

DATABÁZOVÉ SYSTÉMY

## Zadanie č. 33 - **Lékárna**

*Projektová dokumentácia*

Natália Bubáková

xbubak01

Klimecká Alena

xklime47

2. mája 2022

# Obsah

<b>1</b>	<b>Datový model (ERD) a model prípadov užitia</b>	<b>3</b>
1.1	Zadanie . . . . .	3
1.2	Model prípadov užitia a jeho popis . . . . .	3
1.3	Datový model a jeho popis . . . . .	4
<b>2</b>	<b>Implementácia databázy</b>	<b>5</b>
2.1	Základná implementácia tabuliek . . . . .	5
2.2	Triggery . . . . .	5
2.3	Procedúry . . . . .	5
2.4	Explain plan a index . . . . .	5
2.5	Materializovaný pohľad . . . . .	6
2.6	Prístupové práva . . . . .	7

# 1 Datový model (ERD) a model prípadov užitia

Pre realizáciu projektu z predmetu IDS sme si zvolili tému „Lékárna“, a teda vypracovali aktuálne zadanie číslo 33. Pri navrhovaní riešenia sme sa tiež inšpirovali vypracovaním rovnomenného zadania z už absolvovaného predmetu IUS.

Zmýšľanie pri modelovaní IS pre lekárňu a i samotné implementácie jeho modelov s ich popismi sú zachytené v časti 1.2 pre Use Case diagram a 1.3 pre Entity Relationship diagram.

## 1.1 Zadanie

Projekt č.: 33

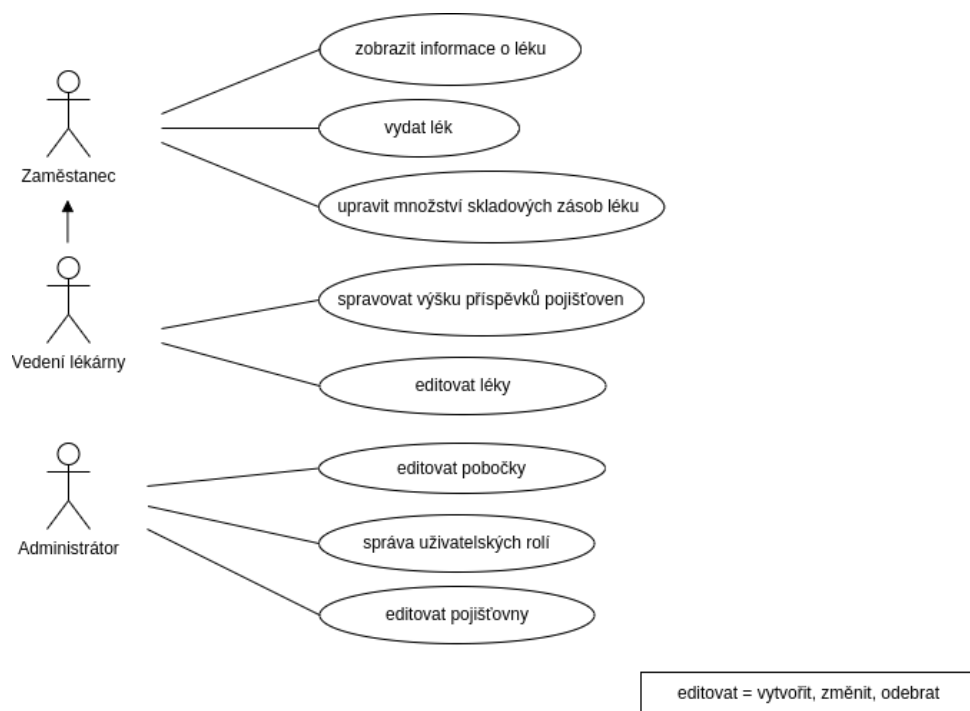
Název projektu: Lékárna

Zadání:

Vášim úkolem je vývoj IS lékárny. Lékárna vydává občanům léky jak na předpis, tak za hotové. U léků vydávaných na předpis může část ceny hradit zdravotní pojišťovna. Některé léky se vydávají pouze na předpis. Systém musí umožnit evidenci vydávaných léků, import příspěvků na léky od zdravotních pojišťoven (může se čas od času měnit), musí poskytovat export výkazů pro zdravotní pojišťovny a musí mít vazbu na skladové zásoby (vidět, zda požadovaný lék je na skladě). Léky jsou identifikovány číselným kódem či názvem.

## 1.2 Model prípadov užitia a jeho popis

Pri modelovaní IS lekárně sa dá dopracovať k jednoznačnému konceptu uzatvoreného systému slúžiacemu len pre evidenciu a správu liekov v rámci jednotlivých pobočiek lekárně. Systém sa síce nezameriava na konkrétnych zamestnancov ako na (jednoznačne identifikované) pobočky, ale stále treba rozoznať práva pre prehliadanie dát v rámci systému.



Obr. 1: Model prípadov užitia pre IS lekárně

Takto v Use Case diagrame 1 môžeme vidieť, že do IS majú prístup radoví zamestnanci pobočky lekárně s povolením základných úkonov v rámci ich náplne práce na konkrétnej pobočke. Ďalej do systému prístupuje i vedúci pobočky, ktorý dedí všetky úkony bežného zamestnanca a zároveň má práva navyše pri dôležitejších úkonoch, ktoré sa môžu sťahovať nie len na konkrétnu pobočku, ale aj celý systém. To sú napríklad úkony ako pridanie, odobranie a upravovanie jednotlivých liekov alebo upravovanie výšky príplatku nahlásaného z poisťovne. Tie sú teda dôležité a pomerne často ich treba spravovať. Naopak poslednému aktérovi tohto diagramu patria síce tiež veľmi významné, ale zriedka potrebné úkony. Administrátor teda nie je vo funkcii priameho zamestnanca pobočky lekárně, ale spravuje užívateľské role pre celý systém a spracúva zriedkavé zmeny v evidencii pobočiek a poisťovní.

### 1.3 Datový model a jeho popis

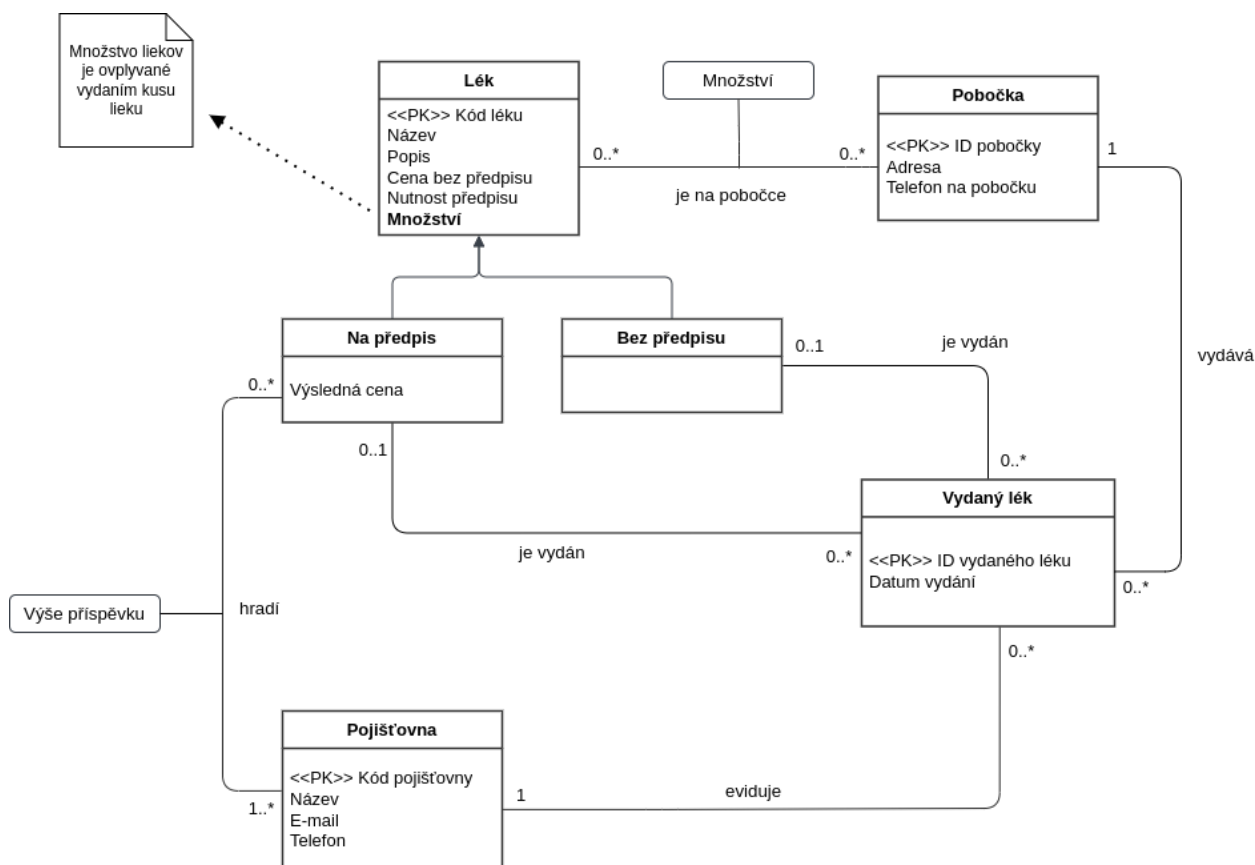
IS lekárne je poňatý ako jednotný informačný systém pre reťazec lekární, a teda umožňuje prístup jednotlivým pobočkám, čo možno sledovať i v ER diagrame 7. Entita pobočky lekárne tak eviduje len základne informácie o nej, ako je telefónne číslo a adresa. Zároveň tiež eviduje množstvo jednotlivých liekov, ktoré sa na pobočke nachádzajú. Takto zamestnanec pobočky prístupuje k atributu entity lieku - množstvo kusov lieku, a pri doložení skladových zásob tento údaj upraví.

Entita lieku je jednoznačne identifikovaná kódom daného druhu lieku, nesie tiež jeho názov, popis a základnú trhovú cenu. Ďalej je však špecializovaná na entitu lieku bez predpisu a lieku na predpis. Liek na predpis tým môže byť previazaný s poisťovňou, a tak výsledná cena tohto lieku je ovplyvnená výškou príspevku od poisťovne. U každého lieku sa vedie tiež či liek môže byť vydaný na predpis aj bez predpisu alebo práve len na predpis, teda či umožňuje väzbu k entite lieku bez predpisu. Tento jav by v praxi znamenal nemožnosť uskutočniť výdaj lieku bez predpisu (napr. zašednuté tlačítko).

Činnosťou zamestnanca lekárne pri výdaji kusu lieku je liek z oboch entít, či už na predpis alebo bez predpisu, ďalej evidovaný ako vydaný liek. Tým možno viesť evidenciu všetkých vydaných liekov, a to vydaných na predpis, bez predpisu a tiež podľa konkrétnych pobočiek, z čoho sa exportuje výkaz vydaných liekov na predpis pre poisťovňu na ekonomické účely.

Vydanie lieku tiež priamo vplýva na množstvo kusov lieku, teda po uskutočnení akcie vydania lieku sa každý kus odráta od množstva kusov liekov na pobočke.

V systéme sú jednotlivé zdravotné poisťovne, ktoré pripievajú na lieky na predpis, vedené pod entitou poisťovne cez jednoznačný identifikátor - kód poisťovne. Táto entita obsahuje len názov poisťovne a základné kontaktné údaje ako sú e-mail a telefónne číslo. Slúži len pre informačné účely pre zamestnancov lekárne.



Obr. 2: Dátový model pre IS lekárne

## 2 Implementácia databázy

### 2.1 Základná implementácia tabuliek

Na základe ER diagramu sme priamo zostavili tabuľky cez `CREATE TABLE` a vyznačili presnejší formát jednotlivých atribútov ako definovanie datového typu, obmedzenie `NULL` hodnoty, či špecifikovanie formátu `REGEX`om s `check(regex_like(...))`. Z tvorby tabuliek bola významná najmä voľba implementácie špecializácie, pri ktorej sme sa rozhodli pre prvú variantu z prednášok. Tou je práve oddelenie špecializácií do samostatných tabuľkových entít s PK zrodným s ich rodičovskou entitou a s reláciami špecifickými pre každú z nich, a to práve zdôvodu, že sa jedná o vzťahy typu lieku na predpis a bez predpisu, ktoré sú čiastočné a zároveň sa prekrývajú. Tie sme ďalej previazali v entite `vydany_liek` dvomi oddelenými cudzími kľúčmi, obmedzenými, aby výlučne jeden bol nosný a druhý obašoval `NULL`.

Tabuľky sme napokon naplnili ukázkovými dátami s `INSERT` a skontrolovali funkčnosť previazania rôznymi typmi implementácie klauzule `SELECT` práve podľa zadania v tretej časti projektu. Takto vypracovanú databázu sme mohli o to ľahšie využiť pri štvrtej časti, kde sme implementovali dva databázové triggery a dve uložené procedúry, využili `EXPLAIN PLAN` s vytvorením indexu a tiež zostavili materializovaný pohľad, na ktorý sme spolu so všetkými tabuľkami a procedúrami definovali prístupové práva pre druhého člena.

### 2.2 Triggery

Pro zadanie lekárne je vhodné implementovať trigger, ktorý umožní upozorniť na nedostatok liekov v momente, keď je na sklade posledný kus.

### 2.3 Procedúry

Podľa zadania sme vytvorili procedúry, ktoré obsahujú kurzor, ošetrovanie výnimky a premennú s odkazom na typ.

Prvá procedúra `liek_na_sklade` je zostavená tak, aby pomocou kurzoru prebehla množstvo liekov uložené na jednotlivých pobočkách a vypísalo koľko kusov daného lieku sa nachádza na danej pobočke spomedzi všetkých, kde požadovaný EAN lieku a ID pobočky je treba vložiť do argumentu. V prípade, že liek (už) nie je k dispozícii na žiadnej z pobočiek, vypíše upozornenie.

Druhá procedúra `export_vykazov_pre_poistovnu` sa zas zameriava na prehľadný výpis výkazov vydaných liekov a poplatkov na ne pre poisťovňu požadovanú argumentom. Vypíše tak zoznam jednotlivých kusov liekov vydaných v jednotlivé dni pre každý z liekov a za týmito skupinkami výpisov nasleduje vždy zhodnotenie a celková úhrada za tento liek.

```
XBURAK01> CALL liek_na_sklade(1, '8595116523847')
[2022-05-02 23:28:12] completed in 30 ms
Na pobočke (1) sa nachádza 63 z 118 ks
XBURAK01> CALL liek_na_sklade(2, '8595116523847')
[2022-05-02 23:28:12] completed in 9 ms
Na pobočke (2) sa nachádza 55 z 118 ks
XBURAK01> CALL liek_na_sklade(2, '111111111111')
[2022-05-02 23:29:04] completed in 12 ms
Liek 111111111111 nie je na sklade
```

Obr. 3: Výpis procedúry `lieknasklade`

### 2.4 Explain plan a index

`EXPLAIN PLAN` jsme použili nad príkazom, ktorý má za úlohu vypísať, koľko kusov jednotlivých liekov celkovo firma vlastní, teda na všetkých jej pobočkách. Po vyvolaní optimalizátoru a vypísaní na výstup vidíme, že metóda spojenia je `HASH JOIN` (u oboch tabuliek sa počítajú hash kľúče pre každý riadok) a výkonnostné ceny by bolo možné znížiť. Preto tu použijeme vytvorenie indexov pro tabuľky `LIEK` a `MNOZSTVO`. Následne znovu vypíšeme a môžeme vidieť, že sa operácie previedli rýchlejšie a s menšou výkonnostnou cenou.

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		8	1264	7 (15)	00:00:01
1	HASH GROUP BY		8	1264	7 (15)	00:00:01
* 2	HASH JOIN		8	1264	6 (0)	00:00:01
3	TABLE ACCESS FULL	LIEK	4	548	3 (0)	00:00:01
4	TABLE ACCESS FULL	MNOZSTVO	8	168	3 (0)	00:00:01

Obr. 4: Explain plan bez indexu

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		8	1264	2 (0)	00:00:01
* 1	HASH JOIN		8	1264	2 (0)	00:00:01
2	INDEX FULL SCAN	LIEK_INDEX	4	548	1 (0)	00:00:01
3	VIEW	VW_GBC_5	8	168	1 (0)	00:00:01
4	HASH GROUP BY		8	168	1 (0)	00:00:01
5	INDEX FULL SCAN	MNOZSTVO_INDEX	8	168	1 (0)	00:00:01

Obr. 5: Explain plan s indexom

## 2.5 Materializovaný pohľad

MATERIALIZED VIEW slúži na uloženie často využívaného pohľadu do formy tabuľky, s rýchlym prístupom a priebežným refreshom na aktuálne dáta. V našej implementácii sme zvolili materializovaný pohľad pre zoznam vydaných liekov pre každú z pobočiek. Toto umožní updateovať priebežný prehľad o tom koľko ktorých liekov sa na jednotlivých pobočkách vydali. Pre kontrolu je spustený SELECT na tento pohľad, vložené dodatočné informácie do tabuliek a vykonaný COMMIT. Následne pre prehľad upravený SELECT vypíše pekne zoradenú tabuľku bez zbytočných informácií, ako je napr. ROWID, ktorý sa využíva pre FAST REFRESH.

ID_POBOCKY	ADRESA	NAZOV	EAN_LIEKU	ID_VYDANEHO_LIEKU	DATUM_VYDANIA	ROWID_POBOCKY	ROWID_VYD_LIEKU	ROWID_LIEKU
1	1 Bašty 413/2, Brno, 62100	Ibaldin 400	3664798833953		1 2021-06-06	AAEQL+AAAAAAAAACAA	AAEQHdAAMAAVPUAA	AAEQL+AMAAAFREAA
2	1 Bašty 413/2, Brno, 62100	Elocom	8584055999424		2 2021-06-09	AAEQL+AAAAAAAAACAA	AAEQHdAAMAAVPUAAB	AAEQL+AMAAAFREAA
3	2 Nádražní 595, Brno, 60200	Elocom	8584055999424		3 2021-06-09	AAEQL+AAAAAAAAACAB	AAEQHdAAMAAVPUAAC	AAEQL+AMAAAFREAA
4	1 Bašty 413/2, Brno, 62100	EXCIPIAL U LIPOLOTIO	7612076354814		4 2021-06-13	AAEQL+AAAAAAAAACAA	AAEQHdAAMAAVPUAAD	AAEQL+AMAAAFREAA
5	1 Bašty 413/2, Brno, 62100	EXCIPIAL U LIPOLOTIO	7612076354814		5 2021-06-13	AAEQL+AAAAAAAAACAA	AAEQHdAAMAAVPUAAE	AAEQL+AMAAAFREAA
6	1 Bašty 413/2, Brno, 62100	Ibaldin 400	3664798833953		21 2021-06-08	AAEQL+AAAAAAAAACAA	AAEQHdAAMAAVPUAAF	AAEQL+AMAAAFREAA
7	1 Bašty 413/2, Brno, 62100	Paralen 500	8595116523847		22 2021-06-08	AAEQL+AAAAAAAAACAA	AAEQHdAAMAAVPUAAG	AAEQL+AMAAAFREAA
8	2 Nádražní 595, Brno, 60200	EXCIPIAL U LIPOLOTIO	7612076354814		23 2021-06-12	AAEQL+AAAAAAAAACAB	AAEQHdAAMAAVPUAAH	AAEQL+AMAAAFREAA
9	1 Bašty 413/2, Brno, 62100	EXCIPIAL U LIPOLOTIO	7612076354814		24 2021-06-09	AAEQL+AAAAAAAAACAA	AAEQHdAAMAAVPUAAI	AAEQL+AMAAAFREAA
10	2 Nádražní 595, Brno, 60200	Paralen 500	8595116523847		25 2021-06-12	AAEQL+AAAAAAAAACAB	AAEQHdAAMAAVPUAAJ	AAEQL+AMAAAFREAA
11	2 Nádražní 595, Brno, 60200	Paralen 500	8595116523847		26 2021-06-13	AAEQL+AAAAAAAAACAB	AAEQHdAAMAAVPUAAK	AAEQL+AMAAAFREAA

Obr. 6: Materializovaný pohľad

	ADRESA	NAZOV	EAN_LIEKU	ID_VYDANEHO_LIEKU	DATUM_VYDANIA
1	Bašty 413/2, Brno, 62100	Ibalgin 400	3664798033953	1	2021-06-06
2	Bašty 413/2, Brno, 62100	Ibalgin 400	3664798033953	21	2021-06-08
3	Bašty 413/2, Brno, 62100	Paralen 500	8595116523847	22	2021-06-08
4	Bašty 413/2, Brno, 62100	EXCIPIAL U LIPOLOTIO	7612076354814	24	2021-06-09
5	Bašty 413/2, Brno, 62100	Elocom	8584055999424	2	2021-06-09
6	Bašty 413/2, Brno, 62100	EXCIPIAL U LIPOLOTIO	7612076354814	4	2021-06-13
7	Bašty 413/2, Brno, 62100	EXCIPIAL U LIPOLOTIO	7612076354814	5	2021-06-13
8	Nádražní 595, Brno, 60200	Elocom	8584055999424	3	2021-06-09
9	Nádražní 595, Brno, 60200	EXCIPIAL U LIPOLOTIO	7612076354814	23	2021-06-12
10	Nádražní 595, Brno, 60200	Paralen 500	8595116523847	25	2021-06-12
11	Nádražní 595, Brno, 60200	Paralen 500	8595116523847	26	2021-06-13

Obr. 7: Upravený výpis materializovaného pohľadu

## 2.6 Prístupové práva

Pomocou príkazu `GRANT ALL ON <nazov_tabulky> TO <login_clena>` sme udelili práva druhému členovi tímu, a to konkrétne všetky práva (na vkladanie, mazanie, ai.) pre každú z tabuliek. Rovnako ako na tabuľku sme definovali práva aj pre materializovaný pohľad, zatiaľčo v prípade umožnenia využívania procedúry sme museli uviesť `GRANT EXECUTE ON <nazov_procedury> TO <login_clena>`.