**Testiranje**

**Točkovanje**

Vsak test se točkuje z določenim število točk pri čemer je maksimalno število točk določeno po ključu.

Vstavljanje (velike količine podatkov): običajna praksa uporabe podatkovne baze vsebuje neprestano vstavljanje podatkov v le-to. Vendar je vstavljanje enega ali več (do nekaj 10) podatkov zanemarljivo hitro in kot tako ni smiselno za testiranje.

V nadaljevanju torej ne testiramo sprotnega vstavljanja v bazo, temveč zgolj polnjenje baze s predpripravljenimi podatki iz csv datotek.

Takšno vstavljanje je v praksi redko, v večini primerov uporabljano zgolj pri migraciji podatkov iz ene v drugo bazo in se zato po našem ključu določi maksimalno število točk 25.

Brisanje velike količine podatkov je v praksi ravno tako redko uporabljeno, zato se v našem ključu določi maksimalno število točk 25.

Z maksimalnim število točk se oceni tudi skupen rezultat kreiranja in brisanja vseh tabel baze (create in drop).

Posodabljanje velike količine podatkov sicer ni najpogostejša praksa, je pa vseeno pogostejša od vstavljanja ali brisanja (velike količine podatkov). v primeru zdravstvenega doma bi takšen primer lahko predstavljala reforma kartic zdravstvenega zavarovanja pri čemer se vsem obstoječim številkam kartic na začetek doda vodilno 0.

Kot takšno, je posodabljanje v ključu vredno več od vstavljanja/brisanja in ima določeno maksimalno število točk 50.

Bralni SQL ukazi so najpogostejši ukazi nad podatki v podatkovni bazi. S pomočjo le teh opravljamo iskalne poizvedbe, pregledujemo podatke za nazaj ter izvajamo statistične vpoglede v podatke.

Običajno rezultatov iskalnih in statističnih poizvedb ne shranjujemo, vendar jih izvajamo sproti. To pomeni, da se pri 10 zaposlenih v zdravstvenem domu lahko pojavi tudi nekaj deset in več poizvedb na sekundo (aplikacija lahko omogča na istem zaslonu prikaz najpogosteje opravljanih preiskav, najpogostejše diagnoze, omogoča iskanje po kodah diagnoz,...).

Kot takšne vsako od teh poizvdb definiramo z maksimalnih številom 50 točk, do maksimalne vsote števila točk 200.

Pri vsakem testu, ki se bo izvedel večkrat, bomo po definiranem ključu določili zmagovalca, ki v dani kategoriji dobil maksimalno število točk. Naslednji dve podatkovni bazi pa bosta 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.

**Kreiranje podatkovne baze**

**Vstavljanje v podatkovno bazo**

V podatkovno bazo se je vstavilo skupaj **xxx** 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 na vseh treh testnih bazah desetkrat v vrstnem redu: Microsoft SQL Server, MySql in PostgreSQL. Po vrsti je program polnil tabele:

1. Pacient
2. Oddelek
3. Obravnava
4. MKB\_koda
5. Dignoza
6. Preiskava
7. Izvid

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.

Iz rezultatov je razvidno, da sta MS SQL Server in MySql približno enako hitra pri vstavljanju, pri čemer ima MS SQL Server manjšo prednost. Na drugi strani je PostgreSQL v povprečju za skoraj xxx % hitrejši.

Iz zgoraj navedenih rezultatov se izračunajo točke: