

Slovenská technická univerzita v Bratislave  
Fakulta informatiky a informačných technológií

Zadanie 4  
Databázové systémy  
Adam Jurčišin

Dátum : 7. 4. 2024

## Obsah

1.	ÚVOD.....	2
2.	ER DIAGRAM .....	2
3.	FYZICKÝ MODEL.....	3
4.	NAPLÁNOVANIE EXPOZÍCIE .....	3
5.	VKLADANIE NOVÉHO EXEMPLÁRU .....	4
6.	PRESUN EXEMPLÁRU DO INEJ ZÓNY .....	4
7.	PREVZATIE EXEMPLÁRU Z INEJ INŠTITÚCIE .....	5
8.	ZAPOŽIČANIE EXEMPLÁRU Z INEJ INŠTITÚCIE.....	5
9.	VYTVORENIE DATABÁZY.....	6
	ZÁVER.....	7

## 1. Úvod

V tomto zadání máme za úlohu vytvoriť návrh databázy pre múzeum. Pre korektný návrh databázy vytvoríme ER diagram a fyzický model.

Zadanie budeme navrhovať v prostredí diagrams.net.

## 2. ER diagram

ER diagram, ktorý vznikol na základe zadania:

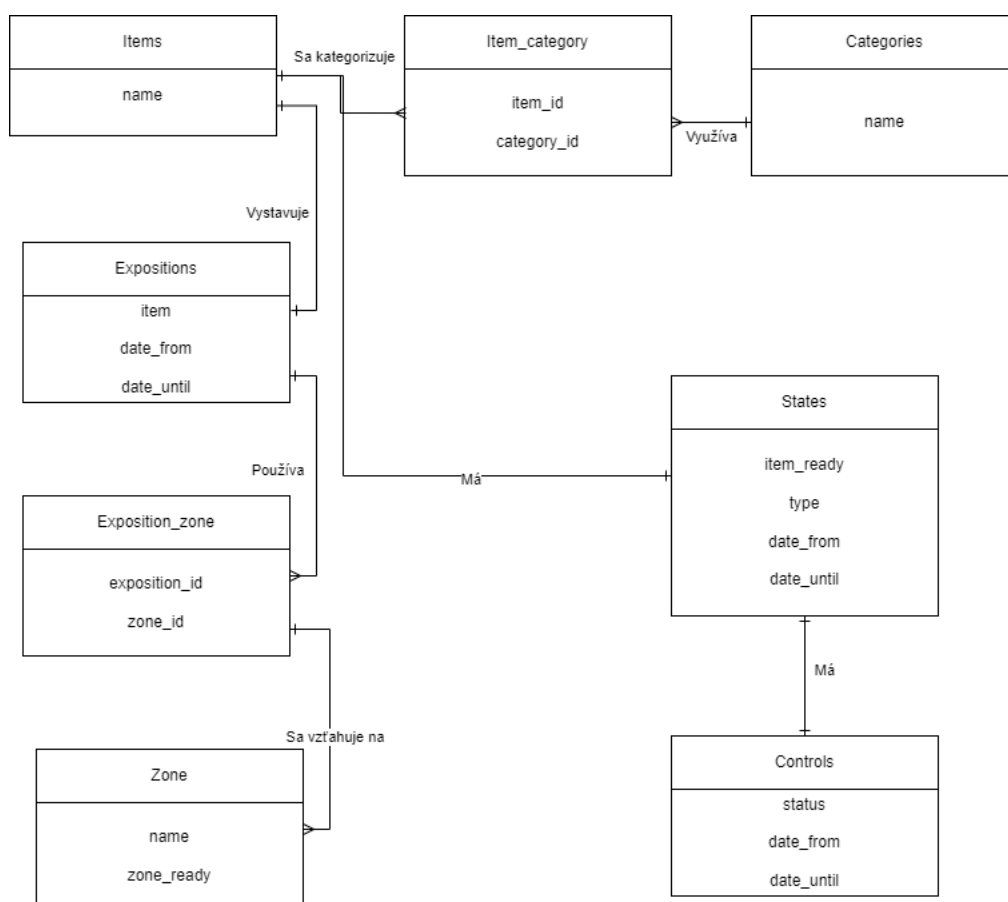


Diagram 1

V ER diagrame sme vytvorili 8 tabuliek s atribútmi potrebnými pre splnenie zadania. V diagrame využívame kardinality opisujúce vzťahy medzi jednotlivými tabuľkami.

### 3. Fyzický model

Fyzický model, ktorý bol vytvorený z ER diagramu:

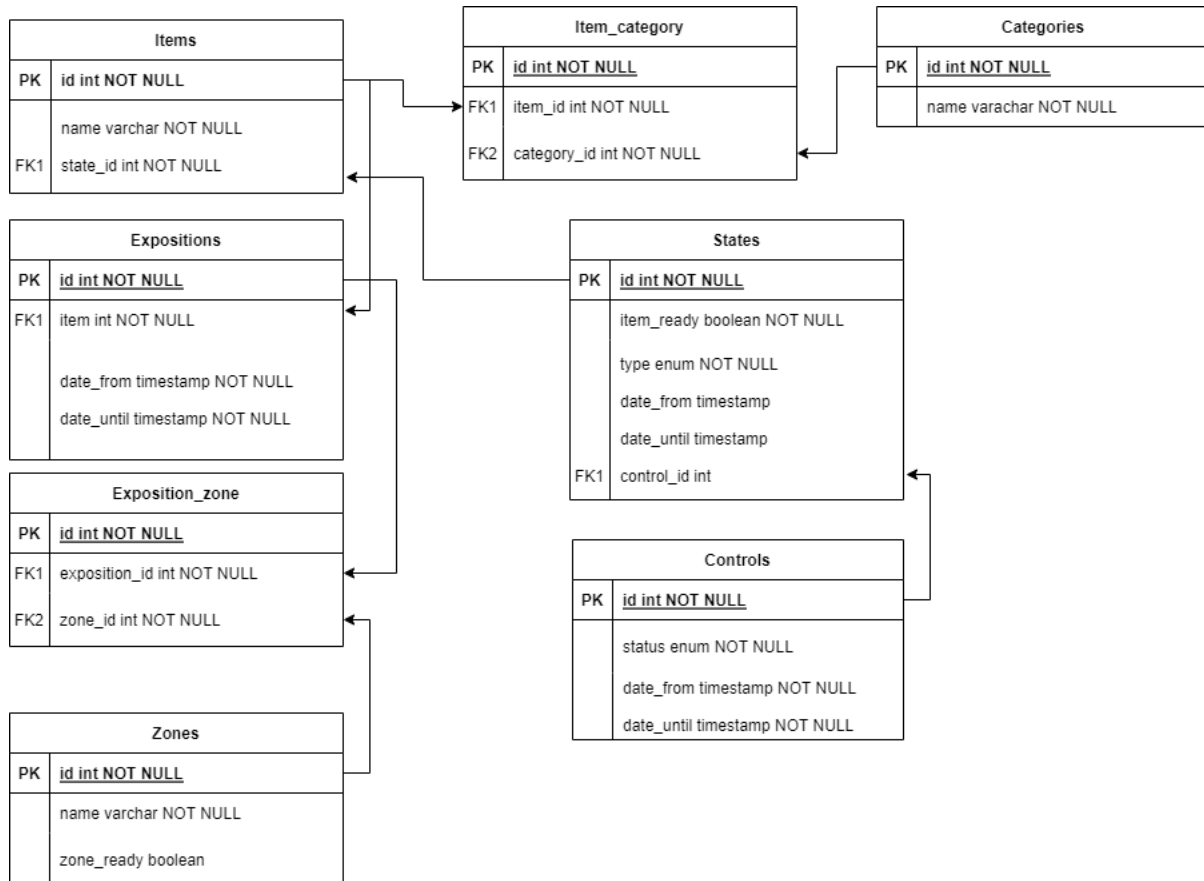


Diagram 2

Vo fyzickom modeli využívame 8 tabuliek. Tabuľky obsahujú atribúty spolu s dátovými typmi, ktoré budú využívané v databáze, zároveň opisujeme primárne kľúče a cudzie kľúče. Taktiež opisujeme atribúty, ktoré nemôžu naberať hodnotu NULL.

### 4. Naplánovanie expozície

Pre naplánovanie expozície je potrebné mať vytvorené všetky tabuľky. Následne pre vytvorenie novej expozície je potrebné mať k dispozícii aspoň 1 exponát, ktorý bude vystavovaný. Pre zistenie dispozícií exponátu je potrebné pomocou tabuľky item\_state získať id pre položku v tabuľke state\_id, pomocou ktorého si v tabuľke states zistíme dostupnosť exponátu. Pokiaľ je pre daný exponát v tabuľke states hodnota stĺpca item\_ready na boolean hodnotu TRUE, potom je exponát k dispozícii.

Následne pre vytvorenie expozície použijeme tabuľku `expositions`, kde vyberieme `id` exempláru, určíme dátum expozície, vytvoríme nový záznam v tabuľke `exposition_zone`, kde vytvoríme vzťah medzi expozíciou a zónou v ktorej bude expozícia, pokiaľ je stĺpec `zone_ready` vybranej zóny v boolean hodnote `TRUE`, potom je zóna dostupná. Vzťahov v tabuľke `exposition_zone` vytvoríme toľko, koľko zón budeme využívať, kde `exposition_id` bude rovné `id` expozície a `zone_id` rovné zónam, ktoré sa budú využívať.

Dostupnosť zón je určená podľa dĺžky expozície, ktorá je nastavená v tabuľke `expositions`.

## 5. Vkladanie nového exempláru

Pre vloženie nového exempláru je potrebné mať vytvorené všetky tabuľky. Následne vytvoríme nový záznam v tabuľke `items`, v tabuľke `item_category` vytvoríme vzťah medzi exemplárom a kategóriou do ktorej sa exemplár radí. Kategóriu exempláru vyberieme z tabuľky `categories`, `id` kategórie uložíme do `category_id` a `id` exempláru do `item_id`.

Následne určíme stav exempláru. Vytvoríme nový záznam v tabuľke `states`, kde určíme aký je typ stavu, či je exemplár vlastný, alebo zapožičaný, na sklade, na ceste a podobne. Pokiaľ je exemplár k dispozícii iba dočasne, v `date_from` a `date_until` máme tento čas uložený, pokiaľ potrebuje exemplár prejsť kontrolou, v tabuľke `controls` vytvoríme nový záznam, kde určíme status kontroly, či je kontrola v procese, hotová alebo zlyhala, taktiež určíme trvanie kontroly v `date_from` a `date_until`. Pokiaľ je všetko v poriadku a exemplár je pripravený k použitiu v tabuľke `states` je `item_ready` nastavený na boolean `TRUE`.

## 6. Presun exempláru do inej zóny

Pokiaľ chceme exemplár presunúť do inej zóny v rámci expozície, musíme mať vytvorené všetky tabuľky.

V tabuľke `zones` skontrolujeme dostupnosť zóny do ktorej má byť exemplár presunutý, následne v tabuľke `exposition_zone` vytvoríme nový záznam vzťahu `id` expozície s `id` inej zóny a záznam predchádzajúcej zóny z tabuľky vymažeme.

## 7. Prevzatie exempláru z inej inštitúcie

Pokiaľ chceme exemplár prevziať z inej inštitúcie musíme mať vytvorené všetky tabuľky a vytvorený záznam o exemplári v tabuľke items podobne ako sa to robilo pri vkladaní nového exempláru.

Prevzatý exemplár musí prebehnúť kontrolou, to znamená, že vytvoríme nový záznam v tabuľke controls a nastavíme trvanie kontroly. Id kontroly je uložené v tabuľke states, kde je uložený záznam o stave exempláru. Pokiaľ kontrola prebehne v poriadku v tabuľke controls sa status zmení na done, v tabuľke states sa item\_ready boolean hodnota zmení na TRUE a exemplár je úspešne prevzatý z inej inštitúcie a k dispozícii.

## 8. Zapožičanie exempláru z inej inštitúcie

Pokiaľ chceme exemplár zapožičať z inej inštitúcie musíme mať vytvorené všetky tabuľky a vytvorený záznam o exemplári v tabuľke items podobne ako sa to robilo pri vkladaní nového exempláru.

Pokiaľ máme záznam o exemplári vytvorený a exemplár k nám EŠTE NEDORAZIL, to znamená, že je na ceste k nám, v tabuľke states je type nastavený na enum hodnotu na\_cestě, rovnako sú nastavené aj date\_from a date\_until, ktoré značia čas cesty.

Pokiaľ máme záznam o exemplári vytvorený a exemplár k nám UŽ DORAZIL, v tabuľke states je type nastavený na enum hodnotu v\_kontrolě, kým sa vykoná kontrola exempláru. Následne sa nastaví type na zapozicany a nastavi sa date\_from a date\_until na dátum odkedy a dokedy máme exemplár zapožičaný.

Pokiaľ čas zapožičania uplynul a exemplár vraciame v tabuľke states je type nastavený na enum hodnotu na\_cestě\_naspat a dátum nastavený na dĺžku cesty.

## 9. Vytvorenie databázy

Na obrázkoch 1-3 máme znázornené SQL pre databázu PostgreSQL pre vytvorenie tabuliek systému pre múzea.

```
CREATE TYPE "state_type" AS ENUM (  
    'vlastny',  
    'zapozicany',  
    'na_ceste',  
    'v_sklade',  
    'na_ceste_naspat',  
    'v_kontrolle'  
);  
  
CREATE TYPE "control_status" AS ENUM (  
    'none',  
    'done',  
    'failed',  
    'in_progress'  
);  
  
CREATE TABLE "items" (  
    "id" integer PRIMARY KEY,  
    "name" varchar NOT NULL,  
    "state_id" integer NOT NULL  
);  
  
CREATE TABLE "item_category" (  
    "id" integer PRIMARY KEY,  
    "item_id" integer NOT NULL,  
    "category_id" integer NOT NULL  
);  
  
CREATE TABLE "categories" (  
    "id" integer PRIMARY KEY,  
    "name" varchar UNIQUE NOT NULL  
);
```

Obrázok 1

```
CREATE TABLE "expositions" (  
  "id" integer PRIMARY KEY,  
  "item" integer NOT NULL,  
  "date_from" timestamp NOT NULL,  
  "date_until" timestamp NOT NULL  
);  
  
CREATE TABLE "exposition_zone" (  
  "id" integer PRIMARY KEY,  
  "exposition_id" integer NOT NULL,  
  "zone_id" integer NOT NULL  
);  
  
CREATE TABLE "zones" (  
  "id" integer PRIMARY KEY,  
  "name" varchar UNIQUE NOT NULL,  
  "zone_ready" boolean DEFAULT 1  
);  
  
CREATE TABLE "states" (  
  "id" integer PRIMARY KEY,  
  "item_ready" boolean DEFAULT 0,  
  "type" state_type NOT NULL,  
  "date_from" timestamp,  
  "date_until" timestamp,  
  "control_id" integer  
);  
  
CREATE TABLE "controls" (  
  "id" integer PRIMARY KEY,  
  "status" control_status NOT NULL,  
  "date_from" timestamp NOT NULL,  
  "date_until" timestamp NOT NULL  
);
```

Obrázok 2

```
ALTER TABLE "item_category" ADD FOREIGN KEY ("item_id") REFERENCES "items" ("id");  
ALTER TABLE "item_category" ADD FOREIGN KEY ("category_id") REFERENCES "categories" ("id");  
ALTER TABLE "states" ADD FOREIGN KEY ("id") REFERENCES "items" ("state_id");  
ALTER TABLE "expositions" ADD FOREIGN KEY ("item") REFERENCES "items" ("id");  
ALTER TABLE "states" ADD FOREIGN KEY ("control_id") REFERENCES "controls" ("id");  
ALTER TABLE "zones" ADD FOREIGN KEY ("id") REFERENCES "exposition_zone" ("zone_id");  
ALTER TABLE "exposition_zone" ADD FOREIGN KEY ("exposition_id") REFERENCES "expositions" ("id");
```

Obrázok 3

## Záver

Úspešne sme navrhli databázu pre systém múzea, vytvorili sme ER diagram a fyzický diagram, vysvetlili sme prácu s databázou a vytvorili SQL kód pre vytvorenie tabuliek.