Sveučilište Josipa Jurija Strossmayera u Osijeku,

Odjel za matematiku,

Preddiplomski studij Matematika i računarstvo

Moderni sustavi baza podataka

Playstation online service

Profesor: Student:

doc.dr.sc. Slobodan Jelić Krunoslav Vrkljan

Asistenti:

Ena Pribisalić

Mateja Đumić

Osijek,srpanj 2020.

**1.Uvod:**

Playstation online service je sustav koji povezuje sve korisnike playstationa te koji omogućuje online kupovanje igara te pretplate na streaming siteove. Preko toga se mogu organizirati zajednice koje povezuju ljude sa sličnim interesima te pravljenje događaja poput turnira i zajedničkog igranja igrica. Sustav omogućuje spremanje kartice radi lakšeg i bržeg nacina plaćanja.

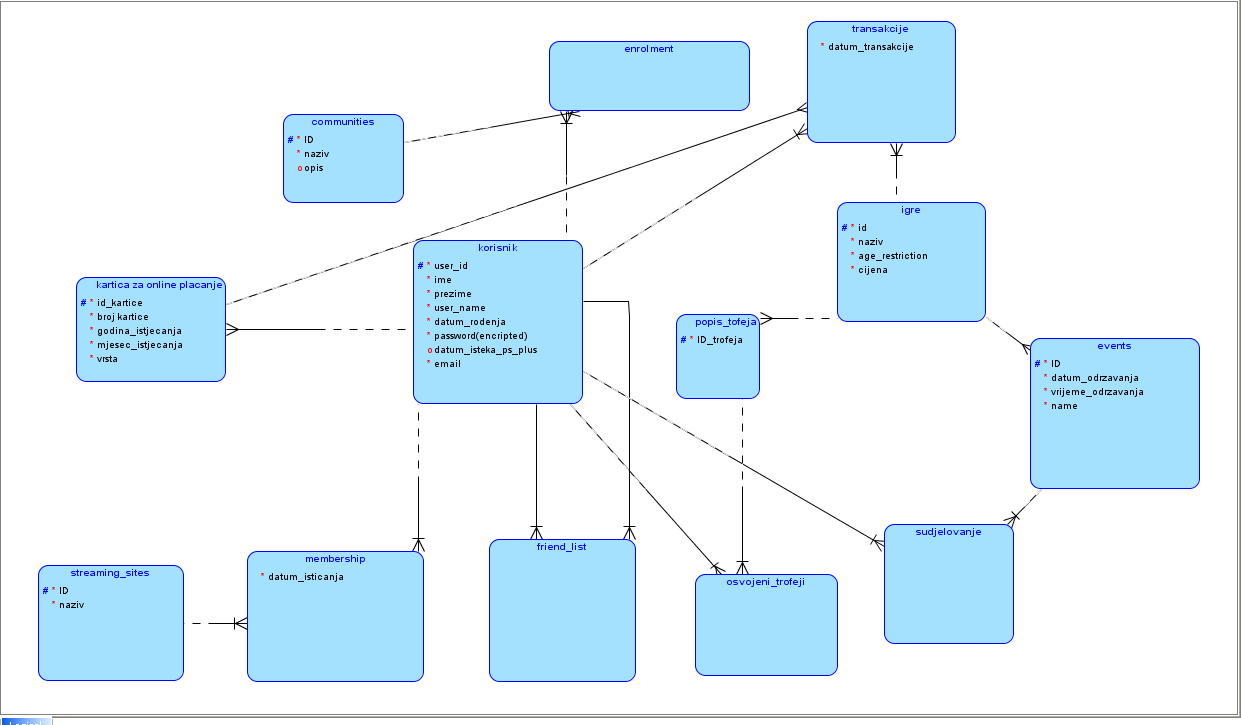
Cilj mog projekta bio je što bolje prikazati bazu podataka za playstationov online sistem.

**2.MEV i Relacijski model:**

Pri izradi baze prije pisanja sql koda bitno je napraviti MEV te iz njega relacijski model za bazu. Pomoću MEV-a možemo jednostavno vizualno prikazati entitete tablice te veze između njih. Relacijski model je koristan jer sadrži koncept relacija između tablica.On podržava programski jezik SQL te tako iz njega možemo napraviti svoju bazu.

Za dizajniranje MEV-a i relacijskog modela koristio sam Oracle SQL Data Developer.

**2a) MEV Playstation online usluge**



Model baze za playstation online sustav ima 13 entiteta. Entiteti su povezani vezama . Glavni entitet u bazi je korisnik(user) te veze iz njega govore sljedeće:

-Svaka katica mora imati svog korisnika , ali svaki korisnik ne mora imati karticu

-korisnik može biti dio communityja , ali i ne mora te svaki community može imati korisnike koji su dio tog communityja , ali I ne mora

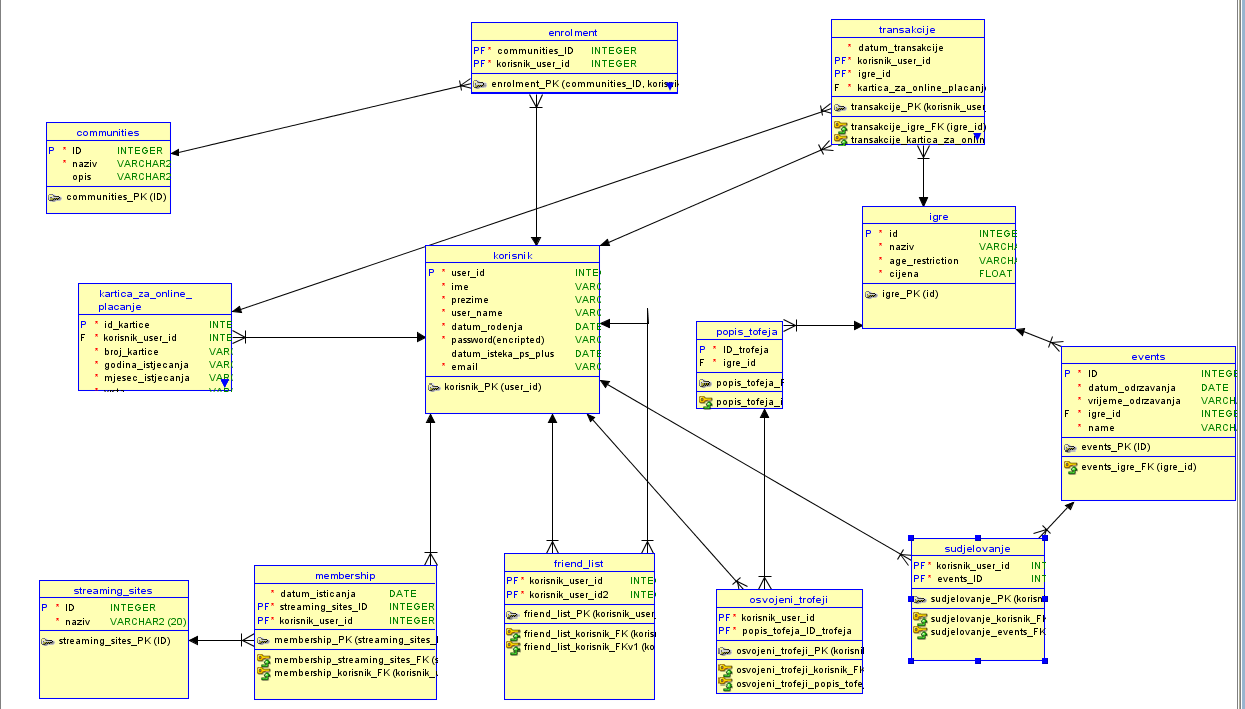
-transakcija je određena korisnikom koji je kupio igru, igrom koju je kupio te karticom koju je upotrijebio

-veza sa friendlistom je zapis dva korisnika koji su ostvarili prijateljstvo

-Praćenje pretplate na streaming siteove određeno je korisnikom I streaming siteom na koji je pretplaćen

-Osvajanje trofeja je određeno korisnikom i popisom trofeja koji se još veže na igru kojoj trofej pripada

-Sudjelovanje u događajima je određeno korisnicima koji sudjeluju u događaju i samim događajem koji se još veže sa igrom u kojoj se sam događaj održava

**2b) Relacijski model Playstation online usluge **

Relacijski model koji smo dobili iz prethodnog mev-a omogućuje nam pravljenje sql baze. Bit će opisani entiteti transakcije ,kartice za online plaćanje ,friend\_list.Ostali se opisuju analogno.

Transakcije se sastoje od tri strana ključa (korisnika,igre i kartice) koji su integeri te primarni kljuc transakcije čine u paru strani ključevi igre i korisnika te se sastoji od atributa datum\_transakcije koji je datum te su svi atributi obvezni.

Kartice za online plaćanje sastoje se od generiranog primarnog ključa koji je integer te stranog ključa korisnika koji je također integer koji su obvezni te od obveznih atributa broj kartice koji je integer, godina isticanja koji je varchar,mjesec isticanja koji je varchar te vrsta kartice koji je varchar.

Friend lista se sastoji od dva strana ključa koji su dva id-a korisnika iz tablice korisnik te su u paru primarni ključ.

**3.Kreiranje tablice:**

Kreiranje tablica služi za pravljenje entiteta u bazi podataka.

Sinstaksa za kreiranje tablica:

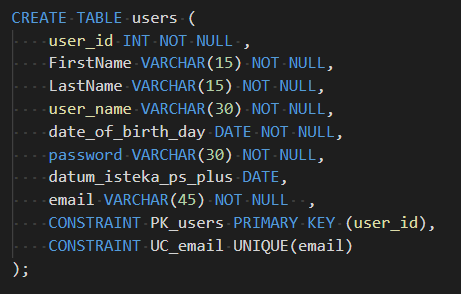
CREATE TABLE table\_name (  
    column1 datatype,  
    column2 datatype,  
    column3 datatype,  
   ....  
);

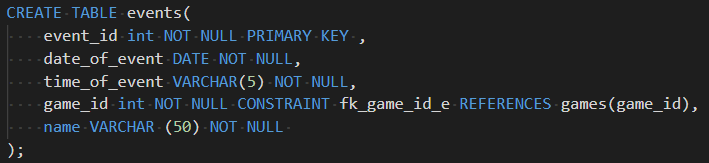
Brisanje tablica:

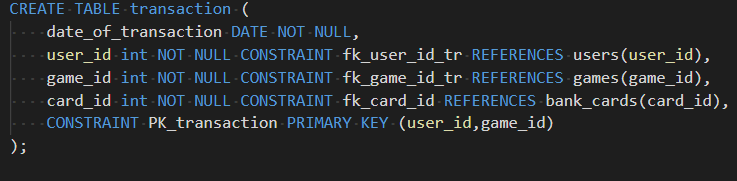
Drop table table\_name;

Mogući tipovi podataka:INT,VARCHAR,NUMBER,BOOL,DATE…

Primjeri:







**4.Unosi u tablicu:**

Unosi služe za punjenje baze sa podatcima.

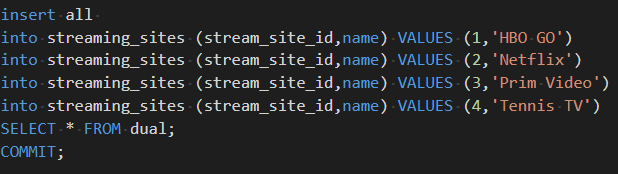
Sintaksa za unos u tablicu:

INSERT INTO table\_name (column1, column2, column3, ...)  
VALUES (value1, value2, value3, ...);

ili

INSERT INTO table\_name  
VALUES (value1, value2, value3, ...);

Pri unosu podataka koristio sam insert all naredbu.

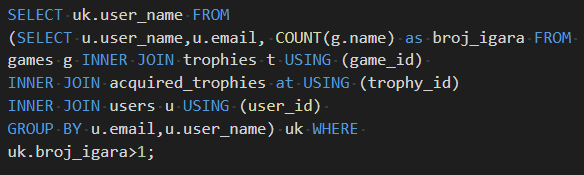
Primjer:

**5.Upiti:**

Upiti nad tablicom služe za dohvaćanje informacija iz baze. Postoje jednostavni i složeni upiti nad tablicom. Jednostavni upiti su upiti nad jednom tablicom dok su složeni upiti upiti nad više od jedne tablice.

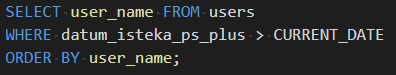
Sintaksa za upite:

SELECT column1, column2, ...  
FROM table\_name;

Primjeri:

Priloženi upit pokazuje korisnike koji imaju trofeje iz više od jedne igre.

Unutar upita napravljen je pod upit iz kojega glavni upit uzima informacije. U podupitu pomoću ključne riječi count brojimo broj igara po korisniku pošto je group by email i user\_name.Tako je napravljen group by jer email mora bit UNIQUE dok user\_name ne mora , a želimo izlistati user\_nameove koji imaju osvojen trofej u više od jedne igre. Tablice se povezane sa INNER JOIN naredbom koji preko stranog ključa povezuje dvije tablice



Priloženi upit propitkuje koji sve korisnici imaju aktivnu ps\_plus pretplatu. Upit je jednostavan te korištenjem ključne riječi WHERE postavljamo uvjet koji provjerava je li zapisan datum isteka pretplate veći od trenutnog dana.

**6.Procedure:**

Procedure su pripremljen sql kod koji se može izvršavati više puta. Proceduri možemo proslijediti vrijednosti tako da nam sql kod može ovisiti o vrijednosti koju smo proslijedili.

Procedure su pisane u jednostavnom proceduralnom jeziku te na njih možemo gledati kao funkcije ili metode.

Prednosti procedura:

* Moguće mnogostruko korištenje
* Optimizacija aplikacije

Nedostatak:

* Spremanje procedure može zauzimati puno memorije

Sintaksa procedure:

CREATE [OR REPLACE] PROCEDURE procedure\_name

[(parameter\_name [IN | OUT | IN OUT]

type [, ...])]{IS | AS}

BEGIN

procedure\_body

END procedure\_name;

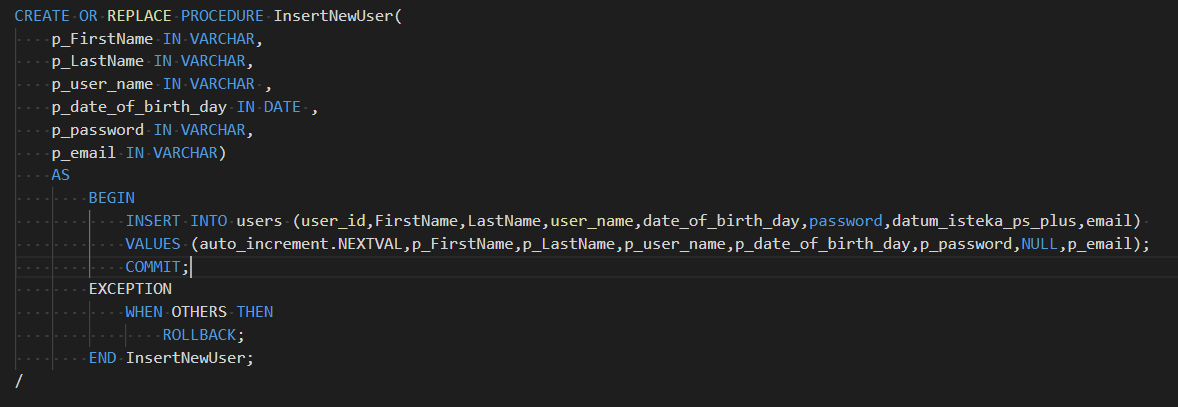
/

Poziv procedure:

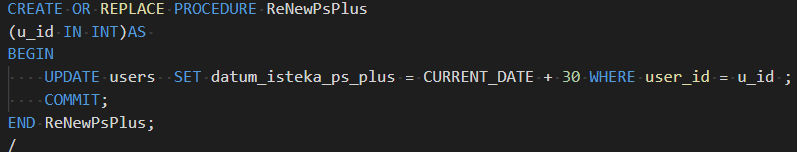
Call procedure\_name(parametar1,…) ;

Brisanje procedure:

DROP PROCEDURE procedure\_name;

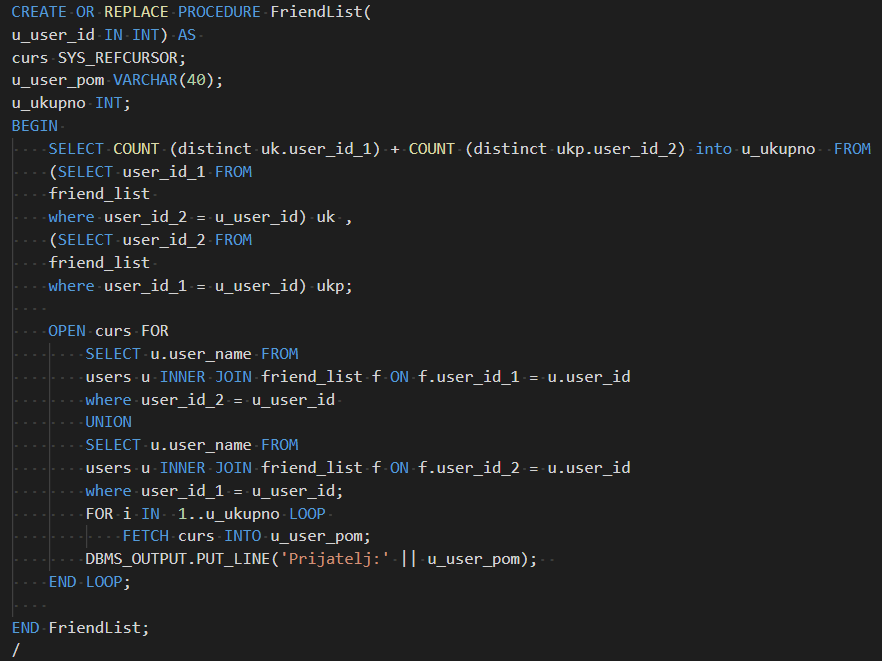
Primjer:

Priložena procedura prihvaća vrijednosti p\_FirstName ,p\_LastName, p\_user\_name, p\_date\_of\_birth\_day, p\_password, p\_email te ih inserta u tablicu users s tim da za id je napravljen SEQUENCE te je datum\_isteka\_ps\_plus uvijek na početku NULL. Naredbom COMMIT potvrđujemo unos novog korisnika u tablicu. Ako se dogodi neka greška pri izvršavanju procedure procedura ulazi u EXCEPTION blok te pomoću ROLLBACK naredbe poništava sve promjene.



Priložena procedura obnavlja pretplatu ps\_plusa te joj stavlja datum isticanja 30 dana nakon što je obnovljen.Procedura prima id od korisnika kojem se produljuje pretplata.

Sa naredbom commit potvrđujemo navedene promjene.



Priložena procedura izlistava sve prijatelje određenog korisnika.Procedura prima ID korisnika kojemu treba izlistati listu prijatelja. Na početku navodimo pomoćne varijable. Prvi upit nam govori koliko prijatelja ima određeni korisnik da bi u idućem upitu pomoću kursora došli do svakog korisnika koji je friendan sa korisnikom te ih sve ispisati.

**7.Okidači:**

Okidači su vrste procedura koji se pokreću nakon naredbi INSERT,DELETE ili UPDATE

Dijelimo ih prema vremenu pokretanja(before/after) i prema načinu djelovanja (row-level/ statement level).

Sintaksa:

CREATE [OR REPLACE] TRIGGER trigger\_name

{BEFORE | AFTER | INSTEAD OF | FOR} trigger\_event

ON table\_name

[FOR EACH ROW]

[{FORWARD | REVERSE} CROSSEDITION]

[{FOLLOWS | PRECEDES} schema.other\_trigger}

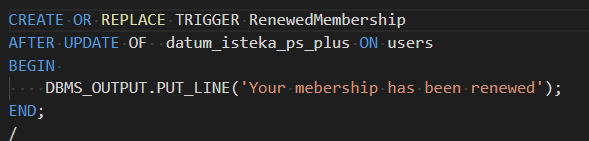
[{ENABLE | DISABLE}]

[WHEN trigger\_condition]]

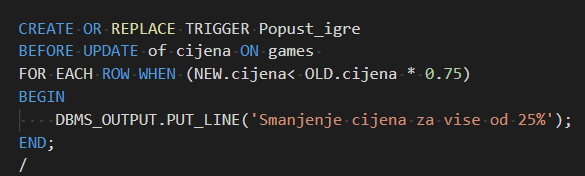
BEGIN

trigger\_body

END trigger\_name;

Primjer: 

Priloženi okidač RenewedMebership je okidač na razini tablice koji se okida nakon nadoplate ps\_plus članstva.



Priloženi okidač Popust\_igre je okidač na razini retka koji javlja korisniku ako je bilo koja igra smanjena za 25 % ili više.

**8.Indeksi:**

Indeksi nam služe za ubrzavanje pretraživanja tablica. Prigodni su za stupce koji imaju velik broj različitih vrijednosti. Njihov način pretraživanja temelji se na strukturi podataka koji nazivamo B-stablo.

Sintaksa:

CREATE [UNIQUE] INDEX index\_name ON

table\_name(column\_name[, column\_name ...])

TABLESPACE tab\_space; Brisanje indeksa:

DROP INDEX index\_name;

Primjer:



Priloženi indeks indeksira trofeje po igrama te ubrzava pretragu trofeja.

**9.)Zaključak:**

Cilj ovog projekta bio je samostalno sa stečenim znanjima i vještinama napraviti bazu podataka , u ovom slučaju za playstation online sistem te usavršiti svoje SQL znanje.

Cjelokupan dojam rada projekta je bio pozitivan zbog samostalnog rad na bazi te samostalno rješavanje problema.