

SQL II.

SQL I., DML, definovanie obmedzení

Obsah prednášky

- Organizácia
- Zadanie 2
- DML
 - Agregácie
 - Pridávanie, modifikovanie a mazanie dát
- DDL
 - Vytváranie
 - Definovanie tabuľky s obmedzeniami

Organizácia

- Nové príklady – cvičenie 3
- Zverejnenie zadania 2
 - g-drive
 - Databáza
 - Yaml
 - Zadanie
 - Link na popis dát <https://ekosystem.slovensko.digital/otvorene-data#ov>
- Nahláste linky
 - Cca 30 ľudí chýba

Zadanie 2

Data Manipulation Language

Agregácie

- Vracajú jednu hodnotu pre zoskupenie riadkov
- Príklad agregáčnych funkcií
 - Priemer – avg(stĺpec) – priemerná hodnota z vybraného stĺpca
 - Minimum – min(stĺpec) – vráti minimálnu hodnotu zo stĺpca
 - Maximum – max(stĺpec) – vráti maximálnu hodnotu zo stĺpca
 - Total: Sum(stĺpec) – vráti sumu
 - Počet: count – vráti počet hodnôt
- link: <https://www.postgresql.org/docs/13/functions-aggregate.html>

Agregácie - GROUP BY

- Možnosť vytvorenia zoskupenia, pre ktorý sa ma vypočítať určitá funkcia

```
SELECT stĺpce tabuľky  
FROM tabuľky  
WHERE atribút  
GROUP BY stĺpce tabuľky
```

Agregácie - COUNT

- COUNT (*)
- COUNT (I)
- COUNT (atribút)
- COUNT (DISTINCT atribút)

Agregácie - HAVING

- Za HAVING nasleduje selekčná podmienka na reláciu, ktorá je výsledkom zoskupenia a vyrátanej agregácie
 - neviaže sa táto podmienka na relácie za FROM

```
SELECT stĺpce tabuľky  
FROM tabuľky  
WHERE atribút  
GROUP BY stĺpce tabuľky  
HAVING podmienka
```

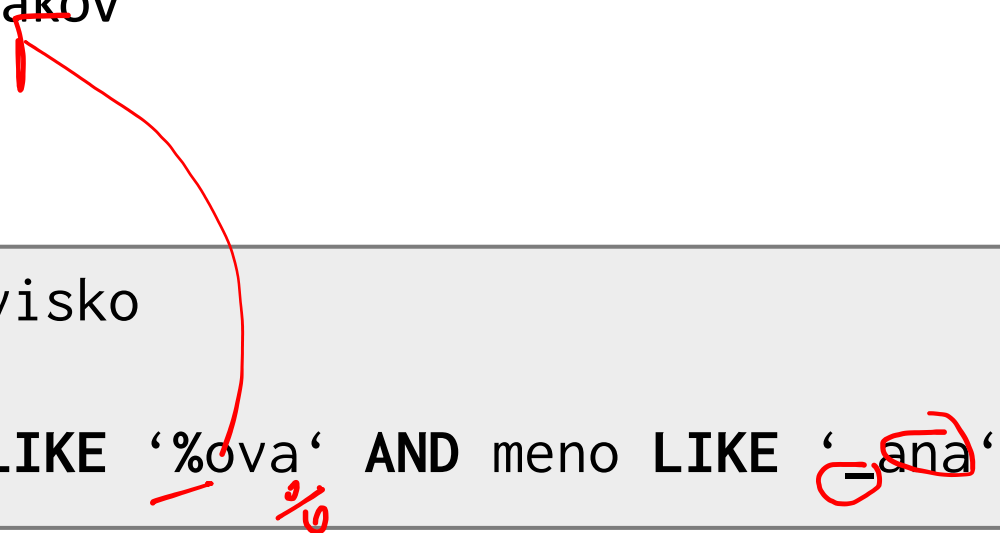
String operácie

- Rôzne operácie pre prácu s textom
 - UPPER
 - CONCAT
 - ...

LIKE

- slúži na vyhľadávanie prostredníctvom vzoru
 - % - 0 až N ľubovoľných znakov
 - _ - 1 ľubovoľný znak

```
SELECT meno, priezvisko  
FROM osoba  
WHERE priezvisko LIKE '%ova' AND meno LIKE '_ana'
```



nájdenie všetkých osôb, ktorých priezvisko končí na *ova* a meno môže byť Jana, Hana prípadne nejaké iné, ktoré sa líši v prvom znaku.

BETWEEN

- slúži na definovanie rozsahu hodnôt od-do (vrátane) alebo časových období (dátumov)
- Príklad:
SELECT priezvisko, meno FROM osoba WHERE vek **BETWEEN 20 AND 30**;

```
SELECT priezvisko, meno  
FROM osoba  
WHERE vek BETWEEN 20 AND 30
```

LIMIT

- Výber len určitého počtu záznamov od začiatku
 - Možnosť posunutia pomocou OFFSET

```
SELECT stĺpce tabuľky  
FROM tabuľky  
WHERE atribút  
LIMIT počet
```

DML - INSERT

- INSERT pridá nový záznam (riadok) alebo záznamy do relácie/tabuľky

→ **INSERT INTO** tabuľka _____
VALUES (hodnota1, hodnota2,, hodnotaN);

new Tid

INSERT INTO tabuľka (názov_atribútu1, názov_atribútu2, ,
názov_atribútuN)

VALUES

(hodnota1, hodnota2, , hodnotaN),
(hodnotax1, hodnotax2, , hodnotaxN);

DML - INSERT

- INSERT pridá nový záznam (riadok) alebo záznamy do relácie/tabuľky

INSERT INTO

tabuľka

VALUES

(hodnota1, hodnota2,, hodnotaN);

Záleží na poradí

Nezáleží na poradí vzhľadom
na poradie stĺpcov v tabuľke

INSERT INTO

tabuľka (názov_atribútu1, názov_atribútu2,,
názov_atribútuN)

VALUES

(hodnota1, hodnota2,, hodnotaN),
(hodnotax1, hodnotax2,, hodnotaxN);

DML - UPDATE

- UPDATE zmení existujúci záznam (riadok) alebo záznamy v relácií
- počet modifikovaný hodnôt je voliteľný a tiež použitie podmienky
- v podmienke WHERE sa môže nachádzať čokoľvek, čo môže byť vo WHERE v rámci príkazu SELECT

UPDATE tabuľka

SET atr1=nová_hodnota1, ..., atrN=nová_hodnotaN

WHERE podmienka;

DML - DELETE

- príkaz DELETE odstráni záznam (riadok) alebo záznamy v rámci relácie
- v podmienke WHERE sa môže nachádzať čokoľvek, čo môže byť vo WHERE v rámci príkazu SELECT

```
DELETE FROM tabuľka  
WHERE podmienka
```

Data Definition Language

Vytvorenie tabuľky - CREATE TABLE

- Vytvorenie relácie/tabuľky
- **Typ dát** – určuje typ hodnôt, ktoré daná atribút môže nadobúdať
- **Obmedzenia (constraint)** - hodnoty pri vkladaní, modifikácií alebo mazaní majú ohraničenie, ktoré musia spĺňať
 - môžu byť napr. povinnosť uvedenia hodnoty, unikátna hodnota, primárny a cudzí kľúč ...

```
CREATE TABLE meno_tabuľky(  
    názov1 typ1 [obmedzenia];  
    názov2 typ2 [obmedzenia];  
    ...  
    názovN typN [obmedzenia];  
);
```

DDL - CREATE TABLE príklad

```
CREATE TABLE osoba (  
    meno          VARCHAR(20),  
    rodne_cislo   INTEGER,  
    vek           SHORTINT  
);
```

Typy dát

- poznáme
 - Reťazce (pevné alebo premenlivej dĺžky) - **TEXT, VARCHAR**
 - Celé čísla – **INTEGER, SHORTINT**
 - Desatinné čísla
 - S pohyblivou desatinou čiarkou – **REAL, DOUBLE**
 - Presné – **NUMERIC, DECIMAL**
 - Dátum a čas – **TIMESTAMP, DATE**
 - A mnoho ďalších
 - Závisle od DBMS
- Rôzne implementácie DBMS môžu mať rôzne dátové typy
 - **SERIAL** (PostgreSQL) vs **AUTO-INCREMENT** (MySQL)
 - SERIAL, BIGSERIAL (PostgreSQL)
 - SERIAL v MySQL – predstavuje ALIAS pre BIGINT UNSIGNED NOT NULL AUTO_INCREMENT UNIQUE.

Numeric vs Real

- **NUMERIC** – presné desatinné číslo,
 - <https://github.com/postgres/postgres/blob/master/src/backend/utils/adt/numeric.c>
 - Pomalšie vykonanie ako REAL
- **REAL** – odchylka, rýchlejšie spracovanie

Vymazanie tabuľky - DROP TABLE

- Odstránenie tabuľky
- **TRUNCATE** vs **DROP** vs **DELETE**
 - **DELETE**
 - DML
 - Možnosť použitia WHERE
 - Možnosť použitia ROLLBACK
 - **TRUNCATE**
 - DDL
 - Nemožnosť použitia WHERE a vymaže všetky záznamy z tabuľky
 - **DROP**
 - DDL
 - Vymazanie celej schémy

```
DROP TABLE názov_tabuľky;  
TRUNCATE názov_tabuľky;  
DELETE FROM názov_tabuľky;
```

Modifikácia tabuľky - ALTER

- Pridanie/vymazanie/modifikovanie atribútu/stĺpca

```
ALTER TABLE tabuľka  
ADD/DROP/MODIFY atribút typ(v prípade ADD/MODIFY)
```



Syntax závisi od DBMS

Modifikácia tabuľky - PostgreSQL

```
ALTER TABLE tabuľka  
ALTER COLUMN atribút(názov stĺpca)  
    TYPE dátový_typ  
    USING expression;
```

USING potrebné pre
konverziu na nový dátový
typ

Obmedzenia

- **NOT NULL** – atribút nemôže obsahovať NULL hodnotu, vždy musí byť vyplnené
- **UNIQUE** – jedinečné hodnoty
- **PRIMARY KEY** – primárny kľúč
- **CHECK** – definovanie určitých hodnôt, ktoré môže nadobúdať daný atribút
- **FOREIGN KEY** – cudzí kľúč
- **DEFAULT** – predvolená hodnota

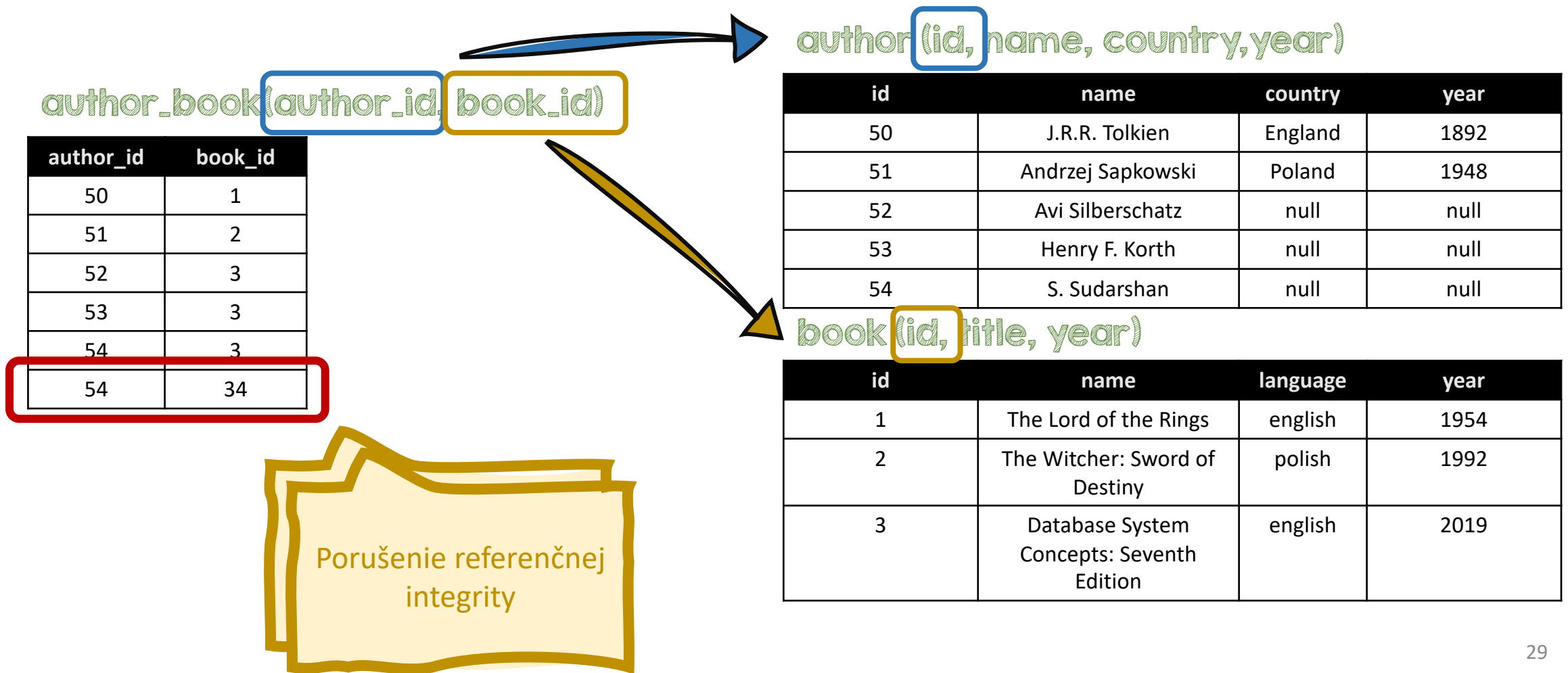
Primárny kľúč

- Je možné aby bol vyskladaný z jedného alebo viacerých parametrov

Referenčná integrita

- Referenčná integrita
 - Predstavuje stav dát, kedy odkazujúce dáta sa odvolávajú na platné záznamy

Porušenie referenčnej integrity



Porušenie referenčnej integrity

- Nastáva
 - odstránenie záznamu, na ktorý odkazuje iný záznam – teda odstránenie parent záznamu s PK, na ktorý odkazuje iný záznam s FK
 - vloženie/aktualizácia referencie na neexistujúci záznam
 - aktualizácia PK záznamu a neaktualizovanie child záznamov, ktoré sa odkazujú na daný parent záznam
 - ak potrebujete aktualizovať PK, je potrebné zamyslieť sa nad tým, či máte vhodný dizajn aplikácie
- Typy prístupov
 - CASCADE
 - reštrikčný RESTRICT
 - SET NULL or SET DEFAULT

Cudzí kľúč - foreign key

- Cudzí kľúč predstavuje vzťah medzi stĺpcami dvoch tabuliek
 - Tabuľka s cudzím kľúčom označovaná ako
 - **child**
 - Tabuľka obsahujúci primárny kľúč označovaná ako
 - **referenced** alebo **parent**
- Kľúčové slovo REFERENCES
 - syntax REFERENCES tabuľka(stĺpec)

Referenčná integrita - prístupy

- Typy prístupov
 - **CASCADE**
 - **RESTRICT**
 - **SET NULL**
 - **SET DEFAULT**
 - **NO ACTION**

CASCADE

- Vymazanie **parent** záznamu spôsobí vymazanie aj všetkých záznamov, ktoré naň odkazujú
- V prípade aktualizácií údajov dochádza k aktualizácií **child** záznamov

RESTRICT

- Nedovoľuje vymazanie alebo mazanie **parent** záznamu v prípade, že naň odkazuje child záznam

SET NULL / SET DEFAULT

- Zmazanie **parent** záznamu spôsobí nastavenie NULL alebo DEFAULT hodnoty pre **child** záznamy
- Rovnako aj v prípade aktualizácie parent záznamu (PK)

Zhrnutie

- DDL
 - Možnosť definovania rôznych obmedzení – presunutie aplikačnej logiky bližšie k DB

Čo ďalej?

- View
- Triggers
- Window functions