

Sveučilište J. J. Strossmayer u Osijeku, Odjel za matematiku

KOLEGIJ: Moderni sustavi baza podatka

SEMINARSKI RAD

BAZA PODATAKA ZA NBA LIGU

Student: David Dujmović

Mentor: Mateja Đumić

SADRŽAJ:

1. Uvod	2
1.1. Ukratko o NBA ligi	2
1.2. Model entiteta i veza – konceptualni model	3
1.3. Relacijski model	4
2. Kreiranje baze u Oracle SQL Developeru	5
2.1. Kreiranje tablica	5
2.2. Unos podataka naredbom INSERT	7
2.3. Upiti	8
2.3.1. Jednostavni upiti nad jednom tablicom	8
2.3.2. Složeni upiti	9
2.3.3. Podupiti	10
2.4. Indeksi	10
2.5. Procedure	11
2.6. Okidači.....	13
3. Zaključak	15

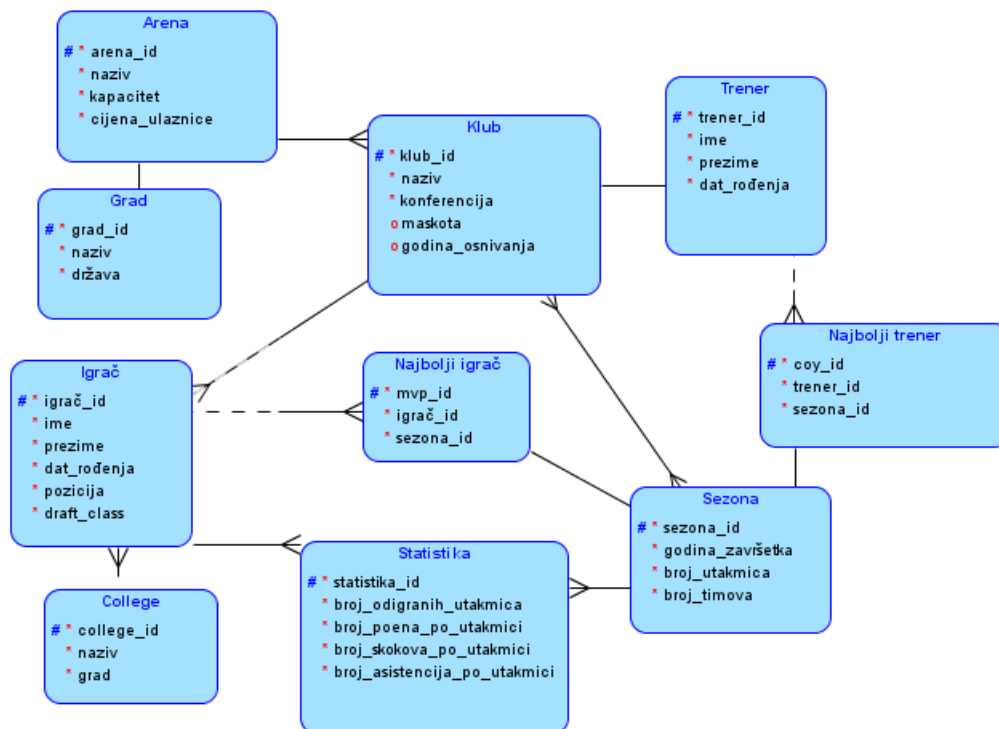
1. UVOD

1.1. UKRATKO O NBA LIGI

Nacionalna košarkaška organizacija, poznatija kao NBA, najistaknutija je muška profesionalna košarkaška liga U Sjevernoj Americi. U njoj sudjeluju razni klubovi iz SAD-a i Kanade. Klubovi su podijeljeni u dvije konferencije, s obzirom gdje se geografski nalaze. One momčadi koje se nalaze bliže istočnoj obali su dio Istočne Konferencije, a oni bliže zapadnoj su dio Zapadne konferencije. Svi klubovi igraju međusobno, bez obzira na konferenciju. Svaki klub se sastoji od barem jednog igrača po poziciji. Pozicije igrača su centar, visoko krilo, plejmejker, nisko krilo, bek šuter. Svaki igrač može i ne mora pohađati college. Atribut `draft_class` podrazumijeva godinu u kojoj je igrač pristupio ligi. Na kraju svake sezone dodjeljuju se nagrade za najboljeg igrača i za trenera godine. Svaka sezona se igra u dvije godine, jer počinje u studenom, a završava u travnju. Statistika igrača se uglavnom sastoji od 4 najvažnije komponente: broj odigranih utakmica, prosjek ubačenih koševa po utakmici, broj skokova po utakmici, te broj asistencija po utakmici

Moj cilj u ovom projektu je bio stvoriti bazu podatka koja će sadržavati popis klubova, igrača, trenera i svih ostalih značajki nekog kluba i koja će pratiti njihove individualne uspjehe kroz statistiku, koja govori koliko sudjeluju u igri

1.2. MODEL ENTITETA I VEZA – KONCEPTUALNI MODEL

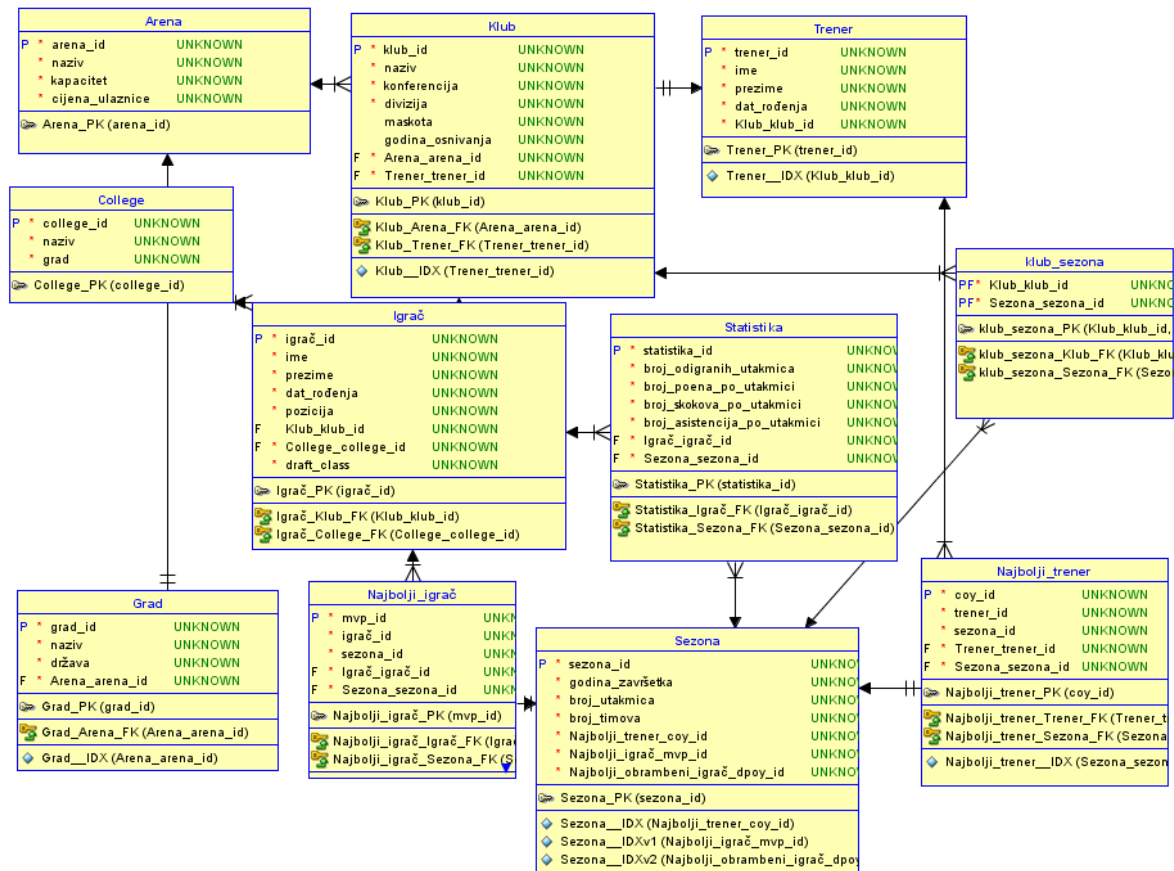


Slika 1. Model entiteta i veza

Konceptualni model koristimo kako bi pokazali entitete, veze i atribute. Model za bazu NBA lige sastoji se od 10 entiteta. Entiteti su povezani vezama, od kojih razlikujemo M:N veze, 1:1 veze i 1:N veze, a to su sljedeće:

- Klub mora imati jednog trenera, te trener smije voditi samo jedan klub. (1:1)
- Klub mora imati arenu u kojoj je domaćin, dok jedna arena može biti središte više klubova. (1:N)
- Arena se uvijek nalazi u jednom gradu, dok jedan grad može imati više arena. (1:N)
- Klub mora imati jednog ili više igrača, dok igrač igra za točno jedan klub. (1:N)
- Svako sveučilište mora imati barem jednog studenta, dok svaki igrač nije morao pohađati sveučilište. (1:N)
- Igrač ima jednu ili više statistika, dok statistika mora biti dodijeljena određenom igraču. (1:N)
- Svaka sezona ima više klubova i svaki klub ima više sezona. (M:N)
- Svaka sezona ima jednog najboljeg trenera i svaka nagrada vezana je uz jednu sezonu (1:1)
- Svaka sezona ima jednog najboljeg igrača i svaka nagrada vezana je uz jednu sezonu (1:1)
- Trener može, ali ne mora, dobiti nagradu za najboljeg igrača jednom ili više puta, te je nagrada dodijeljena točno jednom treneru (1:N)

1.3. RELACIJSKI MODEL



Slika 2. Relacijski model

Ovaj model se dobiva pretvaranjem entiteta i veza iz konceptualnog modela u relacije u kojemu se opisuje atributima iz konceptualnog modela te naglašuje koji su atributi primarni i koji vanjski. **Podebljao** sam primarne ključeve, dok sam strane ključeve *nakosio* u sljedećem dijelu:

- Klub(**klub_id**, naziv, konferencija, divizija, maskota, godina_osnivanja, *arena_id*, *trener_id*)
- Igrač(**igrač_id**, ime, prezime, dat_rođenja, pozicija, *klub_id*, *college_id*, draft_class)
- Arena(**arena_id**, naziv, kapacitet, cijena ulaznice, *grad_id*)
- Grad(**grad_id**, naziv, state, *arena_id*)
- Trener(**trener_id**, ime, prezime, dat_rođenja, *klub_id*)
- College(**college_id**, naziv, grad)
- Sezona(**sezona_id**, godina_održavanja, broj_utmica, broj_timova, *najbolji_trener_id*, *najbolji_igrač_id*)
- Statistika(**statistika_id**, broj_odigranih_utmica, broj_poena_po_utmici, broj_skokova_po_utmici, broj_asistencija_po_utmici, *igrač_id*, *sezona_id*)
- Klub_sezona(**klub_id**, **sezona_id**)
- Najbolji igrač(**najbolji igrač_id**, *igrač_id*, *sezona_id*)
- Najbolji trener(**najbolji trener_id**, *trener_id*, *sezona_id*)

Baza podatka NBA liga

- Klub – sadrži detalje o klubovima NBA lige
- Igrač – sadrži detalje o osobama koje igraju u NBA lizi
- Arena – sadrži detalje o mjestu okupljanja navijača određenog kluba
- Grad – sadrži detalje o mjestu u kojem se nalazi arena
- Sezona – sadrži detalje o vremenskom periodu u kojem se održavaju utakmice
- Trener – sadrži detalje o osoba koja vodi momčad
- Statistika – sadrži podatke o igraču tokom sezone
- College – sadrži detalje o sveučilištu, koje je pohađao igrač
- Najbolji trener – sadrži podatke o osobi kojoj je nagrada za najboljeg trenera u sezoni
- Najbolji igrač – sadrži podatke o osobi kojoj je nagrada za najboljeg igrača u sezoni

2. KREIRANJE BAZE U ORACLE SQL DEVELOPERU

2.1. KREIRANJE TABLICA

Sad kada sam napravio konceptualni i relacijski model za bazu, napravio sam temelj za implementiranje baze. Započeo sam kreiranjem tablica iz relacijskog modela. Opisao sam samo nekoliko tablica.

```
CREATE TABLE klub(  
    klub_id INTEGER CONSTRAINT klub_id_pk PRIMARY KEY,  
    naziv VARCHAR2(60) NOT NULL,  
    konferencija VARCHAR2(60) NOT NULL,  
    maskota VARCHAR2(13) DEFAULT 'Nema maskotu',  
    godina_osnivanja INTEGER,  
    divizija VARCHAR2(60),  
    arena_id INTEGER REFERENCES arena(arena_id),  
    trener_id INTEGER REFERENCES trener(trener_id)  
);
```

Slika 3. Tablica klub

Tablica klub sastoji se od atributa:

- klub_id je tipa INTEGER i on je primarni ključ ove tablice
- naziv, konferencija, divizija su atributi tipa VARCHAR i oni su obvezni atributi ove tablice
- atributi maskota i divizija su opcionalni atributi tipa VARCHAR i INTEGER i oni ne moraju biti uneseni

Sljedeća tablica koju ću izdvojiti je tablica prijava:

```
CREATE TABLE statistika(  
    statistika_id INTEGER CONSTRAINT statistika_pk PRIMARY KEY,  
    status VARCHAR2(20) DEFAULT 'No injury' NOT NULL,  
    games_played DECIMAL(5,2) NOT NULL,  
    points_avg DECIMAL(5,2) NOT NULL,  
    rebounds_avg DECIMAL(5,2) NOT NULL,  
    assists_avg INTEGER NOT NULL,  
    igrac_id INTEGER CONSTRAINT igrac_fk_statistika REFERENCES igrac(igrac_id),  
    sezona_id INTEGER CONSTRAINT sezona_fk_statistika REFERENCES sezona(sezona_id)  
);
```

Slika 4. Tablica statistika

Tablica statistika ima primarni ključ statistika_ID koji je INTEGER, atribut status tipa VARCHAR, games_played koji je tipa VARCHAR2(20) što znači da je dozvoljen samo jedan znak. Također ima i attribute games_played, points_avg, rebounds_avg i assists_avg tipa DECIMAL(5,2) te 2 strana ključa, to su igrač_ID kojim se referencira na tablicu igrač i sezona_ID kojim se referencira na tablicu sezona. Oba strana ključa su obavezna i tipa su INTEGER.

Ostale tablice su priložene u SQL skripti.

2.2. UNOS PODATAKA NAREDBOM INSERT

Za unos podataka koristio sam naredbu INSERT.

```
--arena
INSERT INTO arena VALUES (991, 'Staples Center', 20000, '50', 891);
INSERT INTO arena VALUES (992, 'Barclays Center', 25000, '60', 892);
INSERT INTO arena VALUES (993, 'Welse Fargo Center', 15000, '25', 893);

--treneri
INSERT INTO trener(trener_id, ime, prezime, dat_rodenja) VALUES (441, 'Doc', 'Rivers', TO_DATE('1961/10/13', 'yyyy/mm/dd'));
INSERT INTO trener(trener_id, ime, prezime, dat_rodenja) VALUES (442, 'Frank', 'Vogel', TO_DATE('1973/06/21', 'yyyy/mm/dd'));
INSERT INTO trener(trener_id, ime, prezime, dat_rodenja) VALUES (443, 'Steve', 'Nash', TO_DATE('1974/02/07', 'yyyy/mm/dd'));

--klubovi
INSERT INTO klub(klub_id, naziv, konferencija, godina_osnivanja, divizija, arena_id, trener_id) VALUES (1, 'Lakers', 'Zapadna', null, 'Pacifička', 991, 442);
INSERT INTO klub(klub_id, naziv, konferencija, godina_osnivanja, divizija, arena_id, trener_id) VALUES (2, 'Nets', 'Istočna', null, 'Atlantska', 992, 443);
INSERT INTO klub(klub_id, naziv, konferencija, godina_osnivanja, divizija, arena_id, trener_id) VALUES (3, '76ers', 'Istočna', null, 'Atlantska', 993, 441);
```

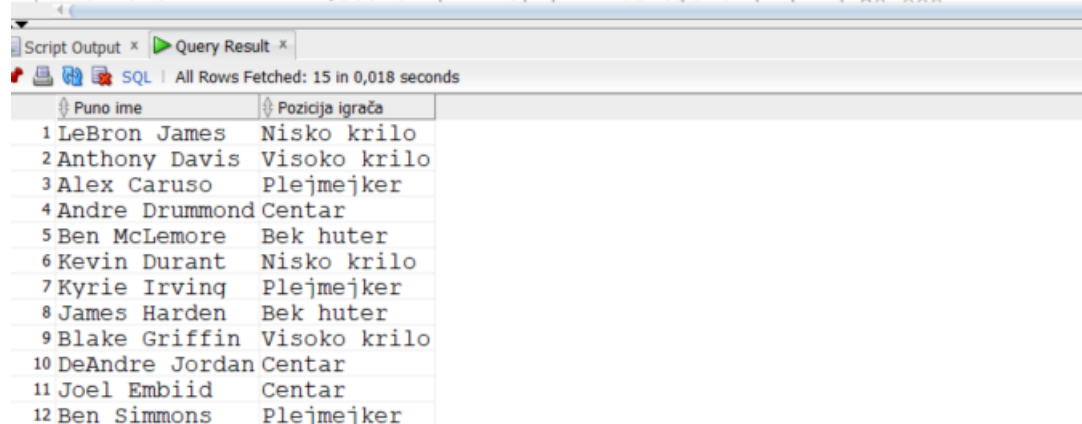
Slika 5. Unos podataka u tablicu klub pomoću naredbe INSERT

Nakon naredbe INSERT potrebno je navesti tablicu u koju se unose podaci. Zatim ide popis vrijednosti koje ćemo unijeti u tablicu, ali oni ne moraju biti navedeni. Vrijednost primarnog ključa ne smije se podudarati niti s jednom vrijednošću tog ključa koja je zapisana u tablici. Ukoliko smo unijeli pogrešan podatak možemo ga promijeniti pomoću naredbe UPDATE. Nakon što sam unio podatke u tablicu pregledao bi ih i ispisao pomoću naredbe SELECT * FROM.

2.3. UPITI

2.3.1. JEDNOSTAVNI UPITI NAD JEDNOM TABLICOM

```
--TOČKA 5
--ispisi ime i prezime svakog igrača, a da pritom koristih jedan stupac
SELECT ime || ' ' || prezime "Puno ime", pozicija "Pozicija igrača"
FROM igrac;
```




	Puno ime	Pozicija igrača
1	LeBron James	Nisko krilo
2	Anthony Davis	Visoko krilo
3	Alex Caruso	Plejmejker
4	Andre Drummond	Centar
5	Ben McLemore	Bek huter
6	Kevin Durant	Nisko krilo
7	Kyrie Irving	Plejmejker
8	James Harden	Bek huter
9	Blake Griffin	Visoko krilo
10	DeAndre Jordan	Centar
11	Joel Embiid	Centar
12	Ben Simmons	Plejmejker

Slika 6. Jednostavni upit nad tablicom vlasnik

Na slici 6 možemo vidjeti primjer jednostavnog upita nad tablicom igrač koji će ispisati tablicu sa dva stupca. U prvom stupcu pomoću operatora konkatencije ispisati će ime i prezime vlasnika kao „Puno ime“, a u drugom stupcu ispisati će poziciju kao „Pozicija igrača“

2.3.2. SLOŽENI UPITI

```
--ispisi datum rođ'enja koharkaha koji nije ihao na college
SELECT ime, prezime, naziv, draft_class
FROM igrac INNER JOIN college
USING (college_id)
WHERE college_id IS NOT NULL;
```



Script Output x Query Result x

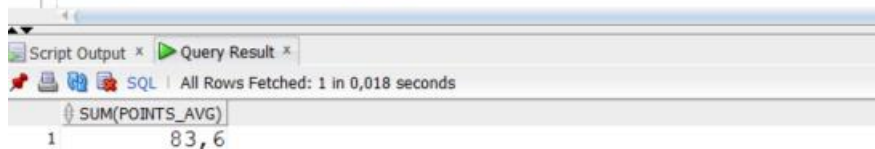
SQL All Rows Fetched: 14 in 0,022 seconds

	IME	PREZIME	NAZIV	DRAFT_CLASS
1	Anthony	Davis	Ketucky	2012
2	Alex	Caruso	Texas	2017
3	Andre	Drummond	UConn	2012
4	Ben	McLemore	Kansas	2013
5	Kevin	Durant	Texas	2008
6	Kyrie	Irving	Duke	2011
7	James	Harden	Arizona	2012
8	Blake	Griffin	Oklahoma	2010
9	DeAndre	Jordan	Texas	2008
10	Joel	Embiid	Kansas	2016
11	Ben	Simmons	LSU	2017
12	Tobias	Harris	Texas	2008
13	Seth	Curry	Duke	2014
14	Tyrese	Maxey	Ketucky	2020

Slika 7. Složeni upit nad tablicama igrač i college

Ovim upitom želio sam vidjeti koji su igrači išli na sveučilište i kada su ušli u ligu. Spojio sam tablice igrač i college preko stranog ključa college_id i to mi je omogućilo željeni ispis.

```
--ispisi igraca koji najvihe zabiye u ekipi zvana 76ers
SELECT SUM(points_avg)
FROM igrac
INNER JOIN statistika
USING(igrac_id)
INNER JOIN klub
USING(klub_id)
WHERE klub.naziv = '76ers';
```



Script Output x Query Result x

SQL All Rows Fetched: 1 in 0,018 seconds

	SUM(POINTS_AVG)
1	83,6

Slika 8. Složeni upit sa INNER JOIN spojem nad tablicama igrač, klub i statistika

Upit sa slike 8 ispisuje prosjek poena po utakmici momčadi koja se naziva 76ers. Spajanje sam napravio na 3 tablice: igrač, klub i statistika. Nakon svake klauzule USING u zagradi se nalazi ime stupca po kojemu se vrši spajanje.

2.3.3. PODUPITI

```
--ispisi sve igrace koji po prosjeku imaju manje koseva od prosjecnog broja skokova nekog igraca
SELECT *
FROM igrac
INNER JOIN statistika
USING (igrac_id)
WHERE points_avg < ANY(SELECT (rebounds_avg)
                        FROM igrac
                        INNER JOIN statistika
                        USING (igrac_id));
```

Script Output * Query Result *

SQL | All Rows Fetched: 5 in 0,045 seconds

IGRAC_ID	IME	PREZIME	DAT_RODENJA	POZICIA	KLUB_ID	COLLEGE_ID	DRAFT_CLASS	STATISTIKA_ID	STATUS	GAMES_PLAYED
1	114	Alex	Caruso	28.02.1994	Plejmejker	1	348	2017	714	No injury 24
2	115	Ben	McLemore	24.07.1995	Bek huter	1	352	2013	715	No injury 24
3	124	Blake	Griffin	16.03.1989	Visoko krilo	2	351	2010	720	No injury 24
4	125	DeAndre	Jordan	21.07.1988	Centar	2	348	2008	719	No injury 24
5	135	Tyrese	Maxey	04.11.2000	Nisko krilo	3	345	2020	725	No injury 24

Slika 9. Podupit

Slika 9. prikazuje podupit koji vraća sve koji imaju manji prosjek koševa od prosječnog broja skokova nekog igrača.

2.4. INDEKSI

```
--točka 12
CREATE INDEX igrac_ime_prezime ON
igrac(ime, prezime);

CREATE INDEX statistika_poeni ON
statistika(points_avg);
```

Slika 10. Indeks koji ubrzava pretragu igrača i njihove statistike

U ovoj tablici imam dosta različitih vrijednosti pa mi je indeks koristan kako bi ubrzao pretragu.

2.5. PROCEDURE

U svojoj bazi kreirao sam nekoliko procedura za unos i ažuriranje podataka.

```
DROP PROCEDURE update_points_avg;
CREATE OR REPLACE PROCEDURE update_points_avg
AS godina_fst INTEGER;
argument_fst INTEGER;
BEGIN
    SELECT FLOOR((TO_DATE('2021/06/30', 'yyyy/mm/dd') - dat_rodenja) / 365)
    INTO godina_fst
    FROM igrac
    WHERE igrac_id = 121;

    --argument_fst := 1.0;

    IF godina_fst > 30 THEN
        argument_fst := 1 - argument_fst;
    ELSE argument_fst := 1 + argument_fst;
    END IF;

    UPDATE statistika
    SET points_avg = points_avg * argument_fst
    WHERE igrac_id = 121;

    COMMIT;

END update_points_avg;
/
```

Slika 11. Procedura za ažuriranje statistike

Svaki igrač pripada nekoj dobnoj kategoriji, a godine igranju značajnu ulogu u efikasnosti na tako visokoj razini košarke. Stoga sam, preko ove procedure napravio da se prosjek broja koševa smanji ili poveća s obzirom da li ga dostižu godine ili još nije napunio 30 godina.

```

SHOW ERRORS;
DROP PROCEDURE injury_update;
CREATE OR REPLACE PROCEDURE injury_update
IS i_status VARCHAR(20);
   i_gms_played INTEGER;
   i_gms_max INTEGER;

BEGIN
    SELECT games_played INTO i_gms_played
    FROM statistika
    WHERE igrac_id = 133;

    SELECT broj_utakmica INTO i_gms_max
    FROM sezona
    WHERE godina_održavanja = '2020/2021';

    IF i_gms_played = i_gms_max THEN
        i_status := 'Not able to play';
    ELSE
        i_status := 'Able to play';
    END IF;

END;
/

```

Slika 12. Procedura za ažuriranje statusa igrača

S obzirom da sezona dugo traje, igrači često nemaju energije igrati za nacionalnu selekciju. S toga sam napravio proceduru kako bi treneri reprezentacija mogli znati koji su igrači igrali sve utakmice sezone. Procedura za ažuriranje i unos podataka radi po principu ako unesete postojeći igrac_id onda će se podaci ažurirati.

```
--točka 13
CREATE OR REPLACE PROCEDURE update_points_average
AS points_average_114 statistika.points_avg%TYPE;
BEGIN
SELECT points_avg INTO points_average_114
FROM statistika INNER JOIN igrac
USING(igrac_id)
WHERE igrac_id = 114;

UPDATE statistika
SET points_avg = (SELECT points_avg FROM statistika WHERE igrac_id = 131)
WHERE igrac_id = 114;

UPDATE statistika
SET points_avg = points_average_114
WHERE igrac_id = 131;

END update_points_average;
/
```

Slika 13. Procedura za zamjenu atributa points_avg kod igrača

Ova procedura radi zamjenu atributa points_avg kod igrača. Pozovete proceduru i upišete id igrača koje želite zamjeniti.

2.6. TRIGGERI

```
CREATE OR REPLACE TRIGGER trigger_fst
AFTER UPDATE
ON statistika
FOR EACH ROW WHEN (new.assists_avg < old.assists_avg)
BEGIN
dbms_output.PUT_LINE('Broj asistencija po utakmici igraca '
|| :old.igrac_id || ' se smanjio s ' || :old.assists_avg ||
' na ' || :new.assists_avg || '.');
END trigger_fst;
/
```

Slika 14. Okidač koji šalje poruku ukoliko se smanjio broj asistencija po utakmici

Na visokoj razini košarke, poput one koja se igra u NBA-u, nije prihvatljivo postajati lošiji u bilo kojem segmentu, jer puno je igrača koji čekaju da im se ukaže prilika. Stoga sam kreirao okidač koji šalje poruku da upozori igrača.

```

--
CREATE OR REPLACE TRIGGER trigger_snd
  AFTER UPDATE
  OF status
  ON statistika
  FOR EACH ROW WHEN (new.status <> 'Able to play' OR new.status <> 'No injury')
BEGIN
  dbms_output.PUT_LINE('Bad luck' );
END trigger_snd;
/

```

Slika 15. Okidač koji šalje poruku ukoliko dođe do povrede

Ponekad steknete mnogo prijatelja kad dobijete na popularnosti. Neki ljudi vam se pretvaraju prijateljima samo kako bi izvukli korist od Vas. Najčešće se događa da shvatite tko su oni pravi tek kada se nađete u nekom problemu. Stoga sam generirao okidač s kojom bi mogli razlikovati lažne od pravih prijatelja kada se dogodi takva situacija.

3.ZAKLJUČAK

Ovim seminarskim radom prikazana je jednostavna baza podataka kakva se može primijeniti na stranice, čiji su posjetitelji košarkaški zaljubljenici. Pri implementaciji MEV modela koristio sam Data Modeler, te sam pomoću Oracle SQL Developera kreirao tablice, unose i postavio upite. Ovim modelom predočio sam košarkašku organizaciju što bi moglo biti zanimljivo fanovima ali i ljudima koji ne prate košarku.