

# **Databázové systémy**

## **Zadanie 4**

### **Dátové modelovanie**

**Filip Híreš**

**Slovenská technická univerzita v Bratislave**

**Fakulta informatiky a informačných technológií**

**[xhires@stuba.sk](mailto:xhires@stuba.sk)**

**5.4.2024**

## Zadanie:

Hlavným cieľom tejto úlohy je pochopiť základné princípy návrhu databázy vytvorením konceptuálneho modelu (ER diagramu) a relačného dátového modelu (fyzický model) v podobe relačného diagramu pre databázu múzea. V rámci ER diagramu sa zamerajte na identifikovanie entít a vzťahov medzi týmito entitami. Okrem kardinality je potrebné uviesť aj popis daného vzťahu. V prípade relačného diagramu uvažujte, že má slúžiť pre implementáciu v rámci PostgreSQL databázového systému. Pri zápise vzťahov medzi reláciami používajte notáciu Crow's foot notation alebo UML notáciu. Pre navrhnutý relačný model pre PostgreSQL databázový systém je potrebné definovať všetky obmedzenia a tiež vytvoriť SQL príkazy, ktorým sa vytvorí samotná databáza. Samotné SQL príkazy musia obsahovať všetky obmedzenia, ktoré sú definované v schéme, napríklad not null, unique, default ... V rámci dokumentácie, je potrebné vysvetliť váš databázový model spolu s odôvodnením výberu jednotlivých atribútov, vzťahov, tabuliek. Dokumentácia musí byť realizovaná ako PDF dokumentácia s tým, že sa bude nachádzať v AIS odovzdaní.

Uvažujte o systéme pre múzeum, ktorý je zameraný na evidovanie exemplárov, ktoré má múzeum k dispozícii. Múzeum vlastní niektoré z exemplárov, ale niektoré exempláre má zapožičané z iných inštitúcií (napríklad iné múzeum, súkromná zbierka ...). Rovnako aj samotné múzeum môže obsahovať exempláre, ktoré zapožičalo nejakej inej inštitúcii. Preto je pre múzeum dôležité evidovať stav jednotlivých exemplárov ako je napríklad to, či sa jedná o vlastný exemplár alebo zapožičaný, rovnako aj to v akom stave sa daný exemplár nachádza tj. zapožičaný exemplár je na ceste, je vystavený, je v sklade, je na ceste naspäť prípadne ďalšie. Jednotlivé exempláre spadajú tiež do rôznych kategórií. Každý exemplár má vždy priradenú nejakú kategóriu na základe, ktorej si vie múzeum uskutočňovať vyhľadávania nad jednotlivými exemplármi. Múzeum tiež organizuje rôzne expozície, do ktorých sú zaradené rôzne exempláre. Samotný exemplár môže byť vystavený len v rámci jednej expozície a teda nie je možné aby jeden exemplár bol v rovnakom čase na dvoch rôznych expozíciách. Systém musí tiež brať do úvahy aj to, že ak exemplár je na ceste, tak nie je možné ho v danom čase vystavovať. Múzeum si drží len odhadovanú informáciu, kedy má prísť a tiež v prípade exemplárov sú tiež uskutočnené procesy kontroly v prípade, že daný exemplár bol zapožičaný inej inštitúcii. Každý exemplár vyžaduje inú dobu kontroly a tieto kontroly sú evidované rovnako s ich dĺžkou. Keďže múzeum obsahuje viacero priestorov/zón, tak expozícia sa môže vyskytovať v rámci jedného alebo viacerých priestorov. Nie je však možné aby v rámci jednej zóny boli v rovnakom čase dve expozície. Múzeum si tiež eviduje jednotlivé expozície, ktoré prebiehajú, ale rovnako aj tie ktoré sú v príprave. Múzeum tiež eviduje všetky historické expozície spolu s prehľadom vystavovaných exemplárov.

V rámci výsledného dokumentu opíšte základné procesy v rámci múzea spolu so životným cyklom jednotlivých entít (kedy vznikajú aké záznamy, prípadne, kde je potrebné vykonať aký SQL dotaz). Menovite nás budú zaujímať nasledujúce procesy:

- Naplánovanie expozície.
- Vkladanie nového exempláru.
- Presun exempláru do inej zóny.
- Prevzatie exempláru z inej inštitúcie.
- Zapožičanie exempláru z inej inštitúcie.

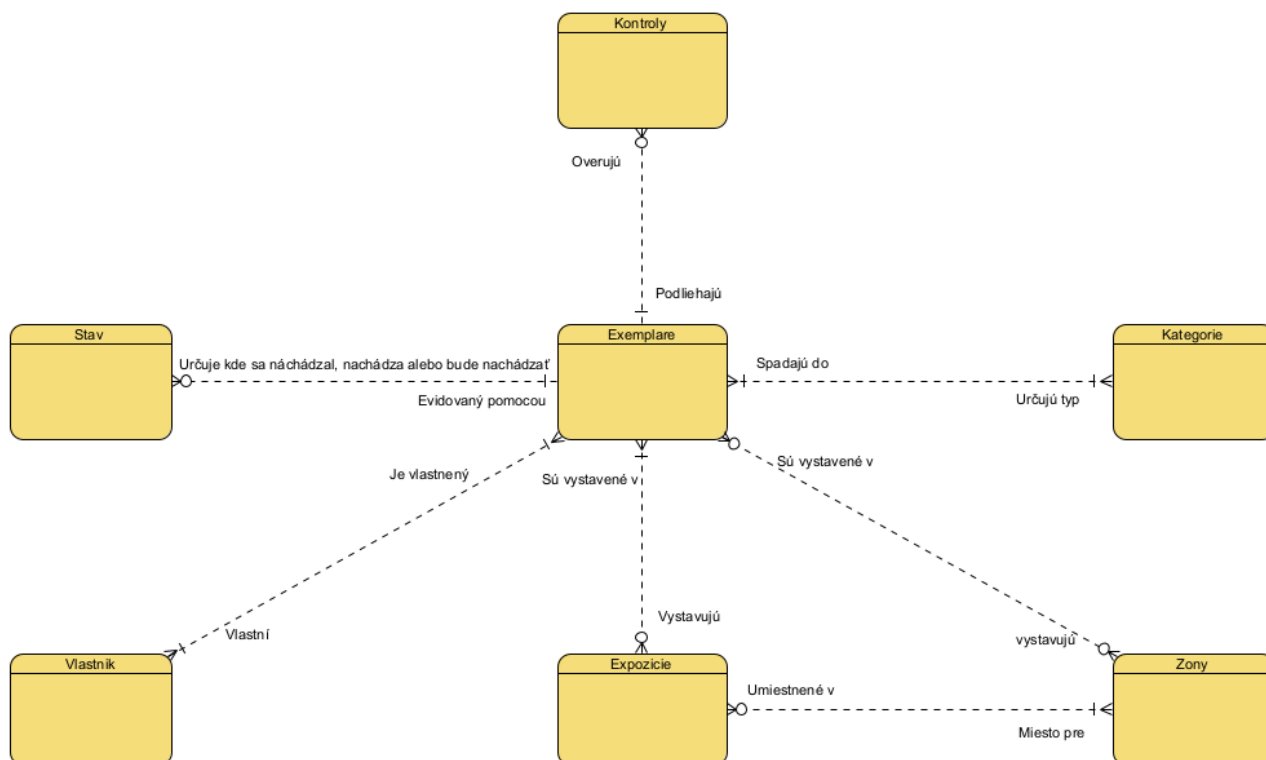
Databázu navrhujte tak, aby malo múzeum k dispozícii vždy prehľad o tom, kde sa nachádzajú jednotlivé exempláre, v akom sú stave a prípadne, kde sa nachádzali v minulosti. Nezabudnite definovať obmedzenia Vášho návrhu.

## Implementácia:

Návrh databázy som realizoval pomocou Visual Paradigm, v ktorom som vytvoril diagramy pre konceptuálny a fyzický relačný model. SQL príkazy som vytvoril a otestoval v pgAdmin4. Pre lepšiu prehľadnosť sú diagramy aj vo forme prílohy.

## Logický model:

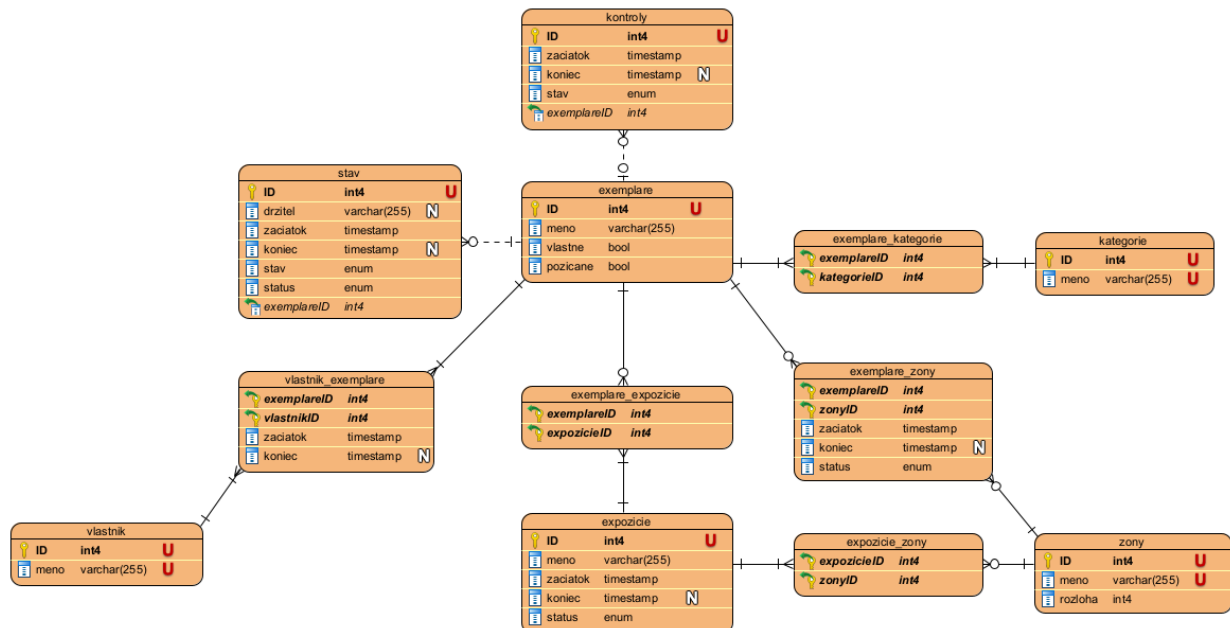
Nasledujúci obrázok zobrazuje diagram logického modelu navrhnuitej databázy.



Ústrednou entitou databázy sú samotné exempláre. Exempláre majú vždy nejakého vlastníka a počas svojho života môžu vystriedať niekoľkých. Rovnako každý vlastník vlastní jeden alebo viac exemplárov. Tieto exempláre môžu spadať do jednej alebo viacerých kategórií. Každá kategória obsahuje jeden alebo viac exemplárov. Exempláre sú vystavované na expozíciách. Exemplár sa v priebehu času môže vyskytnúť na viacerých expozíciách (avšak iba na jednej súčasne) ale taktiež sa nemusí vyskytnúť na žiadnej. Pokiaľ je expozícia naplánovaná vždy sa na nej vyskytne aspoň jeden exemplár. Expozície sa uskutočňujú v určitých zónach budovy. Každá expozícia sa odohráva v jednej alebo viacerých zónach. Zóny v priebehu času môžu hostiť žiadnu alebo viacero expozícií (avšak iba jednu súčasne). História každého exempláru (Uskladnenie, preprava, vystavenie...) je evidovaná pomocou stavu a každý exemplár pravdepodobne počas svojho života prejde niekoľkými stavmi. Pokiaľ je exemplár zapožičaný inej inštitúcii môže prejsť procesom kontroly a to aj viac krát v priebehu času.

## Fyzický model:

Nasledujúci obrázok zobrazuje diagram logického modelu navrhnutej databázy.



Každý exemplár má svoje meno a dve booleovské hodnoty. Vlastné nadobúda hodnotu True ak exemplár patrí nášmu múzeu. Premenná požičané je True ak sa exemplár nenachádza v inštitúcii, ktorá ho vlastní (To znamená, že požičané má hodnotu True ak je exemplár náš a požičali sme ho inej inštitúcii alebo ak nie je náš a máme ho požičaný).

Tabuľka pre vlastníkov exemplárov obsahuje meno inštitúcie alebo múzea. Táto tabuľka je s exemplármi prepojená pomocou tabuľky vlastnik\_exemplare, ktorá udáva odkedy vlastník vlastnil konkrétne exempláre, prípadne kedy ich prevzal iný majiteľ.

Každá kategória má svoje meno a s exemplármi je prepojená pomocou tabuľky exemplare\_kategorie.

Tabuľka expozície obsahuje meno, plánovaný začiatok a koniec expozície. Hodnota koniec môže byť NULL pokiaľ ešte nie je naplánované kedy presne bude ukončená. Hodnota status môže obsahovať 4 rôzne hodnoty: naplánované, ak je expozícia v procese prípravy. Prebiehajúce, ak expozícia už začala ale ešte neskončila. Ukončené, ak expozícia už skončila. Zrušené, ak bola naplánovaná ale z nejakého dôvodu sa neuskutoční, poprípade ak sa zmenil plán expozície, môžeme expozíciu označiť ako zrušenú a vytvoriť nový záznam s upravenými dátami.

Zóny majú okrem mena aj informáciu o ich rozlohe, ktorá pomôže pri plánovaní expozície. S tabuľkou expozície je prepojená pomocou expozicie\_zony.

Tabuľka stav je podobná tabuľke expozícií. Navyše obsahuje údaj o aktuálnom držiteľovi (nie vlastníkovy), stave, ktorý udáva, či bol exemplár v danom čase uskladnený, prepravovaný, alebo vystavený. Táto tabuľka je dôležitá pre plánovanie, aby sa nestalo, že sa exemplár naplánuje na dve expozície naraz, poprípade aby sa nenaplánovalo jeho vystavenie na čas, keď bude prepravovaný.

Posledná tabuľka obsahuje záznamy o kontrolách zapožičaných exemplárov. Čas začiatku a konca kontroly a stav kontrolovaného exempláru (Teda či bol v poriadku poprípade poškodený).

## Vytvorenie databázy:

Tabuľky v databáze som vytvoril pomocou sql kódu, ktorý je v prílohe Zadanie\_4\_sql.

## Opis základných procesov:

V tejto časti sa budem venovať opisu niektorých základných procesov, ktoré môžu nastať pri práci s navrhnutou databázou.

**Naplánovanie expozície** – Pri plánovaní expozície môžeme použiť dotazy na získanie informácií o našich exemplároch. Napríklad vyfiltrovať len exempláre s určitými kategóriami.

Napríklad ak chceme zorganizovať expozíciu vikingských exponátov môžeme si vyfiltrovať záznamy s kategóriu vikingovia, ktoré momentálne nie sú požičané

```
SELECT exemplare.meno as meno_exemplaru, vlastnik.meno as meno_vlastnika,  
vlastnik_exemplare.zaciatok as zaciatok_vlastnictva, vlastnik_exemplare.koniec as  
koniec_vlastnictva, array_agg(kategorie.meno) as kategorie FROM exemplare  
JOIN exemplare_kategorie on exemplare.id = exemplareID  
JOIN kategorie on kategorieID = kategorie.id  
JOIN vlastnik_exemplare on exemplare.id = vlastnik_exemplare.exemplareID  
JOIN vlastnik on vlastnikid = vlastnik.id  
WHERE pozicane = FALSE  
GROUP BY (exemplare.id, vlastnik.meno, vlastnik_exemplare.zaciatok,  
vlastnik_exemplare.koniec)  
HAVING 'vikingovia' = ANY (array_agg(kategorie.meno));
```

	meno_exemplaru character varying (255)	meno_vlastnika character varying (255)	zaciatok_vlastnictva timestamp without time zone	koniec_vlastnictva timestamp without time zone	kategorie character varying[]
1	vikingsky mec	Moje Muzeum	2024-04-06 14:46:25.975644	[null]	{vikingovia,mec}
2	vikingsky stit	Moje Muzeum	2024-04-06 15:05:17.906878	[null]	{stit,vikingovia}
3	dracia lod	Moje Muzeum	2024-04-06 15:06:15.17241	[null]	{lod,doprava,vikingovia}
4	vikingsky roh	Moje Muzeum	2024-04-06 15:07:41.917707	[null]	{stolovanie,vikingovia,jedlo}

Vybrané exempláre budú vystavené, a teda vznikne nový záznam v tabuľke stav. Taktiež vznikne záznam v samotnej tabuľke expozícií. Pre naplánovanie priestorov expozície sa dá využiť údaj o rozlohe daných zón.

Táto query nám umožní vybrať vhodné zóny na základe rozlohy podľa počtu exemplárov, ktoré chceme vystaviť.

```
SELECT meno, rozloha  
FROM zony  
WHERE  
CASE  
WHEN (  
SELECT COUNT(*) AS total_count  
FROM (  
SELECT exemplare.id  
FROM exemplare  
JOIN exemplare_kategorie ON exemplare.id = exemplareID  
JOIN kategorie ON kategorieID = kategorie.id  
WHERE pozicane = FALSE  
GROUP BY exemplare.id  
HAVING 'vikingovia' = ANY (array_agg(kategorie.meno))  
)  
)
```

```

) >= 100
THEN rozloha >= 100
ELSE rozloha < 100
END;

```

	meno character varying (255) 🔒	rozloha integer 🔒
1	B	50
2	C	30

Samozrejme exemplár nemôže byť na dvoch miestach naraz. Systém nám neumožní vytvoriť záznam o expozícii pokiaľ by sa prekrývali časy prepravy alebo inej expozície. To by sa dalo riešiť pomocou triggerov alebo v API.

**Vkladanie nového exempláru** – Pri vkladaní nového exempláru pridávame pomocou funkcie insert nové záznamy do tabuľky exemplárov. Taktiež vzniká záznam v tabuľke vlastník exempláre a tiež treba priradiť exempláru správne kategórie. Nový exemplár bude musieť byť tiež dopravený do nášho múzea a následne uskladnený alebo vystavený takže vzniknú aj záznamy v tabuľke stav.

Pre vkladanie nového exempláru môžeme použiť

```

DO $$
DECLARE
exemplareid INTEGER;
vlastnikid INTEGER;
kategoriaid1 INTEGER;
kategoriaid2 INTEGER;
BEGIN
INSERT INTO exemplare(meno, vlastne, pozicane) VALUES
('vikingsky mec', TRUE, FALSE); SELECT id INTO exemplareid FROM exemplare
WHERE meno = 'vikingsky mec';
SELECT id INTO vlastnikid FROM vlastnik
WHERE meno = 'Moje Muzeum';
SELECT id INTO kategoriaid1 FROM kategorie
WHERE meno = 'mec';
SELECT id INTO kategoriaid2 FROM kategorie
WHERE meno = 'vikingovia';
INSERT INTO vlastnik_exemplare(exemplareid, vlastnikid, zaciatok) VALUES
(exemplareid, vlastnikid, CURRENT_TIMESTAMP);
INSERT INTO exemplare_kategorie(exemplareid, kategorieid) VALUES
(exemplareid, kategoriaid1),
(exemplareid, kategoriaid2);
END $$;

```

Po vyhodnotení query

```

SELECT exemplare.meno as meno_exemplaru, vlastnik.meno as meno_vlastnika,
vlastnik_exemplare.zaciatok as zaciatok_vlastnictva, vlastnik_exemplare.koniec as
koniec_vlastnictva, array_agg(kategorie.meno) as kategorie FROM exemplare
JOIN exemplare_kategorie on exemplare.id = exemplareid
JOIN kategorie on kategorieid = kategorie.id
JOIN vlastnik_exemplare on exemplare.id = vlastnik_exemplare.exemplareid
JOIN vlastnik on vlastnikid = vlastnik.id

```

GROUP BY (exemplare.id, vlastnik.meno, vlastnik\_exemplare.zaciatok, vlastnik\_exemplare.koniec);

môžeme vidieť, že nový exemplár bol pridaný.

	meno_exemplaru character varying (255)	meno_vlastnika character varying (255)	zaciatok_vlastnictva timestamp without time zone	koniec_vlastnictva timestamp without time zone	kategorie character varying[]
1	vikingsky mec	Moje Muzeum	2024-04-06 14:46:25.975644	[null]	{mec,vikingovia}

**Presun exempláru do inej zóny** – Pokiaľ sa zmenia plány expozície a chceme exemplár presunúť do inej zóny, v tabuľke exemplare zóny sa v danom zázname ukončí vystavenie exempláru v pôvodnej zóne (status sa zmení na ukončené alebo zrušené) a vznikne záznam pre vystavenie v novej zóne. Taktiež treba myslieť na to, že pokiaľ sa exemplár presúva do zóny, ktorá nebola pôvodne súčasťou expozície, treba túto zmenu uviesť aj v tabuľke expozícií.

**Prevzatie exempláru z inej inštitúcie** – Keď prevezmeme exemplár, pôvodný majiteľ ho už nevlastní, takže v tabuľke vlastnik exemplár upravíme záznam zmenením hodnoty null v stĺpci koniec na konkrétny čas prevzatia. Následne musí vzniknúť nový záznam, v ktorom bude naše múzeum uvedené ako vlastník daného exempláru.

```
UPDATE vlastnik_exemplare
SET vlastnikid = 1, koniec = CURRENT_TIMESTAMP
WHERE exemplareid = 12
```

```
insert into vlastnik_exemplare(exemplareid, vlastnikid, zaciatok) values
(12, 2, CURRENT_TIMESTAMP);
```

```
select * from exemplare
join vlastnik_exemplare on exemplare.id = vlastnik_exemplare.exemplareid
join vlastnik on vlastnik_exemplare.vlastnikid = vlastnik.id
where exemplare.id = 12
```

	id integer	meno character varying (255)	vlastne boolean	pozicane boolean	exemplareid integer	vlastnikid integer	zaciatok timestamp without time zone	koniec timestamp without time zone	id integer	meno character varying (255)
1	12	vikingsky roh	true	false	12	1	2024-04-06 15:07:41.917707	2024-04-06 15:55:31.217905	1	Moje Muzeum
2	12	vikingsky roh	true	false	12	2	2024-04-06 15:54:58.176866	2024-04-06 15:55:31.217905	2	gregy

**Zapožičanie exempláru z inej inštitúcie** – Pri zapožičaní exempláru sa zmení booleovská hodnota v tabuľke exempláru v stĺpci požičané na True. Taktiež vzniká záznam o preprave, uskladnení a prípadných kontrolách.