

# Dokumentácia k zadaniu 4

## Návrh databázy múzea

**Autor:** Dominik Miškovič  
**Predmet:** Databázové systémy  
**Cvičiaci:** Ing. William Brach  
**Obdobie:** Letný semester 2024

# Obsah

<b>1</b>	<b>Hlavné tabuľky</b>	<b>3</b>
1.1	Opis atribútov . . . . .	3
1.2	Schéma databázy . . . . .	4
1.3	Enumerátory pre typy . . . . .	5
<b>2</b>	<b>Entity</b>	<b>6</b>
2.1	Entitno-Relačný diagram . . . . .	6
2.2	Opis vzťahov medzi entitami . . . . .	6
<b>3</b>	<b>Opis základných procesov</b>	<b>7</b>
3.1	Plánovanie expozície . . . . .	7
3.2	Vkladanie exempláru (do databázy) . . . . .	8
3.3	Vkladanie exempláru (do expozície) . . . . .	8
3.4	Prevzatie exempláru z inej inštitúcie . . . . .	9
3.5	Zapožičanie exempláru z inej inštitúcie . . . . .	10
<b>4</b>	<b>Predbežný návrh query pre vytvorenie databázy</b>	<b>11</b>

# 1 Hlavné tabuľky

## 1.1 Opis atribútov

**Categories:** Tabuľka obsahuje kategórie, do ktorých sa exempláre zaraďujú.

Atribúty:

- id: Identifikátor kategórie (primárny kľúč).
- name: Názov kategórie (jedinečný).

**Exemplars:** Kľúčová tabuľka s informáciami o exemplároch.

Atribúty:

- id: Identifikátor exemplára (primárny kľúč).
- category\_id: Referencia na kategóriu, do ktorej exemplár patrí (cudzí kľúč odkazujúci na categories.id).
- name: Názov exemplára (jedinečný).
- description: Popis exemplára.
- ownership\_status: Stav vlastníctva (vlastnený/zapožičaný).
- current\_status: Aktuálny stav exemplára (v sklade, na vystavení, kontrola, na ceste).
- loan\_id: Referencia na záznam o zapožičaní (cudzí kľúč odkazujúci na loans.id).

**Expositions:** Tabuľka s informáciami o expozíciách.

Atribúty:

- id: Identifikátor expozície (primárny kľúč).
- name: Názov expozície (jedinečný).
- start\_date: Dátum začatia expozície.
- end\_date: Dátum ukončenia expozície.
- status: Stav expozície (plánovaná, prebiehajúca, ukončená).

**Zones:** Tabuľka s informáciami o priestorových zónach.

Atribúty:

- id: Identifikátor zóny (primárny kľúč).
- name: Názov zóny.

**Institutions:** Tabuľka s informáciami o inštitúciách zapojených do zapožičiavania.

Atribúty:

- id: Identifikátor inštitúcie (primárny kľúč).
- name: Názov inštitúcie (jedinečný).

**Loans:** Tabuľka s detailmi o zapožičaných exemplároch.

Atribúty:

- id: Identifikátor záznamu o zapožičaní (primárny kľúč).
- exemplar\_id: Referencia na exemplár, ktorý bol zapožičaný (cudzí kľúč odkazujúci na exemplars.id).
- type: Typ zapožičania (zapožičané inej inštitúcií/od inej inštitúcií).
- involved\_institution\_id: Referencia na zapojenú inštitúciu (cudzí kľúč odkazujúci na institutions.id).
- loan\_start\_date: Dátum začatia zapožičania.

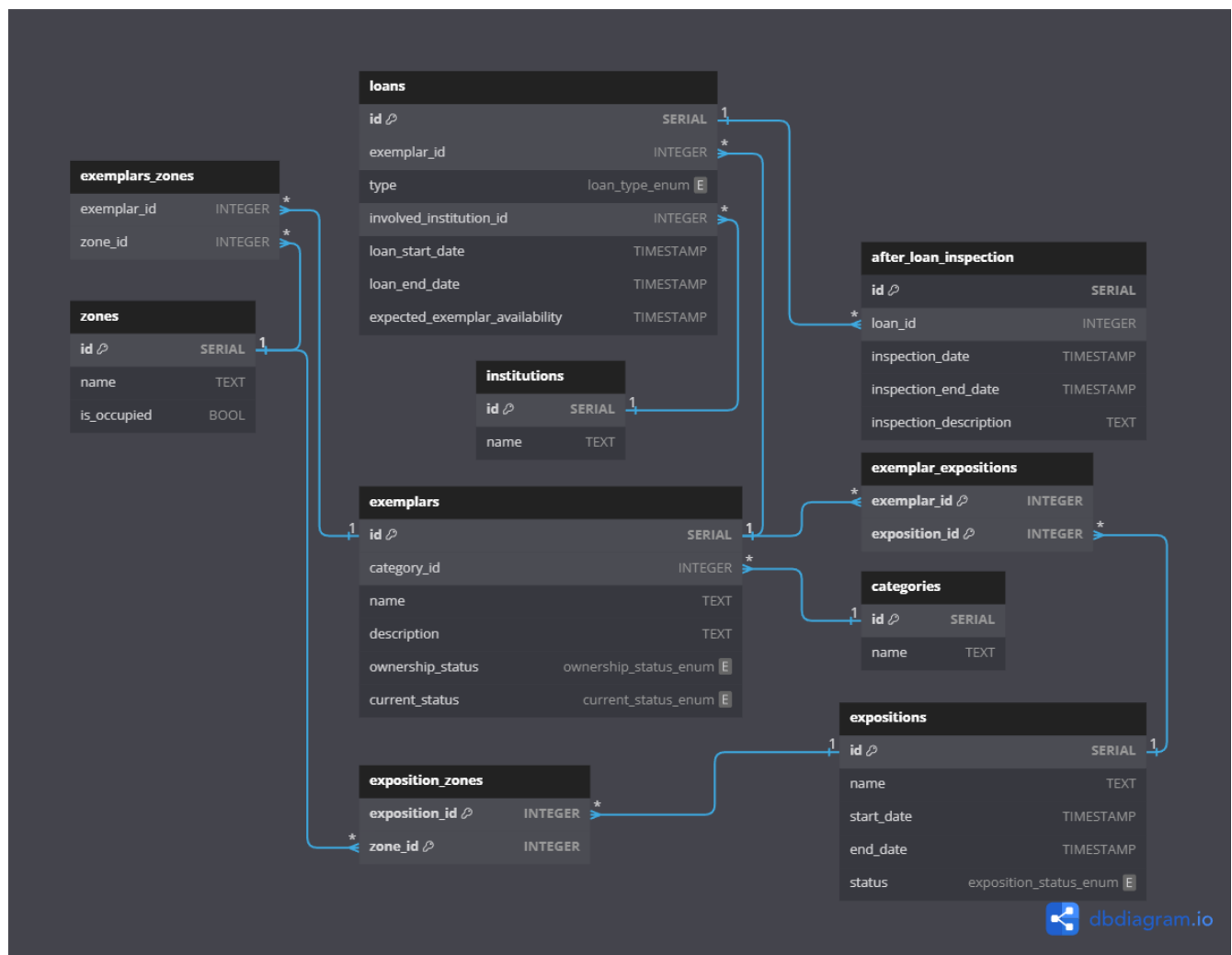
- loan\_end\_date: Dátum ukončenia zapožičania.
- expected\_exemplar\_availability: Očakávaný dátum dostupnosti exemplára po návrate.

**AfterLoanInspection:** Tabuľka s informáciami o kontrolách po ukončení zapožičania.

Atribúty:

- id: Identifikátor záznamu o kontrole (primárny kľúč).
- loan\_id: Referencia na záznam o zapožičaní (cudzí kľúč odkazujúci na loans.id).
- inspection\_date: Dátum kontroly.
- inspection\_end\_date: Dátum ukončenia kontroly.
- inspection\_description: Popis zistení z kontroly.

## 1.2 Schéma databázy



### 1.3 Enumeratory pre typy

```
1 CREATE TYPE "ownership_status_enum" AS ENUM (  
2     'OWNED',  
3     'LOANED'  
4 );  
5  
6 CREATE TYPE "current_status_enum" AS ENUM (  
7     'IN_STORAGE',  
8     'ON_DISPLAY',  
9     'BEING_INSPECTED',  
10    'IN_TRANSIT',  
11    'NONE'  
12 );  
13  
14 CREATE TYPE "exposition_status_enum" AS ENUM (  
15     'PLANNED',  
16     'IN_PROGRESS',  
17     'ENDED'  
18 );  
19  
20 CREATE TYPE "loan_type_enum" AS ENUM (  
21     'LOANED_TO',  
22     'LOANED_IN'  
23 );  
24
```

## 2 Entity

**Category:** Kategórie, do ktorých sa exempláre zaraďujú.

**Exemplar:** Exempláre s rôznymi vlastnosťami (stav, umiestnenie, história).

**Exposition:** Expozície s definovaným dátumom konania a umiestnením v zónach.

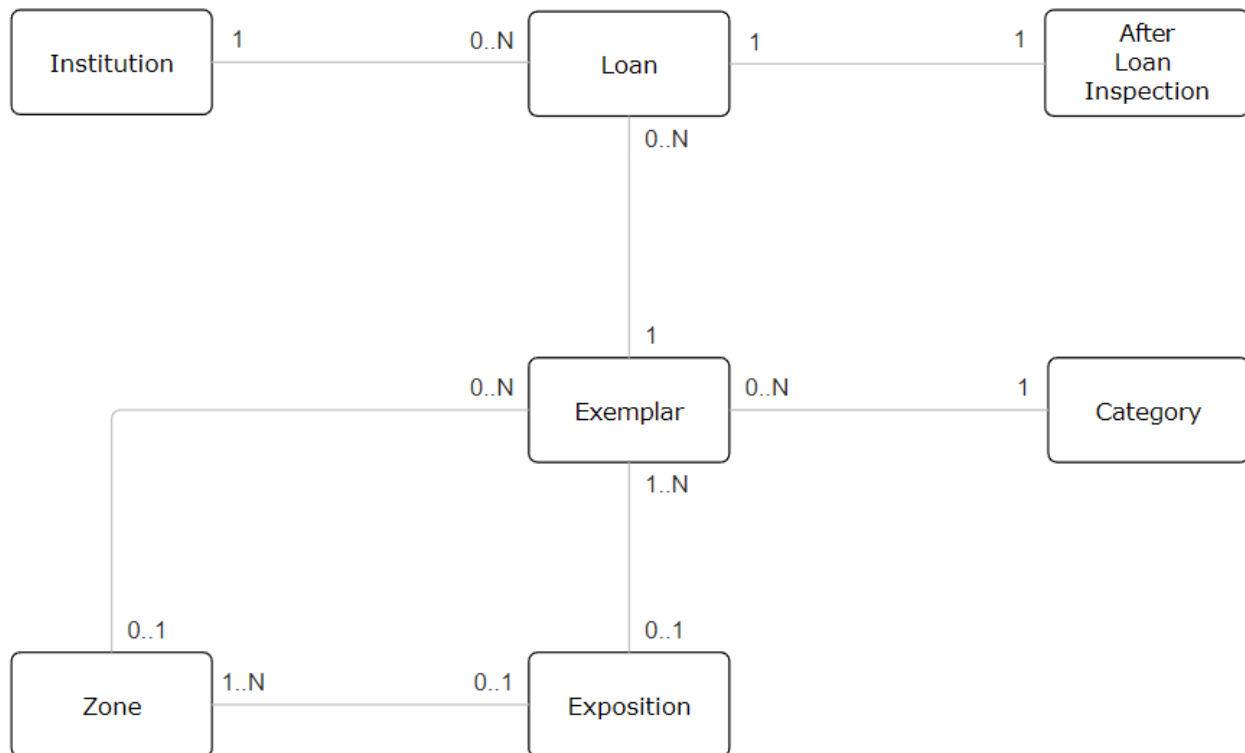
**Zone:** Priestorové zóny, v ktorých prebiehajú expozície.

**Institution:** Inštitúcie zapojené do procesu zapožičiavania exemplárov.

**Loan:** Záznamy o zapožičaných exemplároch s detailmi o inštitúcii a dátume.

**AfterLoanInspection:** Záznamy o kontrolách exemplárov po ich návrate z zapožičania.

### 2.1 Entitno-Relačný diagram



### 2.2 Opis vzťahov medzi entitami

#### **Exemplar** $\longleftrightarrow$ **Category**

Každý exemplár patrí práve do jednej kategórie.

Kategória môže obsahovať viacero exemplárov.

#### **Exemplar** $\longleftrightarrow$ **Exposition**

Exemplár môže byť súčasťou 0 až 1 expozície v daný čas.

Expozícia môže obsahovať 1 až N exemplárov v daný čas.

#### **Exemplar** $\longleftrightarrow$ **Zone**

Exemplár môže byť v 0 až 1 zóne v daný čas.

Zóna môže obsahovať 0 až N exemplárov.

#### **Exemplar** $\longleftrightarrow$ **Loan**

Exemplár môže byť viackrát zapožičaný.

Jedno zapožičanie sa týka 1 exemplára.

### Exposition $\longleftrightarrow$ Zone

Expozícia sa môže rozprestierať vo viacerých zónach.  
Zóna môže hostiť maximálne 1 expozíciu v daný čas.

### Loan $\longleftrightarrow$ Institution

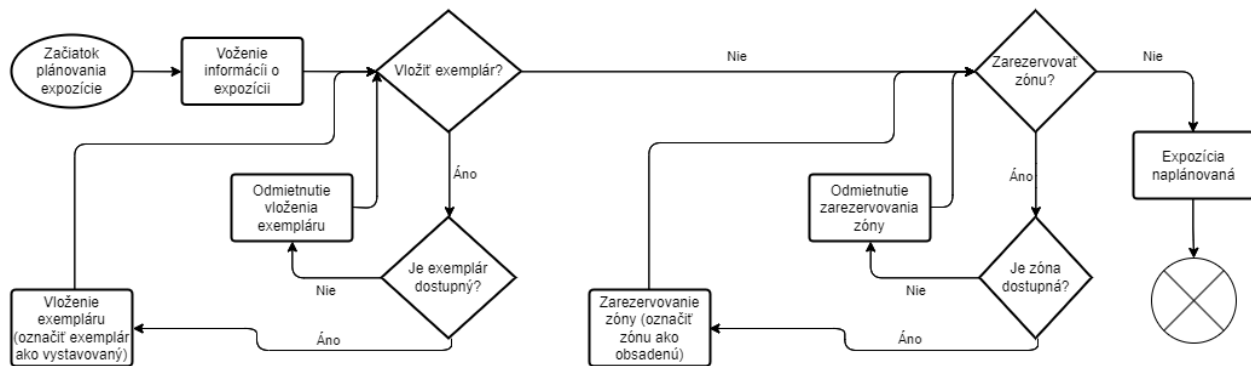
Každé zapožičanie je spojené s 1 inštitúciou.  
Inštitúcia sa zúčastňuje na 0 až N zapožičaniach.

### Loan $\longleftrightarrow$ AfterLoanInspection

Každé zapožičanie má po skončení práve 1 kontrolu.  
Kontrola súvisí len s práve 1 zapožičaním.

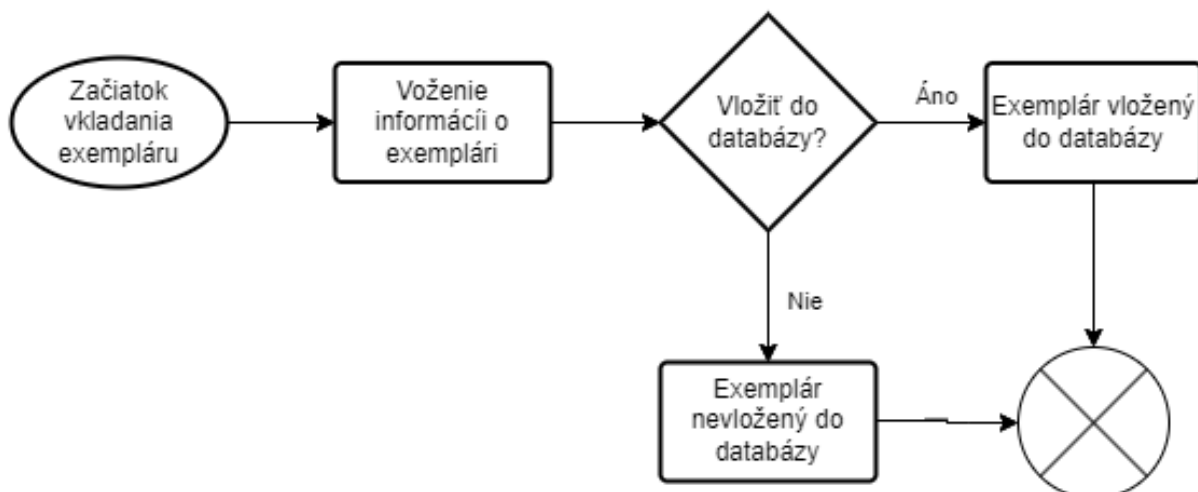
## 3 Opis základných procesov

### 3.1 Plánovanie expozície



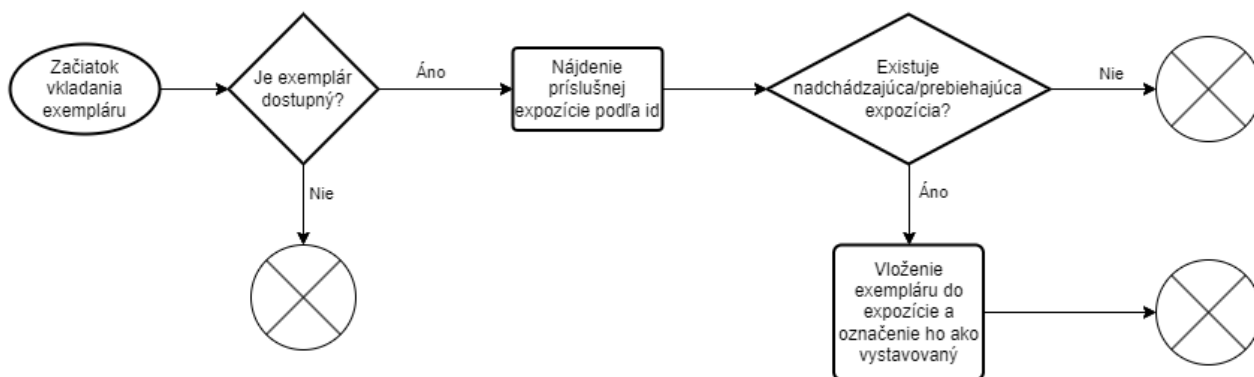
Expozícia sa bude plánovať tak, že sa vytvorí záznam v tabuľke expositions. Následne sa pomocou spojovacej tabuľky budú pridávať do expozície exempláre, ale iba tie, ktoré majú `current_status = 'IN_STORAGE'`. Po pridaní sa exempláru nastaví `current_status` na `'ON_DISPLAY'`. Následne sa bude môcť taktiež pomocou spojovacej tabuľky obsadiť zóna. Ak je dostupná tak sa v spojovacej tabuľke priradí k expozícii a nastaví sa jej hodnota `is_occupied = TRUE`.

### 3.2 Vkladanie exempláru (do databázy)



Klasické vkladanie nového údaje do databázy.

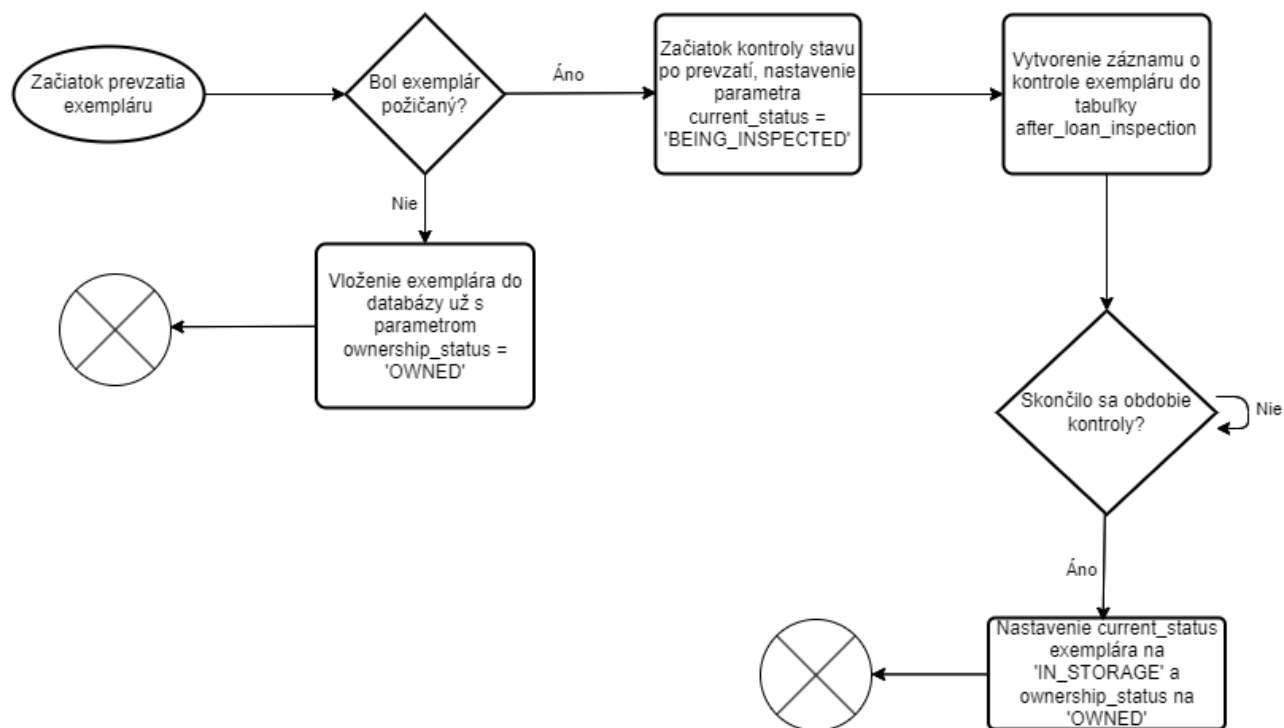
### 3.3 Vkladanie exempláru (do expozície)



Najprv sa zistí, či je exemplár dostupný podľa toho či má parameter `current_status` nastavený na 'IN\_STORAGE'. Ak je dostupný, nájde sa príslušná expozícia (ak existuje) pomocou id a spojí sa s exemplárom v spojovacej tabuľke. Exempláru sa nastaví `current_status` na 'ON\_DISPLAY'.

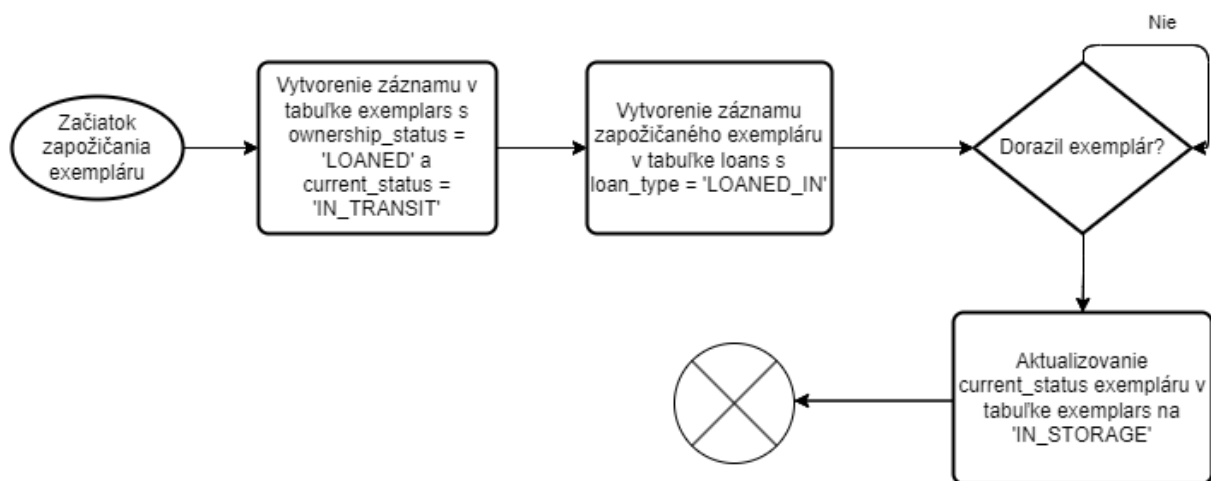


### 3.4 Prevzatie exempláru z inej inštitúcie



Najprv sa zistí či sa údaj o exemplári nachádza v tabuľke loans a teda či bol inštitúcii zapožičaný alebo nie. Ak nie tak sa jednoducho vytvorí záznam o exemplári v databáze s parametrom ownership\_status = 'OWNED'. Ak áno nastaví sa exempláru current\_status na 'BEING\_INSPECTED' až dovtedy dokiaľ sa nedokončí kontrola. Zároveň sa o kontrole vytvorí záznam v tabuľke after\_loan\_inspection. Ak sa kontrola skončí, nastaví sa exempláru current\_status na 'IN\_STORAGE' a ownership\_status na 'OWNED'.

### 3.5 Zapožičanie exempláru z inej inštitúcie



Najprv sa vytvorí nový záznam v tabuľke exemplars. Ownership\_status sa nastaví na 'LOANED' a current\_status na 'IN\_TRANSIT'. Taktiež sa o zapožičaní vytvorí záznam v tabuľke loans typu 'LOANED\_IN'. Potom čo dorazí exemplár sa mu nastaví current\_state na 'IN\_STORAGE'.

## 4 Predbežný návrh query pre vytvorenie databázy

```
1 CREATE TYPE "ownership_status_enum" AS ENUM (  
2     'OWNED',  
3     'LOANED'  
4 );  
5  
6 CREATE TYPE "current_status_enum" AS ENUM (  
7     'IN_STORAGE',  
8     'ON_DISPLAY',  
9     'BEING_INSPECTED',  
10    'IN_TRANSIT',  
11    'NONE'  
12 );  
13  
14 CREATE TYPE "exposition_status_enum" AS ENUM (  
15     'PLANNED',  
16     'IN_PROGRESS',  
17     'ENDED'  
18 );  
19  
20 CREATE TYPE "loan_type_enum" AS ENUM (  
21     'LOANED_TO',  
22     'LOANED_IN'  
23 );  
24  
25 CREATE TABLE "categories" (  
26     "id" SERIAL PRIMARY KEY,  
27     "name" TEXT UNIQUE  
28 );  
29  
30 CREATE TABLE "exemplars" (  
31     "id" SERIAL PRIMARY KEY,  
32     "category_id" INTEGER REFERENCES "categories" ("id"),  
33     "name" TEXT UNIQUE,  
34     "description" TEXT,  
35     "ownership_status" ownership_status_enum DEFAULT 'OWNED',  
36     "current_status" current_status_enum DEFAULT 'IN_STORAGE',  
37     CONSTRAINT compatible_ownership_status CHECK (  
38         (ownership_status = 'OWNED' AND current_status IN ('IN_STORAGE', 'ON_DISPLAY', 'BEING_INSPECTED'))  
39         OR (ownership_status = 'LOANED' AND current_status IN ('IN_STORAGE', 'ON_DISPLAY', 'BEING_INSPECTED', 'IN_TRANSIT'))  
40     )  
41 );  
42  
43 CREATE TABLE "expositions" (  
44     "id" SERIAL PRIMARY KEY,  
45     "name" TEXT UNIQUE,  
46     "start_date" TIMESTAMP,  
47     "end_date" TIMESTAMP,  
48     "status" exposition_status_enum DEFAULT 'PLANNED'  
49 );  
50  
51 CREATE TABLE "zones" (  
52     "id" SERIAL PRIMARY KEY,  
53     "name" TEXT,  
54     "is_occupied" BOOLEAN DEFAULT FALSE  
55 );  
56  
57
```

```

58 CREATE TABLE "exemplars_zones" (
59     "exemplar_id" INTEGER REFERENCES "exemplars" ("id") ON DELETE CASCADE ON UPDATE CASCADE,
60     "zone_id" INTEGER REFERENCES "zones" ("id") ON DELETE CASCADE ON UPDATE CASCADE
61 );
62
63 CREATE TABLE "institutions" (
64     "id" SERIAL PRIMARY KEY,
65     "name" TEXT UNIQUE
66 );
67
68 CREATE TABLE "loans" (
69     "id" SERIAL PRIMARY KEY,
70     "exemplar_id" INTEGER REFERENCES "exemplars" ("id"),
71     "type" loan_type_enum DEFAULT 'LOANED_TO',
72     "involved_institution_id" INTEGER REFERENCES "institutions" ("id"),
73     "loan_start_date" TIMESTAMP,
74     "loan_end_date" TIMESTAMP,
75     "expected_exemplar_availability" TIMESTAMP
76 );
77
78
79 CREATE TABLE "after_loan_inspection" (
80     "id" SERIAL PRIMARY KEY,
81     "loan_id" INTEGER REFERENCES "loans" ("id") ON DELETE CASCADE ON UPDATE CASCADE,
82     "inspection_date" TIMESTAMP,
83     "inspection_end_date" TIMESTAMP,
84     "inspection_description" TEXT
85 );
86
87 CREATE TABLE "exemplar_expositions" (
88     "exemplar_id" INTEGER REFERENCES "exemplars" ("id") ON DELETE CASCADE ON UPDATE CASCADE,
89     "exposition_id" INTEGER REFERENCES "expositions" ("id") ON DELETE CASCADE ON UPDATE
90     CASCADE,
91     PRIMARY KEY ("exemplar_id", "exposition_id")
92 );
93
94 CREATE TABLE "exposition_zones" (
95     "exposition_id" INTEGER REFERENCES "expositions" ("id") ON DELETE CASCADE ON UPDATE
96     CASCADE,
97     "zone_id" INTEGER REFERENCES "zones" ("id") ON DELETE CASCADE ON UPDATE CASCADE,
98     PRIMARY KEY ("exposition_id", "zone_id"),
99     CONSTRAINT one_exp_per_zone UNIQUE (zone_id)
100 );

```