РАЧУНАРСКА ГИМНАЗИЈА

МАТУРСКИ РАД

из предмета

Модели и базе података

База података за кошаркашки клуб

Ученик

Бошко Влаховић IV/3

Ментор

Душа Вуковић

Београд, мај 2021.

Садржај

Релационе базе података

Мали увод

Теорија каже да је база података било каква колекција, било каквих података, записаних на било какав начин, међутим, када се спомене овај термин, обично се мисли на релационе базе података и начин организације и обраде података који је њима својствен. Можда јесте примамљиво податке само "набацати" на гомилу, али, у практичном смислу, такав приступ само може довести до проблема када су у питању колекције података иоле већег обима и значаја. Базе података користе се у већини десктоп апликација и на скоро свим сајтовима, па је њихов значај свакако велик. У наставку, упознаћемо се са релационим базама података и SQL-ом, програмским језиком који омогућава приступ подацима и њихову обраду.

Основно о релационим базама

Релационе базе података представљају скуп података који су међусобно повезани одређеним релацијама тј. везама .Ти подаци су складиштени и организовани уз помоћ табела једноставних структура које називамо релационе табеле. Табеле су одређене колонама и редовима. Помоћу примера из сопствене базе ови појмови ће бити детаљније објашњени. У зависности од тога шта нам је потебно у табели тако и означавамо атрибуте. Атрибут који је означен као обавезан, испред назива атрибута има *****, при креирању мора да се напише да је **NOT NULL**, и обавезно је попунити то поље приликом попуњавања табеле. Када је атрибут означен као опциони податак, он испред назива атрибута има **O**, тако да при креирању табеле није обавезно попунити тај податак. Када у неком реду табеле нећемо да попунимо податак уносимо само реч **NULL**, и тако написано ће у табели бити пириказано као **-**. Један од атрибута се изабере као јединствени идентификатор, примарни кључ и он се означава **#**.

IGRACI							
id_igraca	**ime**	**prezime**	**pozicija**	**broj_dresa**	**broj_telefona**	**datum_uclanjivanja**	**id_ekipe**
101	Luka	Vukadinovic	Plej mejker	4	061190909	04.05.2013.	23
102	Petar	Veruseski	Plej mejker	5	064123321	07.07.2017.	23
103	Strahinja	Milic	Bek suter	6	069743561	13.02.2012.	23
104	Aleksa	Kovacevic	Krilo	7	063477390	21.06.2012.	23
105	Veljko	Vukasinovic	Krilo	8	065147990	/	23
106	Bosko	Vlahovic	Krilo	9	065356811	29.05.2015.	23
107	Strahinja	Jovanovic	Bek suter	10	062342115	18.09.2011.	23

Колона са примарним кључевима

Колона где подаци морају да се унесу

Колона где подаци нису обавезни

Када у дијаграму имамо везу више на према више, тада додајемо нови ентитет који путем барованих веза спајамо са другим ентитетима. Приликом спајања нови ентитет повлачи примарне кључеве других ентитета, у његовој табели су приказани као страни и примарни кључ. Нова табела може али и не мора да садржи додатне податке, потребно је само да има примарне кључеве суседних ентитета. На овом примеру се то види.

IGRACI

id_igraca	ime	prezime	pozicija	broj_dresa	broj_telefona	datum_uclanjivanja	id_ekipe
101	Luka	Vukadinovic	Plej mejker	4	061190909	04.05.2013.	23
102	Petar	Veruseski	Plej mejker	5	064123321	07.07.2017.	23
103	Strahinja	Milic	Bek suter	6	069743561	13.02.2012.	23
104	Aleksa	Kovacevic	Krilo	7	063477390	21.06.2012.	23
105	Veljko	Vukasinovic	Krilo	8	065147990	/	23
106	Bosko	Vlahovic	Krilo	9	065356811	29.05.2015.	23
107	Strahinja	Jovanovic	Bek suter	10	062342115	18.09.2011.	23

IGRACI_UTAKMICE

id_igraca	id_utakmice
101	060417
102	060417
103	060417
104	060417
105	060417
106	060417
107	060417
108	060417
109	060417

UTAKMICE

id_utakmice	liga	datum_uatkmice	ime_delegata
060417	Kvalitetna liga	06.04.2017	Stefan Papic
130417	Kvalitetna liga	13.04.2017	Mirko Rasic
250518	Jedinstvena liga	25.05.2018	Nemanja Maksimovic
010618	Jedinstvena liga	01.06.2018	Ilija Petrovic
180319	Kvalitetna liga	18.03.2019	Svetlana Maric

База података за кошаркашки клуб

Опис пословања

Кошаркашки клуб пружа младима прилику да се опробају у овом спорту. Има доста талената који још нису откривени и забележени. Ради вођења лакше евиденције правимо базу података за клуб. Када се неко учлани у клуб води се и као играч истог. За разлчите категорије постоје лиге које играју. У зависности од детета, тј. узраста, одлази у једну од многих клупских екипа са својим тренерима. Тренери су ту да омогуће праву помоћ при усавршавању кошаркашких вештина. Те екипе играју утакмице и бране боје свог клуба. О томе се води евиденција и на крају гледа ко је најбољи.

Опис модела

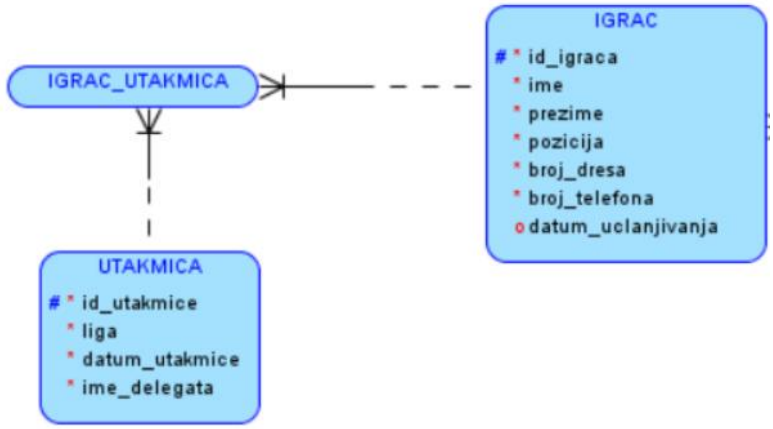
Правимо базу података за кошаркашки клуб. Сваки клуб има више екипа. Сваку екипу чини више играча који играју утакмице. Између утакмица имамо тренинге које воде одабрани тренери у зависности од типа тренинга, тј, жељеног циља(боље технике,кондиције, итд.). Имамо ентитете који су независни. Ентите KLUB садржи атрибуте као што су id_kluba који је обавезан и има #, јер је примаран кључ. Такође поседује обавезне ентите који испред себе имају * као што су naziv, adresa, broj_telefona и mail. Entitet UTAKMICA има id_utakmice као примарн кључ. Обавезни атрибути у овом ентитеу су: liga, datum_utakmice i ime_delegata. Делегат је тај који је одговоран да утакмица прође у најбољем реду. TIP_TRENINGA има id_tipa_treninga као примарни кључ и назив и опис који су обавезни. Опис тренинга нам пружа информације о томе како изгледа тренинг. Остали ентититети имају стране кључеве уз своје већ посотјеће примарне. IGRAC је центар базе јер све креће од њега. Примрни кључ id_igaraca је троцифрени број који сваки играч мора да има. Ime, prezime, pozicija(сваки играч има своју позицију у зависности од предиспозиција), broj_dresa је такође обавезан ради лакшег препознавања играча, broj_telefona постоји због заказивања утакмица, прелазака у друге клубове и слично. На крају datum_uclanjivanja није обавезан, али је добро имати га у бази. Када играч игра утакмицу долазимо до ентитета IGRAC_UTAKMICA који има примарне кључеве табеле IGRAC и UTAKMICA(овде је тип везе више више). EKIPA има id_ekipe као примарни и id_kluba као страни кључ у зависности од клуба за који екипа игра(веза један на према више). Имамо и назив екипе као обавезни атрибут. Свака ЕКИРА има више тренинга тако да TRENING има страни кључ id_ekipe и нарвно свој примарни кључ id_treninga, дан и време као обавезне атрибуте. Ентитет TRENER има broj_licence као примарни кључ, јер без њега не би могао да буде тренер. Остали атрибути ime, prezime и broj_telefona су обавезни поготово због заказивања уткамица.

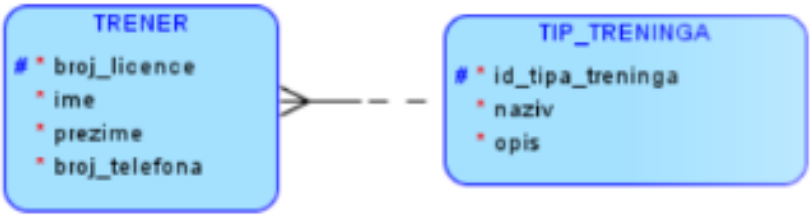
Независни ентитети

На следећим сликама могу да се виде независни ентитети на основу којих ће се креирати прве табеле у бази података.

Примери неких веза

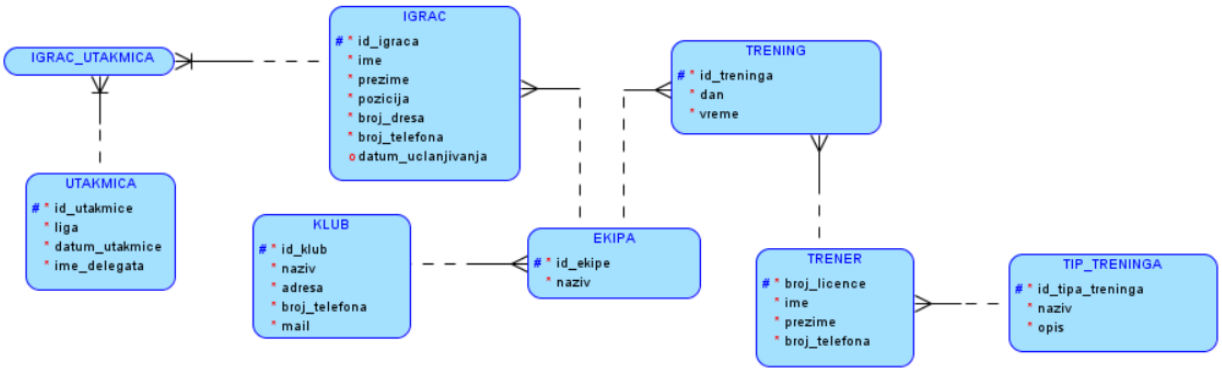
На следеће две слике могу да се виде примери везе M:M (више на према више) и везе 1:M (један на према више).





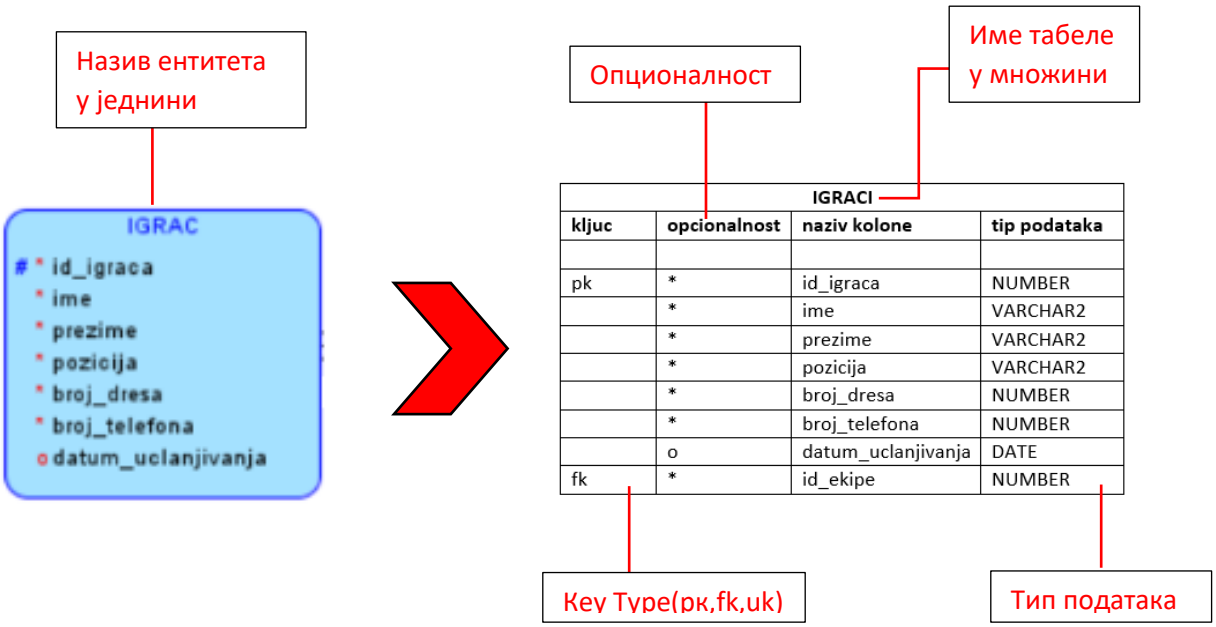
Пример целог логичког модела

Следи приказ целог логичког модела на основу којег ће се креирати база података.



Мапирање

Након детаљне анализе пословања и креирања логичког модела, дијаграма ентитета и веза, потребно је да се уради припрема за креирање базе података. Та припрема се назива мапирање и подразумева да се формира детаљан опис свих табела које ће имати база података. Опис сваке табеле мора да садржи списак колона, њихових типова података и списак свих ограничења, као што су примарни и страни кључеви, ограничење јединствености и ограничење NOT NULL.



Најчешће коришћен типови података:

NUMBER - Број. У загради може да се наведе број цифара. Уколико се наведе још један број у загради, он редставља број децимала.

VARCHAR2 - Кратак текст. У загради може да се наведе број знакова, тако да је, на пример,

VARCHAR2(60) - ознака за текстуални податак који има највише 60 знакова.

CHAR - Кратак текст фиксиране дужине. У загради може да се наведе број знакова.

CLOB - Велики текстуални податак. Скраћеница не енглеском: LOB - Large Object.

DATE - Датум.

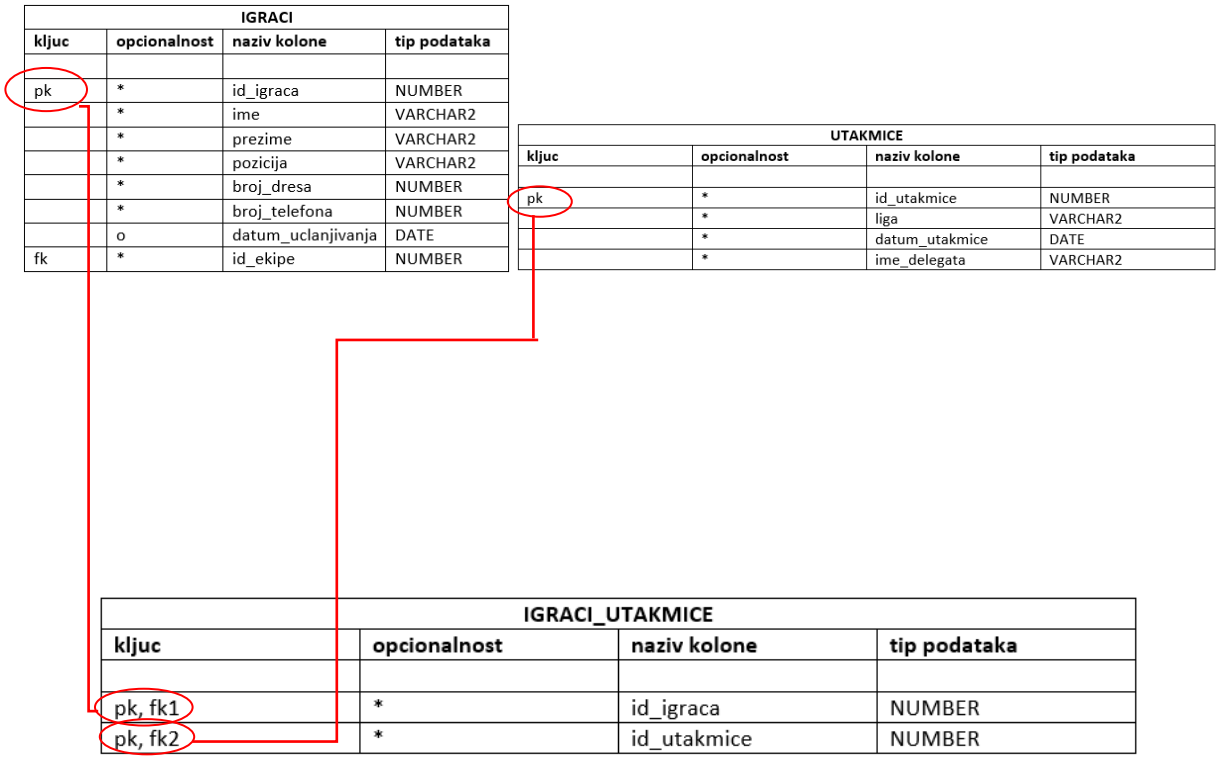
CURRENCY - Валута.

Примарни јединствени идентификатор постаје примарни кључ: **pk - primary key**. Атрибути који су били кандидати за примарни јединствени идентификатор морају да имају особину да су им вредности за сваку инстанцу ентитета јединствене.

Колоне које њима одговарају имају ограничење јединствености: **uk - unique key**.

Свака веза постаје додатна колона у једној од две повезане табеле са додатним ограничењем да је та колона страни кључ: **fk - foreign key**.

Ентитет који разрешава везу више на према више, приликом мапирања табела ће садржати примарне кључеве суседних ентитета, и у њој ће бити приказани као примарни и страни кључ. Ако су ентитети повезани само путем везе један на према више, онда ће тај примарни кључ у његовој табели бити приказан само као страни кључ и имаће ознаку **fk**. Уколико је повезан путем барованих веза онда ће тај ид у табели бити и примаран и страни кључ и имаће ознаку **pk** и **fk**.



IGRACI			
kljuc	**opcionalnost**	**naziv kolone**	**tip podataka**
pk	*	id_igraca	NUMBER
	*	ime	VARCHAR2
	*	prezime	VARCHAR2
	*	pozicija	VARCHAR2
	*	broj_dresa	NUMBER
	*	broj_telefona	NUMBER
	o	datum_uclanjivanja	DATE
fk	*	id_ekipe	NUMBER

UTAKMICE			
kljuc	**opcionalnost**	**naziv kolone**	**tip podataka**
pk	*	id_utakmice	NUMBER
	*	liga	VARCHAR2
	*	datum_utakmice	DATE
	*	ime_delegata	VARCHAR2

IGRACI_UTAKMICE			
kljuc	**opcionalnost**	**naziv kolone**	**tip podataka**
pk, fk1	*	id_igraca	NUMBER
pk, fk2	*	id_utakmice	NUMBER

Узорак података

IGRACI							
id_igraca	**ime**	**prezime**	**pozicija**	**broj_dresa**	**broj_telefona**	**datum_uclanjivanja**	**id_ekipe**
101	Luka	Vukadinovic	Plej mejker	4	061190909	04.05.2013.	23
102	Petar	Veruseski	Plej mejker	5	064123321	07.07.2017.	23
103	Strahinja	Milic	Bek suter	6	069743561	13.02.2012.	23
104	Aleksa	Kovacevic	Krilo	7	063477390	21.06.2012.	23
105	Veljko	Vukasinovic	Krilo	8	065147990	/	23
106	Bosko	Vlahovic	Krilo	9	065356811	29.05.2015.	23
107	Strahinja	Jovanovic	Bek suter	10	062342115	18.09.2011.	23
108	Ognjen	Vasiljevic	Krilo	11	065783192	15.05.2013.	23
109	Uros	Krstic	Bek suter	12	061143389	14.04.2015.	23
110	Nikola	Vucinic	Centar	13	066783456	11.01.2013.	23
111	Filip	Sinobad	Krilo	14	064778221	28.05.2017.	23
112	Nikola	Miholjcic	Centar	15	063455338	/	23
201	Ognjen	Kovacevic	Krilo	4	069789356	13.07.2016.	45
202	Filip	Stojkovic	Krilo	5	064367900	05.06.2013.	45
203	Dragomir	Babic	Krilo	6	065553698	19.03.2016.	45
204	Dusan	Nikolic	Plej mejker	7	065983265	08.08.2014.	45
205	Arsenije	Strbac	Bek suter	8	063215664	/	45
206	Andrija	Cvetkovic	Centar	9	061555333	12.05.2017.	45
207	Aleksa	Markovic	Centar	10	062456654	30.07.2015.	45
208	Djordje	Ciric	Plej mejker	11	063885249	/	45
209	Petar	Petrovic	Krilo	12	066139422	26.06.2019.	45
210	Ljubisa	Karajankovic	Bek suter	13	065143588	/	45
211	Stefan	Matijas	Bek suter	14	066325647	03.09.2018.	45
212	Uros	Kozic	Krilo	15	061548795	13.10.2016.	45

TRENINZI			
id_treninga	**dan**	**vreme**	**id_ekipe**
001	Ponedeljak	21:00	23
012	Utorak	21:30	23
039	Sreda	21:30	23
095	Cetvrtak	21:00	23
007	Petak	21:30	23
023	Subota	12:00	23
091	Ponedeljak	20:30	45
033	Utorak	21:00	45
027	Sreda	21:00	45
005	Cetvrtak	21:30	45
010	Petak	20:30	45
064	Subota	16:30	45

KLUBOVI				
id_kluba	**naziv**	**adresa**	**broj_telefona**	**mail**
13122	KK ,,Zarkovo,,	Olimpijskih igara 14	233678	zarkovokk@gmail.com
99867	KK ,,Crvena Zvezda,,	Mali Kalemegdan 2	455621	kkczv@gmail.com

UTAKMICE			
id_utakmice	**liga**	**datum_uatkmice**	**ime_delegata**
060417	Kvalitetna liga	06.04.2017	Stefan Papic
130417	Kvalitetna liga	13.04.2017	Mirko Rasic
250518	Jedinstvena liga	25.05.2018	Nemanja Maksimovic
010618	Jedinstvena liga	01.06.2018	Ilija Petrovic
180319	Kvalitetna liga	18.03.2019	Svetlana Maric

EKIPE		
id_ekipe	**naziv**	**id_kluba**
23	Zarkovo kadeti	13122
45	Crvena Zvezda kadeti	99867

TRENERI				
broj_licence	**ime**	**prezime**	**broj_telefona**	**id_tipa_treninga**
238	Zlatko	Polak	065556987	1
230	Nikola	Kitanovic	063598633	1
235	Nebojsa	Peric	061233548	2
456	Miroslav	Kljajic	062669339	1
478	Srdjan	Maksic	069788745	2

TIPOVI_TRENINGA		
id_tipa_treninga	**naziv**	**opis**
1	Trening sa loptom	tehnika, sut, taktika, igra jedan na jedan, vezbanje odbrane
2	Kondicioni trening	kondicija, vezbe snage, vezbe izdrljivosti

IGRACI_UTAKMICE	
id_igraca	id_utakmice
101	060417
102	060417
103	060417
104	060417
105	060417
106	060417
107	060417
108	060417
109	060417
110	060417
111	060417
112	060417
201	060417
202	060417
203	060417
204	060417
205	060417
206	060417
207	060417
208	060417
209	060417
210	060417
211	060417
212	060417
101	250518
102	250518
103	250518
104	250518
105	250518
106	250518
107	250518
108	250518
109	250518
110	250518
111	250518
112	250518
201	250518
202	250518
203	250518
204	250518
205	250518
206	250518

SQL језик

SQL (Structured Query Language) је стандардни релациони упитни језик. Његов творац је Чемберлин а настао је у IBM-овој истраживачкој Лабораторији (IBM Research Laboratory) Сан Хозеу, Калифорнија 1974. године, дакле на истом месту где је код 1970. дефинисао основне концепте релационог модела података.

SQL наредбе сврстане су у три категорије:

1. Наредбе за дефинисање података

2. Наредбе за руковање подацима

3. Наредбе за контролне функције

Наредбе за дефинисање података

Следи команда CREATE TABLE језика SQL којом се креира ова табела у релационој бази података. Као што можемо приметити, структура команде одговара оној табели коју смо креирали у процесу мапирања и за коју кажемо да описује дизајн табеле. Када желимо да обележимо да је нешто примарни кључ, после назива атрибута и типа података пишемо PRIMARY KEY. За податке који су обавезни атрибути поред типа података ћемо написати нот нулл, и тај податак је обавезно попунити. Уколико је податак опциони, поред типа података нећемо ништа написати.

```
CREATE TABLE KLUBOVI
( id_kluba NUMBER(5) PRIMARY KEY,
    naziv VARCHAR2(25) NOT NULL,
    adresa VARCHAR2(20) NOT NULL,
    broj_telefona NUMBER(6) NOT NULL,
    mail VARCHAR(25) NOT NULL )
```

```
CREATE TABLE UTAKMICE
( id_utakmice NUMBER(6) PRIMARY KEY,
    liga VARCHAR2(20) NOT NULL,
    datum_utakmice DATE NOT NULL,
    ime_delegata VARCHAR(25))
```

Када креирамо табелу за ентитет који је у дијаграму повезан путем барованих веза са другим ентитетима, тад су нам **id** тих табела и примарни кључ(**pk**) и страни кључ(**fk**). Уписујемо назив колоне у којој ћемо чувати примарни кључ табеле са којом је наша табела повезана, њем тип и кључну реч **REFERENCES** потом назив табеле из које узимамо примарни кључ и у заградама уносимо назив примарног кључа те табеле. Овако унет примарни кључ друге табеле је у нашој табели приказан као страни кључ. Он у насој табели постаје примарни кључ уношењем кључне речи **PRIMARY KEY** у загради називе примарних кључева друге табеле.

```
CREATE TABLE IGRACI_UTAKMICE
( id_igraca NUMBER(3) REFERENCES igraci(id_igraca),
  id_utakmice NUMBER(6) REFERENCES utakmice(id_utakmice),
  PRIMARY KEY(id_igraca,id_utakmice))
```

DROP TABLE наредба

За брисање табеле из базе коју смо креирали, користимо DROP TABLE . Када пишемо наредбу за брисање табеле користимо кључну реч DROP TABLE у наставку наредбе напишемо назив табеле коју желимо да избришемо, при покретању наредбе табела ће бити избрисана.

```
DROP TABLE klubovi
```

Наредбе за руковање података

SELECT наредба

Да би приказали садржај релационе базе података, користимо SELECT. Кад приказујемо садржај целе табеле онда ћемо написати SELECT * FROM потом назив табеле чији садржај желимо да прикажемо.

```
SELECT * FROM IGRACI
```

Ако приказујемо садржај само неких колона табеле онда ћемо поселе SELECT написати називе колона које желимо да прикажемо, потом FROM и назив табеле чији садржај приказујемо.

```
SELECT ime,prezime,pozicija
FROM IGRACI
```

IME	PREZIME	POZICIJA
Ognjen	Vasiljevic	Krilo
Nikola	Vucinic	Centar
Ognjen	Kovacevic	Krilo
Aleksa	Markovic	Centar
Stefan	Matijas	Bek suter

Када желимо да прикажемо само неке редове табеле, користимо WHERE део. У SELECT делу написемо називе свих колона које желимо да прикажемо, док у FROM делу само написемо из које табеле узимамо податке. У WHERE делу наведемо услов по којем ће да се издвоје само неки редови.

```
SELECT ime,prezime,pozicija FROM IGRACI
WHERE id_ekipe = 23
```

IME	PREZIME	POZICIJA
Ognjen	Vasiljevic	Krilo
Nikola	Vucinic	Centar
Strahinja	Milic	Bek suter
Uros	Krstic	Bek suter
Luka	Vukadinovic	Plej mejker

Када приказујемо садржај из више табела, потребно је написати SELECT и називе свих колона које желимо да прикажемо. У FROM делу кода потребно је спојити табеле да би нам садршај био приказан. Табеле спајамо помоћу JOIN функције. Прво напишемо назив једне табеле затим кључну реч JOIN и назив друге табеле потом пишемо USING, где упсујемо назив примарног кључа једне табеле. Пише се ид који је у једној табели примарни, а у другој табели страни кључ.

```
SELECT liga, datum_utakmice, ime_delegata FROM UTAKMICE
JOIN IGRACI_UTAKMICE USING(id_utakmice)
```

LIGA	DATUM_UTAKMICE	IME_DELEGATA
Kvalitetna liga	07/04/2017	Stefan Papic
Jedinstvena liga	05/25/2018	Nemanja Maksimovic

Када желимо да прикажемо само неке редове из табеле корстимо WHERE. У случају да у две различите табеле имамо колоне са истим именом, при писању SELECT упита те колоне са истим именом морамо означити из које су табеле, у супротном нам упит неће радити. Испред назива колоне напишемо назив табеле у којој се налази тај колоне.

```
SELECT KLUBOVI.naziv, adresa, broj_telefona, EKIPE.naziv
FROM KLUBOVI JOIN EKIPE USING(id_kluba) JOIN TRENINZI USING(id_ekipe)
WHERE vreme = '21:00'
```

NAZIV	ADRESA	BROJ_TELEFONA	NAZIV
KK "Zarkovo"	Olimpijskih igara 14	233678	Zarkovo kadeti
KK "Crvena Zvezda"	Mali Kalemegdan 2	455621	Crvena Zvezda kadeti

Команда DELETE

За брисање података из базе користимо DELETE. При брисању неког реда из базе написаћемо кључну реч DELETE и потом назив табеле из које желимо да избришемо неки ред, у WHERE делу написаћемо id реда који желимо да обришемо. Унети подаци могу да се обришу овом наредбом.

```
UPDATE TRENINZI
WHERE id_treninga = 001
```

Команда UPDATE

За измену вредности колона табеле користимо UPDATE. Можемо да измени све податке у табели. Када желимо неки податак да изменимо прво уписујемо UPDATE па назив табеле где желимо да изменимо податак, у SET делу напишемо назив колоне који желимо да променимо и стављамо нови податак који желимо да пише, у WHERE делу уписујемо како је тај податак написан у нашој табели. Унети подаци могу увек бити промењени. На овом примеру се то види.

```
UPDATE UTAKMICE
SET datum_utakmice = '07.04.2017'
WHERE id_utakmice = 060417
```

INSERT наредба

За додавање података у табелу користи се SQL наредба INSERT...INTO. Наредба INSERT...INTO додаје нове врсте у табелу релационе базе података. Вредности колона се дефинишу задавањем вредности у облику константи или коришћењем резултата SQL упита. У зависности од начина задавања вредности колона постоје различити облици INSERT...INTO наредбе. Прво напишемо INSERT INTO и назив табеле где желимо да додамо нови ред, потом у VALUES уносимо податке истим редом као кад смо креирали табелу. Ако нам је поље типа VARCHAR2 онда податак пишемо између апострофа, када нам је податак опциони и то поље уколико не желимо да попунимо онда уписујемо реч нулл. Када нам је поље типа NUMBERонда само уписујемо одређен број који нам треба. Када нам је поље типа DATE, датум се уписује између апострофа.

```
INSERT INTO KLUBOVI(id_kluba,naziv,adresa,broj_telefona,mail)
VALUES(13122,'KK "Zarkovo"', 'Olimpijskih igara 14', 233678, 'zarkovokk@gmail.com')
```

Када попуњавамо нови ред у табели која садржи стране кључеве других табела, онда уносимо вредност примарног кључа једне табеле и вредност примарног кључа друге табеле које желимо да повежемо, уколико табелела садржи и друге атрибуте њих уносимо у зависности од типа података.

```
INSERT INTO IGRACI_UTAKMICE(id_igraca, id_utakmice)
VALUES(109, 060417)
```

Наредбе за контролу функције

Постоје још неке наредбе које користимо за контролне функције.

- GRANT
- REVOKE
- COMMIT
- ROLLBACK

Претрага помоћу PL/SQL језика

Сам PL/SQL језик

PL / SQL има много више опција за обраду базе података од самог језика SQL. Он представља поцедурални језик који је проширење за SQL за релационе базе податакта компаније Орацле. Назив је настао као скраћеница за: „Процедурал Лангуаге фор SQL„ . Помоћу PL / SQL могу се креирати неименовани блокови, функције, процедуре и окидачи. Омогућује декларацију и коришћењеконстанти и променљивих. Када пишемо програме помоћу овог језика користимо наредбе језика SQL за директан рад са базом података, али имамо на располагању и друге елементе процедуралних језика као што су:

- Гранања: IF и CASE
- Циклуси: LOOP, FOR и WHILE

SELECT INTO наредба

Ево и неких задатака

1. Приказати број утакмица које се играју у Квалитетној лиги

```
DECLARE
v_broj NUMBER;
BEGIN
SELECT COUNT(*) INTO v_broj
FROM UTAKMICE WHERE liga ='Kvalitetna liga';
DBMS_OUTPUT.PUT_LINE('Broj utakmica Kvalitetne lige je: '||' '||v_broj);
END;
```

```
Broj utakmica Kvalitetne lige je:  3

Statement processed.
```

2. У овом задатку за одређено име и презиме играча добијамо број утакмица које је играо.

   Користимо SELECT COUNT. Корисник уноси име и презиме играча за ког жели да сазна број одиграних утакмица. Морамо да декларишемо ime, prezime и променљиву v_broj у којој ће се то рачунати. COUNT функцијом ћемо то и бројати. На крају покажемо ime, prezime и сами broj утакмица.

```
DECLARE
v_ime VARCHAR(30);
v_prezime VARCHAR2(30);
v_broj NUMBER;
BEGIN
v_ime:=:Ime;
v_prezime:=:Prezime;
SELECT COUNT(*) INTO v_broj
FROM UTAKMICE JOIN IGRACI_UTAKMICE USING(id_utakmice)
JOIN IGRACI using(id_igraca)
WHERE (ime)|| (prezime)
LIKE '%'||(v_ime)||(v_prezime)||'%';
DBMS_OUTPUT.PUT_LINE('Igrač: '||' '||v_ime||' '||v_prezime);
DBMS_OUTPUT.PUT_LINE('Broj utakmica koje je igrao: '||' '||v_broj);
END;
```

Када покренемо програм, као резултат добијамо информације о играчу чије име и презиме смо унели са тастатуре. Такође нам се и показује број одиграних утакмица што је и био циљ задатка.

```
Igrač:  Bosko Vlahovic
Broj utakmica koje je igrao:  2

Statement processed.
```

Рад са курсорима

3. Приказати име и презиме заједно под једним именом и број телефона за све тренере.

   У овом примеру ћемо користити CURSOR. Прво што треба да урадимо јесте да декларишемо све податке које желимо касније да прикажемо. Треба да декларишемо тренера(име и презиме истог) и број телефона. При декларисању наводимо тип податка, у овом случају је то VARCHAR2 и треба да унесемо број карактера, односно ограничење. Уместо типа података можемо да користимо и %TYPE после назива табеле и колоне које желимо да прикажемо. Кориситмо само један курсор. Пмоћу DBMS_OUTPUT.PUT_LINE приказујемо појединачне појединачно сваки податак.

```
DECLARE
  v_trener VARCHAR2(150);
  v_broj_telefona TRENERI.broj_telefona%TYPE;
  CURSOR kursor_trener IS SELECT ime||' '||prezime, broj_telefona FROM TRENERI;
BEGIN
   OPEN kursor_trener;
   LOOP
    FETCH kursor_trener INTO v_trener, v_broj_telefona;
    EXIT WHEN kursor_trener%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE('Ime i prezime trenera: '||v_trener);
    DBMS_OUTPUT.PUT_LINE('Telefon: '|| v_broj_telefona);
   END LOOP;
   CLOSE kursor_trener;
END
```

Након тачно извршења програма као резултат добијамо све жељене информације. Име и презиме (као један податак) и број телефона. Сваки податак се приказује у новом реду.

```
Ime i prezime trenera: Zlatko Polak
Telefon: 65556987
Ime i prezime trenera: Miroslav   Kljajic
Telefon: 62669339
Ime i prezime trenera: Nebojsa  Peric
Telefon: 61233548
Ime i prezime trenera: Srdjan Maksic
Telefon: 69788745
Ime i prezime trenera: Nikola  Kitanovic
Telefon: 63598633

Statement processed.
```

4. Ako играч има id већи или једнак од 106 прикажи његово име, презиме и id, уз поруку да је играч играо утакмицу. У супротном напиати поруку да играч није играо утакмицу.

```
DECLARE
  CURSOR kursor_igraci IS SELECT ime, prezime, id_igraca FROM IGRACI;
BEGIN
  FOR v_red IN kursor_igraci LOOP
    DBMS_OUTPUT.PUT_LINE('Ime i prezime: '||v_red.ime || ' ' ||v_red.prezime);
    DBMS_OUTPUT.PUT_LINE('ID_IGRACA: '|| v_red.id_igraca);
    IF v_red.id_igraca < 106 THEN
      DBMS_OUTPUT.PUT_LINE('Igrač nije igrao utakmicu');
    ELSIF v_red.id_igraca >= 106 THEN
      DBMS_OUTPUT.PUT_LINE('Igrač je igrao utakmicu');
    END IF;
    DBMS_OUTPUT.PUT_LINE('- - -');
  END LOOP;
END
```

Овде корситимо FOR циклус ради лакшег проласка кроз податке. Помоћу IF услова имамо опцију да тржимо податке по нечему специфичном у више случајева.THEN функције нам само говори да ако податак не ,,упада,, у први услов да пређе на селдећи. ELSIF отвара следећи услов. END IF затвара услове. DBMS_OUTPUT.PUT_LINE('- - -') је ту само ради прегледности што се види у решењу.

Ево га и решење:

```
Ime i prezime: Uros  Krstic
ID_IGRACA: 109
Igrač je igrao utakmicu
- - -
Ime i prezime: Luka Vukadinovic
ID_IGRACA: 101
Igrač nije igrao utakmicu
- - -
Ime i prezime: Petar Veruseski
ID_IGRACA: 102
Igrač nije igrao utakmicu
- - -
Ime i prezime: Aleksa Kovacevic
ID_IGRACA: 104
Igrač nije igrao utakmicu
- - -
Ime i prezime: Strahinja Jovanovic
ID_IGRACA: 107
Igrač je igrao utakmicu
```

Курсори са параметром

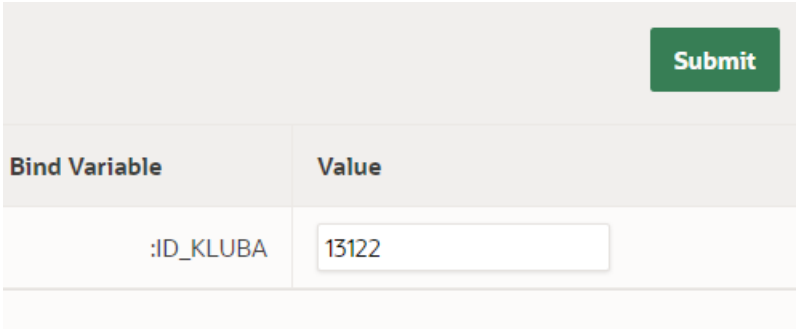
5. Уносимо id_kluba са тастатуре и за тај клуб добијамо id_ekipe и њен naziv. Код ће излистати све екипе клуба чији се id_kluba уноси. У овом случају је то само једна екипа.

У овом примеру користимо курсор са параметорм, а као параметар узимамо id_kluba где за одређени унети податак (id_kluba). За одређени клуб добијамо id_ekipe и naziv екипе у том клубу.

Ево и примера:

```
DECLARE
  v_id_kluba KLUBOVI.id_kluba%TYPE;
  CURSOR kursor_klub (p_id_kluba KLUBOVI.id_kluba%TYPE)
    IS SELECT id_ekipe,naziv FROM EKIPE
    WHERE id_kluba = p_id_kluba;
BEGIN
  v_id_kluba := :id_kluba;
  FOR v_red_ekipe IN kursor_klub(v_id_kluba) LOOP
    DBMS_OUTPUT.PUT_LINE('   ID EKIPE: '||v_red_ekipe.id_ekipe);
    DBMS_OUTPUT.PUT_LINE('   Naziv ekipe: '||v_red_ekipe.naziv);
  END LOOP;
END
```

Унос:



А резултат је:

```
   ID EKIPE: 23
   Naziv ekipe: Zarkovo kadeti

Statement processed.
```

6. У овом задатку исто користимо параметар, али се подаци не уносе са тастатуре. За сваки клуб добијамо његову адресу, број телефона и назив екипе у том клубу.

```
DECLARE
  CURSOR kursor_klub
    IS SELECT id_kluba, adresa, broj_telefona FROM KLUBOVI;
  CURSOR kursor_ekipa (p_id_kluba KLUBOVI.id_kluba%TYPE)
    IS SELECT id_ekipe, naziv FROM EKIPE
    WHERE id_kluba=p_id_kluba;
BEGIN
  FOR v_red_klub IN kursor_klub LOOP
    DBMS_OUTPUT.PUT_LINE('Adresa: '||v_red_klub.adresa);
    DBMS_OUTPUT.PUT_LINE('Telefon: '|| v_red_klub.broj_telefona);
    FOR v_red_ekipa IN kursor_ekipa(v_red_klub.id_kluba) LOOP
      DBMS_OUTPUT.PUT_LINE('Naziv ekipe: '||v_red_ekipa.naziv);
       DBMS_OUTPUT.PUT_LINE('---');
    END LOOP;
  END LOOP;
END
```

Решење за овај задатка је:

```
Adresa: Olimpijskih igara 14
Telefon: 233678
Naziv kluba: Zarkovo kadeti
---
Adresa: Mali Kalemegdan 2
Telefon: 455621
Naziv kluba: Crvena Zvezda kadeti
---

Statement processed.
```

Процедуре

7. Правимо процедуру где ћемо за унети id_kluba са тастатуре, добијамо назив, адресу и број телефона истог.

   За базу података са много клубова је ово јако корисно јер се као пааараметар узима id_kluba.

```
CREATE OR REPLACE PROCEDURE
klub_podatci(p_id_kluba KLUBOVI.id_kluba%TYPE,  p_naziv OUT KLUBOVI.naziv%TYPE,
 p_adresa OUT KLUBOVI.adresa%TYPE,
p_broj_telefona OUT KLUBOVI.broj_telefona%TYPE) AS
BEGIN
  SELECT naziv, adresa, broj_telefona INTO p_naziv, p_adresa, p_broj_telefona
  FROM KLUBOVI WHERE id_kluba = p_id_kluba;
  EXCEPTION
  WHEN NO_DATA_FOUND THEN
  DBMS_OUTPUT.PUT_LINE('Nema tog kluba');
END
```

   Потврда о направљеној процедури:

```
Procedure created.
```

   Решење за id_kluba = 13122:

```
Naziv kluba: KK "Zarkovo"
Adresa: Olimpijskih igara 14
Telefon: 233678

Statement processed.
```

У програму смо ставили да ако не постоји id_kluba да избаци поруку да клуб не постоји.

```
Nema tog kluba
Naziv kluba:
Adresa:
Telefon:

Statement processed.
```

Функције

8. Правимо функцију које ће на основу id_kluba показати његово име, адресу, број телефона и маил.

Слично је као процедуре, јер такође правимо функцију и касније позивамо.

Прављење функције:

```
CREATE OR REPLACE FUNCTION klubovi_funkcija(p_id_kluba KLUBOVI.id_kluba%TYPE)
RETURN KLUBOVI%ROWTYPE AS
  podatci_o_klubu KLUBOVI%ROWTYPE;
BEGIN
  SELECT * INTO podatci_o_klubu FROM KLUBOVI
  WHERE id_kluba = p_id_kluba;
  RETURN podatci_o_klubu;
END
```

Позив функције:

```
DECLARE
  v_klubovi KLUBOVI%ROWTYPE;
BEGIN
  v_klubovi := klubovi_funkcija(13122);
  DBMS_OUTPUT.PUT_LINE('Naziv: '|| v_klubovi.naziv);
  DBMS_OUTPUT.PUT_LINE('Adresa: '|| v_klubovi.adresa);
  DBMS_OUTPUT.PUT_LINE('Broj telefona: '||v_klubovi.broj_telefona);
  DBMS_OUTPUT.PUT_LINE('Mail: '||v_klubovi.mail);
END
```

Решење:

```
Naziv: KK "Zarkovo"
Adresa: Olimpijskih igara 14
Broj telefona: 233678
Mail: zarkovokk@gmail.com

Statement processed.
```

Функције су такође битне, јер што их више имамо посао је лакши. Довољно је да је позовемо и све је ту. Само прављење траје, али касније коришћење ја захвално.

Закључак

Да би база података за кошаркашки клуб нормално функционисала мора бити прегледна. Сваки податак мора бити пажљиво унет. Све наредбе, функције и слично ради те прегледности би требало да буде написана великим словима. Оваква база податак олакшава сам рад клуба. База садржи све потребне информације. Помоћу овакве базе олакшавамо сам рад и преглед рада кошаркашког клуба.

Подаци могу увек да се промене или накнадно унесу. Број играча, клубова и осталих елемната једне базе се стално мењају. Зато је поребно ажурирати базу.

У овом раду сви прикази су урађени у APEX алату који ради са релационим базама података. Прграмирање је у овом алату се реализује помоћу SQL и PL/SQL који је проширена верзија SQL језика.

За лакше руковање базом правимо функције и процедуре. Помоћу њих можемо врло лако и брзо да извучемо неке битне информације. Рад са самим функција и процедурама побољшава рад јер га убрзава.

Литература

- https://apex.oracle.com/en/
- Материјали за предмет ,,Модели и базе података,, на систему https://learning.rg.edu.rs/, школска 2020/2021. година.
- Материјал са часова ,,Модели и базе података,,
- https://sr.wikipedia.org/sr-el/Релационе_базе_података
- https://en.wikipedia.org/wiki/PL/SQL
- https://sr.wikipedia.org/wiki/SQL