**MySQL** tahák

www.mysql.com

dev.mysql.com/doc

Legenda:

**d** – databáze

**t** – tabulka

**a** – atribut

**r** – záznam

**v** – hodnota

**p** – patern

**expr** – výraz

**cond** – podmínky

# DATOVÉ TYPY

**TINYINT** 1B 256 hodnot, **SMALLINT** 2B 65K, **MEDIUMINT** 3B 16M, **INT** 4B 4.29G, **BIGINT** 8B 2E+64, **FLOAT** 4B 6.8E+38, **DOUBLE** (REAL) 8B 3.3E+308

**DECIMAL(m,d)**, **DEC(m,d)**, **NUMERIC(m,d)**

**m** – počet cifer, max 65

**d** – desetinných cifer, max 30

**CHAR(m)** text pevná délka

**VARCHAR(m)** proměnlivá délka

**m** – délka řetězce

**TINYTEXT**, **TEXT**, **MEDIUMTEXT**, **LONGTEXT**

**ENUM(v1,v2…)** výčet, OR, 65K

**SET(v1,v2…)** výčet, AND, 64

**DATE** "RRRR-MM-DD", **TIME** "HH:MM:SS"

**DATETIME** "RRRR-MM-DD HH:MM:SS"

**TIMESTAMP** unixtime

# MODIFIKÁTORY

**PRIMARY KEY** primární klíč

**UNIQUE** jen unikátní hodnoty

**NULL** buňka může být prázdná

**NOT NULL** buňka jen s hodnotou

**UNSIGNED** číslo bez znaménka **AUTO\_INCREMENT** zvyšování hodnoty

AUTO\_INCREMENT=50

**DEFAULT** výchozí hodnota

**FULLTEXT INDEX** rychlejšímu hledání

pomocí příkazů MATCH a AGAINST

**INDEX** rychlejší přístup k datům

**ZEROFILL** doplní před číslo nuly

# DDL PRÁCE S DATABÁZÍ

**CREATE DATABASE** d;

**SHOW DATABASES**; zobrazí jména databází

**USE** d; nastavíme aktivní DB

**SELECT DATABASE()**; název aktuální db

**SHOW TABLES FROM** d;

**DROP DATABASE** d; vymaže db

# ARITM OPER

+ - \* / %

# LOGIC OPER

= <=> != <> < <= > >=

# DDL PRÁCE S TABULKAMI

**CREATE TABLE** t (aname datatype... )

**ALTER TABLE** t ADD COLUMN, CHANGE, MODIFY, DROP, RENAME

**DROP TABLE** t

# DML SELECT

**SELECT** {[DISTINCT] a… | expr…}

[**FROM** t

[**WHERE** cond]

[**GROUP** **BY** {a|expr} [ASC|DESC]…]

[**HAVING** where\_cond]

[**ORDER BY** {a|expr } [ASC|DESC]…]

[**LIMIT** {count|count OFFSET offs}]

]

[] může, {} musí, | nebo, default

# COND WHERE/HAVING

**WHERE** a1=v1 AND a2<v2 OR a3 IS NULL…

**AND** **OR NOT**

**BETWEEN** v1 AND v2 NOT BETWEEN v1 AND v2

**IN** (v1,v2…) a NOT IN **IS NULL** a IS NOT NULL

**LIKE** p, NOT LIKE

% libovolné znaky

\_ právě jeden znak

# DML PRÁCE S DATY

**INSERT INTO** t **VALUES** (r1\_v), (r2\_v)

**INSERT INTO** t (a1,a2) **VALUES** (v1,v2)

**UPDATE** t **SET** a1=v1, a2=v2 **WHERE** cond

**DELETE FROM** t **WHERE** cond

# IF(podmínka, v\_true, v\_false)

SELECT **IF(surname IS NULL, "-", surname)** AS "name" FROM user

# SPOJOVÁNÍ TABULEK JOIN, LEFT JOUN, INNER JOIN

t1 **JOIN** t2 **ON** cond => t (výsledkem je tabulka)

t.a\_count = t1.a\_count + t2.a\_count

t.r\_count = t1.r\_count × t2.r\_count + cond

t1 **LEFT JOIN** t2 – levé polospojení = all r z t1 join to r z t2, RIGHT JOIN analogicky

# AGREGAČNÍ FCE

**AVG(a)** průměr, **COUNT(a)** počet, **MAX(a)** maximum, **MIN(a)** minimum, **SUM(a)** suma

# STRING FCE

**CONCAT(v1,v2…)** spojování textových řetězců, NULL if v je NULL

**CONCAT\_WS(sep, v1, v2…)** spojování s oddělovačem, v s NULL se ignorují

**LENGTH**(str), **SUBSTRING**(str,start)

**REPLACE**(str,search,replace)

**REVERSE**(str), **TRIM**(str), **LTRIM**(str), **RTRIM**(str), **UPPER**(str), **LOWER**(str),

**LEFT**(str,cnt), **RIGHT**(str,cnt)

# SELECT EXAMPLE

**SELECT** 1+1 "dva", u.name, COUNT(l.id) AS "logs"

**FROM** user AS u LEFT JOIN log AS l, (SELECT…) AS t2

**WHERE** u.name LIKE ‘B%‘ AND u.created >= "2023-01-01"

**GROUP** **BY** l.id

**HAVING** logs > 100 OR u.updated IS NULL

**ORDER BY** u.name, CONCAT\_WS("-",u.created,u.id) DESC

**LIMIT** 20,3

# TRIGGER

**CREATE TRIGGER** name

{BEFORE|AFTER} {UPDATE|INSERT|DELETE} ON t1

**FOR EACH ROW**

**BEGIN**

DECLARE *var* INT;

SELECT a1 INTO *var* FROM t2 WHERE t2.id = OLD.id;

UPDATE t3 SET a2 = *var* WHERE t3.id = NEW.id;

**END;**

# UNION/UNION ALL/INTERSECT/EXCEPT

Sjednotí unikátní hodnoty tabulek do jedné:

SELECT a1 FROM t1 **UNION** SELECT a1 FROM t2

SELECT a1 FROM t1 **INTERSECT** SELECT a1 FROM t2

# DATE FCE

NOW() DAY() YEAR() CURRENT\_DATE() DATEDIFF(v1, v2)

DATE\_ADD(v, INTERVAL 10 DAY) DATE\_FORMAT(v,p)

p: "%Y-%m-%d %H:%i:%s" "$e. $c. $y %k.%i"

# DCL Data Control Language

**GRANT** – příkaz pro přidělení oprávnění uživateli k určitým objektům

**REVOKE** – příkaz pro odnětí práv uživateli

# TCL Transaction Control Commands

**BEGIN** – zahájení transakce

**COMMIT** – potvrzení transakce

**ROLLBACK** – zrušení transakce, do původního stavu