**Sveučilište Josipa Jurja Strossmayera**

**u Osijeku**

**Odjel za Matematiku**

**Moderni sustavi baza podataka**

**Baza podataka za Trgovinu Računalne**

**Opreme**

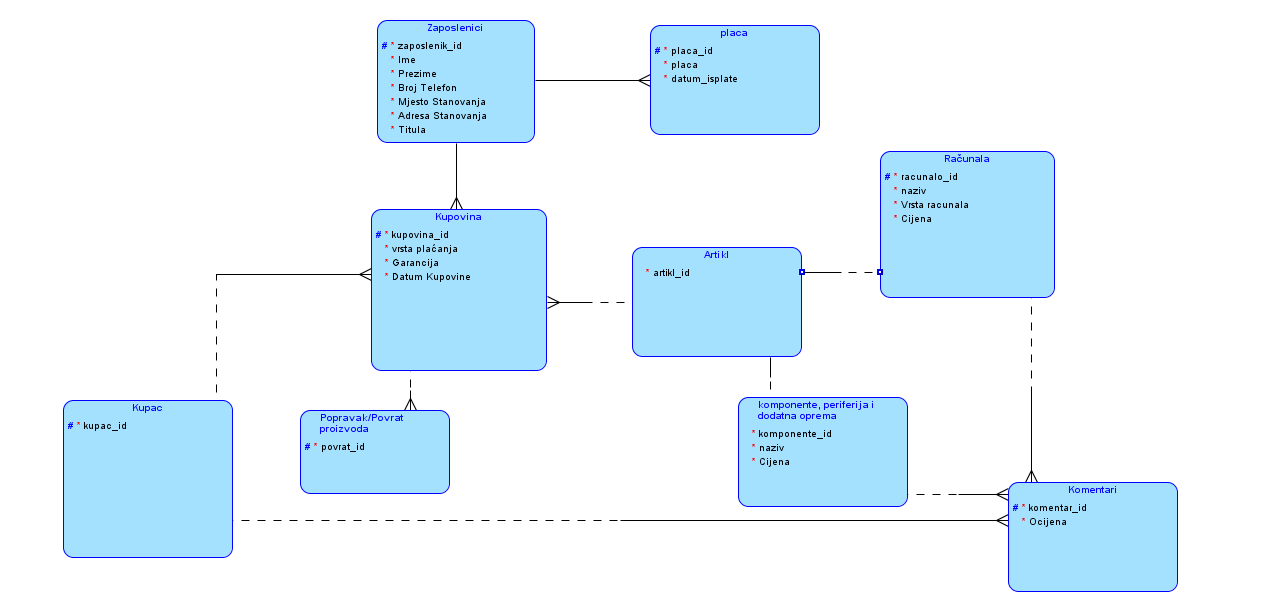
**Stjepan Stojčević**

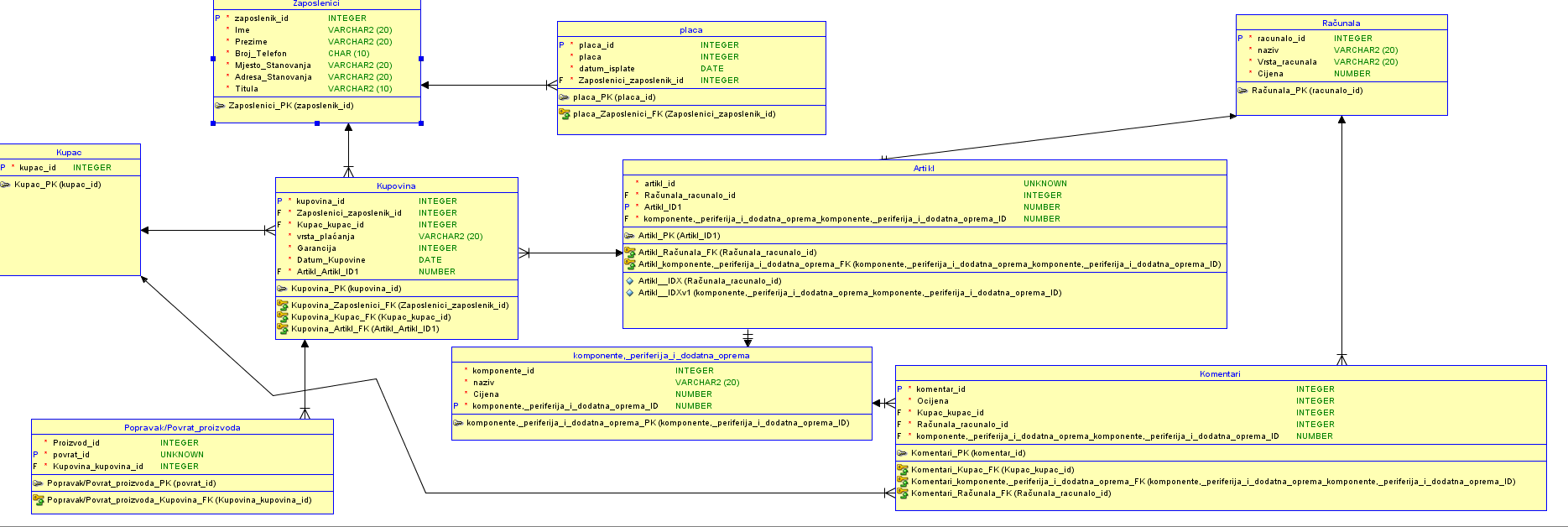
**Seminar**

**UVOD**

Za temu svojeg projekta uzeo sam trgovinu računalne opreme jer kao računarac volim znati o računalima, koje su komponente najnovije, koje se najbolje prodaju i slično. Te sam smatrao da će mi projekt biti zabavan, a osim toga da i ima zanimljivih podataka za obraditi i pokazati.

**MEV I RELACIJSKI MODEL**

****

****

**Kupac:** Tablica kupac sadrži samo podatak kupac\_id koji je dovoljan da pratimo koliko kupaca smo imali jer ćemo ga zapisivati kao integer pa ce najveci broj ujedno biti i broj kupaca, te će biti dovoljan da pratimo što je koji kupac uzeo. Također spajamo kupca s one to many vezom s kupovinom, jer svaki kupac može imati više kupovina, no svaka kupovina mora imat kupca i one to many s komentarima jer ako je jedan kupac kupio više proizvoda može dati komentar svakom od njih, no svaki komentar ima točno jednog usera(kupca).

**Zaposlenici:** Kod zaposlenika pratimo ime, prezime, telefon i slične podatke koje nam bitne kao poslodavcu da imamo da mozemo pristupiti svojim gostima, također svaki gost ima svoju plaću koja se može mjenjati iz mjeseca u mjesec, te smo zbog toga povezali zaposlenika s one to many s plaćom jer zaposlenik mora imati barem jednu plaću, ali svaka plaća ima točno jednog zaposlenika, također smo spojili zaposlenika s kupovina s one to many vezom jer jedan zaposlenik može obraditi više kupovina, dok svaku kupovinu može obraditi samo jedan zaposlenik

**Računala i komponente:** Računala i komponente su vrlo slične tablice i imaju vrlo slične karakteristike. Oboje su spojeni s one to many vezom s komentarima jer svako računalo ili komponenta može imati više ocijena i također su spojeni s one to one vezom s artiklom jer svako računalo i svaka komponenta može biti kao jedan artikl u izlogu. Naziv i cijena su jasna, no u vrstu računala i vrstu komponente spadaju stvari kao “stolno računalo”, “laptop”, “procesor”, “grafička kartica”, itd.

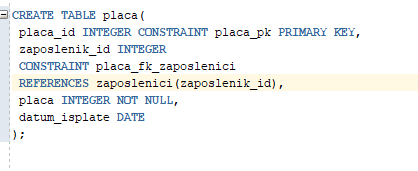
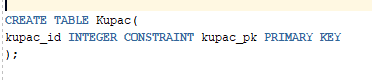
**Artikl:** Artikl nam sadrži sva računala i komponente koje naša trgovina sadrži, te je spojen s one to many vezom s kupovinom jer svaki proizvod se može više puta kupiti, ali sam napravio tako da svaka kupovina može imati jedan artikl

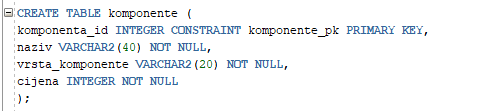
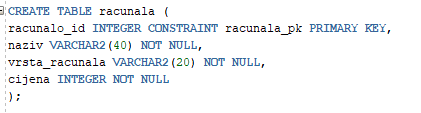
**Kupovina:** Kupovina je prije svega posredni entitet koji razbija many to many vezu između artikla i kupca na dvije one to many veze, no kupovina sadrži podatke kao datum kupovine, garanciju i vrstu plaćanja koje služe kao dodatna informacija za svaki prodani proizvod. Kupovina je spojena s one to many vezom s povratom proizvoda jer svaki artikl može imati više kvarova no svaki zaseban povrat proizvoda može imati samo jedan proizvod.

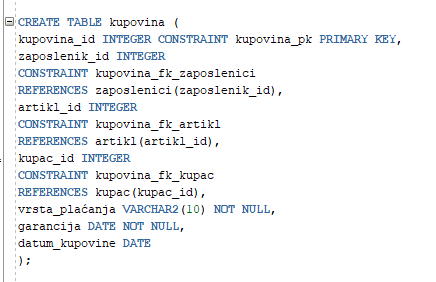
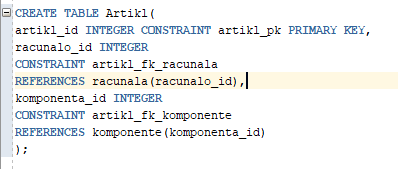
**Komentari:** Komentari služe da budući kupci znaju što je preporučeno i što je dobar proizvod a što ne.

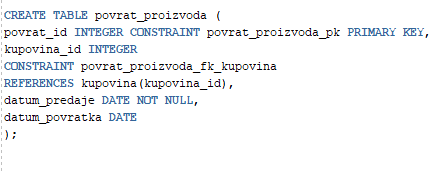
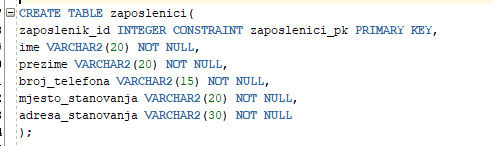
**TABLICE**

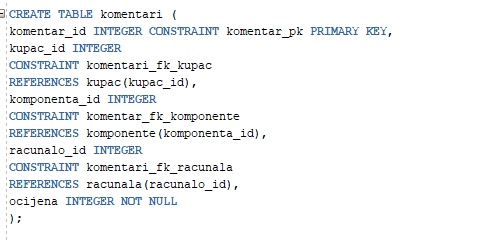
Tablice su u potpunosti kreirane na principu na MEV i relacijski model, svi su inserti poslani zajedno sa seminarom te nema potrebe ništa posebno komentirati.

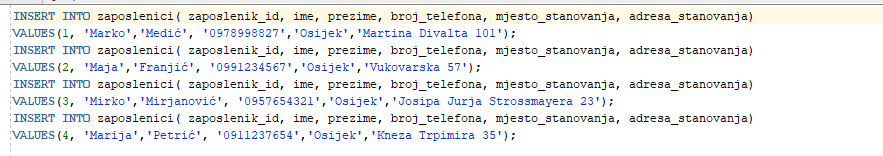
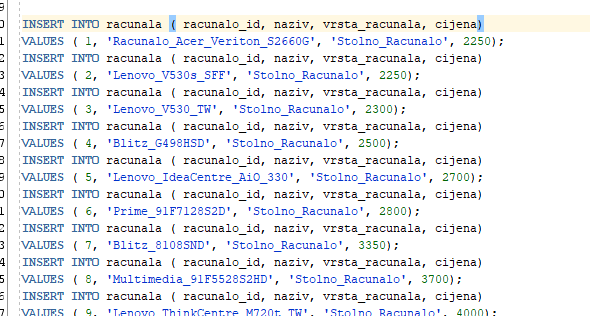
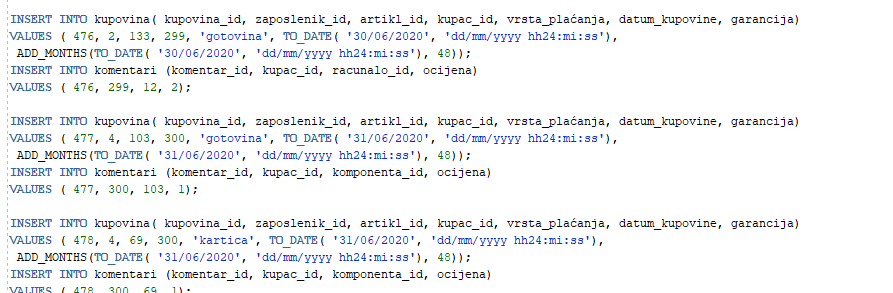
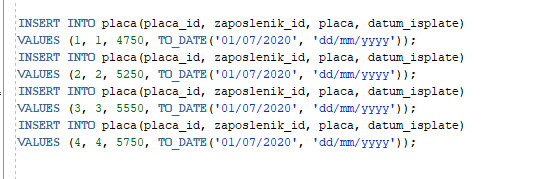
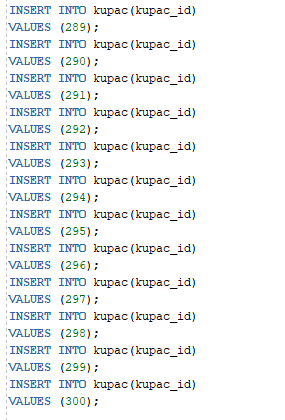
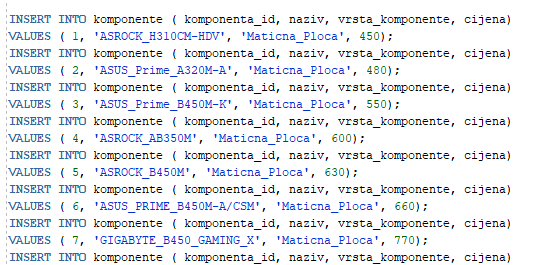




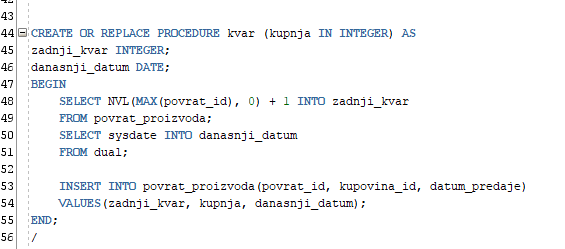






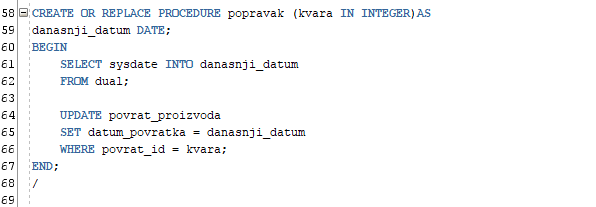


**PROCEDURA UNOSA**



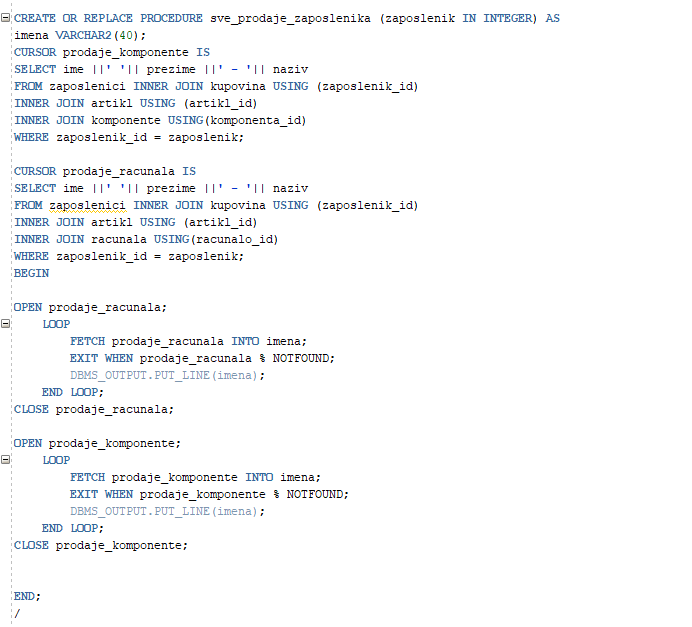
Jednostavna procedura unosa, prima kupnja\_id kao argument funkcije i stavlja ju u tablicu popravka s danasnjim datumom.

**PROCEDURA UPDATE-A**

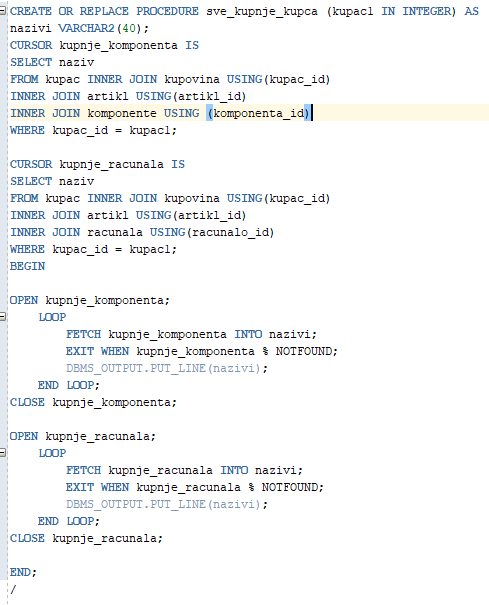


Prosljeđujemo proceduri kvar\_id koji smo u prijašnjoj proceduri uveli te postavljamo datum\_povratka na današnji dan, odnosno dan kada je kupac došao i preuzeo proizvod

**PROCEDURE ISPISA**

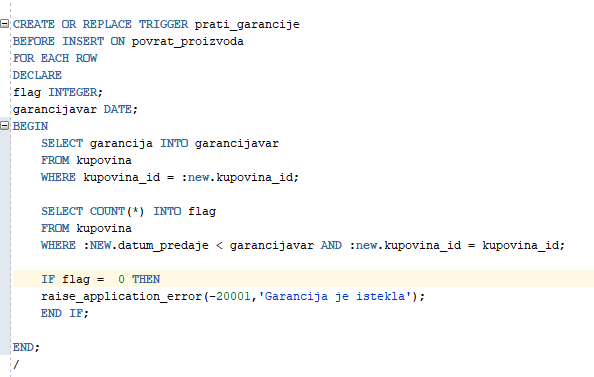
****

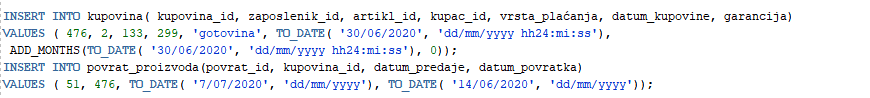
Procedura prima zaposlenika\_id kao argument te radi upit koji traži imena, prezimena i naziv prvo komponente pa onda računala koje su prodali, (ti podaci su lako dostupni iz tablice kupovina) te se JOINA sve do tablice računala odnosno komponente kako bi došli do naziva tih artikla te ih sprema u kursore koje kasnije ispisuje.

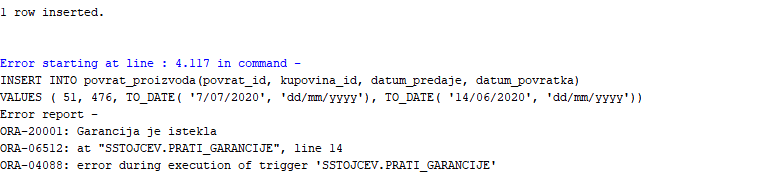


Procedura jako slična prijašnjoj osim što se ovdje za razliku od prošle ne traže prodaje već kupovine, podatak koji je isto vrlo lagano dostupan iz tablice kupovina te ih također sprema u kursore koje kasnije ispisuje

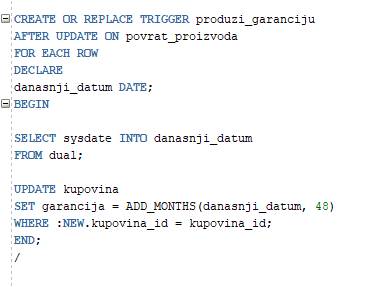
**TRIGGERI**



Trigger prati\_garanciju se uvijek okida prije inserta u povrat\_proizvoda, on gleda da artikl koji se treba popraviti još uvijek pod garancijom tako da gleda doad mu garancije traje i gleda kada je predan i u flag spremamo jeli taj proizvod jos uvijek pod garancijom ako je flag = 0 onda znaci da nije



Primjer rada triggera, znači kupovina je spremljena s garancijom do 30/06/2020, no mi smo htjeli unjeti onda tu kupovinu u povrat\_proizvoda 07/07/2020, no onda nam se javi error “garancija je istekla”



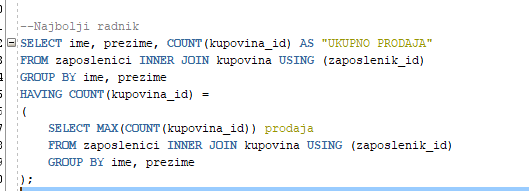
Trigger produzi\_garanciju se javlja nakon update-a na tablici povrat\_proizvoda, a znamo da cemo updateati povrat\_proizvoda u proceduri popravak, odnosno ovaj trigger će se okinuti jedino kad se pokvareni proizvod popravi i vrati vlasniku te će nakon što se artikl vrati ovaj trigger produžiti garanciju tog artikla.

**INDEKS**

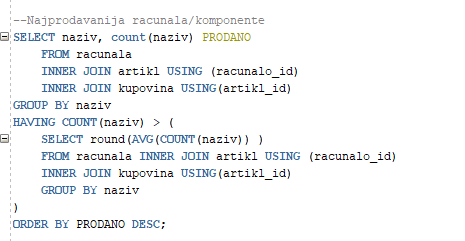


Napravio sam indekse na naziv i vrste komponente i racunala jer njih najviše pozivam u upitima te također u procedurama ispisa tražim naziv računala i komponenti, također indeksi su dobri na podacima koji se ne mijenjaju i ne dodaju često, a recimo iako mogu imati nove proizvode u trgovini te nove vrste opreme, to nisu toliko česte promjene da bi moji indeksi bili problem.

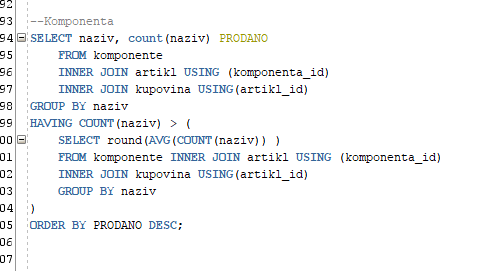
**UPITI**



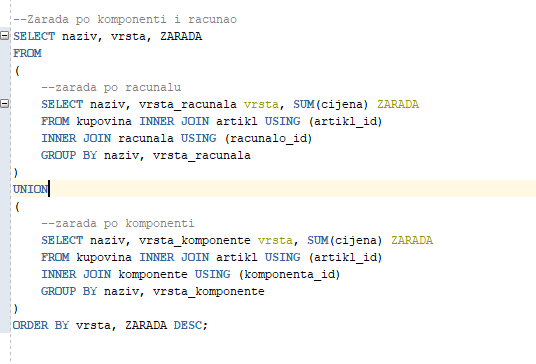
Upit vraća imena i prezimena te njihov zaseban broj prodaja najboljih radnika tako što nađe najboljeg prodavaca i onda trazi sve prodavace koje imaju isto toliko prodaja



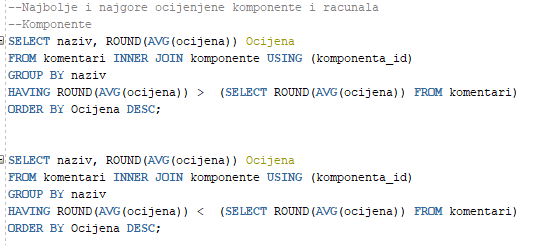
Upit vraća najprodavanija računala tako što izračuna prosjek prodaje svih računala i onda gleda ona koja se prodaju više od prosjeka



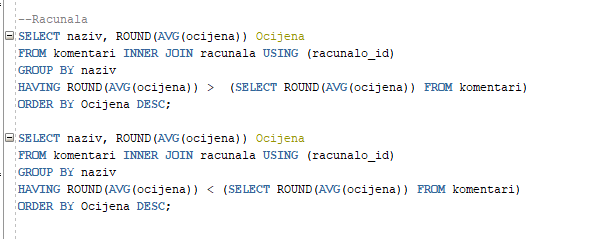
Ovaj upiti je isti kao prijašnji samo što se ne gleda prosjek računala već komponenata



Upit koji vraća svu zaradu svakog pojedinačnog računala i komponente tako što prvo izračuna zaradu svakog računala, pa zaradu svake komponente i onda kroz skupovnu operaciju ‘UNION’ ih spoji sve u jedan upit, slično smo mogli napraviti i s prošla dva upita da smo htjeli vidjeti najprodavanije komponente i računala zajedno



Upit koji vraća najbolje (i najgore) ocijenjenu komponentu tako što gleda prosijek svih ocijena iz tablice komentari i vraća sve komponente koje imaju u prosijeku veće ocijenjenu komponentu odnosno u prosijeku manje ocijenjenu komponentu



Upit identičan kao i prošli osim što se ovaj bazira na računala a ne komponente



Upit koji vraća najlosiji proizvod po pitanju kvarova tako što napravi union između upita najgorih komponenta po kvarovima i upita najgorih računala po kvarovima, koje koje izračuna tako što izbroji koliko koji proizvod ima komponenata te uzme maximum tog broja i ispiše sva računala odnosno komponente koje imaju toliko kvarova