení transakce
- **ROLLBACK;** --odvolání transakce
- Příklad TCL (SAVEPOINT)
 - **BEGIN TRANSACTION** --definice transakce --DDL nebo DML operace 1
 - **SAVE TRANSACTIONS** <název záchranného bodu 1> --DDL nebo DML operace 2
 - **SAVE TRANSACTION** <název záchranného bodu 2>
 - **END TRANSACTION;**
 - **COMMIT;** --potvrzení transakce
 - **ROLLBACK** <název záchranného bodu 1>; --vrátí se k savepoint 1

Insert

- Příkaz přidá nový záznam o tabulky databáze
- **INSERT INTO** <tabulka> [(<sloupec>[,...n])] **VALUES** (<hodnota>[,...n]);
- Vloží do vybraných sloupců tabulky určené hodnoty
- Bez uvedení sloupců se předpokládá vložení celého záznamu
- Počet sloupců a hodnot musí být stejný
- Kopírování tabulky
- Vytvoření tabulky na základě jiných dat
- Pomocí příkazu INSERT lze najednou přidat několik záznamů, což se používá při importu zálohy databáze

Příklad INSERT

jmeno	cislo	ulice	mesto
Jan Novák	257125474	Wikipedistická 28	Pastoriovice
Jana Nováková	574125474	Luční 6	Praha

telefonní seznam

INSERT INTO telefonni_seznam **VALUES** ('John Doe', '555-1212', 'Pařížská 6', 'Aš');

jmeno	cislo	ulice	mesto
Jan Novák	257125474	Wikipedistická 28	Pastoriovice
Jana Nováková	574125474	Luční 6	Praha
John Doe	555-1212	'Pařížská 6	Aš

Více záznamů - INSERT INTO telefonni_seznam

(jmeno, cislo) **VALUES**
 ('John Doe', '555-1212'),
 ('Leona Lewis', '555-112777'),
 ('Joe King', '555-1213');

INSERT INTO telefonni_seznam (jmeno,
 cislo, mesto)
VALUES (
 'Jan Kos', '666-765', 'Liberec')

●

Příklad kopírování pomocí INSERT

Všechny sloupce z tabulky

```
INSERT INTO table2
SELECT * FROM table1
WHERE condition;
```

Některé sloupce

```
INSERT INTO table2 (column1, column2,
column3, ...)
SELECT column1, column2, column3, ...
FROM table1
WHERE condition;
```

-

Příklad INSERT SET

- Jiná forma zápisu INSERT

```
INSERT INTO telefonni_seznam
SET jmeno='John Doe', cislo='555-1212';
```

-

UPDATE

- Příkaz se používá pro modifikování existujícího záznamu v tabulce, který je upřesněn podmínkou where
- **UPDATE** nazev_tabulky **SET** sloupec1 = hodnota1, sloupec2 = hodnota2 **WHERE** podmínka;
- Lze provádět jednoduché matematické operace při záznamu (celého sloupce)

```
UPDATE zamestnanci SET vyplata = vyplata + 500;
```

-

id	jmeno	ulice	vyplata
1	Rudolf	Rudolfova 1	15 000,00 Kč
2	Jirka	Na Zatoči 1440	17 000,00 Kč
3	Jára	Sametová 14	80 000,00 Kč
4	Kazan	Liberecká 591	11 000,00 Kč
5	Hektor	Zahradní 10	- Kč

```
UPDATE zamestnanci SET vyplata = 15000;
```

```
UPDATE zamestnanci SET vyplata = 15000 WHERE jmeno = 'Hektor';
```

id	jmeno	ulice	vyplata
1	Rudolf	Rudolfova 1	15 000,00 Kč
2	Jirka	Na Zatoči 1440	17 000,00 Kč
3	Jára	Sametová 14	80 000,00 Kč
4	Kazan	Liberecká 591	11 000,00 Kč
5	Hektor	Zahradní 10	15 000,00 Kč

id	ulice	vyplata
1	Rudolfova 1	15 000,00 Kč
2	Na Zatoči 1440	15 000,00 Kč
3	Sametová 14	15 000,00 Kč
4	Liberecká 591	15 000,00 Kč
5	Zahradní 10	15 000,00 Kč

-