VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Databázové systémy (IDS) 2022

Projektová dokumentace
Projekt č.: 28

30. dubna 2022

Denis Karev
Vladislav Mikheda

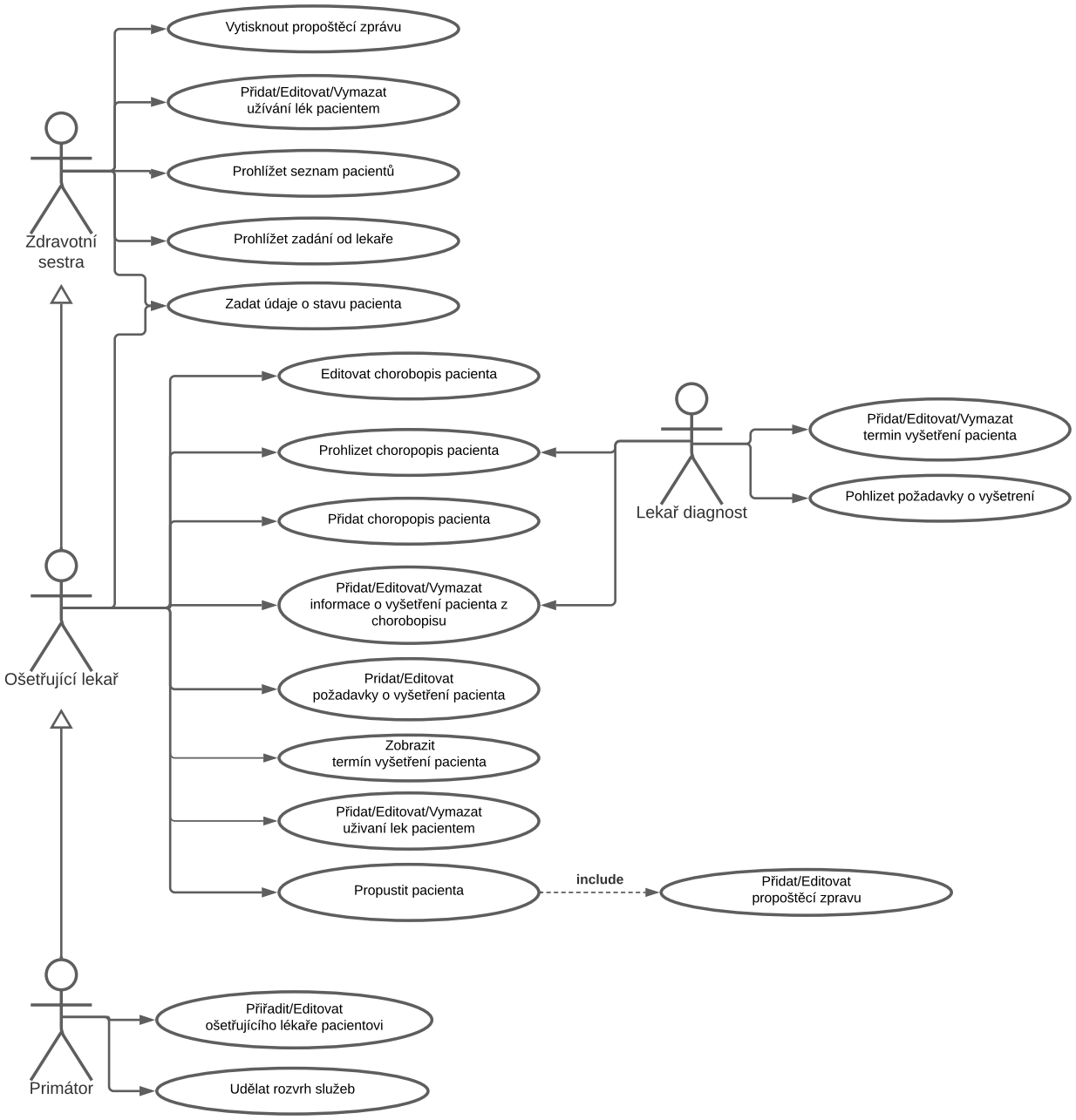
Obsah

1 Popis zadání

Nemocnice[1]:
Navrhněte Datové model a model případu užití malé nemocnice, který by poskytoval základní údaje o lékařích, sestrách či pacientech, kteří jsou a byli hospitalizováni do nemocnice. IS uchovává informace o všech těchto hospitalizacích, přičemž pacient může ve stejný okamžik hospitalizován pouze na jednom oddělení nemocnice. Při každé hospitalizaci je mu určen jeho ošetřující lékař. Lékaři mohou pracovat na více odděleních zároveň. Na každém oddělení má lékař určitý úvazek, telefon atd., zatímco sestry pracují pouze na jednom oddělení. V rámci pobytu v nemocnici může pacient podstoupit různá vyšetření, která byla provedena na určitém oddělení ve stanoveném čase a provedl ji určitý lékař, který také zapisuje výsledky vyšetření do IS. Dále mu mohou být podávány různé léky, každé podávání léku má určité detaily (kdy se podává, kolikrát apod.). Zaměřte se i na otázku ochrany dat tak, aby měl každý lékař přístup pouze k potřebným údajům.

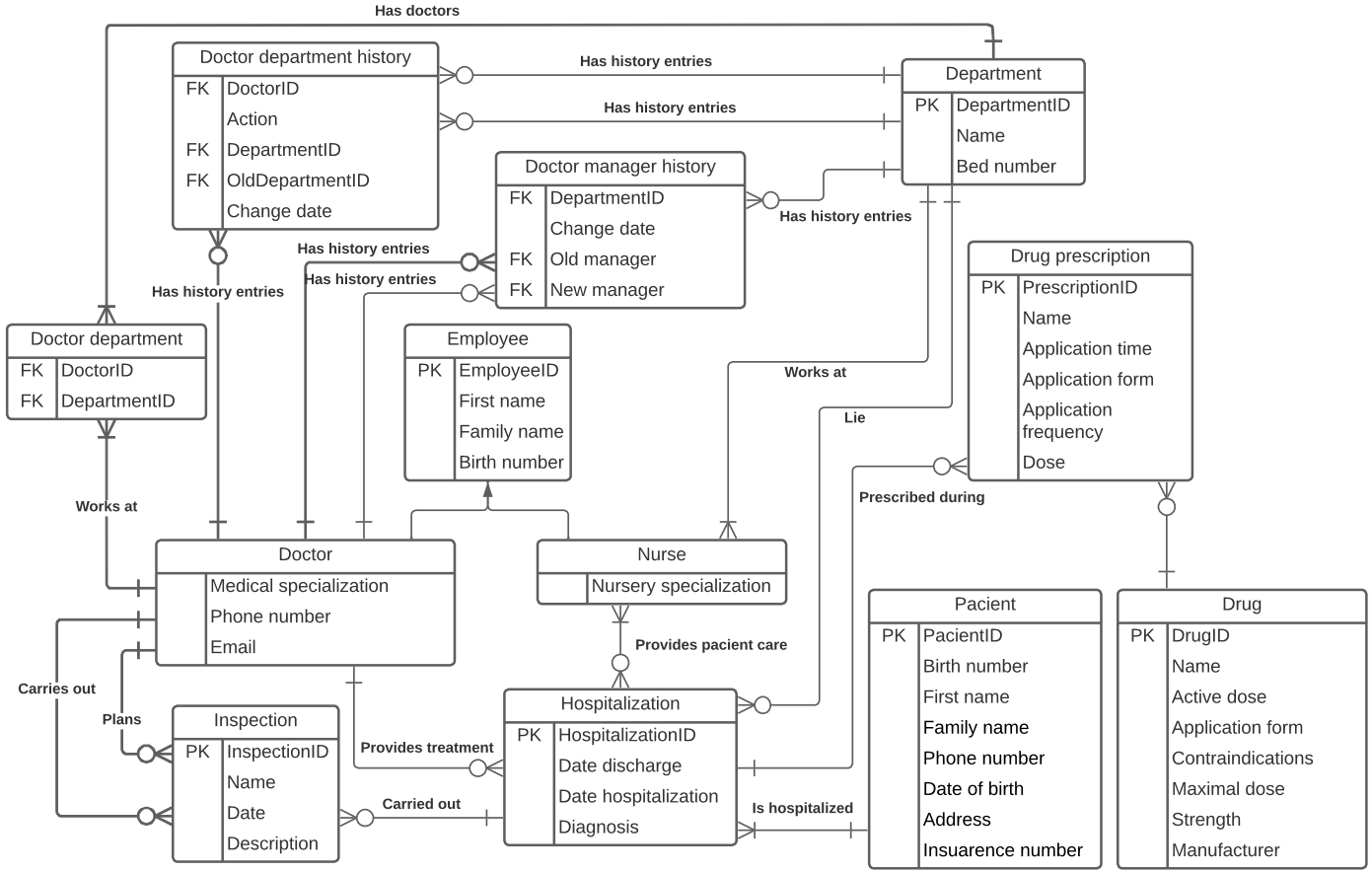
[1] Zadaní je inspirováno projektem IUS

2 Model případů užití (Use Case Diagram)



Obrázek 1: Model případů užití

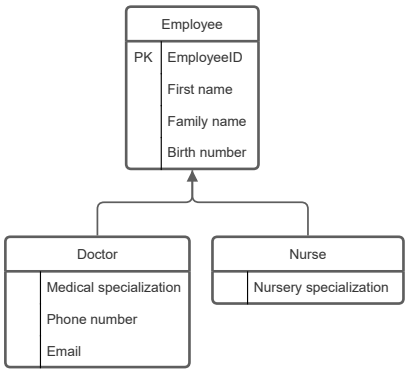
3 Datový model (ERD)



Obrázek 2: Datové model

4 Realizace generalizace/specializace

V ERD od entity `Employee(Personál)` vztahem generalizace/specializace jsou odvedeny dvě entity: `Doctor(Lékař)`, `Nurs(Zdravotní sestá)`.



Při převodě do tabulky v databázi to bylo vyřešeno vytvořením tabulky pro nadtyp a také pro podtypy s primárním klíčem nadtypu. Taková možnost byla zvolena protože potřebujeme dvě různé tabulky pro `lékaře` a `zdravotní sestru`. Tabulka `personál` je využita k ukládání společné informací. Při rozšíření databáze například přidáním tabulky `maséru` není potřeba nic měnit jen přidat novou tabulku.

Employee

ID	first_name	famely_name	birth_number

Doctor

ID	ph_number	email	med_spec

Nurse

ID	specialization

5 Triggery

Byly vytvořeny dva triggery: `DEPARTMENT_MANAGER_HISTORY_T` a `DOCTOR_DEPARTMENT_HISTORY`. Tyto triggery jsou určeny pro logging některých změn v tabulkách.

`DEPARTMENT_MANAGER_HISTORY_T` se vyvolá po změně `MANAGER_ID` v tabulce `DEPARTMENTS`. Tento trigger uloží zkratku oddělení, čas změny, ID starého a nového manažera oddělení do tabulky `DEPARTMENT_MANAGER_HISTORY`.

Druhý trigger se aktivuje po přidaní nových dát, změně zkratky oddělení nebo mazaní dát z tabulky `DOCTORS_DEPARTMENTS`. Tento trigger využije několik proměnných, protože obsah `:NEW` a `:OLD` se mění v závislosti na typu operace. Na konci `DOCTOR_DEPARTMENT_HISTORY` vkládá nový záznam do tabulky `DOCTOR_DEPARTMENT_HISTORY`.

6 Procedury

Byly vytvořeny dvě procedury: `CREATE_EMPLOYEE` a `ASSIGN_DOCTORS`.
Procedura `CREATE_EMPLOYEE` má následující IN parametry:

1. `IN_IS_DOCTOR` - `BOOLEAN`, je-li nový zaměstnanec doktorem.
2. `IN_BIRTH_NUMBER` - rodné číslo.
3. `IN_FIRST_NAME` - jmeno.

4. IN_FAMILY_NAME - příjmení.
5. IN_SPECIALIZATION - VARCHAR, specializace.
6. IN_DEPARTMENT - zkrátka oddělení.
7. IN_PHONE_NUMBER - telefonní číslo, je volitelným parametrem.
8. IN_EMAIL - email, je volitelným parametrem.

Tato procedura na začátku vytvoří nový záznam v tabulce EMPLOYEES. Potom v závislosti na typu zaměstnance vytvoří nové záznamy v příslušných tabulkách.

Procedura ASSIGN_DOCTORS nemá žádné parametry. Tato procedura přiřadí každé hospitalizace bez doktoru (tj. DOCTOR_ID = NULL) lékaře s příslušného oddělení. Procedura používá kurzor, který odkazuje na hospitalizace bez doktorů. Pro každý záznam z kurzoru, ASSIGN_DOCTORS vybere náhodného lékaře (pro náhodný vyber se používá ORDER BY DBMS_RANDOM.RANDOM()) a přiřadí ho.

7 Index, Explain Plan

Indexy mohou být nastaveny za účelem zrychlení provádění konkrétního dotazu. Indexy je potřeba přidávat ne na prázdnou tabulku, ale již s nějakými daty, je lepší, když jsou zřídka aktualizovány. Pro indexaci byl zvolen dotaz množství různých léků, které pacienti potřebují. Pro zjištění jak se dotaz vykonává využijeme Explain Plan

```
-----------------------------------------------------------------------------------------------
| Id  | Operation                    | Name               | Rows  | Bytes | Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |                    |     2 |    24 |     6  (17)| 00:00:01 |
|   1 |  HASH GROUP BY               |                    |     2 |    24 |     6  (17)| 00:00:01 |
|   2 |   NESTED LOOPS               |                    |     2 |    24 |     5   (0)| 00:00:01 |
|   3 |    NESTED LOOPS              |                    |     2 |    24 |     5   (0)| 00:00:01 |
|   4 |     TABLE ACCESS FULL        | DRUG_PRESCRIPTIONS |     2 |    16 |     3   (0)| 00:00:01 |
|*  5 |     INDEX UNIQUE SCAN        | SYS_C001641914     |     1 |       |     0   (0)| 00:00:01 |
|*  6 |    TABLE ACCESS BY INDEX ROWID| HOSPITALIZATIONS   |     1 |     4 |     1   (0)| 00:00:01 |
-----------------------------------------------------------------------------------------------
```

Z plánu lze vidět že jde zpracovaní select dotazů a na začátku bude provedeno grupování položek a pak bude prováděn Procházení každé položky v tabulce drug_prescriptions, a lze vidět že byly využity indexy které samostatné přidala databáze, pomocí nich je provedeno pouze procházení stromu, dál už bude využit ukazatel na údaje v tabulce.
Pro zrychlení dotazu byly zvoleny 2 indexy, první pro sloupce date_disch, id v tabulce hospitalization, druhý pro ABBREVIATION, id_hosp v tabulce drug_prescriptions.

```
-----------------------------------------------------------------------------------------
| Id  | Operation            | Name           | Rows  | Bytes | Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT     |                |     2 |    82 |     1   (0)| 00:00:01 |
|   1 |  SORT GROUP BY NOSORT|                |     2 |    82 |     1   (0)| 00:00:01 |
|   2 |   NESTED LOOPS       |                |     2 |    82 |     1   (0)| 00:00:01 |
|   3 |    INDEX FULL SCAN   | DRUG_PRE_INDEX |     2 |    38 |     1   (0)| 00:00:01 |
|*  4 |    INDEX RANGE SCAN  | HOSP_INDEX     |     1 |    22 |     0   (0)| 00:00:01 |
-----------------------------------------------------------------------------------------
```

Lze vidět že procházení tabulek s využitím indexu zrychli dotazovaní.

8 Materializovaný pohled

Byly vytvořeny dva materializovaných pohledy, byly využity `BUILD IMMEDIATE` co naplní pohled při vytváření, pro první bylo vyžito `REFRESH COMPLETE ON DEMAND` aktualizuje se přepočítáním definujícího dotazu materializovaného pohledu, pro aktualizace je využit postup `DBMS_MVIEW.REFRESH`. V druhém materializovaném pohledu je využito `REFRESH COMPLETE ON COMMIT`. Což vyžaduje využití `COMMIT`, co ukončí transakci a uloží změny.