

Računarska gimnazija

**Maturski rad**  
iz predmeta  
Modeli i baze podataka

**Baza podataka za  
turističku agenciju**

Učenik

Jana Mirčić IV1

Mentor

Duša Vuković

**Beograd, maj 2021.**

# Sadržaj

<b>Uvod .....</b>	<b>3</b>
<b>Modelovanje .....</b>	<b>4</b>
Opis poslovanja.....	4
Opis modela .....	4
Mapiranje.....	6
Uzorak podataka .....	7
<b>Kreiranje baze.....</b>	<b>8</b>
Naredba CREATE TABLE .....	9
Naredba ALTER TABLE.....	13
Naredba DROP TABLE .....	13
Naredba INSERT INTO .....	13
Naredba UPDATE .....	14
Naredba DELETE FROM.....	14
<b>SELECT upiti .....</b>	<b>14</b>
Funkcije .....	16
<b>Programiranje u PL/SQL .....</b>	<b>16</b>
Ukratko o jeziku PL/SQL.....	16
Neimenovani PL/SQL blokovi .....	17
Procedure.....	20
Funkcije .....	21
<b>Zaključak.....</b>	<b>22</b>
<b>Literatura .....</b>	<b>22</b>

## Uvod

**Relaciona baza podataka** predstavlja sistem međusobno povezanih tabela određenim relacijama. Dijagram (Entity Relationship Diagram-ERD) relacione baze je detaljan prikaz podataka preko skupa entiteta, njihovih atributa i međusobnih veza. Svaki entitet ima svoje instance, jedan konkretni primerak. Entiteti su opisani atributima koji mogu biti označeni sa tri simbola: #-primarni jedinstveni identifikator, \*-obavezan atribut i o-opcioni atribut.

Za vezu više više postavlja se novi entitet koji je povezan sa druga dva entiteta. Novi entitet povlači primarne ključeve drugih entiteta. Tabela novog entiteta može da sadrži i dodatne podatke.

Naziv tabele: smeštaji	
id_aranzmana	id_hotela
11	118
12	122
12	115
12	114
12	112
13	124
13	120

Naziv tabele: aranžmani				
id_aranzmana	cena	broj_dana	vrsta_prevoza	opis
11	2200	10	avion	Letovanje u Italiji, Rim, pun pansion
12	4000	12	autobus	Put do Španije, dve noći u Beču, tri noći u Berlinu, dve noći u Parizu i četiri noći u Barseloni
13	1900	8	avion	Tri noći u Ljubljani i četiri noći u Veneciji

Naziv tabele: hoteli				
id_hotela	naziv	adresa	broj_zvezdica	id_destinacije
111	May Ramblas	Carrer del Cardenal Casañas, 6	3	3
112	W Barcelona	Plaça Rosa Del Vents 1, Final, Passeig de Joan de Borbó	5	3
113	Hotel Prelude Opera	8 Cité Bergère	4	1
114	Pullman Paris Eiffel Tower	18 Avenue De Suffren, 22 Rue Jean Rey Entrée Au	4	1
115	Riu Plaza Berlin	Martin-Luther-Str. 1	4	7

# Modelovanje

## Opis poslovanja

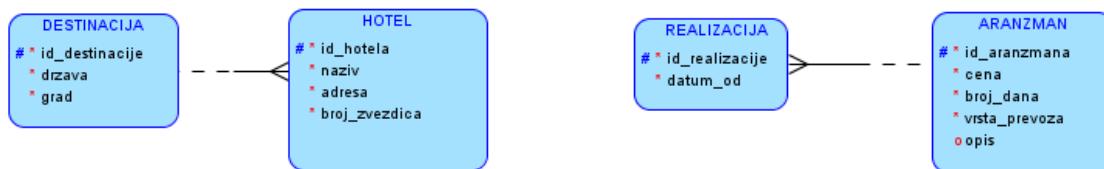
Turistička agencija pruža širok odabir poseta raznim državama. Prilikom rezervacije odredjenog aranžmana putnik unosi svoje lične podatke radi evidencije. Čuva se datum početka putovanja i broj dana provedenih na putu, pomoću kojih se računa datum povratka. Putnikov JMBG je jedinstven i pomoću njega se mogu pretražiti sve rezervacije odredjenog putnika.

## Opis modela

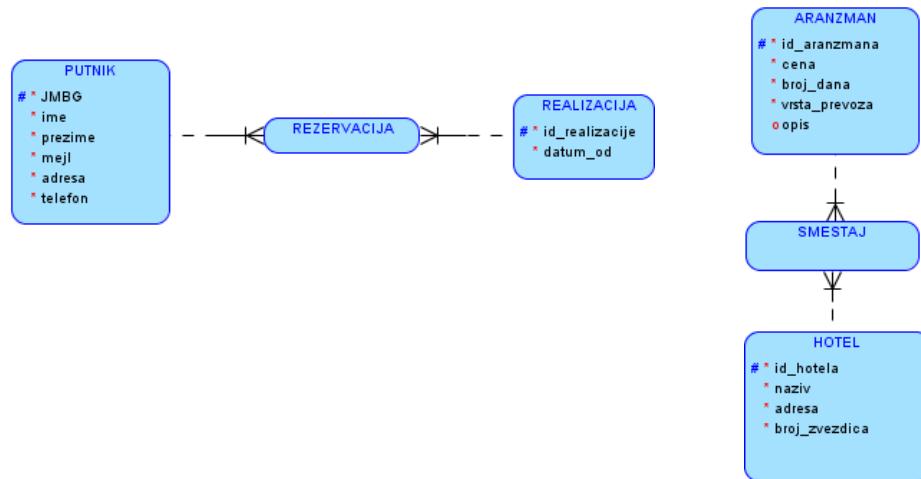
Prvo pravimo entitete koji su nezavisni od drugih, to su entiteti PUTNIK, ARANZMAN i DESTINACIJA. Entitet PUTNIK sadrži atribut JMBG koji je primarni ključ i označavamo ga sa #. Drugi potrebni atributi su ime, prezime, mejl, adresa i telefon. Svi su obavezni atributi pa ih označavamo sa \*. Entitet ARANZMAN ima atribute id\_aranzmana koji je primarni ključ, i ostale atribute cena, broj\_dana, vrsta\_prevoza koji su obavezni atributi, takođe i opis koji je opcionalni atribut pa je obeležen sa o. Entitet DESTINACIJA sadrži atribute id\_destinacije koji je primarni ključ, država i grad koji su obavezni atributi.



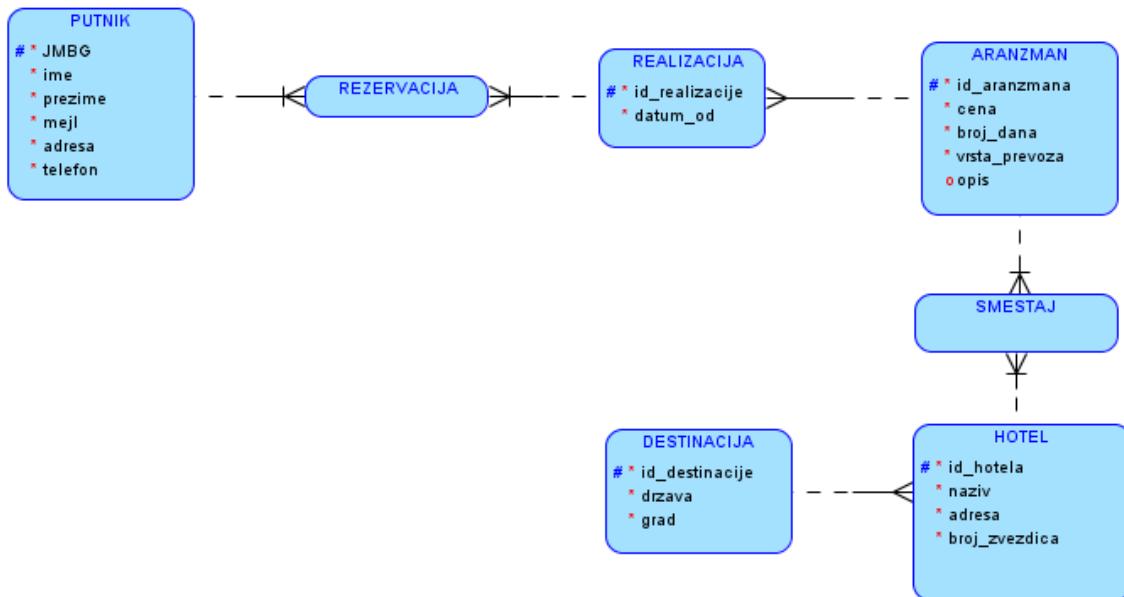
Entitet REALIZACIJA sadrži atribute id\_realizacije kao primarni ključ i datum\_od. Jedan aranžman može da sadrži više realizacija, ali jedna realizacija sadrži tačno jedan aranžman, pa to povezujemo vezom jedan na prema više(1:M). Entitet HOTEL ima atribute id\_hotela kao primarni ključ i ostale naziv, adresa i broj\_zvezdica. Jedna destinacija ima više hotela, ali jedan hotel može da se nalazi samo na jednoj destinaciji, pa to povezujemo vezom jedan na prema više(1:M).



Putnik može da ima više realizacija, ali i više realizacija može da ima jedan isti putnik. Ovaj problem rešavamo dodavanjem novog entiteta REZERVACIJA koji je veza više(M:M) i sadrži primarne ključeve entiteta PUTNIK i REALIZACIJA. Jedan hotel može da se sadrži u planu više aranžmana, takođe i više aranžmana mogu da imaju isti hotel. Ovo rešavamo kreiranjem novog entiteta SMESTAJ koji sadrži primarne ključeve oba entiteta HOTEL i ARANZMAN i čini vezu više(M:M).

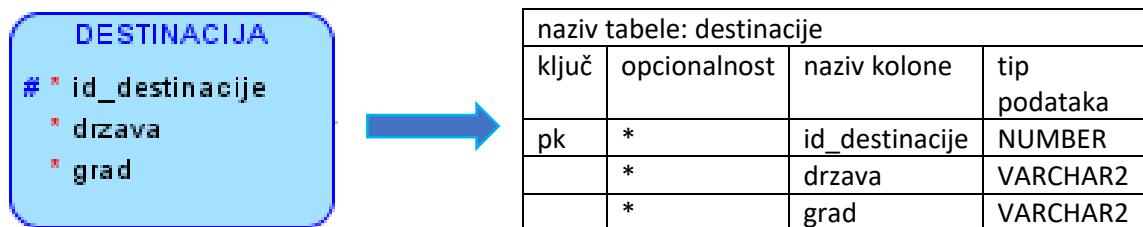


Prikaz celog modela

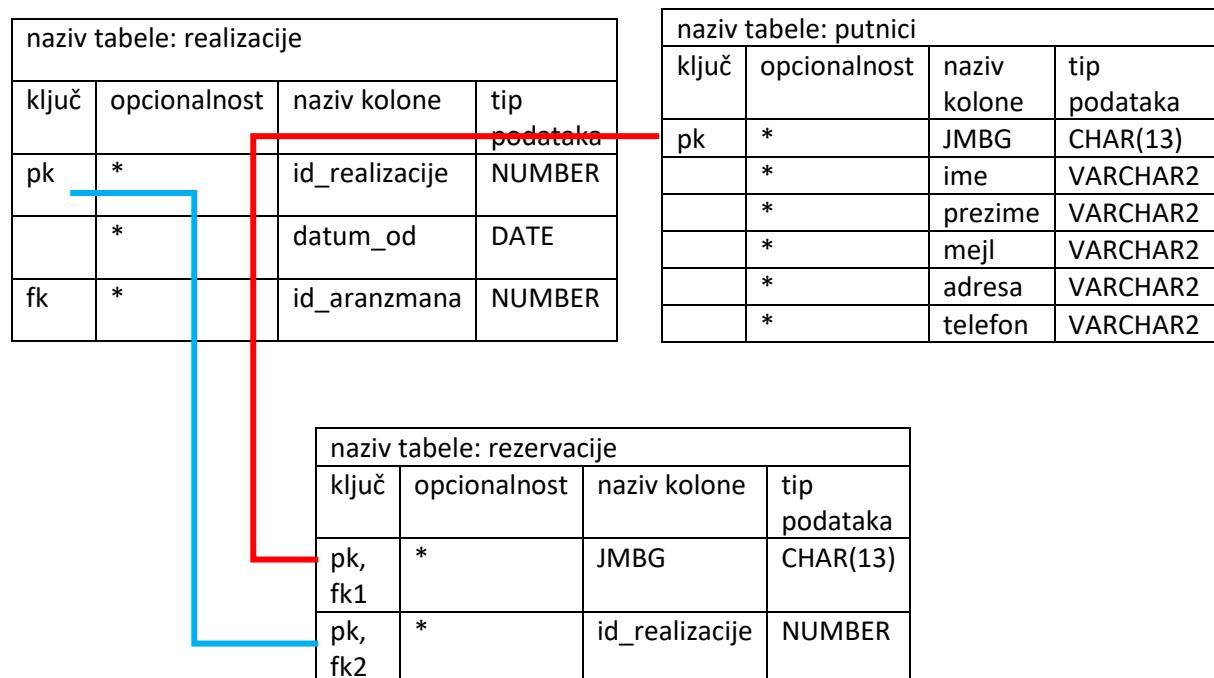


## Mapiranje

Nakon kreiranja logičkog modela, dijagrama entiteta i veza potrebno je da se uradi priprema za kreiranje baze podataka, odnosno mapiranje. Mapiranje podrazumeva detaljan opis svih tabela koje će imati baza. Opis svake tabele mora da ima spisak kolona, njihov tip podataka i spisak svih ograničenja (primarni i strani ključevi, jedinstvenost i not null). Prva kolona ima naziv ključ, gde upisujemo primarni i strani ključ. Ukoliko popunjavamo primarni ključ upisujemo pk, a ukoliko je strani ključ upisujemo fk. Druga kolona je optionalnost i u nju upisujemo kao na dijagramu, može da bude upisana \*-obavezan atribut ili o-optionalni atribut. Treća kolona je naziv kolone gde upisujemo imena atributa iz svakog entiteta. Četvrta je tip podataka u koju upisujemo kog tipa je svaki atribut, neki od najčešćih tipova podataka su NUMBER, VARCHAR2, CHAR, CLOB, DATE, CURRENCY.



Ukoliko entitet ima vezu jedan na prema više u tabelu se dodaje strani ključ iz susednog entiteta i on se upisuje kao fk. Ako su entiteti povezani vezom više više tabela entiteta koji ih povezuje imaće id od oba entiteta kao primarni i kao strani ključ i označavaće se sa pk i fk.



## Uzorak podataka

**PUTNICI**

EDIT	JMBG	IME	PREZIME	MEjl	ADRESA	TELEFON
	2510989547296	Ana	Vuckovic	vuckovica@gmail.com	Francuska 34	0636752839
	1704991545455	Luka	Savic	savicl@outlook.com	Ace Joksimovic 7	0642678390
	0312993561142	Lazar	Filipovic	lazarfilip@hotmail.com	Nemanjina 5	0623457612
	0403996548883	Andjela	Jovanovic	andjelajo@gmail.com	Zmaj Jovina 2	0662735642
	0610992750023	Djordje	Andjelic	andjelic992@gmail.com	Obelskih Radnika 43	0615432876
	1403997258903	Katarina	Antic	antick97@gmail.com	Juria Gagarina 28	0647652958
	2301977061159	David	Boskovic	boskovicc@yahoo.com	Trgovacka 5	0612538946
	0704999105678	Milica	Borkovic	mborkovic@gmail.com	Bulevar Kralja Aleksandra 15	0652215567
	1906998150007	Jovana	Davidovic	jovanadavidovic@gmail.com	Jablanicka 10	0661723846
	1504988672354	Uros	Dimitrijevic	urosdi2@hotmail.com	Dalmatinska 23	0607463529

**DESTINACIJE**

EDIT	ID_DESTINACIJE	DRZAVA	GRAD
	1	Francuska	Pariz
	2	Italija	Rim
	3	Spanija	Barcelona
	4	Italija	Venecija
	6	Slovenija	Ljubljana
	5	Austrija	Bec
	7	Nemacka	Berlin

**ARANZMANI**

EDIT	ID_ARANZMANA	CENA	BROJ_DANA	VRSTA_PREVOZA	OPIS
	12	4000	12	autobus	Put do Spanije, dve noci u Becu, tri noci u Parizu i cetiri noci u Barseloni
	11	2200	10	avion	Letovanje u Italiji, Rim, pun pansion
	13	1900	8	avion	Tri noci u Ljubljani i cetiri noci u Veneciji

**REALIZACIJE**

EDIT	ID_REALIZACIJE	DATUM_OD	ID_ARANZMANA
	22	09/06/2021	12
	25	07/02/2021	11
	27	08/06/2021	13
	29	06/26/2021	13
	20	10/13/2021	12
	21	06/10/2021	11
	28	09/04/2021	11
	30	10/02/2021	11
	23	06/12/2021	13
	24	08/15/2021	12
	26	07/18/2021	12
	31	07/03/2021	12
	32	08/02/2021	11
	33	09/21/2021	13

**REZERVACIJE**

EDIT	JMBG	ID_REALIZACIJE
	031299561142	28
	0405996548883	27
	0610992750023	26
	0704999105678	21
	1403997238903	20
	1504988672354	23
	1704991343455	24
	1704991343455	30
	1906998135007	29
	1906998135007	31
	2301977061139	22
	2301977061139	33
	2510989547296	25
	2510989547296	32

**HOTELI**

EDIT	ID_HOTELA	NAZIV	ADRESA	BROJ_ZVEZDICA	ID_DESTINACIJE
	111	May Ramblas	Carrer del Cardenal Casanás, 6	3	3
	113	Hotel Prelude Opera	8 Cite Bergere	4	1
	118	Atlante Star Hotel	Via Giovanni Vitelleschi, 34	4	2
	123	Hotel Lev	Voršnjakova ulica 1	4	6
	114	Pullman Paris Eiffel Tower	18 Avenue De Suffren, 22 Rue Jean Rey Entrée Au	4	1
	115	Riu Plaza Berlin	Martin-Luther-Str. 1	4	7
	119	Hotel Tre Archi	Cannaregio, 923	3	4
	112	W Barcelona	Plaça Rosa Del Vents 1, Final, Passeig de Joan de Borbo	5	3
	121	Leonardo Hotel Vienna	Matrosengasse 6-8	4	5
	116	Steigenberger Hotel Am Kanzleramt	Ella-Trebe-Strasse 5	5	7
	117	Hotel Serena	Via Principe Amedeo, 64	3	2
	120	Bauer Palazzo	Piscina S. Moise, 1459	5	4
	122	Melia Vienna	Donau-City-Strasse 7	5	5
	124	Hotel Nox	Celovska cesta 469	4	6

**SMESTAJI**

Table		Data	Indexes	Model	Constraints	Grants	Statistics	UI Defaults	Triggers	Dependencies	SQL	REST	Sample Queries	
Query	Count Rows	Insert Row	Load Data											
EDIT		ID_ARANZMANA										ID_HOTELA		
		11											118	
		12											112	
		12											114	
		12											115	
		12											122	
		15											120	
		15											124	

## Kreiranje baze

Nakon analize poslovanja, kreiranog logičkog modela i mapiranja sledi kreiranje same baze. Jezik za rad u sistemima za upravljanje bazama je jezik SQL.

## Naredba CREATE TABLE

Komanda CREATE TABLE počinje nazivom komande praćenom nazivom tabele nakon čega se u zagradama nabrajaju kolone. Za svaku kolonu navodimo naziv i tip podataka, a često dodajemo i ograničenja(NOT NULL, PRIMARY KEY...).

Alat Oracle Application Express(APEX) nam daje mogućnost gotove funkcije CREATE TABLE.

**Create Table**

Columns

\* Table Name: putnici

Preserve Case

Column Name	Type	Precision	Scale	Not Null	Identity	Move
JMBG	CHAR	15		<input checked="" type="checkbox"/>		<input type="button" value="^ \\"/> <input type="button" value=" \\"/>
ime	VARCHAR2	20		<input checked="" type="checkbox"/>		<input type="button" value="^ \\"/> <input type="button" value=" \\"/>
prezime	VARCHAR2	50		<input checked="" type="checkbox"/>		<input type="button" value="^ \\"/> <input type="button" value=" \\"/>
mejl	VARCHAR2	40		<input checked="" type="checkbox"/>		<input type="button" value="^ \\"/> <input type="button" value=" \\"/>
adresa	VARCHAR2	90		<input checked="" type="checkbox"/>		<input type="button" value="^ \\"/> <input type="button" value=" \\"/>
telefon	VARCHAR2	15		<input checked="" type="checkbox"/>		<input type="button" value="^ \\"/> <input type="button" value=" \\"/>

**Create Table**

Primary Key

Table name: PUTNICI

Primary Key:  No Primary Key  
 Populated from a new sequence  
 Populated from an existing sequence  
 Not populated  
 Populated by identity column

\* Primary Key Constraint Name: PUTNICI\_PK

\* Primary Key: JMBG(CHAR)

\* Sequence Name: PUTNICI\_SEQ

**Create Table**

Foreign Key

Foreign Keys

Foreign Key Columns Referenced Table Referenced Columns Action

Add Foreign Key

Name: PUTNICI\_FK

Disallow Delete  
 Cascade Delete  
 Set Null on Delete

Select Key Column(s): JMBG  
IME  
PREZIME  
MEJL  
ADRESA  
TELEFON

## Create Table

Progress: ✓ ✓ ✓ ✓ Constraints

Constraint Name	Type	Column(s)/Check	Add
	<input checked="" type="radio"/> Check <input type="radio"/> Unique	Check Condition	
Key Column(s)	JMBG IME PREZIME MEJL ADRESA TELEFON	↔ ↔> > < <<	↔ ^ ^ v v

## Create Table

Progress: ✓ ✓ ✓ ✓ Confirm

Please confirm your request.

Schema: JMIRCIC17 [?](#)

Table name: PUTNICI [?](#)

> SQL

< Cancel Create Table

```

CREATE TABLE "PUTNICI"
(
    "JMBG" CHAR(13) NOT NULL ENABLE,
    "IME" VARCHAR2(20) NOT NULL ENABLE,
    "PREZIME" VARCHAR2(50) NOT NULL ENABLE,
    "MEJL" VARCHAR2(40) NOT NULL ENABLE,
    "ADRESA" VARCHAR2(90) NOT NULL ENABLE,
    "TELEFON" VARCHAR2(15) NOT NULL ENABLE,
    CONSTRAINT "PUTNICI_PK" PRIMARY KEY ("JMBG")
    USING INDEX ENABLE
)
/

CREATE OR REPLACE EDITIONABLE TRIGGER "BI_PUTNICI"
before insert on "PUTNICI"
for each row
begin
if :NEW."JMBG" is null then
    select "PUTNICI_SEQ".nextval into :NEW."JMBG" from sys.dual;
end if;
end;

/
ALTER TRIGGER "BI_PUTNICI" ENABLE
/

```

Kada kreiramo tabelu koja ima strani ključ neke druge tabele u polju za strani ključ biramo već postojeću tabelu sa id koji je strani ključ našoj novoj tabeli. U kodu pored kolone koja je strani ključ neophodno je dodati reč REFERENCES, a nakon nje naziv tabele i kolone na koju strani ključ pokazuje.

### Create Table

**Columns**

\* Table Name: realizacije (?)

Preserve Case

Column Name	Type	Precision	Scale	Not Null	Identity	Move
id_realizacije	NUMBER		2	<input checked="" type="checkbox"/>	- None -	^ v
datum_od	DATE			<input checked="" type="checkbox"/>		^ v
id_aranzmana	NUMBER		2	<input checked="" type="checkbox"/>	- None -	^ v
	- Select Datatype -					^ v
	- Select Datatype -					^ v
	- Select Datatype -					^ v

### Create Table

**Primary Key**

Table name: REALIZACIJE (?)

Primary Key:  No Primary Key  
 Populated from a new sequence  
 Populated from an existing sequence  
 Not populated  
 Populated by Identity column

\* Primary Key Constraint Name: REALIZACIJE\_PK (?)

\* Primary Key: ID\_REALIZACIJE(NUMBER) (?)

\* Sequence Name: REALIZACIJE\_SEQ (?)

### Create Table

**Foreign Key**

Foreign Keys

Foreign Key	Columns	Referenced Table	Referenced Columns	Action
REALIZACIJE_FK	ID_ARANZMANA	ARANZMANI	ID_ARANZMANA	Default <span>(X)</span>

Add Foreign Key

Name: REALIZACIJE\_FK (?) Add

Select Key Column(s): ID\_REALIZACIJE, DATUM\_OD, ID\_ARANZMANA >> <<

References Table: (?)

Referenced Column(s): >> <<

### Create Table

Progress: Step 1 of 4

Constraint Name	Type	Column(s)/Check
	<input checked="" type="radio"/> Check	

Constraint Type:  Check  Unique

Check Condition:

Key Column(s): ID\_REALIZACIJE, DATUM\_OD, ID\_ARANZMANA

Add

Name: \*

### Create Table

Progress: Step 1 of 4

Please confirm your request.

Schema: JMIRCIC17 [?](#)

Table name: REALIZACIJE [?](#)

> SQL

Cancel Create Table

```

CREATE TABLE "REALIZACIJE"
(
    "ID_REALIZACIJE" NUMBER(2,0) NOT NULL ENABLE,
    "DATUM_OD" DATE NOT NULL ENABLE,
    "ID_ARANZMANA" NUMBER(2,0) NOT NULL ENABLE,
    CONSTRAINT "REALIZACIJE_PK" PRIMARY KEY ("ID_REALIZACIJE")
)
/
ALTER TABLE "REALIZACIJE" ADD CONSTRAINT "REALIZACIJE_FK" FOREIGN KEY ("ID_ARANZMANA")
    REFERENCES "ARANZMANI" ("ID_ARANZMANA") ENABLE
/


CREATE OR REPLACE EDITABLE TRIGGER "BI_REALIZACIJE"
before insert on "REALIZACIJE"
for each row
begin
    if :NEW."ID_REALIZACIJE" is null then
        select "REALIZACIJE_SEQ".nextval into :NEW."ID_REALIZACIJE" from sys.dual;
    end if;
end;

/
ALTER TRIGGER "BI_REALIZACIJE" ENABLE
/

```

## Naredba ALTER TABLE

Ukoliko je pri kreiranju tabele došlo do greške moguće je izvršiti izmene. Pomoću ove naredbe možemo dodati ili obrisati kolonu u tabelu.

```
ALTER TABLE putnici  
ADD (datum_rodjenja DATE)
```

```
ALTER TABLE putnici  
DROP COLUMN prezime
```

## Naredba DROP TABLE

Ova naredba uklanja celu tabelu iz baze, briše kompletan sadržaj (sve podatke iz tabele) i uklanja tabelu kao objekat iz baze.

```
DROP TABLE putnici;
```

## Naredba INSERT INTO

Kada su kreirane tabele podatke unosimo naredbom INSERT INTO.

APEX nam pruža gotovu INSERT INTO komandu za unos podataka.

The screenshot shows the Oracle APEX interface for creating a new row in the 'PUTNICI' table. The table has the following columns and their current values:

Column	Type	Value
Jmbg	Text	1906998135007
Ime	Text	Jovana
Prezime	Text	Davidovic
Mejl	Text	jovanadavidovic@gmail.com
Adresa	Text	Jablanicka 10
Telefon	Text	0661723846

## Naredba UPDATE

Uneti podaci mogu se ažurirati komandom UPDATE. Prvo upisujemo UPDATE pa naziv tabele, nakon SET pišemo naziv kolone koju menjamo i upisujemo novi podatak koji želimo da piše, a nakon WHERE pišemo opet naziv kolone i podatak kakav je trenutno u našoj tabeli.

```
UPDATE putnici
```

```
SET ime = 'Anja'
```

```
WHERE ime = 'Ana'
```

## Naredba DELETE FROM

Podaci iz tabele brišu se komandom DELETE FROM. Nakon naredbe DELETE FROM pišemo tabelu iz koje želimo nešto da obrišemo, a nakon WHERE pišemo id reda koji želimo da obrišemo.

```
DELETE FROM destinacije
```

```
WHERE id_destinacije = 1
```

## SELECT upiti

Pomoću SELECT upita imamo mogućnost efikasne obrade i dobijanja podataka. Ako želimo da prikažemo čitav sadržaj neke tabele koristimo SELECT \* FROM.

```
SELECT * FROM putnici
```

Ukoliko želimo da prikažemo samo određene kolone onda nakon SELECT pišemo nazive kolona koje želimo da dobijemo.

```
SELECT ime, prezime FROM putnici
```

IME	PREZIME
Ana	Vuckovic
Luka	Savic
Lazar	Filipovic
Andjela	Jovanovic

Ukoliko želimo da izdvojimo samo neke redove tabela koristimo WHERE. Nakon SELECT pišemo kolone koje želimo da se prikažu, a nakon WHERE zadajemo uslov koji izdvaja redove koje mi želimo.

```
SELECT id_aranzmana FROM aranzmani WHERE cena < 3000
```

ID_ARANZMANA
11
13

Kada želimo da dobijemo podatke iz dve ili više različitih tabeli koristimo komandu JOIN. Ukoliko se id iz jedne tabele i id iz tabele koja ima strani ključ prve tabele zovu isto koristimo USING, u suprotnom koristimo ON, a u zagradi se upisuje taj id.

```
SELECT naziv, grad FROM hoteli JOIN destinacije USING(id_destinacije)
```

NAZIV	GRAD
May Ramblas	Barcelona
Hotel Prelude Opera	Pariz
Atlante Star Hotel	Rim
Hotel Lev	Ljubljana
Pullman Paris Eiffel Tower	Pariz
Riu Plaza Berlin	Berlin

Kada povezujemo više tabela ali želimo da izdvojimo samo određeno podatke koristimo i JOIN i WHERE. Nakon spajanja tabele korišćenjem JOIN i USING ili ON(id) pišemo WHERE i zadajemo uslov. Primer, ako želimo da prikažemo imena putnika koji su putovali u Barselonu.

```
SELECT ime FROM putnici JOIN rezervacije USING(JMBG) JOIN realizacije  
USING(id_realizacije) JOIN aranzmani USING(id_aranzmana) JOIN smestaji  
USING(id_aranzmana) JOIN hoteli USING(id_hotela) JOIN destinacije  
USING(id_destinacije) WHERE grad = 'Barcelona'
```

IME
Luka
Djordje
Katarina
David
Jovana

## Funkcije

Postoje dve vrste funkcija, jednostavne(primenjuju se na svakom redu rezultata pojedinačno) i grupne(vraćaju jedan rezultat za više redova)

Kada bismo hteli da se za svakog putnika automatski pravi nova mejl adresa od prvog slova imena, prezimena i nastavka '@turisticaagencija.rs' morali bi smo da koristimo funkciju CONCAT(nadovezuje dva prosleđena tekstualna podatka), LOWER(precbacuje sve u mala slova) i SUBSTR(izdvaja deo teksta, navode se pozicija od koje kreće i broj znakova)

```
SELECT CONCAT(LOWER(SUBSTR(ime,1,1)||prezime), '@turisticaagencija.rs')  
"mejl adrese" FROM putnici
```

mejl adrese
avuckovic@turisticaagencija.rs
lsavic@turisticaagencija.rs
lphilovic@turisticaagencija.rs
ajovanovic@turisticaagencija.rs

Ukoliko želimo da prikažemo najskuplji i najjeftiniji aranžman koristimo funkcije MIN i MAX, u zagrade upisujemo od čega uzimamo najmanju i najveću vrednost.

```
SELECT MIN(cena) "najmanja cena", MAX(cena) "najveca cena" FROM aranzmani
```

najmanja cena	najveca cena
1900	4000

## Programiranje u PL/SQL

### Ukratko o jeziku PL/SQL

PL/SQL je proceduralni jezik relacijske baze podataka. Dizajniran je za izgradnju web aplikacije. PL/SQL skraćeno je od engleskog naziva Procedural Language/Sequential Query Language. Pomoću njega mogu se kreirati neimenovani blokovi, procedure, funkcije i okidači.

## Neimenovani PL/SQL blokovi

### SELECT INTO

Kada želimo da prikažemo ime i prezime(spojeno) putnika čiji je JMBG 0704999105678 potrebni su nam podaci iz tačno jednog reda pa koristimo SELECT INTO

```
DECLARE
v_putnik VARCHAR2(150);
BEGIN
SELECT ime||' '||prezime INTO v_putnik
FROM putnici WHERE JMBG = 0704999105678;
DBMS_OUTPUT.PUT_LINE('Ime i prezime putnika: '||v_putnik);
END
```

```
Ime i prezime putnika: Milica Borkovic
```

```
Statement processed.
```

Ukoliko želimo da prikažemo cenu aranžmana datog id koristimo SELECT INTO. Id se unosi sa tastature i za taj zadati id će nam se prikazati cena aranžmana. Prikazujemo dati id i cenu.

```
DECLARE
v_broj aranzmani.id_aranzmana%TYPE;
v_cena aranzmani.cena%TYPE;
BEGIN
v_broj := :id_aranzmana;
SELECT cena INTO v_cena
FROM aranzmani WHERE id_aranzmana=v_broj;
DBMS_OUTPUT.PUT_LINE('Id aranzmana: '||v_broj);
DBMS_OUTPUT.PUT_LINE('Cena datog aranzmana: '||v_cena);
END
```

Kada se program pokrene otvara nam se polje gde unosimo id aranžmana čiju cenu želimo da prikažemo.

```
Id aranzmana: 12
Cena datog aranzmana: 4000
```

### Kursori

Ukoliko želimo da prikažemo imena i prezimena(spojeno) svih putnika moramo da koristimo kurzor. Da bi koristili kurzor neophodni su sledeći koraci:

1. U delu za deklaraciju kreiramo kurzor i vežemo za SELECT upit

2. Otvorimo kurzor u telu PL/SQL bloka
3. U ciklusu se čitaju svi redovi jedan po jedan
4. Kurzor se zatvori

```

DECLARE
    v_putnik VARCHAR2(150);
    CURSOR kurzor_putnik IS SELECT ime||' '||prezime FROM putnici;
BEGIN
    OPEN kurzor_putnik;
LOOP
    FETCH kurzor_putnik INTO v_putnik;
    EXIT WHEN kurzor_putnik%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE('Ime i prezime putnika: '||v_putnik);
END LOOP;
CLOSE kurzor_putnik;
END

```

```

Ime i prezime putnika: Ana Vuckovic
Ime i prezime putnika: Luka Savic
Ime i prezime putnika: Lazar Filipovic
Ime i prezime putnika: Andjela Jovanovic
Ime i prezime putnika: Djordje Andjelic
Ime i prezime putnika: Katarina Antic
Ime i prezime putnika: David Boskovic
Ime i prezime putnika: Milica Borkovic
Ime i prezime putnika: Jovana Davidovic
Ime i prezime putnika: Uros Dimitrijevic

```

Takođe možemo da koristimo promenljivu složenog tipa i da u nju čitamo ceo red. Promenljiva ima onoliko polja koliko ima odgovarajući SELECT upit. Promenljivoj se pristupa tako što napišemo naziv promenljive, tačka i naziv polja. U sledećem primeru želimo da prikažemo imena, prezimena i brojeve telefona svih putnika.

```

DECLARE
    CURSOR kurzor_putnik IS SELECT ime||' '||prezime putnik, telefon FROM putnici;
    v_red kurzor_putnik%ROWTYPE;
BEGIN
    OPEN kurzor_putnik;
LOOP
    FETCH kurzor_putnik INTO v_red;
    EXIT WHEN kurzor_putnik%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE('Ime i prezime putnika: '||v_red.putnik);
    DBMS_OUTPUT.PUT_LINE('Telefon putnika: '||v_red.telefon);
END LOOP;
CLOSE kurzor_putnik;
END

```

```

Ime i prezime putnika: Ana Vuckovic
Telefon putnika: 0636752839
Ime i prezime putnika: Luka Savic
Telefon putnika: 0642678390
Ime i prezime putnika: Lazar Filipovic
Telefon putnika: 0623457612
Ime i prezime putnika: Andjela Jovanovic
Telefon putnika: 0662735642

```

Kursor može i implicitno da se otvori, upotrebom ciklusa FOR čita se red po red

```
DECLARE
    CURSOR kursor_putnik IS SELECT ime||' '||prezime putnik, telefon FROM putnici;
    v_red kursor_putnik%ROWTYPE;
BEGIN
    FOR v_red IN kursor_putnik LOOP
        DBMS_OUTPUT.PUT_LINE('Ime i prezime putnika: '||v_red.putnik);
        DBMS_OUTPUT.PUT_LINE('Telefon putnika: '||v_red.telefon);
    END LOOP;
END
```

Kursor sa parametrom

Kada bi smo hteli da prikažemo datume svake realizacije za svakog putnika morali bi smo da kreiramo kursor sa parametrom koji nazivamo p\_JMBG koji je istog tipa kao kolona putnici.JMBG.

```
DECLARE
    CURSOR kursor_putnik
    IS SELECT JMBG, ime||' '||prezime putnik FROM putnici;
    CURSOR kursor_realizacija (p_JMBG putnici.JMBG%TYPE)
    IS SELECT datum_od FROM realizacije JOIN rezervacije USING(id_realizacije)
    WHERE JMBG = p_JMBG;
BEGIN
    FOR v_red_putnik IN kursor_putnik LOOP
        DBMS_OUTPUT.PUT_LINE('Ime i prezime putnika: '||v_red_putnik.putnik);
        FOR v_red_realizacija IN kursor_realizacija(v_red_putnik.JMBG) LOOP
            DBMS_OUTPUT.PUT_LINE('Datum od kog se ide na put: '||v_red_realizacija.datum_od);
        END LOOP;
    END LOOP;
END
```

```
Ime i prezime putnika: Luka Savic
Datum od kog se ide na put: 08/15/2021
Datum od kog se ide na put: 10/02/2021
Ime i prezime putnika: Lazar Filipovic
Datum od kog se ide na put: 09/04/2021
Ime i prezime putnika: Andjela Jovanovic
Datum od kog se ide na put: 08/06/2021
Ime i prezime putnika: Djordje Andjelic
Datum od kog se ide na put: 07/18/2021
Ime i prezime putnika: Katarina Antic
Datum od kog se ide na put: 10/13/2021
```

## Procedure

Procedure su niz akcija koje se izvršavaju nakon poziva procedure, mogu izvršiti SQL naredbe. Kada bi smo hteli da za dati id aranžmana prikažemo nazine svih hotela koji su u tom aranžmanu koristili bi smo proceduru.

```
CREATE OR REPLACE PROCEDURE spisak_hotela(p_id_aranzmana aranzmani.id_aranzmana%TYPE)
AS CURSOR kursor_hoteli
IS SELECT hoteli.naziv AS naziv
      FROM hoteli JOIN smestaji USING(id_hotela) WHERE id_aranzmana = p_id_aranzmana
      GROUP BY hoteli.naziv;
BEGIN
  FOR v_red_hotel IN kursor_hoteli LOOP
    DBMS_OUTPUT.PUT_LINE('Naziv hotela: '||v_red_hotel.naziv);
  END LOOP;
END
```

Kada je procedura ispravno kreirana treba da izadje poruka Procedure created.

```
Procedure created.
```

U sledećem delu se id aranžmana unosi sa tastature, prikazuje se id aranžmana i opis, a zatim se poziva procedura.

```
DECLARE
  v_id_aranzmana aranzmani.id_aranzmana%TYPE;
  v_aranzman aranzmani%ROWTYPE;
BEGIN
  v_id_aranzmana := :id_aranzmana;
  SELECT * INTO v_aranzman FROM aranzmani
  WHERE id_aranzmana = v_id_aranzmana;
  DBMS_OUTPUT.PUT_LINE('Id aranžmana: '||v_aranzman.id_aranzmana);
  IF v_aranzman.opis IS NOT NULL
  THEN DBMS_OUTPUT.PUT_LINE('Opis: '||v_aranzman.opis);
  END IF;
  spisak_hotela(v_id_aranzmana);
EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Greska');
END
```

Primer za aranžman broj 12

```
Id aranžmana: 12
Opis: Put do Spanije, dve noci u Becu, tri noci u Berlinu, dve noci u Parizu i cetiri noci u Barseloni
Naziv hotela: Pullman Paris Eiffel Tower
Naziv hotela: W Barcelona
Naziv hotela: Riu Plaza Berlin
Naziv hotela: Melia Vienna
```

## Funkcije

Funkcije predstavljaju procedure koje vraćaju neku vrednost. Kada bi smo hteli da za svakog putnika prikažemo broj putovanja koja tek predstoje napisali bi smo funkciju.

```
CREATE OR REPLACE FUNCTION broj_putovanja(p_JMBG putnici.JMBG%TYPE)
RETURN NUMBER
IS
    v_broj NUMBER;
BEGIN
    SELECT COUNT(*) INTO v_broj
    FROM realizacije JOIN rezervacije USING(id_realizacije)
    WHERE JMBG = p_JMBG
    AND datum_od > SYSDATE;
    RETURN v_broj;
END
```

Kada je procedura ispravno kreirana prikazuje se poruka Function created.

```
Function created.
```

Sada koristimo cursor koji se prolazi kroz sve putnike i za svakog se poziva kreirana funkcija.

```
DECLARE
    CURSOR kurzor_putnici IS SELECT JMBG, ime||' '||prezime putnik FROM putnici;
    v_broj NUMBER;
BEGIN
    FOR v_putnik IN kurzor_putnici LOOP
        DBMS_OUTPUT.PUT_LINE('Ime i prezime putnika: '||v_putnik.putnik);
        v_broj:=broj_putovanja(v_putnik.JMBG);
        DBMS_OUTPUT.PUT_LINE('Broj predstoječih putovanja: '||v_broj);
    END LOOP;
END
```

```
Ime i prezime putnika: Ana Vuckovic
Broj predstoječih putovanja: 2
Ime i prezime putnika: Luka Savic
Broj predstoječih putovanja: 2
Ime i prezime putnika: Lazar Filipovic
Broj predstoječih putovanja: 1
Ime i prezime putnika: Andjela Jovanovic
Broj predstoječih putovanja: 1
Ime i prezime putnika: Djordje Andjelic
Broj predstoječih putovanja: 1
```

## Zaključak

Za dobro poslovanje turističke agencije vrlo je važno da baza podataka bude pregledna. Sve podatke unosimo u tabele koje moramo dobro povezati radi kasnijih upita. Dobro odradena baza dosta olakšava rad neke turističke agencije. Svakoj agenciji važno je da čuva podatke o putnicima radi evidencije i slanja promotivnih materijala.

U bazi lako možemo brisati i dodavati nove podatke. Turističke agencije često prave nove aranžmane za koje nekad treba dodati novu državu i grad u bazu. Sve te izmene možemo lako odrediti.

U ovom radu baza je rađena na APEX platformi za razvoj aplikacija baziranih na webu čiji rad je usko povezan sa serverom Oracle baze podataka.

## Literatura

- <https://apex.oracle.com/en/>
- Materijali sa nastave predmeta Modeli i baze podataka, Računarska gimnazija, školska 2020/2021 <https://learning.rg.edu.rs>
- [https://sr.wikipedia.org/sr-ec/Релационале базе података](https://sr.wikipedia.org/sr-ec/Релационале_базе_података)
- <http://www.vps.ns.ac.rs/Materijal/mat794.pdf>