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 primerjivih operacij v različnih SUPB-jih.

Poglavje 1

Način testiranja

1.1 Izbrani SUPB

Testiranje se bo izvajalo na štirih prosto dostopnih podatkovnih bazah: MySQL, MariaDB, PostgreSQL in Micorsoft SQL Server.

Vse štiri podatkovne baze se bodo izvajale istočasno na Dockerju.

1.1.1 Zagon Docker slik

1.1.2 MySql

```
docker run --name TUP-sem-mysql -p 7202:3306 -e
        MYSQL_ROOT_PASSWORD=root -d mysql:latest
```

Baza je dostopna na naslovu http://localhost:7202

1.1.3 MariaDB

```
docker run --name TUP-sem-mariadb -p 7203:3306 -e
        MYSQL_ROOT_PASSWORD=root -d mariadb:latest
```

Baza je dostopna na naslovu http://localhost:7203

1.1.4 PostgreSQL

```
docker run --name tup-sem-postgres
      -e POSTGRES_USER=root -e POSTGRES_PASSWORD=root
      -p 7200:5432 -d postgres
```

Baza je dostopna na naslovu http://localhost:7200

1.1.5 Microsoft SQL Server

```
docker run -e 'ACCEPT_EULA=Y' --name tup-sem-mssql
   -e 'SA_PASSWORD=root_ROOT' -p 7201:1433
   -d mcr.microsoft.com/mssql/server:2017-CU8-ubuntu
```

Baza je dostopna na naslovu http://localhost:7201.

1.2 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[3]	12.1
Verzija MS SQL Server docker slike[4]	2017
Verzija MariaDB docker slike[5]	10.4.11-bionic

1.3 Način točkovanja testiranja in vrste testov

Vsak test se točkuje z določenim število točk pri čemer je maksimalno število točk določeno po ključu, kjer je glavni faktor pogostost izvajanja določene operacije v realnem svetu *(primer: zdravstveni dom)*.

Testira se vse vidike operacij nad podatki podatkovne baze, katere izvajamo v okviru *CRUD (Create, Read, Update, Delete)* operacij.

[1] https://docs.docker.com/docker-for-windows/install/
[2] https://hub.docker.com/_/mysql
[3] https://hub.docker.com/_/postgres
[4] https://hub.docker.com/_/microsoft-mssql-server
[5] https://hub.docker.com/_/mariadb

Operacija	Maksimalno število točk	Primer uporabe
Vstavljanje[6] podatkov	25	migracija podatkov
Brisanje[7] podatkov	25	praznjenje tabel
Posodabljanje[8] podatkov	50	sprememba KZZ
Branje, analiza podatkov	$4 \cdot 50 = 200$	statistika

1.4 Način izvedbe testov in določitve točk

Pri vsakem testu, ki se bo izvedel večkrat, bomo po definiranem ključu določili zmagovalca, ki v dani kategoriji dobil maksimalno število točk. Naslednje podatkovne baze pa bodo od maksimalnega števila točk dobila odbitek, ki je enak odstotku razlike med lastnim rezultatom in rezultatom zmagovalca kategorije.

V zaključku raziskave se bodo točke seštele in tako objektivno določile performančnega zmagovalca testov. Vseeno pa so točke podeljene zgolj informativne narave, saj se izkaže, da pri izbiri pravega SUPB-ja ni pomembna zgolj performanca, temveč tudi izbira posameznika, naj si bo naročnika, upravljalca ali programerja podatkovne baze. Več o tem sledi v zaključku naloge.

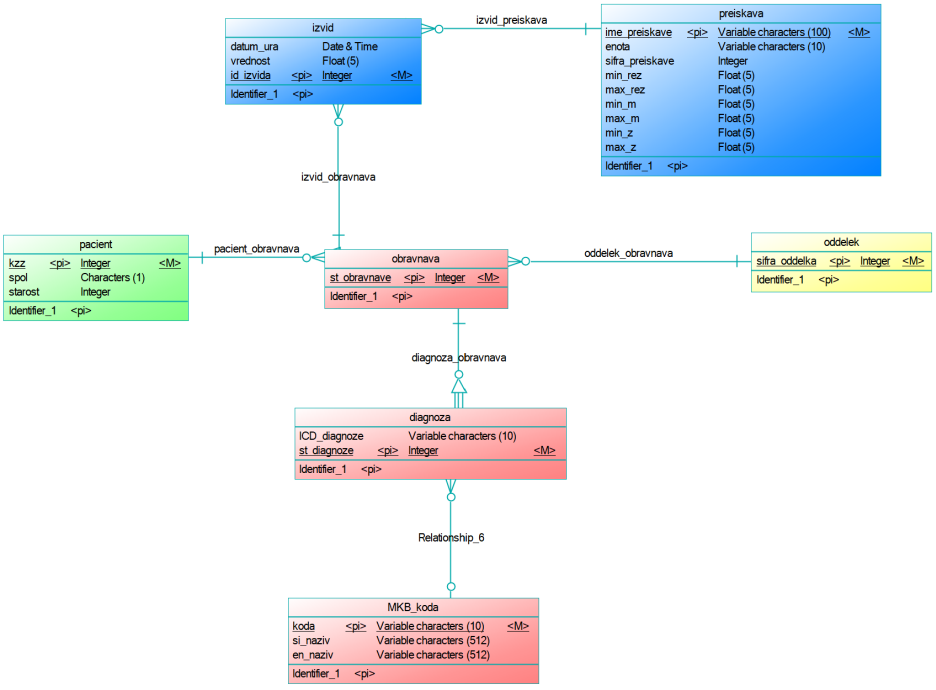
[6] Vstavljanje velike količine podatkov
[7] Brisanje velike količine podatkov
[8] Posodabljanje velike količine podatkov

Poglavje 2

Podatkovna baza - zdravstveni dom

2.1 Konceptni model

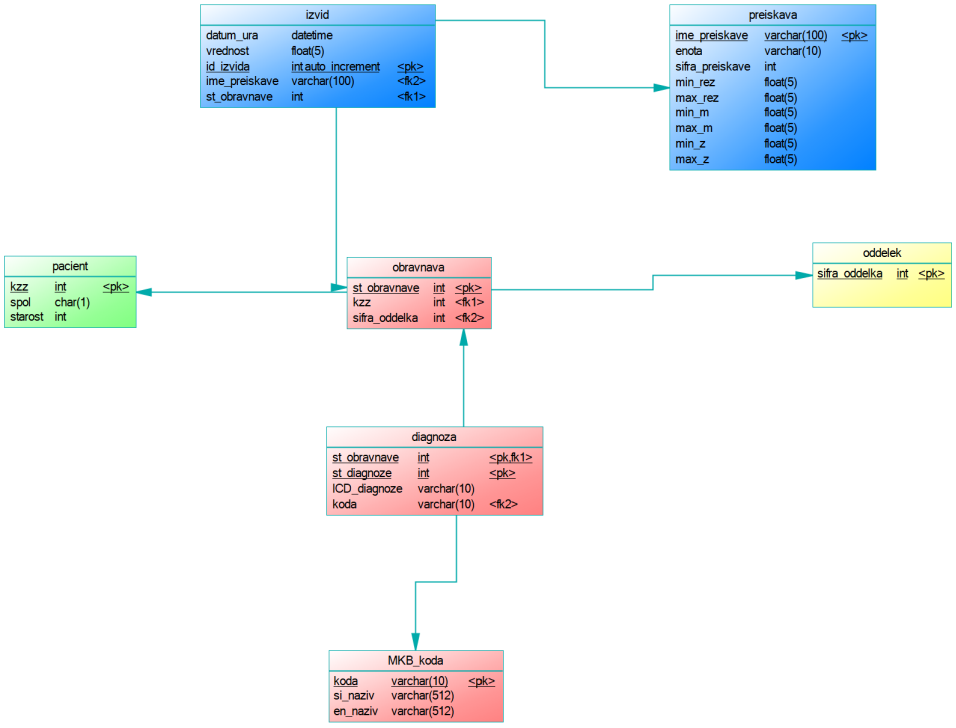
Podatkovna baza za potrebe testiranja je podatkovna baza zdravstvenega doma, ki se je uporabljala pri 3. domači nalogi. Podatkovna baza je razdeljena na 7 tabel in ima skupno 1 028 883 vrstic.



Slika 2.1: Konceptni model podatkovne baze

2.2 Fizični model

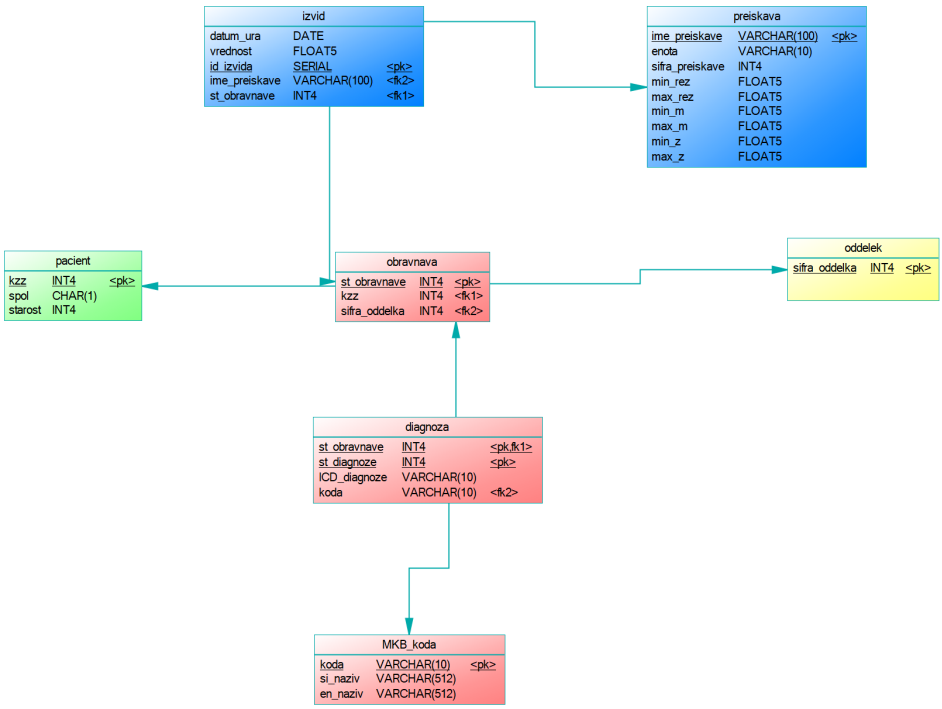
2.2.1 MySQL in MariaDB



Slika 2.2: Fizični model podatkovne baze za MySQL in MariaDB

Skripta za ustvarjanje podatkovne baze je shranjena v datoteki *./sql-skripte/mysql-init.sql*.

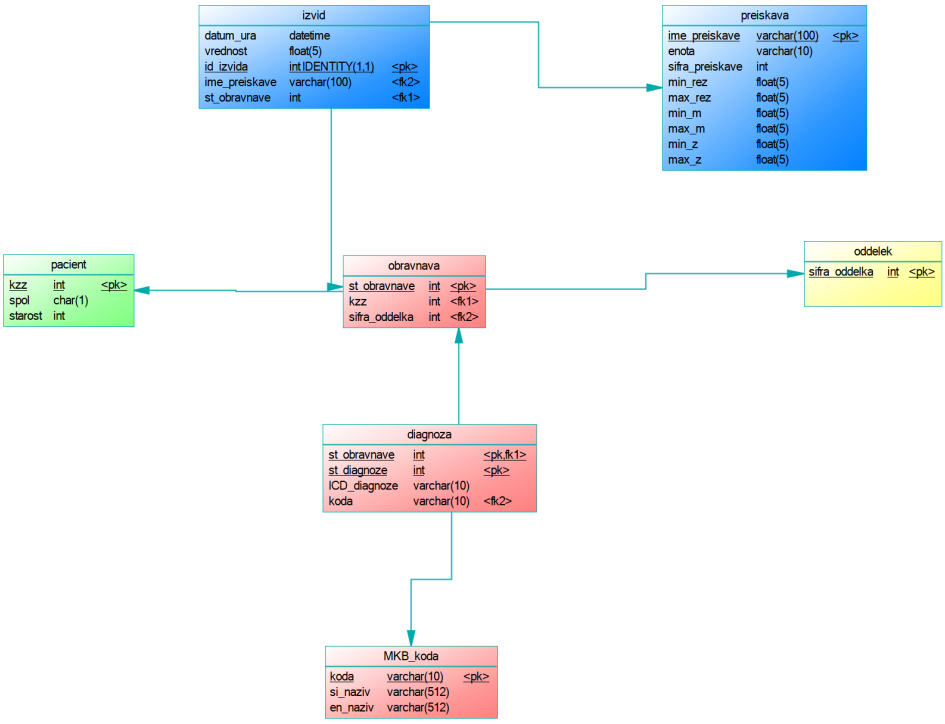
2.2.2 PostgreSQL



Slika 2.3: Fizični model podatkovne baze za PostgreSQL

Skripta za ustvarjanje podatkovne baze je shranjena v datoteki *./sql-skripte/postgres-init.sql*.

2.2.3 Microsoft SQL Server



Slika 2.4: Fizični model podatkovne baze za MS SQL Server

Skripta za ustvarjanje podatkovne baze je shranjena v datoteki *./sql-skripte/mssql-init.sql*.

Poglavje 3

Testiranje

3.1 Vstavljanje (velike količine) zapisov v podatkovno bazo

V podatkovno bazo se je vstavilo skupaj 1 028 883 zapisov. Vstavljanje je potekalo prek Python skripte filanje.py po postopku:

1. Iz csv (coma seperated values) datoteke je program prebral vrstico, jo razdeli in pretvoril v pravilen zapis spremelnjivke (s pomočjo vgrajenih ali pomožnih funkcij)
2. Program shrani trenutni sistemski čas
3. Program izvede INSERT
4. Program od trenutnega sistemskega časa odšteje sistemski čas iz točke 3 in ga prišeje k števcu skupnega časa
5. Program se premakne na naslednjo vrstico csv datoteke
6. Ko program obdela vse datoteke, funkcija vrne celoten seštevek časov izvajanja

Test se je avomatsko izvedel desetkrat na vseh testnih bazah. Po polnjenju vseh treh podatkovnih baz je program podatkovne baze izpraznil pri čemer je bil uporabljen ukaz *DELETE * FROM —ime tabele—* .

Pri obdelavi rezultatov se najboljši in najslabši čas nista upoštevala, iz ostalih pa se je izračunalo povprečje.

SUPB	Najboljši čas [s]	Najslabši čas [s]	Povprečje [s]
MariaDB	482,683	563,107	488,029
PostgreSQL	512,716	520,791	515,921
Microsoft SQL Server	592,745	608,040	596,453
MySQL	619,255	630,775	623,806

Ob tem gre poudariti, da se je pri vseh uporabljal *običajen INSERT* in ne kakšna posebna oblika vstavljanja velike količine podatkov (npr. *BULK INSERT* pri Microsoft SQL Server).

Iz zgoraj navedenih rezultatov se, kot je zapisano v sekciji 1.4.1, določi točke po naslednjem ključu:

SUPB	Rezultat testiranja[ms]	Odstotek točk [%]	Število točk
MariaDB	484 050	100	25
PostgreSQL	515 921	93,8	23,5
Micorsoft SQL Server	596 453	81,2	20,3
MySql	623 806	77,6	19,4

3.2 Brisanje (velike količine) podatkov iz podatkovne baze

Brisanje podatkov je potekalo avtomatsko z uporabo Python skripte *filanje.py*. Testiranje se je izvedlo destkrat.

Testiranje je potekalo po naslednjem postopku:

1. Program za izvede ukaz s katerim pridobi imena vseh tabel v podatkovni bazi
2. Program za vsako izmed tabel izvede:
3. Program shrani trenutni sistemski čas
4. Program izvede ukaz *DELETE FROM —ime-tabele—*
5. Program od trenutnega sistemskega časa odšteje shranjen čas
6. Program vrne seštevek vsote časov izvajanja funkcije brisanja podatkov

Rezultati izračunani iz povprečja so naslednji:

SUPB	Testiranje [ms]	Odstotek točk [%]	Število točk
Microsoft SQL Server	1087	100	25
PostgreSQL	1490	72,9	18,2
MariaDB	2418	45,0	11,2
MySQL	5373	20,2	5,1

Iz zgornjih rezultatov lahko sklepamo, da je brisanje podatkov iz podatkovne baze veliko hitrejši postopek od vstavljanja. Ravno zaradi tega so relativne razlike med testiranimi SUPB-ji izjemno velike v primerjavi z razlikami pri vstavljanju, čeprav so absolutne razlike (predvsem med MS SQL Server in PostgreSQL) zanemarljive.

3.3 Branje podatov iz podatkovne baze

3.3.1 Povezovanje tabel in primerjava med *WHERE* in *INNER JOIN*

SQL poizvedba Poizvedba za vsakega izmed pacientov prikaže pacientovo številko zdravstvenega zavarovanja, spol, številko obravnave ter kakšna vrsta preiskave je bila opravljena in njeno vrednost. V testiranju sta bili uporabljeni dve različni poizvedbi ena z uporabo *INNER JOIN* (poizvedba 1) in druga z uporabo *WHERE* stavka (poizvedba 2).

```
Poizvedba 1:
   select
      p.kzz,
      p.spol,
      o.st_obravnave,
      i.ime_preiskave,
      i.vrednost
   from pacient p
   inner join obravnava o on o.kzz = p.kzz
   inner join izvid i on i.st_obravnave=o.st_obravnave
   order by p.kzz, o.st_obravnave
```

```
Poizvedba 2:
    select
        p.kzz,
        p.spol,
        o.st_obravnave,
        i.ime_preiskave,
        i.vrednost
    from
         pacient p,
         obravnava o,
         izvid i
    where
        p.kzz = o.kzz and
        o.st_obravnave = i.st_obravnave
    order by p.kzz, o.st_obravnave
```

Rezultati poizvedbe 1 (*INNER JOIN*) Test se je izvedel desetkrat, avtomatsko z uporabo Python skripte *poizvedbe.py* za vse štiri podatkovne baze.

SUPB	Najboljši [ms]	Najslabši [ms]	Povprečje [ms]
Microsoft SQL Server	312	361	334
MySql	701	838	725
PostgreSQL	879	1040	925
MariaDB	1188	1374	1237

Primerjava hitrsoti izvedbe SQL stavka z uporabo *INNER JOIN*

Rezultati poizvedbe 1 (*WHERE*)

SUPB	Najboljši [ms]	Najslabši [ms]	Povprečje [ms]
Microsoft SQL Server	303	375	336
MySql	702	729	710
PostgreSQL	885	973	909
MariaDB	1204	1223	1212

Primerjava hitrsoti izvedbe SQL stavka z uporabo *WHERE* Razlike med

hitrostjo poizvedb pri uporabi *WHERE* in *INNER JOIN* so pri vseh štirih bazah zanemarljive:

SUPB	***INNER JOIN* [ms]**	***WHERE* [ms]**	**Razlika [%]**
Microsoft SQL Server	334	336	0,5
MySql	725	710	-2,1
PostgreSQL	925	909	-1,8
MariaDB	1237	1212	-2,1

Prikaz razlike v povprečni hitrosti izvedbe SQL poizvedbe z uporabo *INNER JOIN* in *WHERE* stavka.

Določitev točk Čeprav se poizvedba deli na dve sintaktično različni, je razultat obeh poizvedb identičen. Kot takega ga torej štejemo kot eno postavko izračuna točk, pri čemer se upošteva boljši rezultat obeh poizvedb.

SUPB	**Testiranje**	**Odstotek točk [%]**	**Število točk**
Microsoft SQL Server	334	100	50
MySql	710	47,0	23,5
PostgreSQL	909	36,7	18,4
MariaDB	1212	27,6	13,8

3.3.2 Uporaba funkcije *COUNT* v povezanih tabelah in predpomnjenje

Test se je izvedel desetkrat, avtomatsko z uporabo Python skripte *poizvedbe.py* za vse tri podatkovne baze.

SQL poizvedba Poizvedba izpiše ime preiskave in število izvidov, katerih vrednost je višja od priporočene vrednosti glede na spol. Izpis je urejen padajoče po številu prekoračenih izvidov.

```
select
    i.ime_preiskave,
    count(i.ime_preiskave)
from
    izvid i
inner join obravnava o on
        i.st_obravnave = o.st_obravnave
inner join pacient p
        on p.kzz = o.kzz
inner join preiskava p2
        on i.ime_preiskave = p2.ime_preiskave
where
    (p.spol = 'M' and p2.max_m <= i.vrednost
        and p2.max_m is not null)
    or
    (p.spol = 'Z' and p2.max_z <= i.vrednost
        and p2.max_z is not null)
group by i.ime_preiskave
order by count(i.ime_preiskave) desc
```

SUPB	Najboljši [ms]	Najslabši [ms]	Povprečje [ms]
MariaDB	1	1166	118
PostgreSQL	314	416	362
Microsoft SQL Server	490	584	518
MySql	1534	2567	1784

SUPB	Testiranje	Odstotek točk [%]	Število točk
MariaDB	118	100	50
PostgreSQL	362	32,4	16,2
Microsoft SQL Server	518	22,7	11,2
MySql	1704	6,5	3,4

V tem testu se je prvič videla moč predpomnjenja, ki ga uporablja MariaDB, ki si shranjuje rezultate poizvedbe. Baza naslednjič servira že predpomnjene podatke in ne izvaja SQL poizvedbe ponovno. To je uporabno predvsem v podatkovnih bazah, kjer je pogostejše branje iz baze, kot je pisanje v njo.

3.3.3 Uporaba funkcij *COUNT, MAX, MIN, AVG*

Test se je izvedel desetkrat, avtomatsko z uporabo Python skripte *poizvedbe.py* za vse štiri podatkovne baze.

SQL poizvedba Poizvedba izpiše število opravljenih preiskav, z minimalno, maksimalno in povprečno vrednostjo, kjer je število izvedb večje od 200 in je povprečna vrednost večja od polovice maksiamlne vrednosti.

```
Select
    ime_preiskave,
    count(vrednost),
    min(vrednost),
    max(vrednost),
    avg(vrednost)
from izvid
group by ime_preiskave
having
    count(vrednost) > 200 and
    avg(vrednost) * 2 <= max(vrednost)
order by count(vrednost) desc, ime_preiskave asc
```

SUPB	Najboljši [ms]	Najslabši [ms]	Povprečje [ms]
MariaDB	1	622	63
PostgreSQL	237	307	362
Microsoft SQL Server	230	316	265
MySql	898	984	927

SUPB	Testiranje	Odstotek točk [%]	Število točk
MariaDB	118	100	50
PostgreSQL	362	24,9	12,5
Microsoft SQL Server	518	23,7	11,8
MySql	1704	6,8	3,4

Zopet ima veliko prednost MariaDB, predvsem zaradi predpomnjenja, v kolikor le tega ne bi uporabljali (ali bi se med branji zgodilo novo pisanje) bi čas poizvedbe bil primerjiv najslabšemu rezultatu v testiranju,

3.3.4 Gnezdene SQL poizvedbe

Test se je izvedel desetkrat, avtomatsko z uporabo Python skripte *poizvedbe.py* za vse štiri podatkovne baze.

SQL poizvedba Poizvedba poišče vse diagnoze v katerih se nahaja ključna beseda, v gnezdenih skriptah pridobimo najmaljšo, najstarejšo in povprečno starost pacientov z iskano diagnozo.

```
select m.si_naziv, count(o.st_obravnave) as Stevilo,
(
    select min(p1.starost)
        from
            pacient p1,
            obravnava o1,
            diagnoza d1,
            MKB_koda m1
        Where
            p1.kzz = o1.kzz and
            d1.st_obravnave = o1.st_obravnave and
            d1.icd_diagnoze = m1.koda and
            m1.si_naziv = m.si_naziv
) as min,
(
    select max(p1.starost)
        from
            pacient p1,
            obravnava o1,
            diagnoza d1,
            MKB_koda m1
        Where
            p1.kzz = o1.kzz and
            d1.st_obravnave = o1.st_obravnave and
            d1.icd_diagnoze = m1.koda and
            m1.si_naziv = m.si_naziv
) as max,
(
    Select avg(p1.starost)
        from
            pacient p1,
            obravnava o1,
            diagnoza d1,
            MKB_koda m1
        Where
            p1.kzz = o1.kzz and
```

```
                d1.st_obravnave = o1.st_obravnave and
                d1.icd_diagnoze = m1.koda and
                m1.si_naziv = m.si_naziv
    ) as povprecje
from pacient p
    inner join obravnava o
        on p.kzz = o.kzz
    Inner join diagnoza d
        on o.st_obravnave = d.st_obravnave
    inner join MKB_koda m
        on m.koda = d.icd_diagnoze
    inner join izvid i
        on d.st_obravnave = i.st_obravnave
    inner join preiskava p2
        on i.ime_preiskave = p2.ime_preiskave
    where
        p.starost between 21 and 40 and
        m.si_naziv like '%tr_en%'
    group by m.si_naziv, p.starost
    order by count(o.st_obravnave) desc
```

SUPB	Najboljši [ms]	Najslabši [ms]	Povprečje [ms]
Microsoft SQL Server	374	568	407
PostgreSQL	7045	7243	7100
MariaDB	1	164 669	16 468
MySql	139 951	144 340	141 129

SUPB	Testiranje	Odstotek točk [%]	Število točk
Microsoft SQL Server	407	100	50
PostgreSQL	7100	5,8	2,9
MariaDB	16 486	2,4	1,2
MySql	1704	0,1	0,1

Test preverja predvsem hitrost izvedbe gnezdenih SQL poizvedb in primerjave nizov znakov. Izkaže se, da Microsoft SQL server uporablja veliko bolj optimizirano proceduro za izvajanje takšne poizvedbe. Hkratu se zopet izkaže kakšno moč ima predpomnjenje, ki ga uporablja MariaDB, vendar je ob tem potrebno predpostaviti, da se med samimi bralnimi poizvedbami ne izvede pisanje (nov zapis ali posodabljanje), zaradi česar bi predpomnjen rezultat postal neveljaven.

3.4 Posodabljanje zapisov v podatkovni bazi

SQL poizvedba Poizvedba poišče vse diagnoze v katerih se nahaja ključna beseda, v gnezdenih skriptah pridobimo najmaljšo, najstarejšo in povprečno starost pacientov z iskano diagnozo.

```
update preiskava
   set min_z = min_m, max_z = max_m
where min_z is null and max_z is null
```

SUPB	Vstavljanje [ms]	Commit [ms]	Skupaj [ms]
MySql	3	2	5
Microsoft SQL Server	7	3	10
PostgreSQL	11	10	21
MariaDB	15	9	24

SUPB	Testiranje	Odstotek točk [%]	Število točk
MySql	5	100	50
Microsoft SQL Server	10	50	25
PostgreSQL	21	23,8	11,9
MariaDB	24	20,8	10,4

Poglavje 4

Zaključek

4.1 Seštevek točk performančnih testov

Test	MSSQL Server	MariaDB	PostgreSQL	MySQL
Vstavljanje	20,3	25	23,5	19,4
Brisanje	25	11,5	18,2	5,1
Branje 1	50	18,4	13,8	23,5
Branje 2	11,2	50	16,2	3,4
Branje 3	11,8	50	12,5	3,4
Branje 4	50	1,2	2,9	0,1
Posodabljanje	25	10,4	11,9	50
Vsota	193,3	166,5	99	104,9

Razumevanje rezultatov performančnih testov Izvedeni testi v okviru seminarkse naloge zagotavljajo primerjivo testno okolje za vse podatkovne baze, vendar ne zagotavljajo optimalne konfiguracije baz. Ravno tako v testih niso uporabljene poseben inačice SQL jezika, ampak se (kjer je to mogoče) uporablja splošno sprejeta oblika SQL jezika *(ne uporablja se posebnih ukazov kot so INSERT IGNORE ali BULK INSERT)*, ravno tako so rezultati močno podvrženi sistematičnsti testov, ki omogočajo višjo performanso podatkovnih baz z vgrajenim predpomnjenjem (izrazito MariaDB).

4.2 Izbira SQL podatkovne baze

Dasiravno je in tudi bo izbira podatkovne baze odvisna od želja in izkušenj programerjev, naročnikov in uporabnikov informacijske rešitve, lahko iz performančnih testov potegnemo nekaj zaključkov.

4.2.1 Microsoft SQL Server

Microsoft SQL Server ima izmed 4 testiranih najboljšo performanco, saj je poleg hitrega vstavljanja in brisanja, povprečno hiter tudi pri vstavljanju podatkov v tabelo. Najboljšo optimiziranost pa zagotavlja tudi pri branju podatkov z uporabo gnezdenih SQL poizvedb in poizvedb z uporabo agregatnih funkcij, brez prevelikega zanašanja na predpomnjenje.

Največja težava MSSQL Serverja je njegov SQL jezik, ki se predvsem pri kreiranju in brisanju tabel močno razlikuje od ostalih testiranih. V velikem produkcijskem okolju moramo tudi predvideti možnost, da brezplačna verzija ne bo omogočala dovolj funkcij, ki so potrebne pri takšni vrsti sistemov.

4.2.2 MySQL

Čeprav je MySQL "de-factoštandard v industriji se je (presenetljivo) izkazal za najmanj optimiziranega. Edina kategorija v kateri je izstopal je bila posodabljanje podatkov.

Pri naslednjem projektu, je zato mogoče bolje razmisliti o smiselni alternativi.

4.2.3 MariaDB

MariaDB je pogostokrat označena kot mlajša sestra MySQL (kar tudi je), vendar se je v testih izkazala kot boljša alternativa, saj je konstantno dosegala performančno boljše rezultate.

Zaradi svojega velikega zanašanja na predpomnilnik je najboljša izbira za baze, kjer se zapisi dogajajo dokaj redko, veliko pa je ponovljivih branj. Zaradi pododbnosti zgradbe z MySQL je prehod med eno in drugo instanten, saj uporabljata (večino) enake gonilnike kakor tudi SQL jezik.

4.2.4 PostgreSQL

PostgreSQL se izkaže za nekoliko slabšo alternativo vsem zgoraj omenjenim podatkovnim bazam, vendar vseeno zagotavlja povprečno performanco.

PostgreSQL lahko predstavlja odprtokodno alternativo MySQL/MariaDB in je ravno zaradi svoje "objektne-orientacije"pogosto smiselna podporna baza za aplikacije spisane v objektno-orientiranem modelu.

4.3 Zaključek

Izbira SUPB-ja bo še naprej ostala primarno odvisna od kompetenc naročnika-izvajalca, vendar se vseeno zdi, da ima vsaka podatkovna baza svoje prednosti in slabosti, ki niso nujno povezane (samo) z golo performanco.

Programerji pa si lahko kot izziv vedno znova pri svojih projektih izbirajo drugo rešitev, ki jim omogoča, da spoznajo alternative kakor tudi možnosti, ki jih alternative ponujajo.