

ŽILINSKÁ UNIVERZITA V ŽILINE
FAKULTA RIADENIA A INFORMATIKY



ŽILINSKÁ UNIVERZITA V ŽILINE
Fakulta riadenia
a informatiky

Nemocničný informačný systém
Semestrálna práca

Vypracoval: **Maroš Gorný, Patrik Ištivanko, Samuel Vrana**

Študijná skupina: **5ZIB11**

Predmet: **Pokročilé databázové systémy**

Cvičiaci: Ing. **Miroslav Potočár**

OBSAH

1.	Úvod	4
2.	Technické parametre a zdroje dát	5
2.1.	Databáza	5
2.2.	Pôvodné dáta	5
2.3.	Zber a spracovanie dát	5
2.4.	Vývojárske prostredie	5
2.5.	Početnosť dát v tabuľkách	6
3.	Databázový model	7
4.	Opis databázy	8
4.1.	Objektový atribút tabuľky	12
4.2.	JSON atribút tabuľky	13
4.3.	Nested table atribút tabuľky	13
5.	XML Report – Lieky pacienta	14
5.1.	SQL Dotaz	14
5.2.	Grafické rozhranie	15
5.3.	Ukážka vygenerovaného XML dokumentu	15
6.	Správa súborov – Obrázok v zdravotnej karte	16
6.1.	SQL Dotaz	16
6.2.	Grafické rozhranie – Detail zákroku	16
7.	Prístup k dátam cez vzdialený server	17
7.1.	Vytvorenie pripojenia	17
7.2.	Tabuľka a trigger na vzdialenom serveri	17
8.	Výstupy informačného systému	18
8.1.	Výpis pacientov s možnosťou filtrovania, zoradovania podľa rôznych atribútov	18
8.2.	Výpis ambulancií s možnosťou filtrovania, zoradovania podľa rôznych atribútov	18
8.3.	Výpis ktorý zobrazuje informácie o lôžkových izbách (Voľné miesta, obsadené miesta, kapacitu a percentuálnu obsadenosť)	19
8.4.	Výpis oddelení podľa počtu lekárov. Výpis je usporiadaný podľa počtu lekárov zostupne.	19
8.5.	Výpis, ktorý ukazuje počet zamestnancov v jednotlivých vekových kategóriách	19
8.6.	Výpis na zobrazenie top 10 diagnóz podľa priemernej dĺžky pobytu	20
8.7.	Výpis, pre Top 10 pacientov s najvyšším počtom darovaní krvi	20
8.8.	Výpis pre typy darovanej krvi aj s počtom darovaní	21
8.9.	Výpis pacientov podľa vekovej kategórie	21
8.10.	Výpis pacientov podľa pohlavia	21
8.11.	Výpis najstarších pacientov	22
8.12.	Množstvo pacientov narodených v jednotlivých mesiacoch konkrétneho roka	22

8.13.	Výpis počtu dnešných stretnutí	22
9.	SQL dotazy zložitejších výstupov	23
9.1.	Získanie počtu voľných miest pre ambulancie	23
9.2.	Získanie oddelení podľa počtu lekárov	24
9.3.	Získanie vekových kategórií pre zamestnancov	24
9.4.	Získanie top N diagnóz podľa priemernej dĺžky pobytu v nemocnici	25
9.5.	Získanie top 10 ľudí, s najviac darovaním krvi	26
9.6.	Získanie počtu darovania krvi podľa typu krvi	27
9.7.	Získanie počet pacientov podľa vekových kategórií	27
9.8.	Získanie počtu pacientov podľa pohlavia	27
9.9.	Získanie top N najstarších pacientov pre nemocnicu	28
9.10.	Získanie množstva ľudí, podľa narodenia v mesiacoch	28
9.11.	Získanie počtu stretnutí za daný deň	29
9.12.	Získanie platných receptov pre daného pacienta	29
9.13.	Získanie historických stretnutí pre pacienta	30
10.	Analýza optimalizácie príkazu SELECT	31
10.1.	Bez explicitného indexu	32
10.2.	S explicitným indexom pre podmienku WHERE	32
10.3.	S indexami na cudzie kľúče	33
10.4.	S indexom na tabuľku <i>PATIENT</i>	33
11.	Popis uložených procedúr a funkcií	35
11.1.	Procedúry	35
11.2.	Funkcie	35

1. ÚVOD

Cieľom tejto semestrálnej práce je vytvorenie informačného systému určeného pre nemocničné prostredie. Systém je navrhnutý tak, aby efektívne podporoval správu pacientov, informatívny náhľad stavu nemocnice a operácie súvisiace s ambulanciami. Kľúčové funkcie zahŕňajú:

- **Správa Pacientov:** Umožňuje vytváranie, úpravu a odstraňovanie záznamov pacientov. Tento modul je navrhnutý s dôrazom na jednoduchosť a efektívnosť, čím zaisťuje hladký chod nemocničných procesov.
- **Prístup Lekárov:** Lekári majú prístup k detailným informáciám o pacientoch, vrátane možnosti zaznamenávania poznámok do zdravotných kariet a vystavovania lekárskeho predpisu.
- **Správa Ambulancií:** Poskytuje možnosti na pridávanie, úpravu a mazanie ambulancií, čo umožňuje nemocnici udržiavať aktuálne informácie o dostupných službách a oddeleniach.
- **Informatívny Náhľad:** V hlavnom menu je k dispozícii prehľadný výpis o aktuálnom stave nemocnice a jej pacientoch, lekároch, diagnózach a ambulanciách.

Systém bol vyvinutý s použitím moderných technológií, ako sú **React** a **NodeJS**, s databázou hostovanou na serveri **asterix.fri.uniza.sk**.

2. TECHNICKÉ PARAMETRE A ZDROJE DÁT

2.1. DATABÁZA

- **Hostiteľský Server:** asterix.fri.uniza.sk
- **Použité Technológie:** SQL pre generovanie a spracovanie dát.

2.2. PÔVODNÉ DÁTA

- **Primárne Zdroje:** [Národné centrum zdravotníckych informácií](#) (NCZI).
- **Vlastné Generované Dáta:** Využitím pokročilých SQL dotazov na vytvorenie, prepojenie a spracovanie komplexných dátových súborov.

2.3. ZBER A SPRACOVANIE DÁT

- **Extrahovanie Dát z Externých Zdrojov:** Použili sme širokú škálu dátových súborov z NCZI, ktoré zahŕňajú detailné informácie ako stav kategorizácie, stav registrácie liekov, typy očkovania, a mnoho ďalších zdravotníckych a liekových informácií.
- **Prístup k Dátam:** Dáta sme získali z rôznych sekcií webstránky NCZI, vrátane sekcií ako 'Štandardy zdravotníckej informatiky', 'Číselníky zdravotníckej informatiky', a mnohých ďalších.
- **Využitie SQL Developer Dotazov:** Naša práca zahŕňala vytváranie komplexných SQL dotazov pre efektívne generovanie a analýzu dát, čím sme dosiahli presné prepojenia a zabezpečili integritu dát.

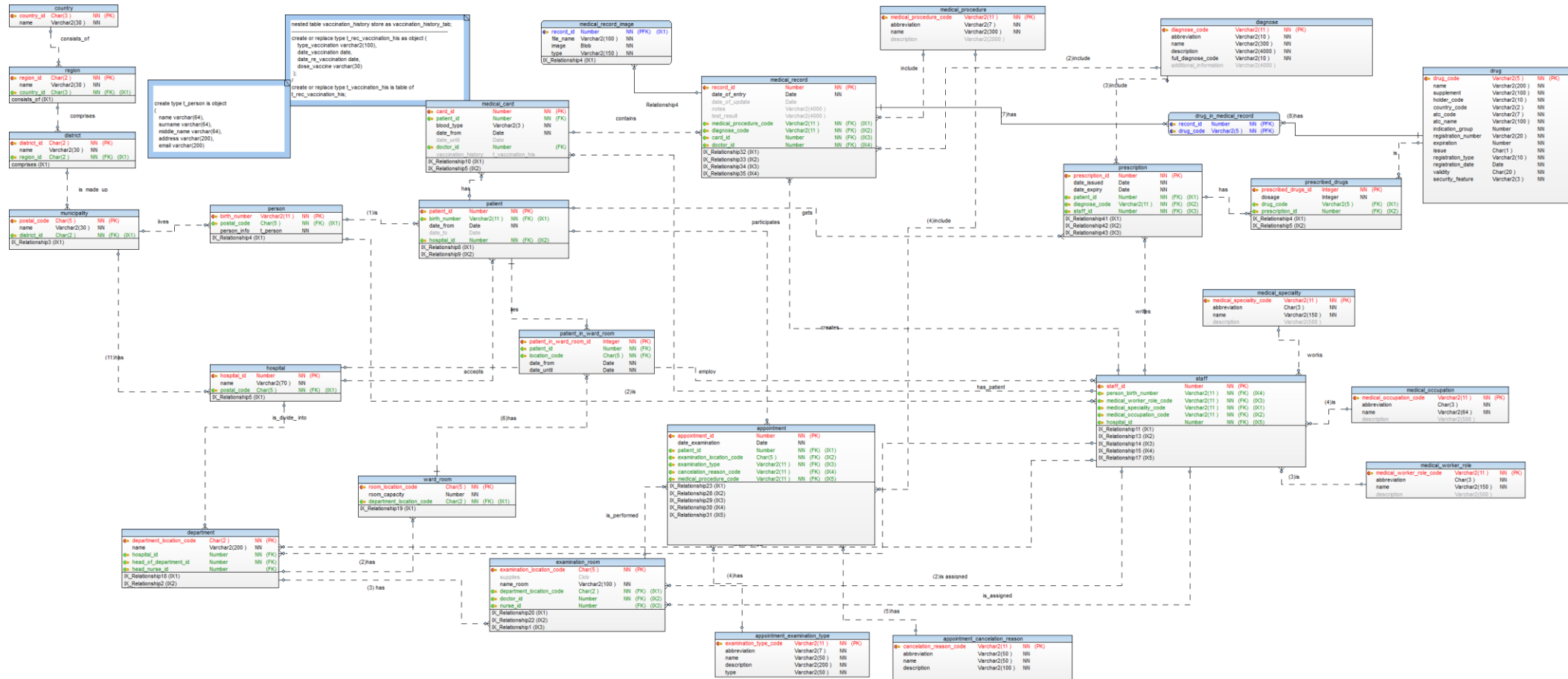
2.4. VÝVOJÁRSKE PROSTREDIE

- **Hlavné Nástroje:** **Visual Studio Code** a **WebStorm** použité pre vývoj a optimalizáciu aplikácie.

2.5. POČETNOST DÁT V TABUĽKÁCH

Tabuľka	Počet dát
country	2
region	10
district	20
municipality	846
hospital	2
person	26436
patient	22493
medical_card	22493
medical_record	6704
medical_record_image	3
medical_procedure	7234
diagnose	19649
drug	50992
prescribed_drugs	94698
prescription	42545
patient_in_ward_room	8243
ward_room	202
department	48
examination_room	148
appointment	6
appointment_examination_type	9
appointment_cancelation_reason	9
staff	3956
medical_speciality	621
medical_occupation	36
medical_worker_role	141
Počet celkových záznamov	307546

3. DATABÁZOVÝ MODEL



4. OPIS DATABÁZY

country - reprezentuje štát

- **country_id**(primárny kľúč) - identifikátor pre konkrétny štát
- **name** - názov štátu

region - reprezentuje konkrétny kraj v štáte

- **region_id**(primárny kľúč) – identifikátor pre konkrétny kraj
- **name** – názov kraja
- **country_id**(cudzí kľúč) – prepojenie na štát v ktorom sa kraj nachádza

district - reprezentuje konkrétny okres

- **district_id**(primárny kľúč) – identifikátor pre konkrétny okres
- **name** – názov okresu
- **region_id**(cudzí kľúč) – prepojenie na kraj v ktorom sa okres nachádza

municipality – reprezentuje danú obec/mesto

- **postal_code**(primárny kľúč) – identifikátor pre konkrétnu obec (PSČ)
- **name** – názov obce/mesta
- **district_id**(cudzí kľúč) – prepojenie na okres v ktorom sa obec nachádza

hospital – reprezentuje konkrétnu nemocnicu

- **hospital_id**(primárny kľúč) – identifikátor pre konkrétnu nemocnicu
- **name** – názov nemocnice
- **postal_code**(cudzí kľúč) – prepojenie na obec v ktorom sa nemocnica nachádza

person – reprezentuje zamestnanca alebo pacienta v danej nemocnici

- **birth_id**(primárny kľúč) – identifikátor pre konkrétnu osobu (rodné číslo)
- **postal_code**(cudzí kľúč) – prepojenie na obec v ktorom osoba žije
- **person_info** – objektový atribút v ktorom sa nachádza meno, priezvisko, stredné meno, adresa a email pacienta

patient – reprezentuje konkrétneho pacienta v nemocnici

- **patient_id**(primárny kľúč) – identifikátor pre konkrétneho pacienta
- **birth_number**(cudzí kľúč) – prepojenie na osobu, kvôli základným informáciám
- **date_from** – dátum odkedy je pacientom danej nemocnice
- **date_to** – dátum dokedy bol pacientom danej nemocnice
- **hospital_id**(cudzí kľúč) – prepojenie na nemocnicu v ktorej je pacientom

medical_card – reprezentuje pacientovu zdravotnú kartu

- **card_id**(primárny kľúč) – identifikátor pre konkrétnu zdravotnú kartu
- **patient_id**(cudzí kľúč) – prepojenie na pacienta, ktorému zdravotná karta patrí

- **blood_type** – krvná skupina pacienta
- **date_from** – dátum odkedy má pacient danú kartu
- **date_until** – dátum dokedy mal pacient danú kartu
- **doctor_id(cudzí kľúč)** – id všeobecného lekára ktorému pacient patrí
- **vaccination_history** – história vakcín, ktoré má pacient(nested table)

medical_record – reprezentuje konkrétny záznam v zdravotnej karte

- **record_id(primárny kľúč)** – identifikátor pre konkrétny záznam v zdravotnej karte
- **date_entry** – dátum, kedy bol záznam vytvorený
- **date_of_update** – dátum, kedy bol záznam aktualizovaný
- **notes** – poznámky lekára v zázname
- **test_result** – výsledky testov napr. pri odbere krvi
- **medical_procedure_code(cudzí kľúč)** – prepojenie do tabuľky medical_procedure, ktorá reprezentuje procedúru, ktorý uskutočnil lekár na pacientovi
- **diagnose_code(cudzí kľúč)** – prepojenie do tabuľky diagnose, ktorá reprezentuje určitú diagnózu, ktorú ma pacient
- **card_id(cudzí kľúč)** – prepojenie na zdravotnú kartu, ktorej záznam patrí
- **doctor_id(cudzí kľúč)** – prepojenie na lekára, ktorý vykonal daný záznam

medical_record_image – reprezentuje obrázok v zdravotnej karte

- **record_id(primárny cudzí kľúč)** – identifikátor, ktorému záznamu patrí daný obrázok
- **file_name** – cesta k súboru, kde sa obrázok nachádza
- **image(Blob)** – konkrétny obrázok
- **type** – reprezentuje o aký typ obrázku ide

medical_procedure – reprezentuje lekárske postupy

- **medical_procedure_code(primárny kľúč)** – identifikátor pre konkrétny lekárske postupy
- **abbreviation** – skratka daného lekárskeho postupu
- **name** – názov konkrétneho lekárskeho postupu
- **description** – opis konkrétneho lekárskeho postupu

diagnose – reprezentuje určitú diagnózu

- **diagnose_code(primárny kľúč)** – identifikátor pre konkrétnu diagnózu
- **abbreviation** – skratka danej diagnózy
- **name** – názov danej diagnózy
- **description** – opis danej diagnózy
- **full_diagnose_code** – celý kód diagnózy
- **additional_information** – ďalšie informácie o danej diagnóze
-

drug – reprezentuje konkrétny liek

- **drug_code(primárny kľúč)** – identifikátor pre konkrétny liek
- **name** – názov daného lieku
- **supplement** – množstvo tabliet lieku
- **holder_code** – unikátny kód pre balenie lieku
- **country code** – reprezentuje krajinu pôvodu lieku

- **atc_code** – reprezentuje do akej kategórie patrí liek
- **atc_name** – reprezentuje informácie o lieku
- **indication_group** – poskytuje ďalšie informácie o diagnóze pre ktorú je liek určený
- **registration_number** – dodatočné identifikačné číslo pre registrovaný liek
- **expiration** – dátum vypršania lieku
- **issue** – typ oznamu, napr. či je liek stiahnutý z predaja
- **registration_type** – typ registrácie, ktorú liek absolvoval
- **registration_date** – dátum kedy bol liek zaregistrovaný
- **validity** – informácie o platnosti registrácie lieku
- **security_feature** – informácie o bezpečnostných prvkoch, ktoré sú integrované do obalu alebo balenia lieku

drug_in_medical_record – reprezentuje lieky v zdravotnom zázname

- **record_id**(primárny cudzí kľúč) – reprezentuje záznam v ktorom sa nachádza liek
- **drug_code**(primárny cudzí kľúč) – reprezentuje konkrétny liek v zázname

prescribed_drugs – reprezentuje lieky v recepte

- **prescription_id**(primárny kľúč) – identifikátor pre konkrétny recept
- **dossage** – množstvo lieku, koľkokrát ho má daný pacient užiť za 1 deň.
- **drug_code**(cudzí kľúč) – prepojenie na liek, ktorý identifikuje o aký liek ide
- **prescription_id**(cudzí kľúč) – prepojenie na recept, ktorému liek patrí

prescription – reprezentuje recept na lieky

- **prescription_id**(primárny kľúč) – identifikátor pre konkrétny recept
- **date_issued** – dátum, kedy bol recept vytvorený
- **date_expiry** – dátum, kedy bude recept zrušený
- **patient_id**(cudzí kľúč) – prepojenie na pacienta pre ktorého je recept určený
- **diagnose_code**(cudzí kľúč) – prepojenie do tabuľky diagnózy, ktorá reprezentuje určitú diagnózu pre daný recept
- **staff_id**(cudzí kľúč) – prepojenie na lekára, ktorý vykonal daný recept

patient_in_ward_room – reprezentuje pacientov, ktorí sú v lôžkovej časti nemocnice

- **patient_in_ward_room**(primárny kľúč) – identifikátor pre konkrétneho pacienta v izbe
- **patient_id**(cudzí kľúč) – prepojenie na pacienta, ktorý sa nachádza na izbe
- **location_code**(cudzí kľúč) – prepojenie na izbu v ktorej sa pacient nachádza
- **date_from** – dátum, odkedy pacient leží v danej izbe
- **date_until** – predpokladaný dátum, dokiaľ pacient bude ležať v danej izbe

ward_room – reprezentuje konkrétnu izbu s lôžkami

- **room_location_code**(primárny kľúč) – identifikátor pre konkrétnu izbu s lôžkami
- **room_capacity** – kapacita danej izby
- **department_location_code** (cudzí kľúč) – prepojenie na oddelenie, v ktorom sa izba nachádza

department – reprezentuje konkrétne oddelenie nemocnice

- **department_location_code (primárny kľúč)** – identifikátor pre konkrétne oddelenie nemocnice
- **name** – názov oddelenia nemocnice
- **head_of_department_id (cudzí kľúč)** – prepojenie na lekára, ktorý je primárom daného oddelenia
- **head_nurse_id (cudzí kľúč)** – prepojenie na zdravotnú sestru, ktorá je vedúca sestry

examination_room – reprezentuje konkrétnu ambulanciu

- **examination_location_code (primárny kľúč)** – identifikátor pre konkrétnu ambulanciu v nemocnici
- **name_room** – názov danej ambulancie
- **department_location_code (cudzí kľúč)** – prepojenie na oddelenie, v ktorom sa ambulancia nachádza
- **doctor_id (cudzí kľúč)** – prepojenie na lekára, ktorý je pridelený danej ambulancii
- **nurse_id (cudzí kľúč)** – prepojenie na zdravotnú sestru, ktorý je pridelená danej ambulancii
- **supplies** – atribút typu JSON, ktorý reprezentuje zdravotné pomôcky v ambulancii.

appointment – reprezentuje konkrétne lekárske vyšetrenie

- **appointment_id (primárny kľúč)** – identifikátor pre konkrétne lekárske vyšetrenie
- **date_examination** – dátum vyšetrenia
- **examination_location_code (cudzí kľúč)** – prepojenie na ambulanciu, v ktorom sa vykoná lekárske vyšetrenie
- **examination_type (cudzí kľúč)** – prepojenie na tabuľku appointment_examination_type, ktorá určuje o aký typ vyšetrenia ide
- **cancelation_reason_code (cudzí kľúč)** – prepojenie na tabuľku appointment_cancelation_reason, ktorá určuje dôvod zrušenia lekárskeho vyšetrenia
- **medical_procedure_code** – prepojenie na lekárske postupy, ktorý bol vykonaný pri vyšetrení

appointment_examination_type – reprezentuje konkrétny typ lekárskeho vyšetrenia

- **examination_type_code (primárny kľúč)** – identifikátor pre konkrétny typ lekárskeho vyšetrenia
- **abbreviation** – skratka pre daný typ lekárskeho vyšetrenia
- **name** – názov vyšetrenia
- **description** – popis daného lekárskeho vyšetrenia
- **type** – typ daného lekárskeho vyšetrenia

appointment_cancelation_reason - reprezentuje konkrétny dôvod zrušenia vyšetrenia

- **cancelation_reason_code (primárny kľúč)** – identifikátor pre konkrétny dôvod zrušenia lekárskeho vyšetrenia
- **abbreviation** – skratka pre daný dôvod zrušenia lekárskeho vyšetrenia

- **name** – názov dôvodu zrušenia lekárskeho vyšetrenia
- **description**– popis daného zrušenia lekárskeho vyšetrenia

staff – reprezentuje zamestnancov danej nemocnice

- **staff_id** (primárny kľúč) – identifikátor pre konkrétneho zamestnanca
- **person_birth_number**(cudzí kľúč)– prepojenie na osobu
- **medical_worker_role** (cudzí kľúč)– prepojenie na typ profesie, ktorú zamestnanec vykonáva
- **medical_speciality_code**(cudzí kľúč)– prepojenie na špecializáciu, ktorej zamestnanec pracuje
- **medical_occupation_code** (cudzí kľúč)– prepojenie na povolanie, ktoré zamestnanec vykonáva
- **hospital_id**(cudzí kľúč)– prepojenie na nemocnicu v ktorej zamestnanec pracuje

medical_speciality – reprezentuje medicínske špecializácie pre zamestnanca

- **medical_speciality_code** (primárny kľúč) – identifikátor pre konkrétnu medicínsku špecializáciu
- **abbreviation** – skratka pre konkrétnu medicínsku špecializáciu
- **name** – názov medicínskej špecializácie
- **description**– popis danej medicínskej špecializácie

medical_occupation – reprezentuje určité povolanie

- **medical_occupation_code** (primárny kľúč) – identifikátor pre konkrétne povolanie
- **abbreviation** – skratka pre konkrétne povolanie
- **name** – názov povolania
- **description**– popis daného povolania

medical_worker_role – reprezentuje konkrétny typ profesie v danej kategórii

- **medical_worker_role** (primárny kľúč) – identifikátor pre konkrétny typ profesie
- **abbreviation** – skratka pre konkrétny typ profesie
- **name** – názov profesie
- **description**– popis danej profesie

4.1. OBJEKTOVÝ ATRIBÚT TABUĽKY

V tabuľke person sa nachádza atribút person_info, ktorý je typu t_person.

```
create type t_person is object
(
  name varchar(64),
  surname varchar(64),
  middle_name varchar(64),
  address varchar(200),
  email varchar(200)
```

```
ISTVANKO.SP.T_PERSON(
  'Olympia',
  'Prodajova',
  'L.Ondrejova 38/14',
  'olympia.prodajova@gmail.com')
```

4.2. JSON ATRIBÚT TABUĽKY

V tabuľke examination_room je atribút supplies typu JSON

```
{
  "Stetoskop":100,
  "Pulzny_oxymeter":1,
  "Teplomer":2,
  "EKG_prístroj":2,
  "Obvazy":200,
  "Gaza":10000
}
```

4.3. NESTED TABLE ATRIBÚT TABUĽKY

V tabuľke medical_card je atribút vaccination_history ako nested table.

```
nested table vaccination_history store as vaccination_history_tab;

create or replace type t_rec_vaccination_his as object (
  type_vaccination varchar2(100),
  date_vaccination date,
  date_re_vaccination date,
  dose_vaccine varchar(30)
);

create or replace type t_vaccination_his is table of
t_rec_vaccination_his;
```

ISTVANKO.SP.T_REC_VACCINATION_HIS
ISTVANKO.SP.T_REC_VACCINATION_HIS(1, 'rotavírusové infekcie', '1997-07-06 00:00:00.0', '1997-08-03 00:00:00.0', 'I. dávka')
ISTVANKO.SP.T_REC_VACCINATION_HIS(2, 'rotavírusové infekcie', '1997-08-03 00:00:00.0', '1997-08-31 00:00:00.0', 'II. dávka')
ISTVANKO.SP.T_REC_VACCINATION_HIS(3, 'záškrt', '1997-08-25 00:00:00.0', '1997-11-25 00:00:00.0', 'I. dávka')
ISTVANKO.SP.T_REC_VACCINATION_HIS(4, 'tetanus', '1997-08-25 00:00:00.0', '1997-11-25 00:00:00.0', 'I. dávka')
ISTVANKO.SP.T_REC_VACCINATION_HIS(5, 'čierny kašeľ', '1997-08-25 00:00:00.0', '1997-11-25 00:00:00.0', 'I. dávka')
ISTVANKO.SP.T_REC_VACCINATION_HIS(6, 'rotavírusové infekcie', '1997-08-31 00:00:00.0', NULL, 'III. dávka')
ISTVANKO.SP.T_REC_VACCINATION_HIS(7, 'vírusová hepatitída B', '1997-11-25 00:00:00.0', NULL, 'II. dávka')
ISTVANKO.SP.T_REC_VACCINATION_HIS(8, 'hemofilové invazívne infekcie', '1997-11-25 00:00:00.0', NULL, 'II. dávka')
ISTVANKO.SP.T_REC_VACCINATION_HIS(9, 'detská obrna', '1997-11-25 00:00:00.0', '1998-05-25 00:00:00.0', 'I. dávka')
ISTVANKO.SP.T_REC_VACCINATION_HIS(10, 'vírusová hepatitída A', '1998-05-25 00:00:00.0', '1998-11-25 00:00:00.0', 'I. dávka')
ISTVANKO.SP.T_REC_VACCINATION_HIS(11, 'pneumokokové invazívne ochorenia', '1998-05-25 00:00:00.0', NULL, 'III. dávka')
ISTVANKO.SP.T_REC_VACCINATION_HIS(12, 'vírusová hepatitída A', '1998-11-25 00:00:00.0', NULL, 'II. dávka')
ISTVANKO.SP.T_REC_VACCINATION_HIS(13, 'záškrt', '2003-05-25 00:00:00.0', '2010-05-25 00:00:00.0', 'preočkovanie')
ISTVANKO.SP.T_REC_VACCINATION_HIS(14, 'tetanus', '2003-05-25 00:00:00.0', '2010-05-25 00:00:00.0', 'preočkovanie')
ISTVANKO.SP.T_REC_VACCINATION_HIS(15, 'čierny kašeľ', '2003-05-25 00:00:00.0', '2010-05-25 00:00:00.0', 'preočkovanie')
ISTVANKO.SP.T_REC_VACCINATION_HIS(16, 'detská obrna', '2010-05-25 00:00:00.0', '2025-05-25 00:00:00.0', 'preočkovanie')
ISTVANKO.SP.T_REC_VACCINATION_HIS(17, 'záškrt', '2010-05-25 00:00:00.0', '2025-05-25 00:00:00.0', 'preočkovanie')
ISTVANKO.SP.T_REC_VACCINATION_HIS(18, 'tetanus', '2010-05-25 00:00:00.0', '2025-05-25 00:00:00.0', 'preočkovanie')
ISTVANKO.SP.T_REC_VACCINATION_HIS(19, 'čierny kašeľ', '2010-05-25 00:00:00.0', '2025-05-25 00:00:00.0', 'preočkovanie')
ISTVANKO.SP.T_REC_VACCINATION_HIS(20, 'detská obrna', '2010-05-25 00:00:00.0', '2025-05-25 00:00:00.0', 'preočkovanie')
ISTVANKO.SP.T_REC_VACCINATION_HIS(21, 'ovcie kiahne', '2011-05-25 00:00:00.0', '2011-06-22 00:00:00.0', 'I. dávka')
ISTVANKO.SP.T_REC_VACCINATION_HIS(22, 'ovcie kiahne', '2011-06-22 00:00:00.0', NULL, 'II. dávka')

5. XML REPORT – LIEKY PACIENTA

5.1. SQL DOTAZ

SELECT

```
p.prescription_id prescription_id,
XMLROOT(
  XMLELEMENT(
    "prescription",
    XMLATTRIBUTES(p.prescription_id AS "PID"),
    XMLELEMENT("date_issued", p.date_issued),
    XMLELEMENT("date_expiry", p.date_expiry),
    XMLELEMENT(
      "prescribed_by",
      pe.person_info.name || ' ' || pe.person_info.surname
    ),
    XMLELEMENT(
      "drugs",
      XMLAGG(
        XMLELEMENT(
          "drug",
          XMLATTRIBUTES(pd.drug_code AS "dc"),
          XMLELEMENT("name", d.name || d.supplement),
          XMLELEMENT("expiration", d.expiration || ' days'),
          XMLELEMENT("dosage", pd.dosage || ' a day')
        )
      )
    )
  ),
  version '1.0'
) xml
```

FROM

```
prescription p
JOIN staff s ON s.staff_id = p.staff_id
JOIN person pe ON pe.birth_number = s.person_birth_number
JOIN prescribed_drugs pd ON pd.prescription_id = p.prescription_id
JOIN drug d ON pd.drug_code = d.drug_code
```

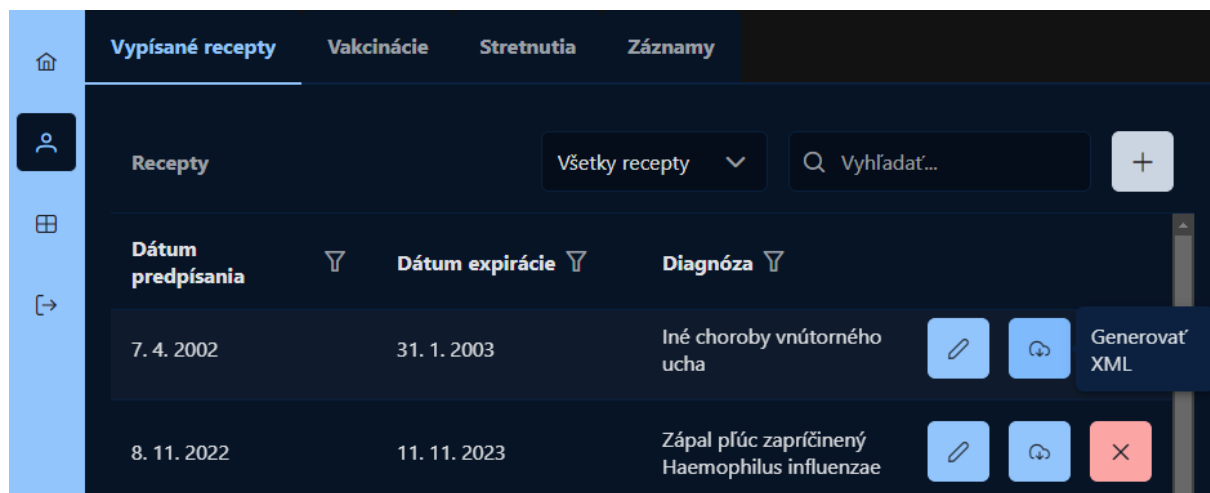
WHERE

```
p.prescription_id = :prescription_id
```

GROUP BY

```
p.prescription_id,
p.date_issued,
p.date_expiry,
pe.person_info.name,
pe.person_info.surname
```

5.2. GRAFICKÉ ROZHRAŇIE



5.3. UKÁŽKA VYGENEROVANÉHO XML DOKUMENTU

```
<?xml version="1.0"?>
<prescription PID="8019">
  <date_issued>2002-04-07</date_issued>
  <date_expiry>2003-01-31</date_expiry>
  <prescribed_by>Ladislava Sarkozyova</prescribed_by>
  <drugs>
    <drug dc="52518">
      <name>Zypsilan 40 mgcps dur 60x40 mg</name>
      <expiration>60 days</expiration>
      <dosage>2 a day</dosage>
    </drug>
    <drug dc="5558B">
      <name>Pregabalin Sandoz GmbH 300 mg tvrde kapsulycps dur 84(2x42x1)x300
mg (blis.PVC/PVDC/Al-perf.)</name>
      <expiration>36 days</expiration>
      <dosage>5 a day</dosage>
    </drug>
  </drugs>
</prescription>
```

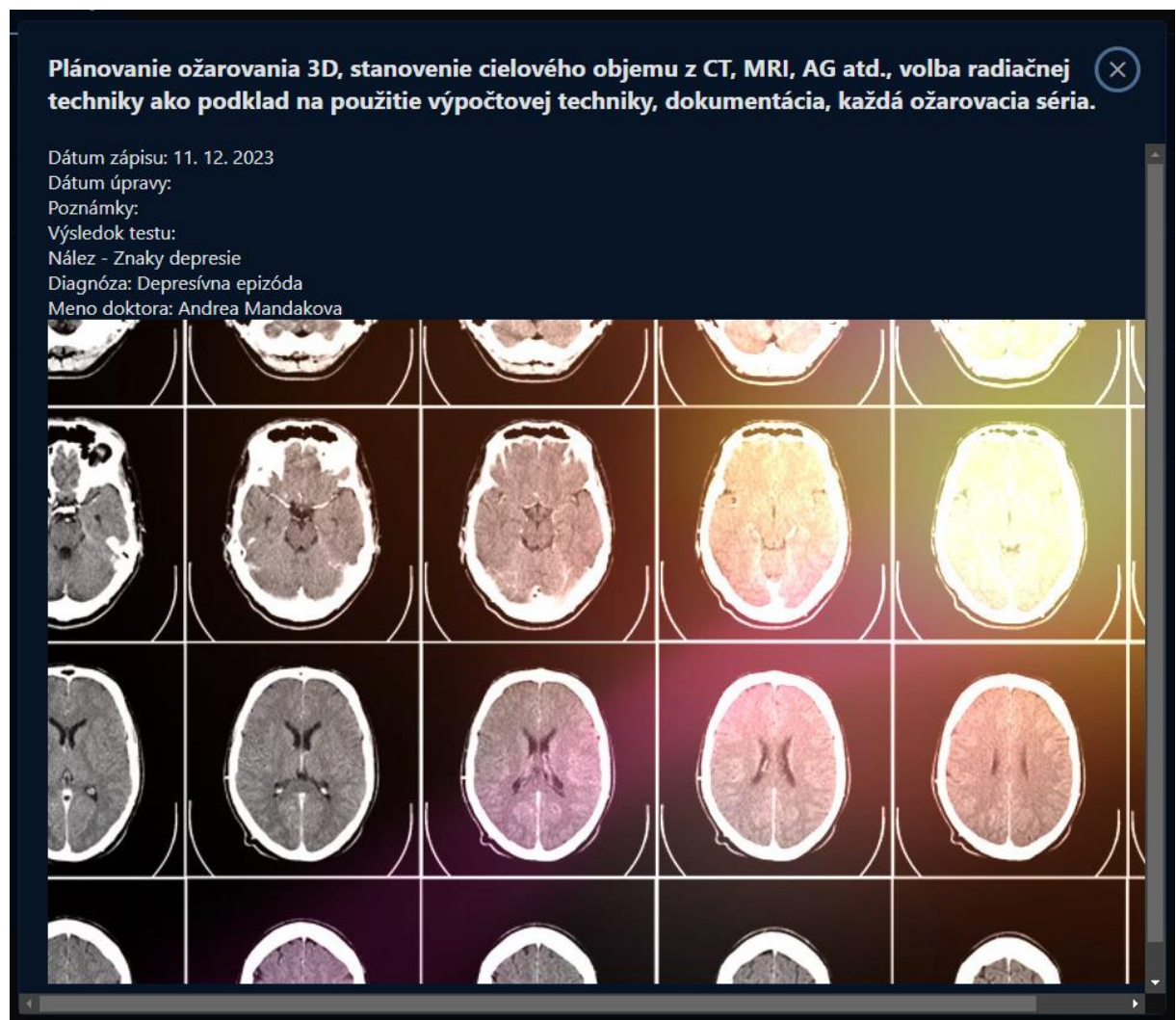
6. SPRÁVA SÚBOROV – OBRÁZOK V ZDRAVOTNEJ KARTE

V zdravotnej karte je možnosť pridať obrázok ku záznamu. Čím sa pacientovi umožní náhľad na rôzne naskenované obrázky alebo rôzne doplňujúce obrázky z vyšetrenia.

6.1. SQL DOTAZ

```
SELECT
    image
FROM
    medical_record_image
WHERE
    record_id = :recordId
```

6.2. GRAFICKÉ ROZHRAVIE – DETAIL ZÁKROKU



7. PRÍSTUP K DÁTAM CEZ VZDIALENÝ SERVER

Vzdialený prístup na server slúži na zálohovanie dát. V prípade pridanie vloženia nového pacienta sa spustí TRIGGER ktorý automaticky zálohuje dáta na vzdialenom serveri.

7.1. VYTvorenie PRIPOJENIA

```
CREATE DATABASE LINK obelix_backup
CONNECT TO istvanko_sp IDENTIFIED
BY ***** USING '(DESCRIPTION =
                      (ADDRESS = (PROTOCOL = TCP)(HOST =
obelix.fri.uniza.sk)(PORT = 1521))
                      (CONNECT_DATA =
                      (SERVER = DEDICATED)
                      (SERVICE_NAME = orcl.fri.uniza.sk)
                      )
                      )';
```

7.2. TABUĽKA A TRIGGER NA VZDIALENOM SERVERI

```
CREATE TABLE person_type (
    BIRTH_NUMBER VARCHAR2(11 BYTE) PRIMARY KEY,
    POSTAL_CODE CHAR(5 BYTE) NOT NULL,
    name VARCHAR2(20) NOT NULL,
    surname VARCHAR2(30) NOT NULL,
    address VARCHAR2(50) NOT NULL,
    email VARCHAR2(65) NOT NULL,
    CONSTRAINT person_type_postal_code FOREIGN KEY (postal_code) REFERENCES
municipality (postal_code)
);
/

CREATE OR REPLACE TRIGGER trigger_person_type_after_insert
AFTER
INSERT
ON person_type FOR EACH ROW DECLARE BEGIN
INSERT INTO
    person
VALUES
    (
        :new.birth_number,
        :new.postal_code,
        t_person(
            :new.name,
            :new.surname,
            :new.address,
            :new.email
        )
    );
END;
/
```

8. VÝSTUPY INFORMAČNÉHO SYSTÉMU

Náš informačný systém poskytuje týchto 13 hlavných informatívnych výstupov o nemocnici:

8.1. VÝPIS PACIENTOV S MOŽNOSŤOU FILTROVANIA, ZORAĐOVANIA PODĽA RÔZNYCH ATRIBÚTOV.

Rodné číslo	Meno	Priezvisko	PSČ		
035829/8457	Nelka	Saligova	1854	Info	Delete
165505/9406	Anabela	Stilova	3842	Info	Delete
877731/6988	Karolína	Rackova	97245	Info	Delete
105804/6165	Branislava	Puckova	95148	Info	Delete
106118/8678	Mia	Janotkova	75661	Info	Delete
797320/0631	Hilda	Schimikova	97245	Info	Delete
026031/8366	Radovana	Valjaskova	94145	Info	Delete
786226/4509	Milota	Slivkova	97231	Info	Delete

8.2. VÝPIS AMBULANCIÍ S MOŽNOSŤOU FILTROVANIA, ZORAĐOVANIA PODĽA RÔZNYCH ATRIBÚTOV

Číslo ambulancie	Názov ambulancie	Oddelenie	Lekár	Zdravotná sestra			
AB001	Endoskopické pracovisko	AB	503308	503814	Info	Edit	Delete
AD001	Prijímová a konziliárna ambulancia	AD	501103	500420	Info	Edit	Delete
AD007	Ambulancia klinickej psychológie I	AD	500176	503081	Info	Edit	Delete
AD010	Psychiatrická príjmová a konziliárna ambulancia II	AD	503227	503723	Info	Edit	Delete
AE004	Hepatologická ambulancia	AE	501438	503815	Info	Edit	Delete
AE005	Ambulancia klinickej imunológie a alergológie	AE	501399	500590	Info	Edit	Delete

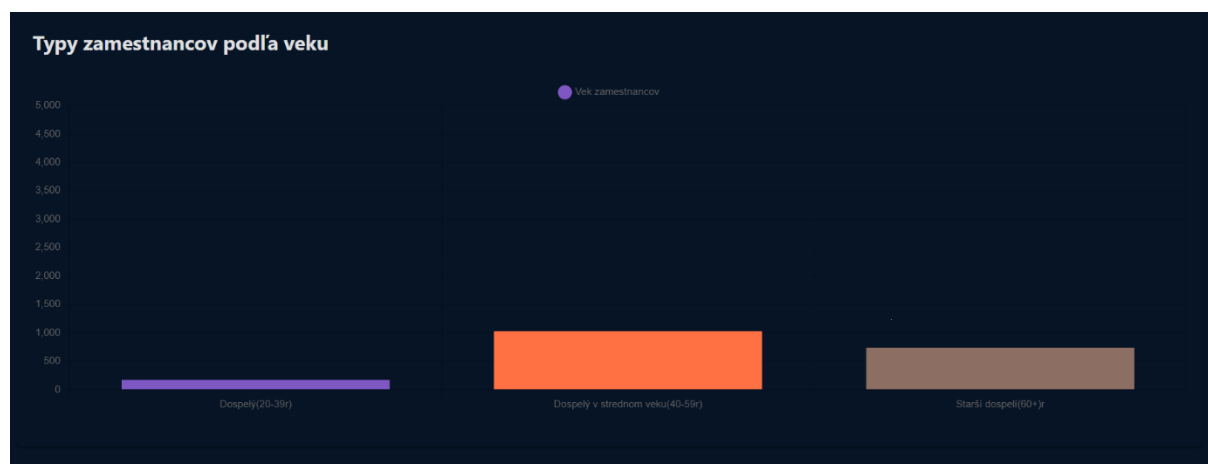
8.3. VÝPIS KTORÝ ZOBRAZUJE INFORMÁCIE O LÔŽKOVÝCH IZBÁCH (VOĽNÉ MIESTA, OBSADENÉ MIESTA, KAPACITU A PERCENTUÁLNU OBSADENOSŤ)

Využitie nemocničných priestorov				
Názov oddelenia	Voľné miesta	Obsadené miesta	Kapacita	Obsadenosť (%)
Gynekologicko-pôrodné oddelenie	68	22	90	24.44%
I. chirurgická klinika	91	13	104	12.50%
I. interná klinika	56	16	72	22.22%
II. chirurgická klinika	93	21	114	18.42%
Klinika stomatológie a maxilofaciálnej chirurgie	20	7	27	25.93%
Neurologické oddelenie	35	10	45	22.22%
Oddelenie rádiologických a zobrazovacích metód	9	7	16	43.75%
Urologické oddelenie	31	2	33	6.06%

8.4. VÝPIS ODDELENÍ PODĽA POČTU LEKÁROV. VÝPIS JE USPORIADANÝ PODĽA POČTU LEKÁROV ZOSTUPNE.

Top 20 oddelení s najväčším počtom lekárov v ambulanciách			
Limit:	20		
Skratka oddelenia	Názov oddelenia	Počet lekárov	Počet sestričiek
EB	Gynekologicko-pôrodné oddelenie	7	7
EA	Klinika stomatológie a maxilofaciálnej chirurgie	6	6
AH	Klinika pracovného lekárstva a klinickej toxikológie	5	5
EF	Neurologické oddelenie	5	5
EG	Očné oddelenie	4	4
FB	Oddelenie rádiologických a zobrazovacích metód	4	4
AD	I. psychiatrická klinika	3	3
FC	Urologické oddelenie	3	3
AE	I. interná klinika	3	3
AB	I. chirurgická klinika	1	1

8.5. VÝPIS, KTORÝ UKAZUJE POČET ZAMESTNANCOV V JEDNOTLIVÝCH VEKOVÝCH KATEGÓRIÁCH



8.6. VÝPIS NA ZOBRAZENIE TOP 10 DIAGNÓZ PODĽA PRIEMERNEJ DĹŽKY POBYTU

Top 10 diagnózy podľa priemernej dĺžky pobytu

Limit: 10

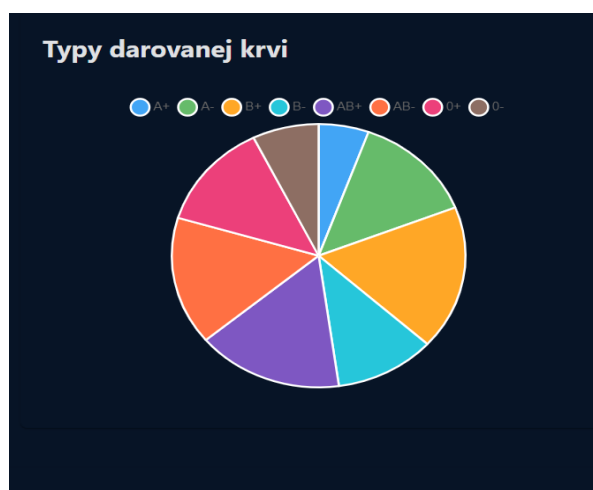
Diagnóza	Priemerná dĺžka pobytu (dni)
Choroba ekkrinných potných žliaz	18.09962962962962
Akútny zápal hrtana	17
Endometrióza panvového peritonea	17
Hypoparatyreóza, bližšie neurčená	17
Intersticiálna a hlboká keratitída	17
Iný problém s výživou (kŕmením) u novorodenca	17
Myasthenia gravis	17
Nedostatok mangánu	17
Trhlínka bradavky v súvislosti s pôrodom, bez ťažkostí pri príložení	17
Voľné teleso v kĺbe predlaktia (laktová kosť, vretenná kosť, zápästný kĺb)	17

8.7. VÝPIS, PRE TOP 10 PACIENTOV S NAJvyšším POČTOM DAROVANÍ KRVÍ

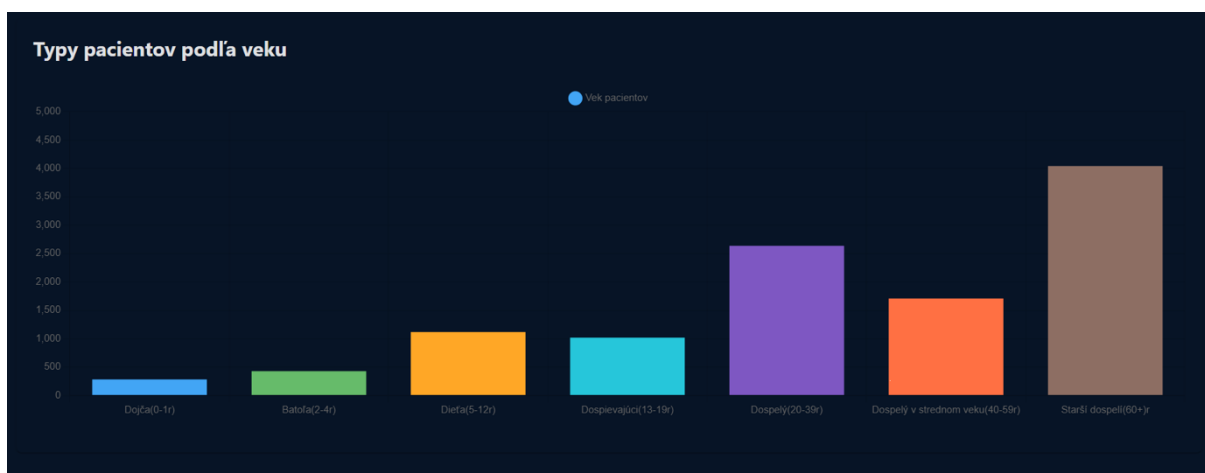
Top 10 pacientov s najvyšším počtom darovaní krvi

Poradie	Počet darovaní	Meno
1	4	Barbara Junosova
2	4	Jana Matusova
3	4	Ida Michalikova
4	4	Kornel Skovajsa
5	4	Sandra Polohova
6	3	Klára Herudova
7	3	Želmíra Junosova
8	3	Ema Slizova
9	3	Miriama Simasekova
10	3	Denis Dubravka

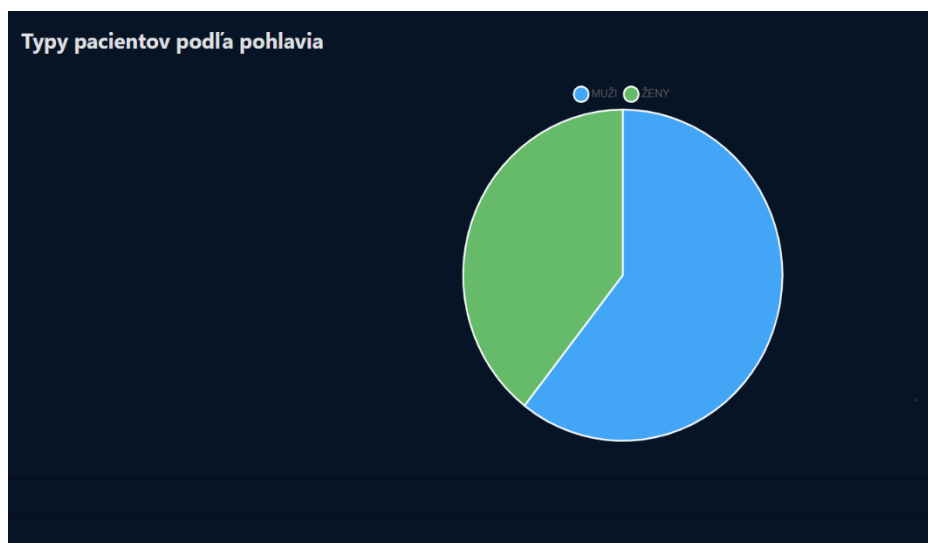
8.8. VÝPIS PRE TYPY DAROVANEJ KRVÍ AJ S POČTOM DAROVANÍ



8.9. VÝPIS PACIENTOV PODĽA VEKOVEJ KATEGÓRIE



8.10. VÝPIS PACIENTOV PODĽA POHLAVIA



8.11. VÝPIS NAJSTARŠÍCH PACIENTOV

Top 10 najstarších pacientov

Limit: 10

Meno	Priezvisko	Rodné číslo	Dátum narodenia	Vek
Miloš	Studený	300317/405	17.03.1930	93
Irma	Balcarkova	305318/516	18.03.1930	93
Filip	Gajdosik	300319/548	19.03.1930	93
Ernest	Dubovsky	300324/896	24.03.1930	93
Medard	Majtan	300328/489	28.03.1930	93
Lubor	Chudoba	300330/609	30.03.1930	93
Ignác	Backady	300401/801	01.04.1930	93
Lukas	Schelling	300401/284	01.04.1930	93
Bohuslav	Velky	300401/358	01.04.1930	93
Hubert	Psenak	300404/035	04.04.1930	93

8.12. MNOŽSTVO PACIENTOV NARODENÝCH V JEDNOTLIVÝCH MESIACHOCH KONKRÉTNÉHO ROKA



8.13. VÝPIS POČTU DNEŠNÝCH STRETNUTÍ

Nemocnica

Rok

Počet dnešných appointmentov

0

9. SQL DOTAZY ZLOŽITEJŠÍCH VÝSTUPOV

9.1. ZÍSKANIE POČTU VOĽNÝCH MIEST PRE AMBULANCIE

```
create or replace FUNCTION "GETFREESPACESINHOSPITAL" (p_hospital_id IN NUMBER)
RETURN SYS_REFCURSOR
IS
    rc SYS_REFCURSOR;
BEGIN
    OPEN rc FOR
        SELECT
            d.name AS department_name,
            d.hospital_id,
            SUM(wr.room_capacity) AS total_capacity,
            (SUM(wr.room_capacity) - COUNT(pwr.patient_id)) AS free_spaces
        FROM
            department d
        JOIN
            ward_room wr ON d.department_location_code = wr.department_location_code
        LEFT JOIN
            patient_in_ward_room pwr ON wr.room_location_code = pwr.location_code
            AND pwr.date_from <= SYSDATE
            AND (pwr.date_until IS NULL OR pwr.date_until >= SYSDATE)
        WHERE
            d.hospital_id = p_hospital_id
        GROUP BY
            d.name, d.hospital_id
        ORDER BY
            d.name;

    RETURN rc;
END GetFreeSpacesInHospital;
```

9.2. ZÍSKANIE ODDELENÍ PODĽA POČTU LEKÁROV

```

SELECT
    DEPARTMENT_LOCATION_CODE,
    DEPARTMENT_NAME,
    numbers_doctors,
    numbers_nurses
FROM (
    SELECT
        d.DEPARTMENT_LOCATION_CODE,
        d.NAME AS DEPARTMENT_NAME,
        COUNT(DISTINCT er.DOCTOR_ID) AS numbers_doctors,
        COUNT(DISTINCT er.NURSE_ID) AS numbers_nurses,
        ROW_NUMBER() OVER (ORDER BY COUNT(DISTINCT er.DOCTOR_ID) DESC) AS
row_number
    FROM
        DEPARTMENT d
    LEFT JOIN
        EXAMINATION_ROOM er ON d.DEPARTMENT_LOCATION_CODE =
er.DEPARTMENT_LOCATION_CODE
    WHERE
        d.HOSPITAL_ID = :hospitalId
    GROUP BY
        d.DEPARTMENT_LOCATION_CODE, d.NAME
)
WHERE
    row_number <= :limitRows
    
```

9.3. ZÍSKANIE VEKOVÝCH KATEGÓRIÍ PRE ZAMESTNANCOV

```

SELECT
    COUNT(CASE WHEN age BETWEEN 20 AND 39 THEN 1 ELSE NULL END) AS "Dospelý(20-
39r)",
    COUNT(CASE WHEN age BETWEEN 40 AND 59 THEN 1 ELSE NULL END) AS "Dospelý v
strednom veku(40-59r)",
    COUNT(CASE WHEN age >= 60 THEN 1 ELSE NULL END) AS "Starší dospelí(60+)r"
FROM (
    SELECT
        FLOOR(MONTHS_BETWEEN(SYSDATE, F_BIRTH_DATE(person_birth_number)) / 12) AS
age
    FROM
        staff
    WHERE staff.hospital_id = :hospitalId
)
    
```


9.4. ZÍSKANIE TOP N DIAGNÓZ PODĽA PRIEMERNEJ DĹŽKY POBYTU V NEMOCNICI

```
SELECT
    pr.diagnose_code,
    d.name AS diagnosis_name,
    AVG(pwr.date_until - pwr.date_from) AS average_stay_duration
FROM
    prescription pr
JOIN
    diagnose d ON pr.diagnose_code = d.diagnose_code
JOIN
    patient_in_ward_room pwr ON pr.patient_id = pwr.patient_id
JOIN
    patient pt ON pwr.patient_id = pt.patient_id
JOIN
    hospital h ON pt.hospital_id = h.hospital_id
WHERE
    pwr.date_from IS NOT NULL AND
    pwr.date_until IS NOT NULL AND
    h.hospital_id = :hospitalId
GROUP BY
    pr.diagnose_code, d.name
ORDER BY
    average_stay_duration DESC, d.name ASC
FETCH FIRST :limitRows ROWS ONLY
```

9.5. ZÍSKANIE TOP 10 ĽUDÍ, S NAJVIAC DAROVANÍM KRVÍ

```

SELECT
    *
FROM
    (
        SELECT
            mr.card_id,
            pe.person_info.name || ' ' || pe.person_info.surname person_name,
            h.hospital_id,
            ROW_NUMBER() OVER(PARTITION BY h.hospital_id ORDER BY COUNT(*))
        DESC) rn,
            COUNT(*) num_blood_donations
        FROM
            medical_record mr
        JOIN medical_card mc ON mc.card_id = mr.card_id
        JOIN patient p ON p.patient_id = mc.patient_id
        JOIN person pe ON pe.birth_number = p.birth_number
        JOIN hospital h ON h.hospital_id = p.hospital_id
        WHERE
            h.hospital_id = :hospitalId
            AND EXTRACT(YEAR FROM mr.date_of_entry) = EXTRACT(YEAR FROM
TO_DATE(:year, 'MM-DD-YYYY'))
        GROUP BY
            mr.card_id,
            h.hospital_id,
            pe.person_info.name,
            pe.person_info.surname
    ) rank_tab
WHERE rn <= 10
ORDER BY rn

```

9.6. ZÍSKANIE POČTU DAROVANIA KRVÍ PODĽA TYPU KRVÍ

```

SELECT
    SUM(CASE WHEN mc.blood_type = 'A+' THEN 1 ELSE 0 END) "A+",
    SUM(CASE WHEN mc.blood_type = 'A-' THEN 1 ELSE 0 END) "A-",
    SUM(CASE WHEN mc.blood_type = 'B+' THEN 1 ELSE 0 END) "B+",
    SUM(CASE WHEN mc.blood_type = 'B-' THEN 1 ELSE 0 END) "B-",
    SUM(CASE WHEN mc.blood_type = 'AB+' THEN 1 ELSE 0 END) "AB+",
    SUM(CASE WHEN mc.blood_type = 'AB-' THEN 1 ELSE 0 END) "AB-",
    SUM(CASE WHEN mc.blood_type = '0+' THEN 1 ELSE 0 END) "0+",
    SUM(CASE WHEN mc.blood_type = '0-' THEN 1 ELSE 0 END) "0-"
FROM medical_card mc
JOIN medical_record mr ON mr.card_id = mc.card_id
JOIN patient p ON p.patient_id = mc.patient_id
JOIN hospital h ON h.hospital_id = p.hospital_id
WHERE
    mr.medical_procedure_code IN (02089715581, 00000468124)
    AND h.hospital_id = :hospitalId
    AND EXTRACT(YEAR FROM mr.date_of_entry) = EXTRACT(YEAR FROM TO_DATE(:year,
'MM-DD-YYYY'))

```

9.7. ZÍSKANIE POČET PACIENTOV PODĽA VEKOVÝCH KATEGÓRIÍ

```

SELECT
    COUNT(CASE WHEN age BETWEEN 0 AND 1 THEN 1 ELSE NULL END) AS "Dojča(0-1r)",
    COUNT(CASE WHEN age BETWEEN 2 AND 4 THEN 1 ELSE NULL END) AS "Batoľa(2-4r)",
    COUNT(CASE WHEN age BETWEEN 5 AND 12 THEN 1 ELSE NULL END) AS "Dieťa(5-
12r)",
    COUNT(CASE WHEN age BETWEEN 13 AND 19 THEN 1 ELSE NULL END) AS
"Dospievajúci(13-19r)",
    COUNT(CASE WHEN age BETWEEN 20 AND 39 THEN 1 ELSE NULL END) AS "Dospelý(20-
39r)",
    COUNT(CASE WHEN age BETWEEN 40 AND 59 THEN 1 ELSE NULL END) AS "Dospelý v
strednom veku(40-59r)",
    COUNT(CASE WHEN age >= 60 THEN 1 ELSE NULL END) AS "Starší dospelí(60+)r"
FROM (
    SELECT
        FLOOR(MONTHS_BETWEEN(SYSDATE, F_BIRTH_DATE(birth_number)) / 12) AS age
    FROM
        patient
    WHERE patient.hospital_id = :hospitalId
)

```

9.8. ZÍSKANIE POČTU PACIENTOV PODĽA POHLAVIA

```

SELECT
    SUM(CASE WHEN SUBSTR(birth_number, 3, 1) IN (0, 1, 2, 3) THEN 1 ELSE 0
END) Muži,
    SUM(CASE WHEN SUBSTR(birth_number, 3, 1) IN (5, 6, 7, 8) THEN 1 ELSE 0
END) Ženy
    FROM patient where hospital_id=:hospitalId

```

9.9. ZÍSKANIE TOP N NAJSTARŠÍCH PACIENTOV PRE NEMOCNICU

```

SELECT
    name,
    surname,
    birth_number,
    TO_CHAR(birth_date, 'DD.MM.YYYY') AS birth_date,
    age
FROM (
    SELECT
        p.person_info.name AS name,
        p.person_info.surname AS surname,
        birth_number,
        F_BIRTH_DATE(birth_number) AS birth_date,
        FLOOR(MONTHS_BETWEEN(SYSDATE, F_BIRTH_DATE(birth_number)) / 12) AS age
    FROM
        patient
    JOIN
        person p USING (birth_number)
    WHERE
        patient.hospital_id = :hospitalId
    ORDER BY birth_date ASC
)
WHERE ROWNUM <= :limitRows
    
```

9.10. ZÍSKANIE MNOŽSTVA ĽUDÍ, PODĽA NARODENIA V MESIACOCH

```

SELECT
    COUNT(CASE WHEN TO_CHAR(birth_date, 'MM') = '01' THEN 1 END) AS january,
    COUNT(CASE WHEN TO_CHAR(birth_date, 'MM') = '02' THEN 1 END) AS february,
    COUNT(CASE WHEN TO_CHAR(birth_date, 'MM') = '03' THEN 1 END) AS march,
    COUNT(CASE WHEN TO_CHAR(birth_date, 'MM') = '04' THEN 1 END) AS april,
    COUNT(CASE WHEN TO_CHAR(birth_date, 'MM') = '05' THEN 1 END) AS may,
    COUNT(CASE WHEN TO_CHAR(birth_date, 'MM') = '06' THEN 1 END) AS june,
    COUNT(CASE WHEN TO_CHAR(birth_date, 'MM') = '07' THEN 1 END) AS july,
    COUNT(CASE WHEN TO_CHAR(birth_date, 'MM') = '08' THEN 1 END) AS august,
    COUNT(CASE WHEN TO_CHAR(birth_date, 'MM') = '09' THEN 1 END) AS september,
    COUNT(CASE WHEN TO_CHAR(birth_date, 'MM') = '10' THEN 1 END) AS october,
    COUNT(CASE WHEN TO_CHAR(birth_date, 'MM') = '11' THEN 1 END) AS november,
    COUNT(CASE WHEN TO_CHAR(birth_date, 'MM') = '12' THEN 1 END) AS december
FROM (
    SELECT
        F_BIRTH_DATE(birth_number) AS birth_date
    FROM
        patient
    WHERE
        patient.hospital_id = :hospitalId
        AND TO_CHAR( F_BIRTH_DATE(birth_number), 'YYYY') = :year
)
    
```

9.11. ZÍSKANIE POČTU STRETNUTÍ ZA DANÝ DEŇ

```
create or replace FUNCTION "GETAPPOINTMENTSCOUNTBYDATE" (p_hospital_id IN
NUMBER, p_date IN DATE DEFAULT TRUNC(SYSDATE))
RETURN NUMBER
IS
    appointments_count NUMBER;
BEGIN
    SELECT COUNT(*)
    INTO appointments_count
    FROM appointment a
    JOIN examination_room er ON a.examination_location_code =
er.examination_location_code
    JOIN department d ON er.department_location_code =
d.department_location_code
    WHERE d.hospital_id = p_hospital_id
    AND TRUNC(a.date_examination) = p_date;

    RETURN appointments_count;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RETURN 0;
    WHEN OTHERS THEN
        RAISE;
END GetAppointmentsCountByDate;
```

9.12. ZÍSKANIE PLATNÝCH RECEPTOV PRE DANÉHO PACIENTA

```
SELECT
    pr.prescription_id,
    pr.patient_id,
    pr.date_issued,
    pr.date_expiry,
    d.name
FROM
    prescription pr
    JOIN patient p ON pr.patient_id = p.patient_id
    JOIN diagnose d ON pr.diagnose_code = d.diagnose_code
WHERE
    p.hospital_id = :hospitalId
    AND pr.patient_id = :patientId
    AND pr.date_expiry < CURRENT_DATE
ORDER BY
    pr.date_expiry DESC
```

9.13. ZÍSKANIE HISTORICKÝCH STRETNUTÍ PRE PACIENTA

```
SELECT
    a.date_examination,
    er.name_room as "Ambulancia",
    er.examination_location_code as "AmbulanciaKod",
    aet.name as "TypVesetrenia",
    mp.name as "LekarskyZakrok"
FROM
    appointment a
    JOIN examination_room er ON a.examination_location_code =
er.examination_location_code
    JOIN appointment_examination_type aet ON a.examination_type =
aet.examination_type_code
    JOIN medical_procedure mp ON a.medical_procedure_code =
mp.medical_procedure_code
WHERE
    a.patient_id = :patientId
    AND a.date_examination < SYSDATE
ORDER BY
    a.date_examination DESC `;
```

10. ANALÝZA OPTIMALIZÁCIE PRÍKAZU SELECT

Naše testy zamerané na optimalizáciu príkazu **SELECT** boli zosúladené s potrebami získania údajov o platných receptoch pre pacientov. Významnou súčasťou príkazu je podmienka založená na aktuálnom dátume, ktorý musí spadať medzi dátum vydania a expirácie predpisu, ako aj filtrácia podľa konkrétneho nemocničného zariadenia. SQL príkaz je nasledovný:

```
SELECT
    pr.prescription_id,
    pr.date_issued,
    pr.date_expiry,
    d.name AS valid_prescriptions
FROM
    prescription pr
JOIN patient p ON pr.patient_id = p.patient_id
JOIN diagnose d ON d.diagnose_code = pr.diagnose_code
WHERE
    CURRENT_DATE BETWEEN pr.date_issued
    AND pr.date_expiry
    AND p.hospital_id = 1
ORDER BY
    pr.date_issued;
```

Aby sme dosiahli zlepšenie výkonu, uvažujeme o implementácii vhodného indexu, ktorý by mohol znížiť čas vykonávania a zlepšiť efektivitu prístupu k dátam. Predpokladáme, že vytvorenie indexu na stĺpcoch *date_issued*, *date_expiry*, a možno aj *patient_id* a *diagnose_code*, ktoré sú súčasťou klauzuly **WHERE**, by mohlo významne ovplyvniť výkonnosť dotazu.

Na analýzu a porovnanie rôznych behov SQL príkazov sme použili funkciu **Explain Plan** dostupnú v prostredí **Oracle SQL Developer**, ktorá nám umožnila posúdiť plány vykonávania SQL príkazov a kvantifikovať metriky ako sú čas a náklady.

10.1. BEZ EXPLICITNÉHO INDEXU

Pri behu zvoleného **SELECT** príkazu **bez špecifického indexu** sme zaznamenali nasledovné ukazovatele výkonnosti, získané priemerovaním výsledkov z 10 samostatných spustení:

- **Priemerný čas:** 1.219 sekúnd
- **Priemerné náklady:** 747

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				747
SORT		ORDER BY		747
HASH JOIN				9700
Access Predicates				9700
D.DIAGNOSE_CODE=PR.DIAGNOSE_CODE				310
NESTED LOOPS				9700
NESTED LOOPS				310
STATISTICS COLLECTOR				
HASH JOIN				19648
Access Predicates				
PR.PATIENT_ID=P.PATIENT_ID				
TABLE ACCESS PATIENT		FULL		11224
Filter Predicates				33
P.HOSPITAL_ID=1				
TABLE ACCESS PRESCRIPTION		FULL		39074
Filter Predicates				106
AND				
PR.DATE_EXPIRY>=CURRENT_DATE				
PR.DATE_ISSUED<=CURRENT_DATE				
INDEX SYS_C00139904		UNIQUE SCAN		
Access Predicates				
D.DIAGNOSE_CODE=PR.DIAGNOSE_CODE				
TABLE ACCESS DIAGNOSE		BY INDEX ROWID		1
TABLE ACCESS DIAGNOSE		FULL		19649

10.2. S EXPLICITNÝM INDEXOM PRE PODMIENKU WHERE

Po vytvorení explicitného indexu s názvom *valid_prescription_index* na stĺpcoch *prescription_id(PK)*, *date_expiry* a *date_issued* z tabuľky *prescription* sme očakávali zlepšenie výkonu príkazu **SELECT**. Index bol navrhnutý tak, aby umožnil databáze efektívnejšie vyhľadávať a filtrovať záznamy, čo by malo viesť k zníženiu času vykonávania dotazu.

```
CREATE INDEX valid_prescription_index
ON prescription (prescription_id, date_expiry, date_issued);
```

- **Priemerný čas:** 1.227 sekúnd
- **Priemerné náklady:** 747

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				747
SORT		ORDER BY		747
HASH JOIN				9700
Access Predicates				9700
D.DIAGNOSE_CODE=PR.DIAGNOSE_CODE				310
NESTED LOOPS				9700
NESTED LOOPS				310
STATISTICS COLLECTOR				
HASH JOIN				19647
Access Predicates				
PR.PATIENT_ID=P.PATIENT_ID				
TABLE ACCESS PATIENT		FULL		11224
Filter Predicates				33
P.HOSPITAL_ID=1				
TABLE ACCESS PRESCRIPTION		FULL		39070
Filter Predicates				106
AND				
PR.DATE_EXPIRY>=CURRENT_DATE				
PR.DATE_ISSUED<=CURRENT_DATE				
INDEX SYS_C00139904		UNIQUE SCAN		
Access Predicates				
D.DIAGNOSE_CODE=PR.DIAGNOSE_CODE				
TABLE ACCESS DIAGNOSE		BY INDEX ROWID		1
TABLE ACCESS DIAGNOSE		FULL		19649

Avšak po implementácii a testovaní sme zistili, že priemerný čas vykonávania dotazu sa signifikantne nezmenil a priemerné náklady aj kardinalita ostali porovnateľné s behom bez indexu, čo naznačuje, že optimalizátor SQL sa rozhodol nový index nepoužiť.

10.3. S INDEXAMI NA CUDZIE KLÚČE

Ako ďalší krok v našej optimalizačnej stratégii sme sa rozhodli vytvoriť indexy na stĺpcoch, ktoré sú cudzími kľúčmi a súčasne sa používajú v **JOIN** operáciách nášho **SELECT** príkazu. Očakávali sme, že indexy na cudzích kľúčoch zlepšia výkon dotazu vďaka efektívnejšiemu spájaniu tabuliek. Boli vytvorené nasledujúce indexy:

```
CREATE INDEX valid_prescription_index_dc
ON prescription (prescription_id, diagnose_code);
CREATE INDEX valid_prescription_index_patient
ON prescription (prescription_id, patient_id);
```

Tieto indexy mali za cieľ zrýchliť vyhľadávanie záznamov na **základe cudzích kľúčov** použitých v príkaze **SELECT**.

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				747
SORT		ORDER BY		747
HASH JOIN			9700	310
Access Predicates				
D.DIAGNOSE_CODE=PR.DIAGNOSE_CODE				
NESTED LOOPS			9700	310
STATISTICS COLLECTOR				
HASH JOIN			19648	139
Access Predicates				
PR.PATIENT_ID=P.PATIENT_ID				
TABLE ACCESS	PATIENT	FULL	11224	33
Filter Predicates				
P.HOSPITAL_ID=1				
TABLE ACCESS	PRESCRIPTION	FULL	39073	106
Filter Predicates				
AND				
PR.DATE_EXPIRY>=CURRENT_DATE				
PR.DATE_ISSUED<=CURRENT_DATE				
INDEX	SYS_C00139904	UNIQUE SCAN		
Access Predicates				
D.DIAGNOSE_CODE=PR.DIAGNOSE_CODE				
TABLE ACCESS	DIAGNOSE	BY INDEX ROWID	1	170
TABLE ACCESS	DIAGNOSE	FULL	19649	170

Po testovaní s novými indexmi sme však zistili, že optimalizátor sa rozhodol indexy nepoužiť. Priemerný čas a priemerné náklady vykonávania dotazu boli:

- **Priemerný čas:** 1.295 sekúnd
- **Priemerné náklady:** 747

Výsledky ukázali mierne zhoršenie výkonu oproti predchádzajúcemu testu bez indexov. Toto zhoršenie však môže byť spôsobené viacerými faktormi, vrátane variácie v záťaži systému, správanie cache a vykonávacie plány, ktoré môžu byť ovplyvnené najnovšími zmenami v databáze.

10.4. S INDEXOM NA TABUĽKU PATIENT

V rámci nášho pokračujúceho úsilia o optimalizáciu výkonu SQL príkazov sme implementovali **index na tabuľke patient**, ktorý bol zameraný na stĺpce *patient_id* a *hospital_id*. Tento index bol navrhnutý s cieľom zlepšiť výkon dotazu, kde sú tieto stĺpce použité v klauzule **WHERE**, a očakávali sme, že zlepšenie výkonu sa prejaví nielen v našom konkrétnom dotaze, ale aj v ďalších dotazoch, ktoré používajú tieto stĺpce. Príkaz na vytvorenie indexu bol nasledovný:

```
CREATE INDEX patient_hospital_id_index
ON patient (patient_id, hospital_id);
```

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				733
SORT				733
HASH JOIN		ORDER BY		9700
Access Predicates				9700
D.DIAGNOSE_CODE=PR.DIAGNOSE_CODE				296
NESTED LOOPS				9700
NESTED LOOPS				296
STATISTICS COLLECTOR				
HASH JOIN				19648
Access Predicates				
PR.PATIENT_ID=P.PATIENT_ID				
INDEX	PATIENT_HOSPITAL_ID_INDEX	FAST FULL SCAN		11224
Filter Predicates				
P.HOSPITAL_ID=1				
TABLE ACCESS	PRESCRIPTION	FULL		39073
Filter Predicates				
AND				
PR.DATE_EXPIRY>=CURRENT_DATE				
PR.DATE_ISSUED<=CURRENT_DATE				
INDEX	SYS_C00139904	UNIQUE SCAN		
Access Predicates				
D.DIAGNOSE_CODE=PR.DIAGNOSE_CODE				
TABLE ACCESS	DIAGNOSE	BY INDEX ROWID		1
TABLE ACCESS	DIAGNOSE	FULL		19649

S novým indexom sme zaznamenali nasledujúce výsledky:

- **Priemerný čas:** 1.1512 sekúnd
- **Priemerné náklady:** 733

V porovnaní s predchádzajúcimi testami bez indexu a s indexmi, ktoré optimalizátor nepoužil, sme zaznamenali **zlepšenie výkonu**. Nižší priemerný čas a náklady naznačujú, že optimalizátor tento index úspešne využil pri vykonávaní dotazu. Tento pozitívny výsledok potvrdzuje, že vhodné indexovanie cudzích kľúčov, ktoré sú často používané v dotazoch, môže byť efektívnou stratégiou pre zlepšenie výkonu databáze.

11. POPIS ULOŽENÝCH PROCEDÚR A FUNKCIÍ

11.1. PROCEDÚRY

- **ADD_APPOINTMENT:** Vloží nové stretnutie do tabuľky APPOINTMENT, obsahuje detaily ako dátum vyšetrenia, identifikácie pacienta a typu vyšetrenia.
- **ADD_EXAMINATION_ROOM:** Pridá novú ambulantnú miestnosť, pokiaľ ešte neexistuje, zahŕňa informácie ako kód miestnosti a súvisiace údaje.
- **ADD_EXAMINATION_ROOM_WITH_SUPPLIES:** Vytvorí ambulantnú miestnosť a pridá do nej zásoby, obsahuje podrobnosti ako kód miestnosti a zoznam zásob.
- **ADD_PATIENT:** Vloží nového pacienta do databázy a synchronizuje informácie s pridruženými tabuľkami.
- **ADD_UPDATE_PATIENT:** Aktualizuje alebo pridáva pacienta v databáze na základe poskytnutých informácií.
- **CHECK_JSON_KEY_EXISTS:** Kontroluje, či konkrétny kľúč existuje v poskytnutom JSON objekte.
- **DELETE_EXAMINATION_ROOM:** Odstráni záznam ambulantnej miestnosti na základe poskytnutého kódu.
- **DELETE_PRESCRIPTION:** Odstráni predpis a súvisiace záznamy o predpísaných liekoch.
- **DELETE_SUPPLY:** Odstráni položku zásob z konkrétnej ambulantnej miestnosti.
- **DELETE_VACCINATION_FOR_PATIENT:** Odstráni záznam o vakcinácii pacienta.
- **INSERT_MEDICAL_RECORD:** Vloží alebo aktualizuje lekársky záznam pacienta vrátane priloženia obrázkových súborov.
- **INSERT_PRESCRIPTION:** Vytvorí nový predpis a pridá do neho predpísané lieky.
- **INSERT_SUPPLY:** Pridá nové zásoby do ambulantnej miestnosti.
- **INSERT_VACCINATION_TO_PATIENT:** Pridá záznam o vakcinácii do lekárskej karty pacienta.
- **P_BIRTH_DATE:** Vypočíta dátum narodenia z rodného čísla.
- **SEED_HOSPITAL1_SPACE_DATA:** Generuje testovacie údaje pre umiestnenie pacientov v nemocnici.
- **UPDATE_EXAMINATION_ROOM:** Aktualizuje údaje o ambulantnej miestnosti vrátane informácií o zdravotníckom personále.
- **UPDATE_PATIENT:** Aktualizuje údaje o pacientovi vrátane osobných a kontaktných informácií.
- **UPDATE_SUPPLY:** Aktualizuje množstvo zásob v konkrétnej ambulantnej miestnosti.

11.2. FUNKCIE

- **F_BIRTH_DATE:** Vráti dátum narodenia vypočítaný z rodného čísla. Zohľadňuje špecifické formáty rodného čísla a príslušné storočie.
- **GET_APPOINTMENTS_COUNT_BY_DATE:** Spočíta počet stretnutí v nemocnici na určitý deň. Výsledok je číselná hodnota predstavujúca počet stretnutí.
- **GET_DEPARTMENT_USAGE_PERCENTAGE:** Vráti kurzor s údajmi o využití oddelení v nemocnici na zadaný deň, vrátane celkovej kapacity a percentuálneho využitia.
- **GET_FREE_SPACES_IN_HOSPITAL:** Vráti kurzor s informáciami o voľných miestach v nemocnici pre každé oddelenie.
- **GET_PATIENT_ADMISSIONS_BY_MONTH:** Vráti kurzor s informáciami o počte prijatí pacientov do nemocnice za mesiac a rok.

- **GET_THE_MOST_PRESCRIBED_DRUG_PER_YEAR:** Vrátí kurzor s informáciami o najčastejšie predpísanom lieku v nemocnici za určitý rok.
- **GET_TODAYS_APPOINTMENTS_COUNT:** Vrátí počet stretnutí naplánovaných na dnešný deň v zadanom zariadení.
- **GET_TOTAL_PATIENT_ADMISSIONS_BY_MONTH:** Vrátí kurzor s celkovým počtom prijatí pacientov do nemocnice za mesiac a rok.
- **INSERT_PRESCRIPTION_F:** Funkcia slúžiaca na testovanie alebo demonštračné účely, ktorá vráti kód lieku z testovacích údajov.