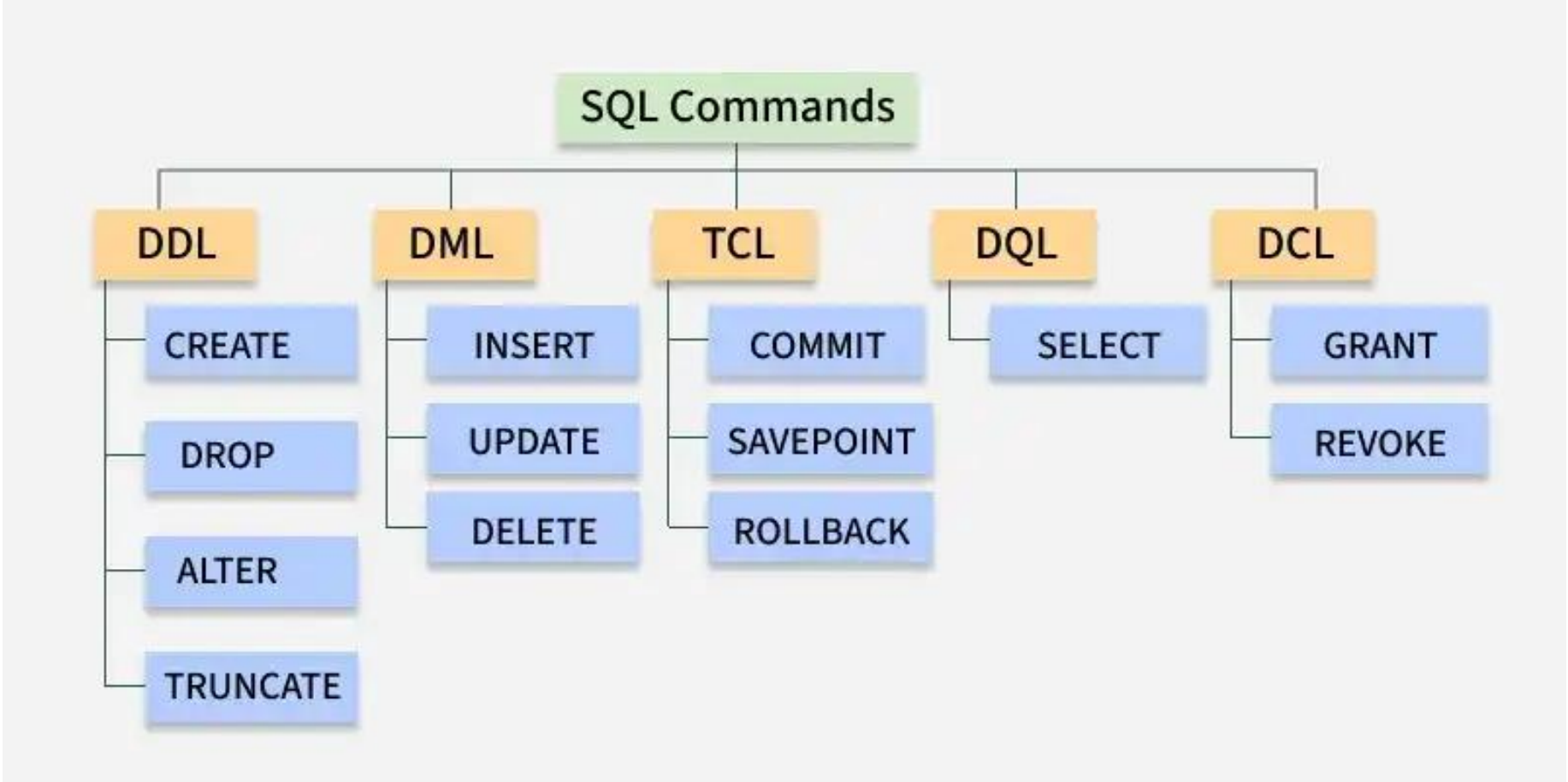
Databázové systémy

SELECT

2026 Lukáš Miňo

SQL príkazy



https://media.geeksforgeeks.org/wp-content/uploads/20260522121146555616/commands.webp

DQL - Data Query Language

Príkaz	Popis	Syntax
SELECT	Používa sa na získanie (výber) dát z databázy	`SELECT column1, column2, ... FROM table_name WHERE condition;`
FROM	Označuje tabuľku (alebo tabuľky), z ktorej sa majú dáta získať	`SELECT column1 FROM table_name;`
WHERE	Filtruje riadky pred akýmkoľvek zoskupovaním alebo agregáciou	`SELECT column1 FROM table_name WHERE condition;`
GROUP BY	Zoskupuje riadky, ktoré majú rovnaké hodnoty v špecifikovaných stĺpcoch	`SELECT column1, AVG_FUNCTION(column2) FROM table_name GROUP BY column1;`
HAVING	Filtruje výsledky príkazu GROUP BY	`SELECT column1, AVG_FUNCTION(column2) FROM table_name GROUP BY column1 HAVING condition;`
DISTINCT	Odstraňuje duplicitné riadky z množiny výsledkov	`SELECT DISTINCT column1, column2, ... FROM table_name;`
ORDER BY	Zoraďuje množinu výsledkov podľa jedného alebo viacerých stĺpcov	`SELECT column1 FROM table_name ORDER BY column1 [ASC | DESC];`
LIMIT	Používa sa na obmedzenie počtu vrátených riadkov v dopyte SELECT (bežne podporované v MySQL a PostgreSQL)	`SELECT * FROM table_name LIMIT number;`

Zdroj

- https://databazy.ics.upjs.sk/OsobaDB.txt

```
SELECT * FROM osoba;
```

id	meno	priezvisko	rodne_priezvisko	dat_nar	dat_smrti	pohlavie	vyska	vaha	otec	matka
1	Adam	Prvý	NULL	1918-05-11	1968-10-01	m	180.0	80.0	NULL	NULL
2	Eva	Prvá	Druhá	1919-01-09	1988-07-22	z	160.0	60.0	NULL	NULL
3	Zoly	Múdry	NULL	1918-04-07	1990-09-23	m	175.5	75.0	NULL	NULL
4	Nasa	Kováèová	Rostová	1928-02-05	1965-03-11	z	155.0	99.0	NULL	NULL
5	Jozef	Urban	NULL	1922-10-19	NULL	m	199.5	NULL	NULL	NULL
6	Mária	Urbanová	Nováková	1937-12-08	NULL	z	172.5	57.5	1	2
7	Patrik	Novák	Novák	1945-06-19	NULL	m	182.5	89.5	1	2
8	Patrícia	Nováková	Halušková	1952-01-08	NULL	z	143.5	35.0	NULL	NULL
9	Michal	Kováè	Kováè	1942-04-10	NULL	m	167.0	88.0	3	2
10	Roman	Kováè	Kováè	1948-05-20	NULL	m	179.5	78.5	3	4
11	Peter	Horváth	Horváth	1959-07-02	2000-12-31	m	193.0	110.5	NULL	NULL
12	Lucia	Horváthová	Urbanová	1959-01-13	NULL	z	156.5	45.5	5	6
13	Urban	Urban	Urban	1957-03-31	NULL	m	138.2	24.5	5	6
14	DÁŠa	Nováková	Novaková	1970-07-17	NULL	z	167.0	55.0	7	8
15	Viera	Silná	Novaková	1973-02-13	NULL	z	169.5	63.0	7	8
16	Vladimír	Silný	Silný	1974-08-01	2002-12-04	m	175.5	73.0	NULL	NULL
17	Milena	Slabá	Slabá	1979-09-14	NULL	z	164.0	64.0	NULL	NULL
18	Ján	Horváth	Horváth	1982-01-16	NULL	m	159.5	65.5	11	12
19	Zuzana	Silná	Silná	2002-03-01	NULL	z	158.5	60.0	16	15
20	Zuzana	Slabá	Slabá	1999-12-16	NULL	z	171.5	54.5	16	17
21	Zuzana	Pravá	Pravá	1990-11-26	NULL	z	170.5	60.5	16	17
22	Zuzana	¼avá	¼avá	1931-01-14	NULL	z	195.5	58.5	16	17
23	Zuzana	Stredná	Stredná	1945-04-08	NULL	z	150.5	87.0	16	17
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
SELECT

        meno,

        priezvisko AS 'Priezvisko‘,

        dat_nar 'Dátum narodenia‘
FROM

        osoba
WHERE

        pohlavie = 'z' AND priezvisko <> rodne_priezvisko AND dat_smrti IS NULL
        AND dat_nar < '1990-01-01' AND vyska > 160
ORDER BY priezvisko ASC, 1 DESC, 'Dátum narodenia';
```

meno	Priezvisko	Dátum narodenia
Viera	Silná	1973-02-13
Mária	Urbanová	1937-12-08

SELECT ... FROM ...

- SELECT * FROM osoba;

- ```
  SELECT
        meno,
        priezvisko
  FROM
        osoba;
  ```

- ```
  SELECT
        meno AS name,
        priezvisko ' Surname'
  FROM
        osoba;
  ```

SELECT ... FROM ...

- SELECT - ktoré stĺpce chceme (alebo * pre všetky)
- FROM - z ktorej tabuľky dáta berieme
- AS (Alias) - dočasné premenovanie stĺpca vo výstupe pre lepšiu čitateľnosť

- SELECT
  meno AS Name,
  priezvisko ' Surname’
  dat_nar AS "Dátum narodenia"
  FROM
  osoba;

SELECT DISTINCT … FROM …

- Zabezpečí, aby sa v konečnom výsledku dopytu **každý riadok** objavil **iba raz**.
- Odstraňuje duplicity až po tom, čo databáza vyžiada stĺpce (aplikuje sa na celý vybraný riadok).
- Píše sa hneď za kľúčové slovo SELECT.

```
-- Vráti všetkých 23 krstných mien (vrátane 5x 'Zuzana')
SELECT meno FROM osoba;
-- Vráti zoznam unikátnych mien bez opakovania (Zuzana bude iba raz)
SELECT DISTINCT meno FROM osoba;
-- Pozor: DISTINCT pre kombináciu stĺpcov!
-- Vráti unikátne dvojice (meno + priezvisko).
-- Ak máme dve Zuzany Silné, vo výsledku bude iba jedna.
-- Ak máme Zuzanu Silnú a Zuzanu Slabú, ostanú obe.
SELECT DISTINCT meno, priezvisko FROM osoba;
```

WHERE

- SELECT
    meno,
    priezvisko,
    vyska
  FROM osoba
  WHERE pohlavie = 'm' AND vyska > 180.0;

Operátor

Aritmetické

+

-

*

/

%

Porovnávacie

=

>

<

>=

<=

<>

!=

Zložené

+=

-=

*=

/=

%=

Logické

ALL

AND

ANY

BETWEEN

EXISTS

IN

LIKE

NOT

OR

SOME

NULL: Ako pracovať s (ne)známymi dátami?

- **Čo je to NULL?** Reprezentuje chýbajúcu, neznámu alebo neaplikovateľnú hodnotu. *Pozor: NULL nie je nula (0) ani prázdny reťazec ('')!*
- **NOT NULL** constraint (obmedzenie) – databáza vynucuje, že stĺpec *musí* mať zadanú hodnotu (napr. meno, dat_nar).
- **Logika Troch Hodnôt (3-valued logic):** Porovnanie s NULL nevráti TRUE ani FALSE, ale UNKNOWN.
- **Zlaté pravidlo syntaxe:** S hodnotou NULL nikdy neporovnávame pomocou = alebo !=. Vždy používame operátory **IS NULL** a **IS NOT NULL**.

- -- Tento dopyt nevráti NIČ (ani ľudí, čo žijú)

~~SELECT meno, priezvisko FROM Osoba WHERE dat_smrti = NULL;~~

- -- SPRÁVNE: Výber žijúcich ľudí (stĺpec dat_smrti nemá hodnotu)
  SELECT meno, priezvisko FROM Osoba WHERE dat_smrti IS NULL;

- -- SPRÁVNE: Výber ľudí, u ktorých poznáme váhu
  SELECT meno, priezvisko, vaha FROM Osoba
  WHERE vaha IS NOT NULL;

```
SELECT
        *
FROM osoba
WHERE
        pohlavie='M'
        AND vyska<=180
        AND datum_narodenia<='1980-01-13'
        OR datum_smrti IS NULL;
```

Pokročilé filtrovanie

- SELECT * FROM osoba WHERE vaha >= 50 AND vaha <= 80;
- SELECT * FROM osoba WHERE vaha BETWEEN 50 AND 80;

- SELECT * FROM osoba
  WHERE vyska = 167 OR vyska = 169 OR vyska = 182);
- SELECT * FROM osoba WHERE vyska IN (167, 169, 182);

LIKE

- Slúži na vyhľadávanie textových reťazcov, ktoré nezodpovedajú presnej hodnote, ale určitej šablóne (**pattern**u).
- Využíva dva základné žolíkové znaky (**wildcards**):
  - % (percento) - reprezentuje **0 alebo viac** ľubovoľných **znakov**.
  - _ (podčiarkovník) - reprezentuje **presne jeden** ľubovoľný **znak**.
- **Pozor na citlivosť**: V závislosti od nastavenia databázy (Collation) môže byť LIKE citlivý na veľkosť písmen a diakritiku (Case/Accent Sensitive). -- 1. Začína na... (Vráti ľudí s priezviskom Novák, Nováková, Nováková) SELECT meno, priezvisko FROM Osoba WHERE priezvisko LIKE 'Nov%'; -- 2. Končí na... (Vráti ženy s priezviskom končiacim na 'á' - napr. Prvá, Slabá) SELECT meno, priezvisko FROM Osoba WHERE priezvisko LIKE '%á'; -- 3. Obsahuje... (Vráti kohokoľvek, kto má v priezvisku text 'orv' - napr. Horváth) SELECT meno, priezvisko FROM Osoba WHERE priezvisko LIKE '%orv%'; -- 4. Presný počet znakov (Hľadá 4-písmenové priezviská - napr. Prvý, Novák) SELECT meno, priezvisko FROM Osoba WHERE priezvisko LIKE '____';

LIKE

- -- Začína na...
  -- Vráti ľudí s priezviskom Novák, Nováková, Nováková
  SELECT meno, priezvisko FROM osoba WHERE priezvisko LIKE 'Nov%’;
- -- Končí na...
  -- Vráti ženy s priezviskom končiacim na 'á' - napr. Prvá, Slabá
  SELECT meno, priezvisko FROM osoba WHERE priezvisko LIKE '%á’;
- -- Obsahuje...
  -- Vráti kohokoľvek, kto má v priezvisku text 'orv' - napr. Horváth
  SELECT meno, priezvisko
  FROM osoba
  WHERE priezvisko LIKE '%orv%’;
- -- Presný počet znakov
  -- Hľadá 4-písmenové priezviská - napr. Prvý, Novák
  SELECT meno, priezvisko FROM osoba WHERE priezvisko LIKE '____';

ORDER BY

- **Základný princíp:** SQL tabuľky nemajú garantované prirodzené poradie. Ak chcete mať istotu, v akom poradí dáta prídu, musíte použiť ORDER BY.
- Smery radenia:
  - ASC (Ascending) – vzostupne (A ➔ Z, 0 ➔ 9). Je **predvolené** (nemusí sa písať).
  - DESC (Descending) – zostupne (Z ➔ A, 9 ➔ 0).
- **Viacúrovňové usporiadanie**: Môžete usporiadať podľa viacerých stĺpcov naraz. Ak nastane zhoda v prvom, databáza použije druhý.
- **Správanie NULL hodnôt:** Hodnota NULL sa správa ako **najmenšia** hodnota a pri ASC usporiadaní sa objaví na úplnom začiatku.

ORDER BY - možnosti usporiadania

- **Podľa ALIASU (Názvu zo SELECTu):** V ORDER BY môžete použiť alias vytvorený pomocou AS. Na rozdiel od WHERE to tu funguje, pretože databáza usporiada výsledky až po priradení aliasov v SELECTe.
- **Podľa PORADIA stĺpca (Čísla):** Namiesto názvu stĺpca môžete napísať jeho poradové číslo z klauzuly SELECT (napr. **1 pre prvý stĺpec**).
  - Pozor: V praxi sa neodporúča, kód je potom zle čitateľný.
- **Podľa VÝRAZU (Expression):** Môžete usporiadať podľa matematických výpočtov alebo SQL funkcií spustených nad stĺpcami za behu dopytu.

ORDER BY

- -- Zoradenie podla priezviska vzostupne, mena zostupne a datumu narodenia vzostupne
- SELECT
  meno,
  priezvisko,
  dat_nar AS 'Dátum narodenia'
- FROM
  osoba
- ORDER BY priezvisko, 1 DESC, 'Dátum narodenia' ASC;

ORDER BY

```
-- Usporiadanie podľa VÝRAZU (expression)
-- Zoradenie ľudí podľa indexu BMI od najchudších
-- Využíva matematický výraz: váha / (výška v metroch na druhú)
SELECT meno, priezvisko, vaha, vyska
FROM
        osoba
WHERE
         -- vynecháme NULL hodnoty pre presnosť výpočtu
        vaha IS NOT NULL
ORDER BY (vaha / ((vyska/100) * (vyska/100))) ASC;
```

Obmedzenie riadkov a stránkovanie v SQL

- **Problém štandardizácie:** Princíp obmedzenia riadkov (LIMIT) a ich "preskakovania" (OFFSET) je v každej databáze rovnaký, ale syntax sa výrazne líši podľa výrobcu databázového systému (RDBMS)
- **Tri hlavné prístupy v praxi:**.
  - **LIMIT / OFFSET:** Používa sa v open-source systémoch. Píše sa na úplný koniec dopytu.
  - **OFFSET / FETCH:** Oficiálny ANSI/ISO štandard. Najčistejší prístup pre moderné enterprise databázy.
  - **TOP:** Špecifická klauzula pre MS SQL Server. Píše sa hneď na začiatok za SELECT. Pri samostatnom použití nevie stránkovať, preto sa od verzie 2012 kombinuje s klauzulou OFFSET.
- **Pozor:** Stránkovanie bez definovaného usporiadania (ORDER BY) nedáva zmysel. Databáza by pri každom načítaní stránky mohla vrátiť iný výsledok.

Podľa výšky preskoč 10 ľudí, zobraz nasledujúcich 5

RDBMS	Prístup k obmedzeniu riadkov a stránkovaniu (Príklady)
MySQL, PostgreSQL, SQLite (**Open-source štandard**)	SELECT meno, priezvisko FROM osoba ORDER BY vyska DESC LIMIT 5 OFFSET 10;
Oracle DB, Moderný MS SQL, ANSI (**Oficiálny ISO/ANSI štandard**)	SELECT meno, priezvisko FROM osoba ORDER BY vyska DESC OFFSET 10 ROWS FETCH NEXT 5 ROWS ONLY;
Microsoft SQL Server (T-SQL) (Klauzula **TOP** a stránkovanie)	-- Iba obmedzenie (Top 5 najvyšších): SELECT TOP 5 meno, priezvisko FROM osoba ORDER BY vyska DESC; -- Stránkovanie (Moderný MS SQL od 2012): SELECT meno, priezvisko FROM osoba ORDER BY vyska DESC OFFSET 10 ROWS FETCH NEXT 5 ROWS ONLY;

Zhrnutie: Syntax

- SELECT
- [ALL | DISTINCT | DISTINCTROW]
- select_expr [, select_expr] ...
- [FROM table_references]
- [WHERE where_condition]
- [ORDER BY {col_name | expr | position} [ASC | DESC], ...]