

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Jakob Marušič
**Primerjava performance SQL
podatkovnih baz**

Seminarska naloga pri predmetu Tehnologija upravljanja podatkov

Ljubljana, 2020

Uvod

V poplavi različnih sistemov za upravljanje s podatkovnimi bazami (SUPB) razvijalci pogosto izbirajo glede na dosedanje izkušnje pri delu s SUPB-ji ali glede na poslovna pravila določena s strani organizacije.

Odločitev je pogosto sprejeta s predpostavko, da večina rešitev za upravljanje s podatki omogoča primerjivo performanco. V seminarski nalogi, ki je nastala v okviru predmeta Tehnologija upravljanja podatkov, želim preveriti to predpostavko s testiranje primerljivih operacij v različnih SUPB-jih.

Poglavje 1

Način testiranja

1.1 Izbrani SUPB

Testiranje bom izvajal v treh prosto dostopnih SQL SUPB-jih. Izbrane SUPB-je sem izbral zaradi svoje dostopnosti in razširjenosti med razvijalci in uporabniki informacijskih sistemov.

MySQL je bila v decembru 2019, po podatkih *db-engines.com*, druga najbolj uporabljena podatkovna baza z rezultatom 1275.67 točk. Je brezplačna odprtokodna rešitev, ki jo razvija podjetje Oracle. MySQL je relacijska podatkovna baza napisana v programskem jeziku C in C++.

PostgreSQL je bila v decembru 2019, po podatkih *db-engines.com*, na 4. mestu popularnosti z rezultatom 503.37. Je brezplačna odprtokodna rešitev, za razvoj skrbi PostgreSQL Global Development Group. PostgreSQL je objetna podatkovna baza napisana v programskem jeziku C.

Microsoft SQL Server je bila v decembru 2019, po podatkih *db-engines.com*, na 3. mestu popularnosti z rezultatom 1096.20. MS SQL Server je izdelek, ki ga razvija Microsoft in ni odprtokodne narave. MS SQL Server je relacijska podatkovna baza napisana v programskem jeziku C in C++.

1.2 Način namestitve

Za zagotovitev čim primerljivejših rezultatov so vse baze nameščene kot docker slike na operacijskem sistemu Windows 10 PRO. Docker in docker slike podatkovnih baz so nameščene po navodilih in priporočilih proizvajalcev, nameščene so bile zadnje verzije, ki so bile na voljo 18. decembra 2019.

1.3 Način poganjanja podatkovne baze

Podatkovna baza je bila nameščena in se je izvajala na računalniku s sledečimi specifikacijami:

Procesor	Intel Core i7-8705G 3.10GHz
Delovni pomnilnik	16 GB
Virtualni delovni pomnilnik	28,9 GB
Operacijski sistem	Windows 10 Education
Verzija operacijskega sistema	10.0.18363 Build 18363
Verzija Docker okolja ¹	19.03.5
Verzija MySQL docker slike ²	8.0.18
Verzija PostgreSQL docker slike	
Verzija MS SQL Server docker slike	

V času testiranja na sistemu niso bile pognane drugi docker vmesniki ali drugi programi, ki bi lahko znatno vplivali na performanco sistema.

¹<https://docs.docker.com/docker-for-windows/install/>

²https://hub.docker.com/_/mysql

Poglavje 2

Priprava podatkovnih baz

2.1 Namestitev Docker slik

2.1.1 MySql

```
docker run --name TUP-seminar-mysql -p 52000:3306 -e  
MYSQLROOTPASSWORD=root -d mysql:latest
```

Baza je dostopna na naslovu <http://localhost:52000>

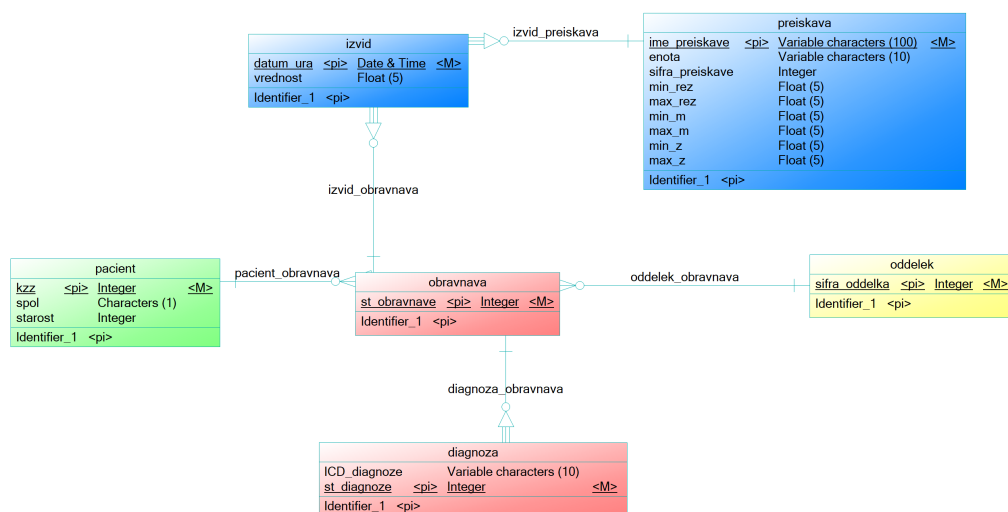
2.2 Podatkovna baza

2.2.1 Konceptualni model podatkovne baze

Podatkovna baza za potrebe testiranja je podatkovna baza zdravstvenega doma, ki se je uporabljala pri 3. domači nalogi. Podatkovna baza je razdeljena na 6 tabel in ima skupno xx vrstic.

2.2.2 Sestava podatkovne baze

Tabela	Število zapisov v tabeli
Pacient	6 071
Obravnava	6 643
Diagnoza	34 274
Oddelek	10
Izvid	846 906
Preiskava	274



Slika 2.1: Konceptni model podatkovne baze

Naloga (a)

Navodila V relacije preslikajte s pomočjo orodja *Power Designer* (*Generate physical model*), kjer iz slike dobljenega logičnega modela prepisete strukturo relacij, npr. *Relacija(A, B, #C)*.

Relacije
kraj(<u>postna_stevilka</u> , kraj)
naslov(<u>naslovID</u> , #postna_stevilka, ulica, hisna_stevilka)
pacient(<u>stevilka_KZZ</u> , #naslovID, ime, priimek, rojstni_datum, spol)
diagnoza_pacient(<u>#stevilka_KZZ</u> , #id_diagnoze, stevilka_diagnoze)
diagnoza(<u>id_diagnoze</u> , slovenska_kvalifikacija, angleska_kvalifikacija, mednarodna_kvalifikacija)
obravnavava(<u>stevilka_obravnave</u> , #stevilka_obravnave)
cakalna_vrsta(<u>#stevilka_obravnave</u> , #stevilka_odelka, datum_vpisa, cas_vpisa, predviden_datum_sprejema, predviden_cas_sprejema)
oddelek(<u>stevilka_odelka</u> , ime_odelka)
izvid_preiskave(<u>stevilka_preiskave</u> , #id_samoplacnika, #sifra_preiskave, #stevilka_obravnave, datum_sreiskave, cas_preiskave, rezultat_preiskave)
samoplacnik(<u>id_samoplacnika</u>)
preiskava(<u>sifra_preiskave</u> , ime_preiskave, min_dopustni_interval, max_dopustni_interval, cena_za_samoplacnike)
referencna_vrednost_moski(<u>sifra_preiskave</u> , ime_preiskave, min_dopustni_interval, max_dopustni_interval, cena_za_samoplacnike, min_referencna_vrednost, max_referencna_vrednost)
referencna_vrednost_zenske(<u>sifra_preiskave</u> , ime_preiskave, min_dopustni_interval, max_dopustni_interval, cena_za_samoplacnike, min_referencna_vrednost, max_referencna_vrednost)

Naloga (b)

Navodila V relacije preslikajte ročno, po navodilih in priporočilih s predavanj.

Relacije
kraj(<u>postna_stevilka</u> , kraj)
naslov(<u>naslovID</u> , #postna_stevilka, ulica, hisna_stevilka)
pacient(<u>stevilka_KZZ</u> , #naslovID, ime, priimek, rojstni_datum, spol)
diagnoza_pacient(<u>#stevilka_KZZ</u> , <u>#id_diagnoze</u> , stevilka_diagnoze)
diagnoza(<u>id_diagnoze</u> , slovenska_kvalifikacija, angleska_kvalifikacija, mednarodna_kvalifikacija)
obravnavava(<u>stevilka_obravnave</u> , #stevilka_obravnave)
cakalna_vrsta(<u>#stevilka_obravnave</u> , <u>#stevilka_oddelka</u> , datum_vpisa, cas_vpisa, predviden_datum_sprejema, predviden_cas_sprejema)
oddelek(<u>stevilka_oddelka</u> , ime_oddelka)
izvid_preiskave(<u>stevilka_preiskave</u> , #id_samoplacnika, #sifra_preiskave, #stevilka_obravnave, datum_sreiskave, cas_preiskave, rezultat_preiskave)
samoplacnik(<u>id_samoplacnika</u>)
preiskava(<u>sifra_preiskave</u> , ime_preiskave, min_dopustni_interval, max_dopustni_interval, cena_za_samoplacnike, min_referencna_vrednost_moski, max_referencna_vrednost_moski, min_referencna_vrednost_zenska, max_referencna_vrednost_zenska)

Naloga (c)

Navodila Primerjajte dobljeni množici relacij, naštejte razlike in jih poskušajte razložiti (po možnosti tudi odpraviti s spremembami v gradnikih konceptualnega modela).

Edina razlika se pojavi pri pretvorbi hierhije iz konceptnega model v logični. Pri ročni pretvorbi sem se odločil za združevanje entitetnih tipov, medtem, ko je avtomatska pretvorba v PowerDesigner-ju ohranila vse tri tabele, akr povzroči redundanco v tabelah *referencna_vrednost_moski* in *referencna_vrednost_zenska*.

Ker pri preslikavi hierhije ne obstaja določen postopek, vendar jih je več, se konceptnega modela (razen, da bi hierhijo popolnoma odstranili) ne da spremeniti do tega, da bi logična modela bila enaka.

Naloga (d)

Navodila Dobljene relacije z orodjem PowerDesigner pretvorite v SQL skripto (za MySQL), kreirajte tabele in jih napolnite (lahko avtomatsko).

```
create table diagnoza
(
  "id_diagnoze"          int not null,
  "slovenska_kvalifikacija" varchar(50),
  "angleska_kvalifikacija" varchar(50),
  "mednarodna_kvalifikacija" varchar(50),
  primary key ("id_diagnoze")
);
create table "diagnoza_pacient"
(
  "stevilka_KZZ"          int not null,
  "id_diagnoze"          int not null,
  "stevilka_diagnoze"     int not null,
  primary key ("stevilka_KZZ", "id_diagnoze")
);
create table "izvid_preiskave"
(
  "datum_preiskave"      date,
```

```

        "cas_preiskave"      time,
        "rezultat_preiskave" float(5),
        "stevilka_izvida"   int not null,
        "id_samoplacnika"   int,
        "sifra_preiskave"   int not null,
        "stevilka_obravnav" int,
        primary key ("stevilka_izvida")
    );
create table kraj
(
    "postna_stevilka" int not null,
    kraj              varchar(20),
    primary key ("postna_stevilka")
);
create table naslov
(
    naslovID int not null,
    "postna_stevilka" int not null,
    ulica      varchar(20),
    "hisna_stevilka" varchar(5),
    primary key (naslovID)
);
create table obravnava
(
    "stevilka_obravnav" int not null,
    "stevilka_KZZ"      int not null,
    primary key ("stevilka_obravnav")
);
create table oddelek
(
    "stevilka_oddelka" int not null,
    "ime_oddelka"      varchar(50),
    primary key ("stevilka_oddelka")
);
create table pacient
(
    "stevilka_KZZ" int not null,
    naslovID      int not null,
    ime           varchar(20),
    priimek       varchar(50),
    "rojstni_datum" date,

```

```

        spol                char(1),
        primary key ("stevilka_KZZ")
    );
create table preiskava
(
    "sifra_preiskave"        int not null,
    "ime_preiskave"          varchar(20),
    "min_dopustni_interval" float(5),
    "max_dopustni_interval" float(5),
    "cena_za_samoplacnike"   float(10),
    primary key ("sifra_preiskave")
);
create table "referencna_vrednost_moski"
(
    "sifra_preiskave"        int not null,
    "ime_preiskave"          varchar(20),
    "min_dopustni_interval" float(5),
    "max_dopustni_interval" float(5),
    "cena_za_samoplacnike"   float(10),
    "min_referencna_vrednost" float(4),
    "max_referencna_vrednost" float(4),
    primary key ("sifra_preiskave")
);
create table "referencna_vrednost_zenske"
(
    "sifra_preiskave"        int not null,
    "ime_preiskave"          varchar(20),
    "min_dopustni_interval" float(5),
    "max_dopustni_interval" float(5),
    "cena_za_samoplacnike"   float(10),
    "min_referencni_interval" float(4),
    "max_referencni_interval" float(4),
    primary key ("sifra_preiskave")
);
create table samoplacnik
(
    "id_samoplacnika"        int not null,
    primary key ("id_samoplacnika")
);
create table "cakalna_vrsta"
(

```

```

    "stevilka_obravnave" int not null,
    "stevilka_odelka"    int not null,
    "datum_vpisa"        date,
    "cas_vpisa"          time,
    "predviden_datum_sprejema" date,
    "predviden_cas_sprejema" time,
    primary key
    ("stevilka_obravnave", "stevilka_odelka")
);

alter table "diagnoza_pacient"
add constraint FK_ima
foreign key ("stevilka_KZZ")
references pacient ("stevilka_KZZ")
on delete restrict on update restrict;

alter table "diagnoza_pacient"
add constraint "FK_je_definirana_z"
foreign key ("id_diagnoze")
references diagnoza ("id_diagnoze")
on delete restrict on update restrict;

alter table "izvid_preiskave"
add constraint FK_definira
foreign key ("sifra_preiskave")
references preiskava ("sifra_preiskave")
on delete restrict on update restrict;

alter table "izvid_preiskave"
add constraint "FK_pripada_obravnavi"
foreign key ("stevilka_obravnave")
references obravnava ("stevilka_obravnave")
on delete restrict on update restrict;

alter table "izvid_preiskave"
add constraint "FK_se_prijavlja"
foreign key ("id_samoplacnika")
references samoplacnik ("id_samoplacnika")
on delete restrict on update restrict;

alter table naslov

```

```

add constraint "FK_se_nahaja_v"
foreign key ("postna_stevilka")
references kraj ("postna_stevilka")
on delete restrict on update restrict;

alter table obravnava
add constraint "FK_je_obravnavan"
foreign key ("stevilka_KZZ")
references pacient ("stevilka_KZZ")
on delete restrict on update restrict;

alter table pacient
add constraint FK_pripada
foreign key (naslovID)
references naslov (naslovID)
on delete restrict on update restrict;

alter table "referencna_vrednost_moski"
add constraint FK_referenca
foreign key ("sifra_preiskave")
references preiskava ("sifra_preiskave")
on delete restrict on update restrict;

alter table "referencna_vrednost_zenske"
add constraint FK_referenca2
foreign key ("sifra_preiskave")
references preiskava ("sifra_preiskave")
on delete restrict on update restrict;

alter table "cakalna_vrsta"
add constraint "FK_je_umescen"
foreign key ("stevilka_obravnavave")
references obravnava ("stevilka_obravnavave")
on delete restrict on update restrict;

alter table "cakalna_vrsta"
add constraint "FK_pripada_oddelku"
foreign key ("stevilka_oddelka")
references oddelek ("stevilka_oddelka")
on delete restrict on update restrict;

```

Naloga (e)

Navodila Opišite pet (5) smiselnih netrivialnih operacij nad podanimi tabelami, ter jih implementirajte v obliki SQL poizvedb (vsaka nad najmanj dvema tabelama, vsaj 3 gnezdene ali agregatne poizvedbe).

Poizvedba 1

Opis poizvedbe Poizvedba za vsakega izmed samoplačnikov v bazi izračuna skupno ceno izvedenih preiskav.

SQL poizvedba

```
select
    id_samoplacnika ,
    sum(cena_za_samoplacnika)
from izvid_preiskave
join preiskava p on
    izvid_preiskave.sifra_preiskave = p.sifra_preiskave
group by id_samoplacnika;
```

Poizvedba 2

Opis poizvedbe Poizvedba prikaže osebne podatke vseh pacientov, ki so potrebni za pošiljanje izvidov po pošti.

SQL poizvedba

```
select
    p.ime,
    p.priimek,
select
    p.ime,
    p.priimek,
    n.ulica,
    n.hisna_stevilka,
    n.postna_stevilka
from
    pacient p
inner join naslov n on
    p.naslovID = n.naslovID;
```

Poizvedba 3

Opis poizvedbe Poizvedba prikaže število pacientov, ki so ovrščeni na čakalno listo oddelka. Oddelki so razvrščeni od oddelka z najdaljšo čakalno vrsto do oddelka z najkrajšo.

SQL poizvedba

```
select
    o.ime_oddelka as 'Ime_oddelka',
    count(c.stevilka_obravnave) as 'V_cakalni_vrsti'
from
    cakalna_vrsta c
inner join oddelek o on
    c.stevilka_oddelka = o.stevilka_oddelka
group by c.stevilka_oddelka
order by count(c.stevilka_obravnave) DESC;
```

Poizvedba 4

Opis poizvedbe Poizvedba za vsakega izmed obravnavanih pacientov prikaže število obravnav.

SQL poizvedba

```
select
  p.stevilka_KZZ ,
  count(o.stevilka_KZZ)
from pacient p
inner join obravnava o on
  p.stevilka_KZZ = o.stevilka_KZZ
group by o.stevilka_KZZ;
```

Poizvedba 5

Opis poizvedbe Poizvedba za izbranega pacienta izpiše seznam diagnoz s slovensko kvalifikacijo. Diagnoze so urejene po številki diagnoze.

SQL poizvedba

```
select
  p.stevilka_KZZ ,
  p.ime ,
  p.priimek ,
  dp.stevilka_diagnoze ,
  d.slovenska_kvalifikacija
from diagnoza_pacient dp
inner join diagnoza d on dp.id_diagnoze = d.id_diagnoze
inner join pacient p on dp.stevilka_KZZ = p.stevilka_KZZ
where p.stevilka_KZZ = :izbrani_pacient;
```


3. domača naloga

Naloga (a)

Navodila Definirajte mehanizem, ki bo preprečeval, da bi pacientu v okviru ISTE obravnave ISTI test naredili več kot dvakrat.

```
DELIMITER $$

CREATE TRIGGER ista_obravnavava
BEFORE INSERT
ON izvid
BEGIN

    DECLARE @test int

    SELECT @retVal = COUNT(vrednost)
    FROM izvid
    WHERE st_obravnavave = :obravnavava
           AND ime_preiskave = :preiskava

    IF ( @retVal < 2 ) THEN
    BEGIN
        —DOVOLIMO IN VSTAVIMO NOV IZVID
    END IF

END $$
```

Naloga (b)

Navodila Definirajte operacije (transakcije), ki jih je potrebno izvesti brez prekinjanja. *Baza avtomatsko preveri ali vnos novega podatka ne krši primarnih in tujih ključev. Predvideva se, da so vsi oddelki, vrste preiskav in kode diagnoz v bazi že vnešeni*

Dodajanje novega izvida 1 V kolikor obravnava še ni odprta za danega pacienta moramo kot del transakcije izvesti:

1. Pacientu dodamo novo obravnavo na danem oddelku
2. Pacientu določimo nov izvid (rezultat preiskave)

Dodajanje novega izvida 2 V kolikor pacient v bazi še ne obstaja moramo kot transakcijo izvesti:

1. V bazo dodamo pacienta
2. Pacientu dodamo novo obravnavo na danem oddelku
3. Pacientu določimo nov izvid (rezultat preiskave)

Dodajanje nove diagnoze k obravnavi 1 V kolikor obravnava še ni odprta (vnešena v bazo) moramo kot transakcijo (brez prekinitev) izvesti:

1. V bazo dodamo obravnavo na danem oddelku za danega pacienta
2. Pacientu dodamo novo diagnozo za dano obravnavo

Dodajanje nove diagnoze k obravnavi 2 V kolikor pacient v bazi še ne obstaja moramo kot transakcijo (brez prekinitev) izvesti:

1. V bazo dodamo pacienta
2. V bazo dodamo obravnavo na danem oddelku za danega pacienta
3. Pacientu dodamo novo diagnozo za dano obravnavo

Naloga (c)

Navodila Naredite graf v Excelu, ki prikazuje 10 najpogostejših bolezni (z imeni) po oddelkih in po spolih.

Naloga (d)

Navodila Ugotovite razliko med MKB kodami in WHO ICD kodami (katere so tiste, ki jih drugi ne pozna), ter jih predstavite v Excelovi tabeli urejene glede na število pojavitev med diagnozami.

Kode, ki so del WHO, niso del MKB Tabela prikazuje kode, ki so del nabora kod WHO, vendar niso nabor kod MKB. *Kode, ki med diagnozami nimajo vnosa niso navedene:*

WHO koda	Število diagnoz
N18.8	59
N18.0	20
T28.6	10
T28.7	7
R73.9	4
R50.0	4
T28.5	3
U80.1	2
N08.3	2
R73.0	1
E90	1
J99.0	1
H36.0	1
H82	1
G63.2	1

Kode, ki so del MKB, niso del WHO Tabela prikazuje 20 najpogostejših kod, ki se pojavljajo med diagnozami, ki so del nabora kod MKB, vendar niso nabor kod WHO. Vseh kod, ki so del nabora MKB, vendar ne WHO, je 1146. *Kode, ki med diagnozami nimajo vnosa niso navedene:*

MKB koda	Število diagnoz
I48	1739
J96.0	1148
I50.0	972
F03	383
N40	287
X61	230
I69.4	224
E11.7	208
Z98.8	182
T42.4	180
J45.9	174
E14.9	151
T58	141
I69.3	135
J96.9	134
I25.1	132
I25.0	132
K70.3	130
E11.71	120
Z92.1	106