

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Databázové systémy (IDS) 2022  
Projektová dokumentace  
Projekt č.: 28

29. dubna 2022

Denis Karev  
Vladislav Mikheda

## Obsah

1	Popis zadání .....	2
2	Model případů užití (Use Case Diagram) .....	3
3	Datový model (ERD) .....	4
4	Realizace generalizace/specializace .....	5
5	Triggery .....	5
6	Procedury .....	5
7	Index, Explain Plan .....	5
8	Materializovaný pohled .....	6

# 1 Popis zadání

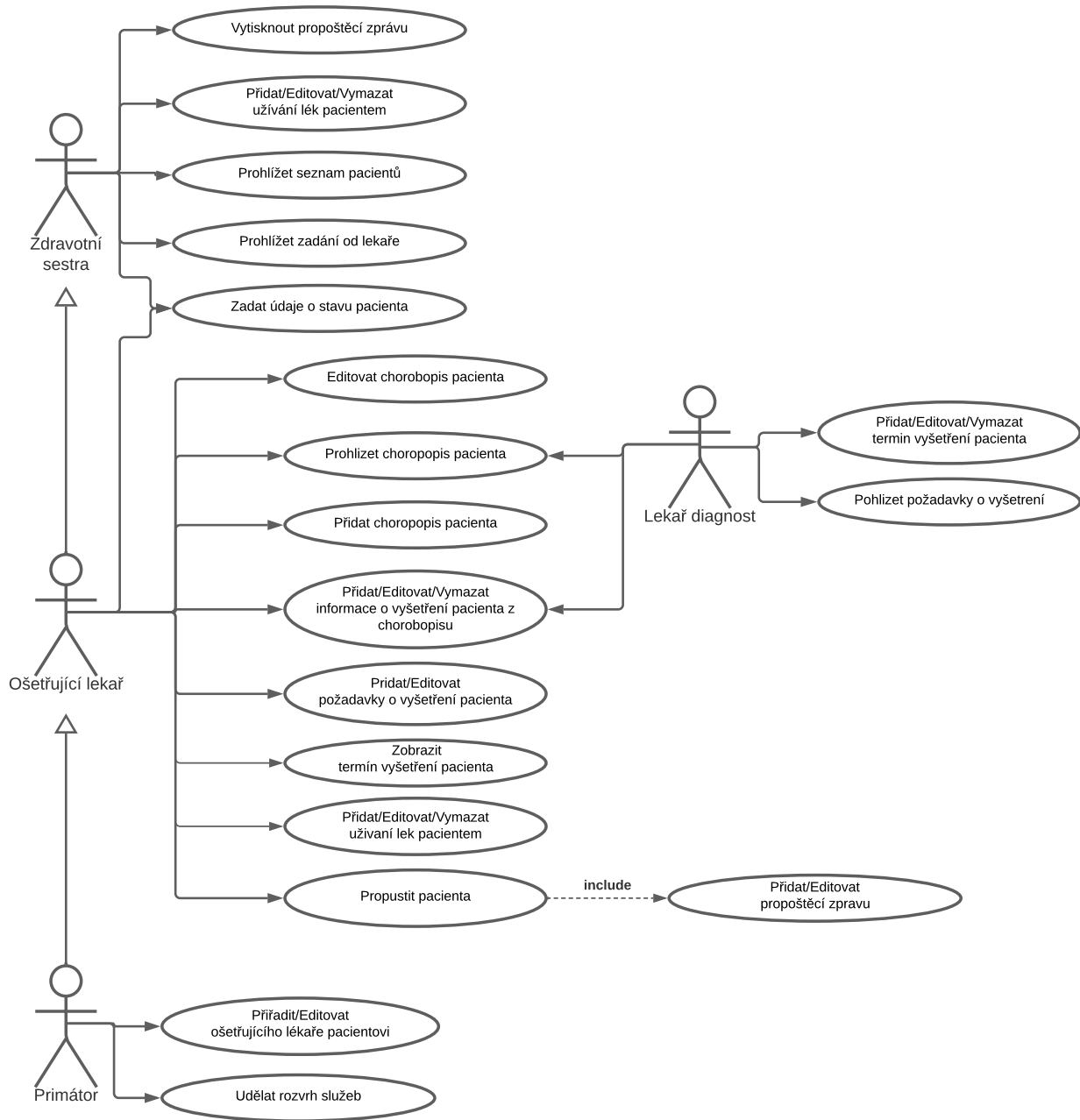
Nemocnice<sup>1</sup>:

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.

---

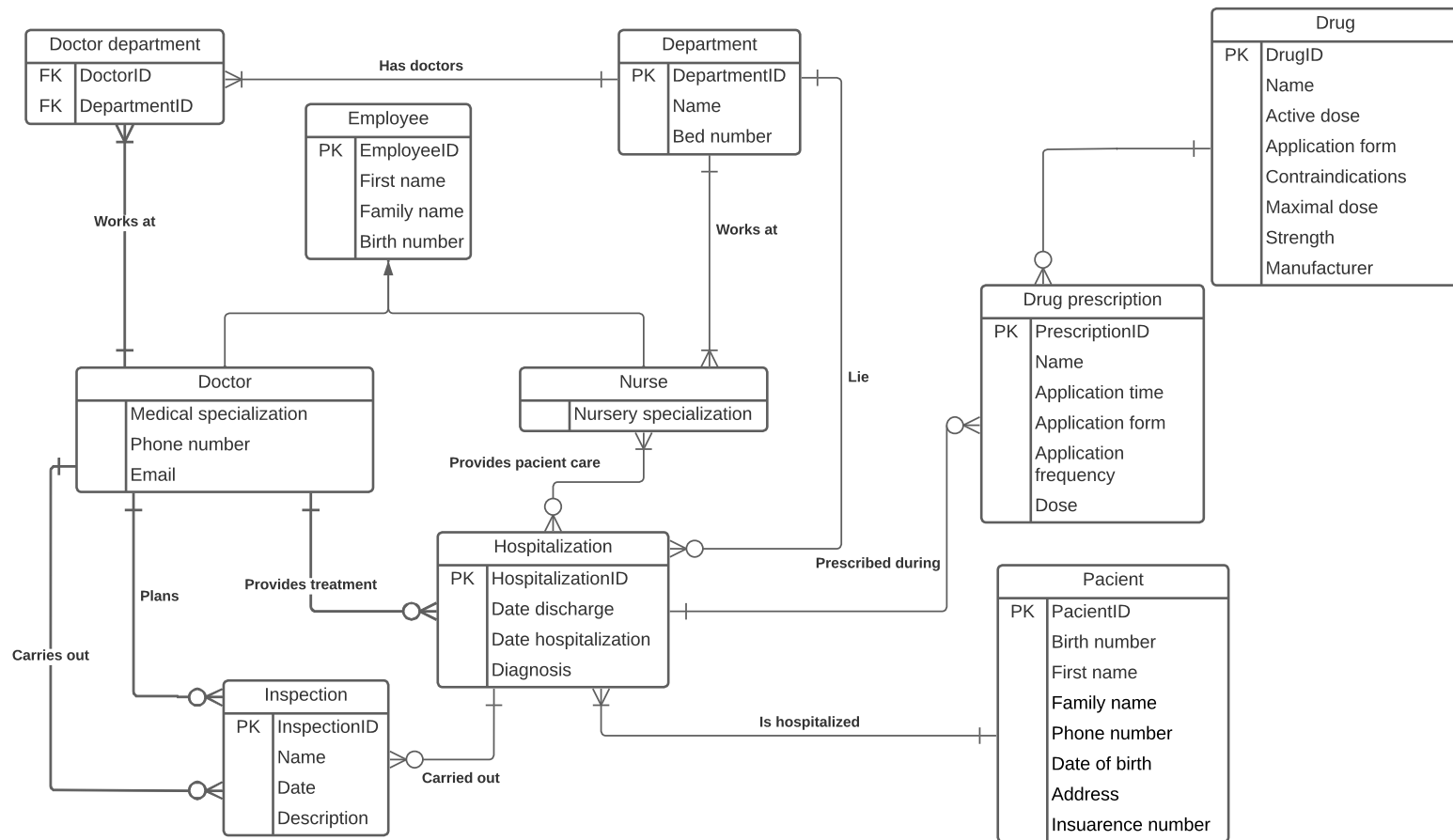
<sup>1</sup>Zadání je inspirováno projektem IUS

## 2 Model případů užití (Use Case Diagram)



Obrázek 1: Model případů užití

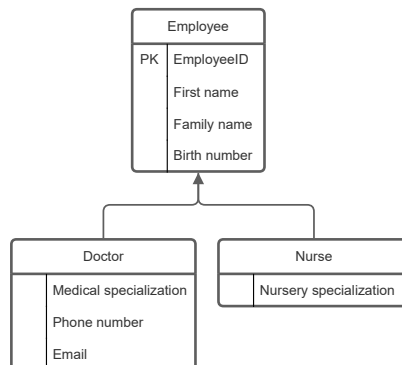
### 3 Datový model (ERD)



Obrázek 2: Datový model

## 4 Realizace generalizace/specializace

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



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 a tabulka `personál` je využita k uložení společné informací. Při rozšíření databáze například dodání maséru není potřeba nic měnit jen přidat novou tabulku.

Employee				Doctor				Nurse	
ID	first_name	famely_name	birth_number	ID	ph_number	email	med_spec	ID	specialization

## 5 Triggery

## 6 Procedury

## 7 Index, Explain Plan

Index mohou být nastaveny, za účelem zrychlení provádění konkrétního dotazu. Indexy je třeba 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 dotaz správce 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 pohledu lze vidět že jde zpracování `select` dotazu a na začátku bude provádění grupování položek a pak bude procházena každá položka v tabulce `drug_prescriptions`, a lze vidět že byli využity indexy který

samostatné přidala databáze, s pomoci nich je provedeno pouze procházení stromu, dal už bude využit ukazatel na údaje v tabulce.

Pro zrychlení dotazu byly zvoleny 2 indexu, první o sloupcích `date_disch`, `id` v tabulce `hospitalization`, druhý o `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 ze už prováděn obchod tabulek s využitím indexu co zrychli dotazování.

## 8 Materializovaný pohled

Byli vytvořeny dva materializovaných pohledu, byli využity `BUILD IMMEDIATE` co naplní pohled při vytváření, pro první bylo využito `REFRESH COMPLETE ON DEMAND` aktualizuje se přepočítáním definujícího dotazu materializovaného pohledu, pro aktualizace využita procedura `dbms_mview.refresh`. V druhem materializovaném pohledu využito `REFRESH COMPLETE ON COMMIT`. Což vyžaduje využití `COMMIT`, co ukončí transakci a uloží změny.