



# **Projektovanje i implementacija Oracle® baze podataka za "MinMax A.D."**

©2010 Darko Petrović  
[www.darkopetrovic.com](http://www.darkopetrovic.com)

## Sadržaj

1. Uvod.....	3
2. Poslovni zahtev .....	3
3. Konceptualni model .....	4
4. Relaciona šema.....	5
5. Implementacija .....	6
5.1. Kreiranje tabela i stranih ključeva .....	7
5.2. Punjenje tabela podacima.....	9
5.3. Izrada aplikacije .....	11
5.4. Pokretanje aplikacije .....	17
6. Zaključak.....	18

## 1. Uvod

Predmet projekta je izrada i implementacija baze podataka za samostalno trgovinsko preduzeće MinMax a.d. Preduzeće posluje na teritoriji cele Republike Srbije kroz minimarkete koji se snabevaju robom iz centralnog skladišta preduzeća. Vlasnik preduzeća želi da vodi kompletnu evidenciju o zahtevima minimarketa za robu iz centralnog skladišta, kao i evidenciju o robi koja se dostavlja minimarketima.

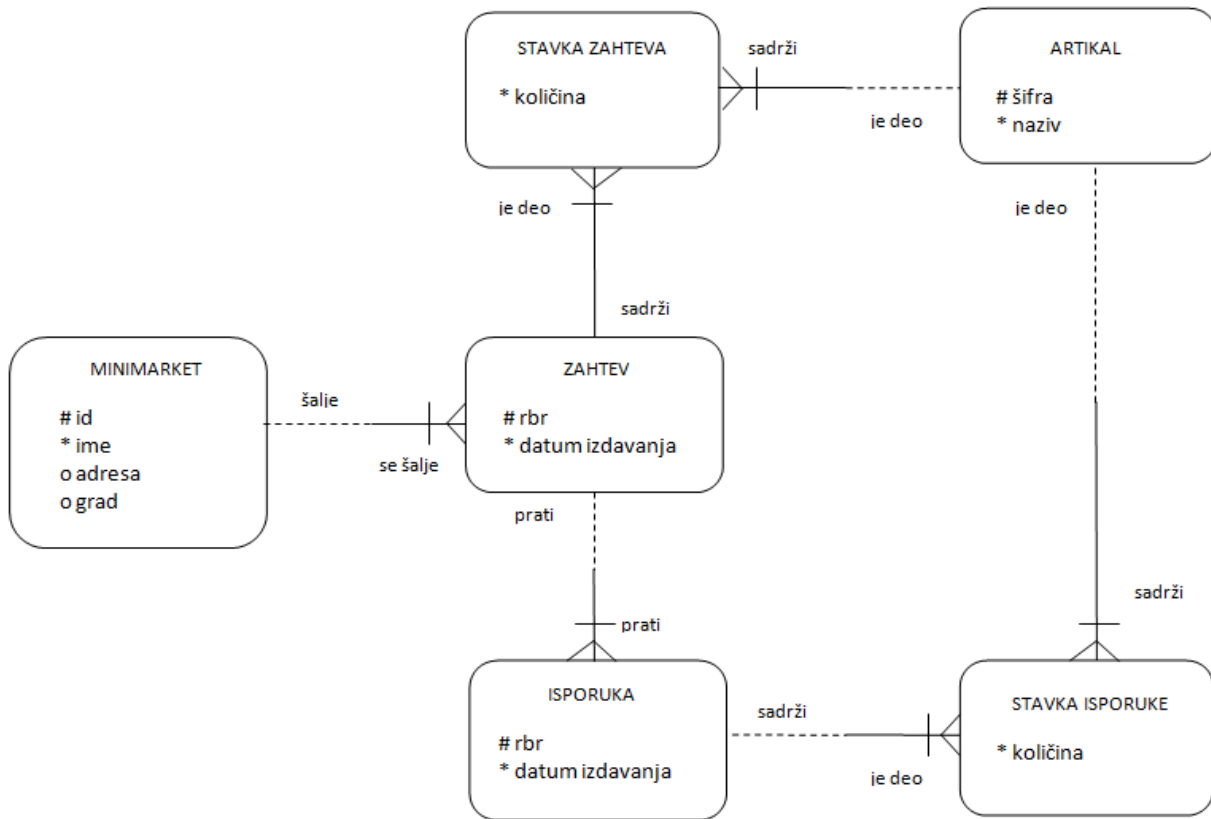
Vlasnik preduzeća očekuje da kompletna implementacija bude završena za mesec dana od dana prihvatanja projekta. U tom periodu bi se izvršila i obuka korisnika aplikacije.

## 2. Poslovni zahtev

Samostalno trgovinsko preduzeće MiniMax a.d. posluje preko svojih minimarketa u više gradova Republike Srbije i ima centralno skladište iz kojeg snabdeva minimarkete. Minimarketi šalju centralnom skladištu zahteve za dostavom robe. Svaki minimarket numeriče svoje zahteve od 1 nadalje i na svaki zahtev upisuje datum izdavanja. Svaki zahtev sadrži spisak sa šiframa, nazivima i količinama traženih artikala. Minimarket može tražiti dostavu nekog artikla svaki put kada količina tog artikla u minimarketu padne ispod određenog nivoa. Za svaki primljeni zahtev centralno skladište izdvaja, pakuje i dostavlja traženu robu kroz jednu ili više isporuka u zavisnosti od stanja zaliha. Svaku isporuku prati otpremnica, koja ima svoj broj i datum. Otpremnica sadži spisak i količine isporučenih artikala. Jedna otpremnica se može odnositi na samo jedan zahtev za isporukom, a jedan artikal se može naći na samo jednoj otpremnici po istom zahtevu.

### 3. Konceptualni model

Na osnovu poslovnog zahteva formiran je konceptualni model baze podataka koji je grafički prikazan na slici 3.1. Osnovna namena baze podataka je evidencija ZAHTEVA koje šalju MINIMARKETI za ARTIKLIMA kao i prikaz ISPORUKA koje se odnose na pomenute ZAHTEVE.



Slika 3.1. ER dijagram za poslovni zahtev preduzeća MinMax a.d.

Sledi objašnjenje nekih veza sa dijagrama.

Pošto je ZAHTEV postavljen kao najbitniji entitet, najpre da pogledamo njegovu vezu sa MINIMARKETOM. Svaki MINIMARKET može da šalje jedan ili više zahteva, a svaki ZAHTEV mora da se šalje od jednog i samo jednog MINIMARKETA. Pošto svaki MINIMARKET numerički svoje zahteve, šifra svakog zahteva će se sastojati od identifikatora (id) MINIMARKETA i rednog broja (rbr) ZAHTEVA.

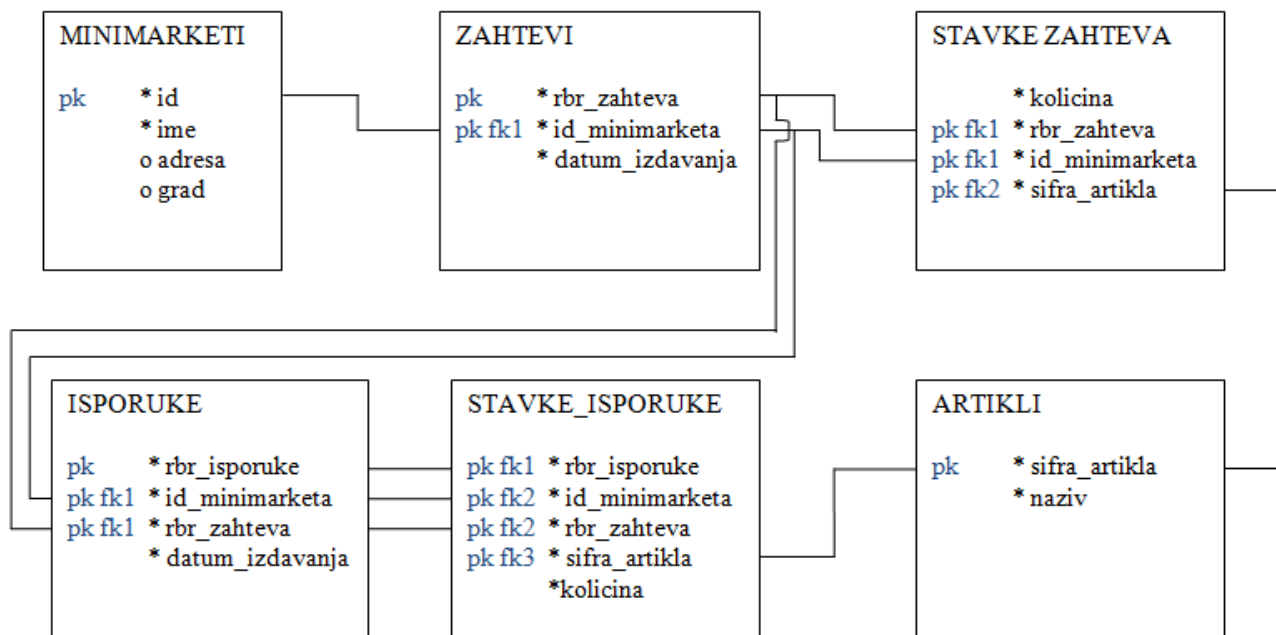
Veza između ZAHTEVA i ARTIKLA je više na prema više (M:M) te sam uveo entitet STAVKA ZAHTEVA koji će rešiti tu nepravilnost. Na isti način razdvojio sam entitete ARTIKAL i ISPORUKA novim entitetom STAVKA ISPORUKE. Oba novouvedena entiteta imaju po jedan mandatorni atribut (količina) a primarni ključ nasleđuju preko bariranih veza.

Informaciju da "minimarket može tražiti dostavu nekog artikla svaki put kada količina tog artikla u minimarketu padne ispod određenog nivoa" ne možemo predstaviti u ERD-u, te ga moramo naknadno dodati u dokumentaciju sistema, ili pak iskoristiti neki proceduralni programski jezik (npr. PL/SQL) da se prevaziđe problem.

## 4. Relaciona šema

Konceptualni model baze podataka se dalje prevodi u relacioni model baze podataka. To znači da će navedeni entiteti, atributi, veze i jedinstveni identifikatori biti prevedeni u objekte relacione baze podataka.

Svaki od entiteta će biti preveden u šemu tabele, atribut u kolonu, jedinstveni identifikatori će postati primarni ključevi (PK – Primary Key), a veze među entitetima će se transformisati u strane ključeve (FK – Foreign Key). Na slici 4.1. je prikazan dijagram relacionog modela baze podataka za MinMax A.D. koji se dobija transformacijom konceptualnog modela (ERD-a) sa slike 3.1.



Slika 4.1. Relacioni dijagram za poslovni zahtev preduzeća MinMax A.D.

## 5. Implementacija

Najpre, evo prikaza izgleda tabela i primera podataka u njima koje će sadržati buduća baza podataka:

Tabela 5.1. MINIMARKETI

id	ime	adresa	grad
1	Minimarket 1	B.Vasovića 1	Čačak
2	Minimarket 2	K.Miloša 2	Beograd
3	Minimarket 3	S.Stepanovića 3	Niš

Tabela 5.2. ZAHTEVI

id_minimarketa	rbr_zahteva	datum_izdavanja
1	1	18-DEC-10
1	2	18-DEC-10
2	1	18-DEC-10
3	1	18-DEC-10

Tabela 5.3. STAVKE\_ZAHTEVA

id_minimarketa	rbr_zahteva	sifra_artikla	kolicina
1	1	101	4
1	1	102	5
1	2	103	6
2	1	104	5
2	2	101	7
3	1	103	16

Tabela 5.4. ARTIKLI

sifra	naziv
101	Brasno
102	Secer
103	Ulje
104	Kuhinjska so

Tabela 5.5. STAVKE\_ISPORUKE

rbr_iskoruke	id_minimarketa	rbr_zahteva	sifra_artikla	kolicina
1	1	1	101	4
2	1	2	102	4
3	1	1	102	1
4	1	1	103	6
5	2	1	104	5
6	2	2	101	7

Tabela 5.6. ISPORUKE

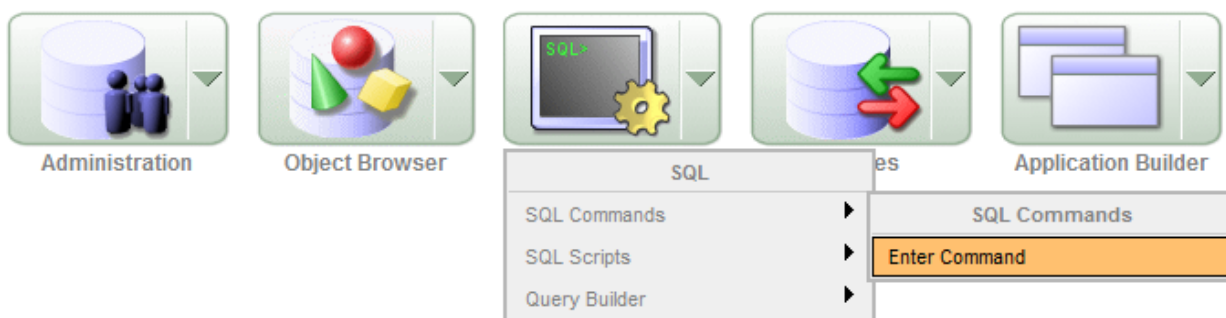
rbr_iskoruke	id_minimarketa	rbr_zahteva	datum_izdavanja
1	1	1	18-DEC-10
2	1	2	18-DEC-10
3	1	1	18-DEC-10
4	1	1	18-DEC-10
5	2	1	18-DEC-10
6	2	2	18-DEC-10

Baza podataka za MinMax A.D. implementirana je korišćenjem *Oracle Application Express* (u daljem tekstu APEX) alata. Implementacija je izvršena kroz nekoliko koraka.

## 5.1. Kreiranje tabela i stranih ključeva

Postupak kreiranja tabela izvršen je unošenjem SQL komandi.

Nakon logovanja na APEX pokrenuta je ikona SQL/SQL Commands/Enter Command (Slika 5.1.).



Slika 5.1. Početna stranica APEX-a

Kreirano je ukupno **6** tabela, na osnovu relacionog modela. SQL kôd na osnovu koga su kreirane table je sledeći:

- 1 | 

```
CREATE TABLE minimarketi
(id NUMBER(6) CONSTRAINT min_id_pk PRIMARY KEY,
ime VERCHAR2(40) NOT NULL,
adresa VARCHAR2(100),
grad VARCHAR2(30));
```
- 2 | 

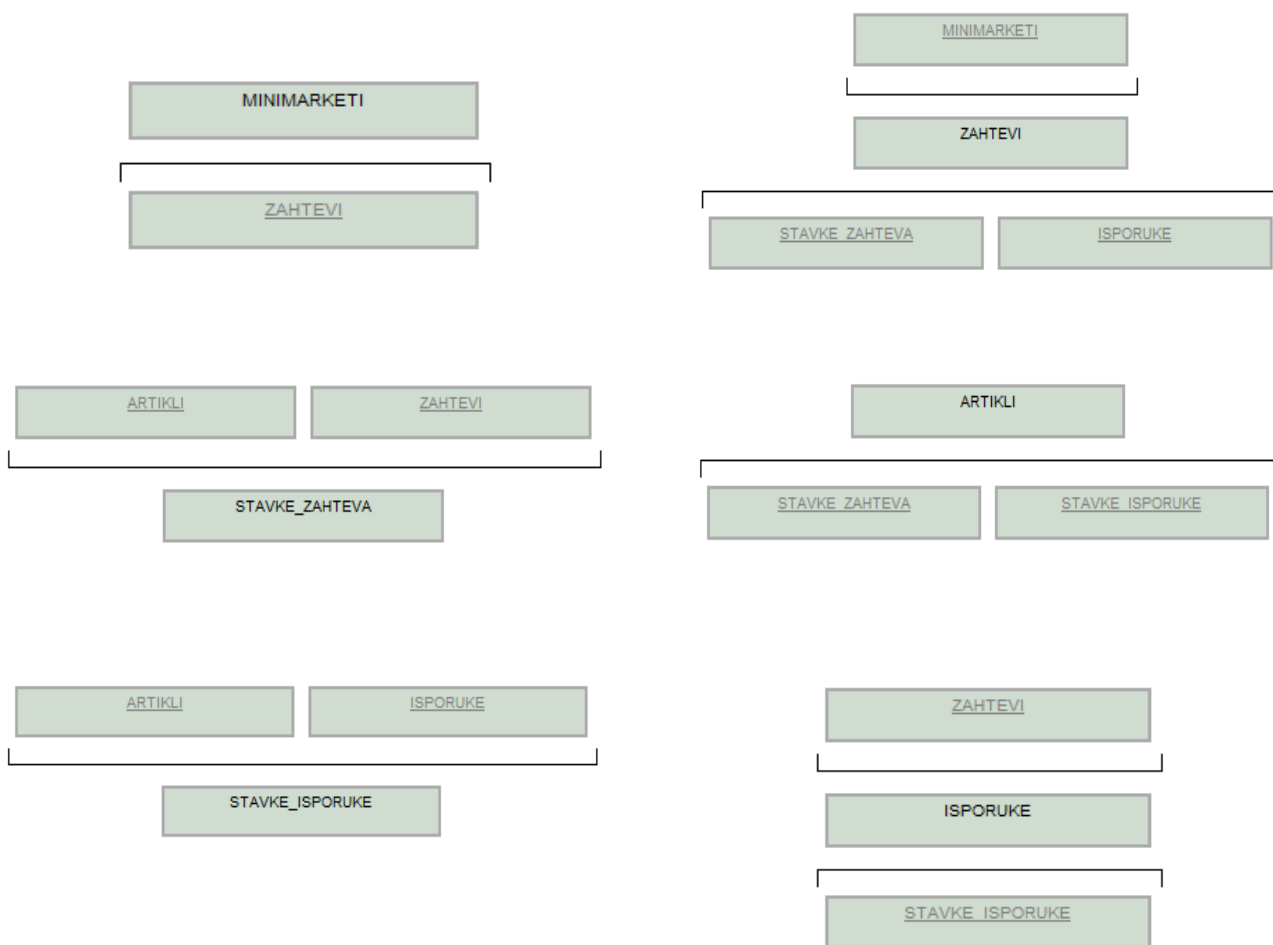
```
CREATE TABLE zahtevi
(id_minimarketa NUMBER(6),
rbr_zahteva NUMBER(6),
datum_izdavanja DATE,
CONSTRAINT zaht_pk PRIMARY KEY(id_minimarketa, rbr_zahteva),
CONSTRAINT zaht_min_fk FOREIGN KEY (id_minimarketa) REFERENCES minimarketi(id));
```
- 3 | 

```
CREATE TABLE stavke_zahteva
(id_minimarketa NUMBER (6),
rbr_zahteva NUMBER (6),
sifra_artikla NUMBER (6),
kolicina NUMBER (6) NOT NULL,
CONSTRAINT stavke_zahteva_pk1 PRIMARY KEY (id_minimarketa, rbr_zahteva, sifra_artikla),
CONSTRAINT stav_zah_idrbr_fk FOREIGN KEY (id_minimarketa, rbr_zahteva) REFERENCES
zahtevi(id_minimarketa, rbr_zahteva),
CONSTRAINT stav_art_fk FOREIGN KEY (sifra_artikla) REFERENCES artikli(sifra));
```
- 4 | 

```
CREATE TABLE artikli
(sifra NUMBER (6) CONSTRAINT art_sifra_pk PRIMARY KEY,
naziv VARCHAR2(100) NOT NULL);
```

5 CREATE TABLE isporuke  
 (rbr\_isporuke NUMBER (6),  
 id\_minimarketa NUMBER (6),  
 rbr\_zahteva NUMBER(6),  
 datum\_izdavanja DATE,  
 CONSTRAINT isporuke\_pk PRIMARY KEY (rbr\_isporuke, id\_minimarketa, rbr\_zahteva),  
 CONSTRAINT ispo\_zahf\_fk FOREIGN KEY (id\_minimarketa, rbr\_zahteva) REFERENCES  
 zahtevi(id\_minimarketa, rbr\_zahteva));

6 CREATE TABLE stavke\_isporuke  
 (rbr\_isporuke NUMBER (6),  
 id\_minimarketa NUMBER (6),  
 rbr\_zahteva NUMBER (6),  
 sifra\_artikla NUMBER (6),  
 kolicina NUMBER (6) NOT NULL,  
 CONSTRAINT stavke\_isporuke\_pk PRIMARY KEY (rbr\_isporuke, id\_minimarketa, rbr\_zahteva,  
 sifra\_artikla),  
 CONSTRAINT stavke\_ispo\_zahf\_fk FOREIGN KEY (rbr\_isporuke, id\_minimarketa, rbr\_zahteva)  
 REFERENCES isporuke(rbr\_isporuke, id\_minimarketa, rbr\_zahteva),  
 CONSTRAINT stavke\_ispo\_art\_fk FOREIGN KEY (sifra\_artikla) REFERENCES artikli(sifra));



Slika 5.2. Grafički prikaz veza među tabelama



## 5.2. Punjenje tabela podacima

U drugom koraku implementacije, u tabele sam uneo demo podatke, i to preko SQL kôdova. Evo i par primera uz grafički prikaz podataka u tabelama.

```
INSERT INTO minimarketi (id, ime, adresa, grad)
VALUES (1, 'Minimarket 1', 'B.Vasovića 1', 'Čačak');
```

```
INSERT INTO minimarketi (id, ime, adresa, grad)
VALUES (2, 'Minimarket 2', 'K.Miloša 2', 'Beograd');
```

```
INSERT INTO minimarketi (id, ime, adresa, grad)
VALUES (3, 'Minimarket 3', 'S.Stepanovića 3', 'Niš');
```

EDIT	ID	IME	ADRESA	GRAD
	1	Minimarket 1	B.Vasovica 1	Cacak
	2	Minimarket 2	K.Miloša 2	Beograd
	3	Minimarket 3	S.Stepanovica 3	Niš
row(s) 1 - 3 of 3				

Slika 5.3. Tabela "MINIMARKETI"

```
INSERT INTO zahtevi (id_minimarketa, rbr_zahteva,
datum_izdavanja)
VALUES (1, 1, '18-DEC-10');
```

```
INSERT INTO zahtevi (id_minimarketa, rbr_zahteva,
datum_izdavanja)
VALUES (1, 2, '18-DEC-10');
```

```
INSERT INTO zahtevi (id_minimarketa, rbr_zahteva,
datum_izdavanja)
VALUES (2, 1, '18-DEC-10');
```

EDIT	ID_MINIMARKETA	RBR_ZAHTEVA	DATUM_IZDAVANJA
	1	1	18-DEC-10
	1	2	18-DEC-10
	2	1	18-DEC-10
	2	2	18-DEC-10
	2	3	18-DEC-10
	3	1	18-DEC-10
row(s) 1 - 6 of 6			

Slika 5.4. Tabela "ZAHTEVI"

```
INSERT INTO stavke_zahteva (id_minimarketa,
rbr_zahteva, sifra_artikla, kolicina)
VALUES (1, 1, 101, 4);
```

```
INSERT INTO stavke_zahteva (id_minimarketa,
rbr_zahteva, sifra_artikla, kolicina)
VALUES (2, 1, 104, 5);
```

```
INSERT INTO stavke_zahteva (id_minimarketa,
rbr_zahteva, sifra_artikla, kolicina)
VALUES (3, 1, 103, 16);
```

EDIT	ID_MINIMARKETA	RBR_ZAHTEVA	SIFRA_ARTIKLA	KOLICINA
	1	1	102	5
	1	2	103	6
	2	1	104	5
	2	2	101	7
	3	1	103	16
row(s) 1 - 5 of 5				

Slika 5.5. Tabela "STAVKE\_ZAHTEVA"

Istim principom punim podacima i preostale tri tabele. Strani ključevi koje sam kreirao obezbediće konzistentnost novounetih podataka. Dakle:

INSERT INTO artikli (sifra, naziv)  
VALUES (101, 'Brašno');

INSERT INTO artikli (sifra, naziv)  
VALUES (102, 'Šećer');

INSERT INTO artikli (sifra, naziv)  
VALUES (103, 'Ulje');

INSERT INTO artikli (sifra, naziv)  
VALUES (104, 'Kuhinjska so');

EDIT	SIFRA	NAZIV
	101	Brašno
	104	Kuhinjska so
	102	Šećer
	103	Ulje
row(s) 1 - 4 of 4		

Slika 5.6. Tabela "ARTIKLI"

INSERT INTO isporuke (rbr\_iskoruke, id\_minimarketa, rbr\_zahteva, kolicina)  
VALUES (1, 1, 1, '18-DEC-10');

INSERT INTO isporuke (rbr\_iskoruke, id\_minimarketa, rbr\_zahteva, kolicina)  
VALUES (1, 1, 2, '18-DEC-10');

INSERT INTO isporuke (rbr\_iskoruke, id\_minimarketa, rbr\_zahteva, kolicina)  
VALUES (3, 1, 1, '18-DEC-10');

EDIT	RBR_ISPORUKE	ID_MINIMARKETA	RBR_ZAHTEVA	DATUM_IZDAVANJA
	1	1	1	18-DEC-10
	1	1	2	18-DEC-10
	3	1	1	18-DEC-10
	2	1	1	18-DEC-10
	2	2	1	18-DEC-10
row(s) 1 - 5 of 5				

Slika 5.7. Tabela "ISPORUKE"

INSERT INTO stavke\_iskoruke (rbr\_iskoruke, id\_minimarketa, rbr\_zahteva, sifra\_artikla, kolicina)  
VALUES (1, 1, 1, 101, 4);

INSERT INTO stavke\_iskoruke (rbr\_iskoruke, id\_minimarketa, rbr\_zahteva, sifra\_artikla, kolicina)  
VALUES (2, 1, 1, 103, 6);

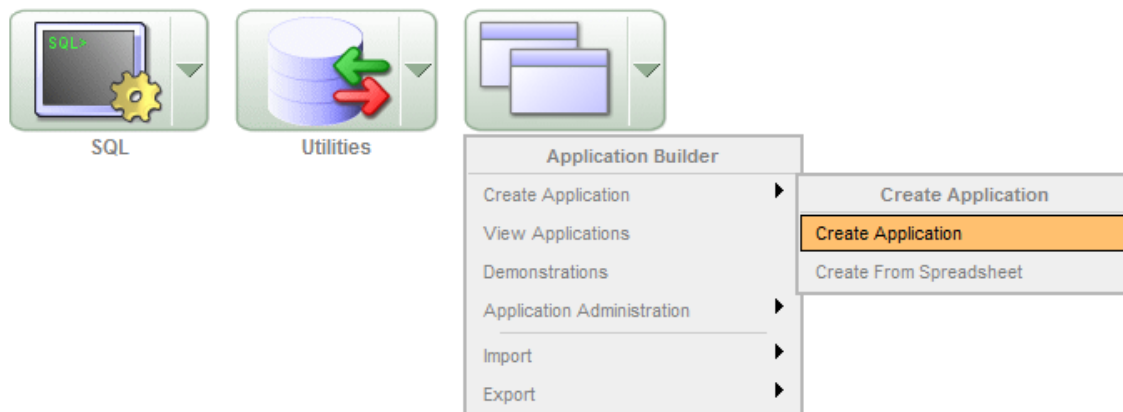
EDIT	RBR_ISPORUKE	ID_MINIMARKETA	RBR_ZAHTEVA	SIFRA_ARTIKLA	KOLICINA
	2	1	2	102	4
	4	1	1	103	6
	5	2	1	104	5
	1	1	1	101	4
	3	1	1	102	1
	6	2	2	101	7
row(s) 1 - 6 of 6					

Slika 5.8. Tabela "STAVKE\_ISPORUKE"

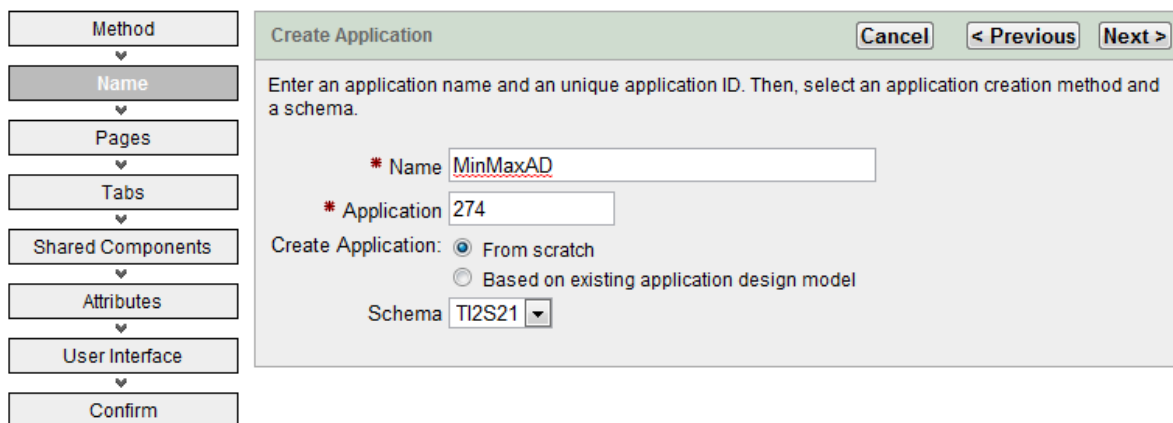
Kao što se vidi iz poslednje tabele, svaka isporuka će imati svoj **jedinstveni redni broj** a pri tome će preko stranih ključeva ostati povezana sa zahtevom na koji se odnosi. Na ovaj način se proces numerisanja isporuka ubrzava a ne narušava integritet veze sa zahtevom. Takođe, primećujemo i da se primarni ključ tabele ZAHTEVI, a koji se sastoji iz dve kolone: id\_minimarketa i rbr\_zahteva, takoreći "provlači" kroz gotovo sve tabele (izuzev tabele ARTIKLI). Razlog za to je neraskidiva veza između zahteva i isporuke, a koja je predviđena u samom poslovnom zahtevu projekta.

### 5.3. Izrada aplikacije

Iz Home prozora APEX-a birao sam ikonu Application Builder/Create. Otvorio se prozor za kreiranje aplikacije. Prvi korak u kreiranju aplikacije je dodeljivanje imena aplikaciji. U ovom slučaju to je MinMaxAD.

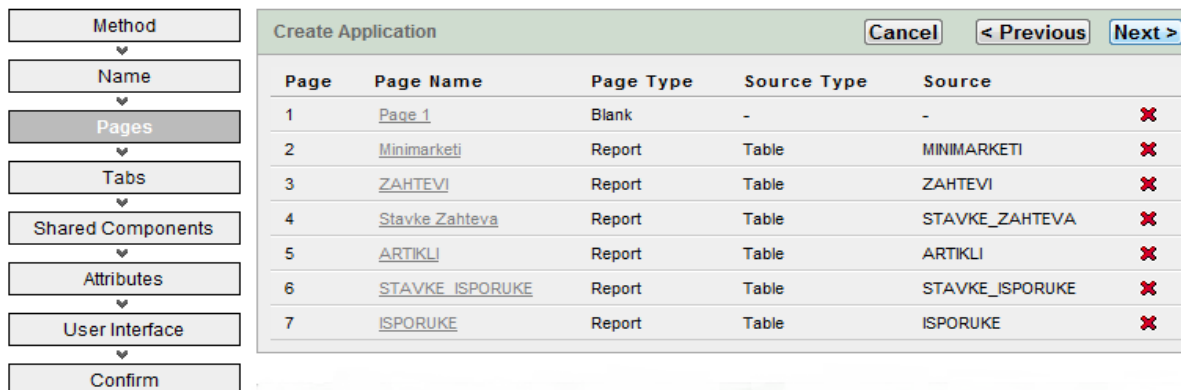


Slika 5.9. Početna stranica APEX-a i izbor stavke "Create Application"



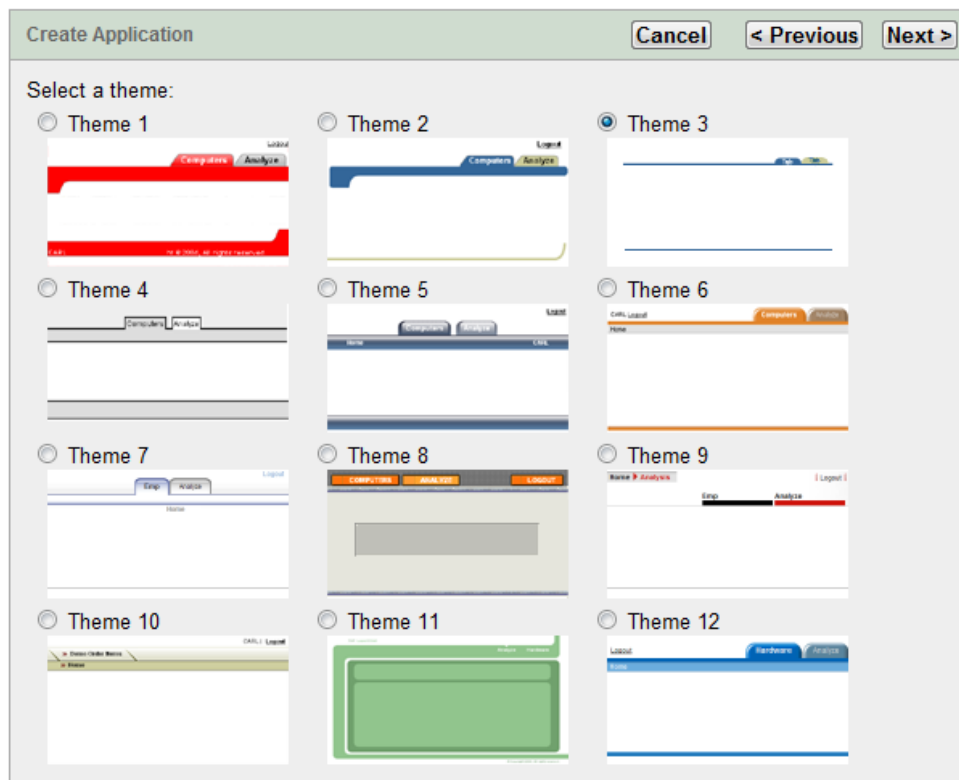
Slika 5.10. Dodeljivanje imena aplikaciji

Stranica "Pages" nudi forumular za kreiranje stranica. Kreirao sam sedam stranica, jednu praznu (Blank) i šest izveštaja (Report) koje će prikazivati sadržaj tabela na koje se odnose.



Slika 5.11. Stranice koje sam kreirao u formularu "Pages"

Sledeće što formular "Create Application" zahteva je izbor kartica (tabova) gde sam izabrao opciju "One Level of Tabs". Izborom ove opcije omogućio sam brz pristup kreiranih stranica preko kartica koje će se nalaziti na vrhu aplikacije. Ostale opcije u okviru formulara za kreiranje aplikacije sam ostavio na "default" vrednostima, sem što sam postavio temu korisničkog interfejsa na "Theme 3". Na slici 5.12. prikazan je izgled svih raspoloživih tema kojima mogu da zamenim već izabranu po želji naručioca projekta ili korisnika aplikacije.



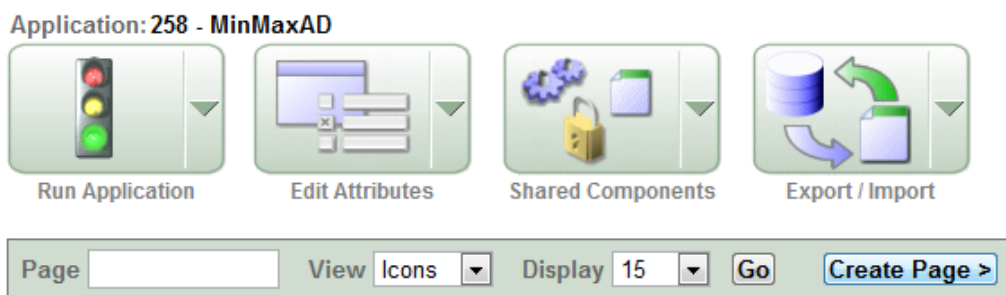
Slika 5.12. Izbor tema korisničkog interfejsa

Poslednja stavka ovog procesa je potvrđivanje kreiranja aplikacije gde se klikom na dugme "Create" momentalno kreira nova aplikacija pod nazivom "MinMaxAD".

Međutim, aplikacija još uvek nije spremna da prikaže sve što se od nje zahteva. Ona trenutno može da zadovolji jedino potrebu naručioca da vrši pregled tabela i podataka u njima. Da bi aplikacija bila potpuno funkcionalna, moramo naručiocu odnosno korisnicima aplikacije obezbediti i manipulaciju podacima u tabelama, konkretno, dodavanje novih podataka, brisanje postojećih, ažuriranje podataka itd. Za tu svrhu, kreiraćemo po jednu dodatnu stranu za svaku stranu koju smo već kreirali, a koja će sadržati formular (Forms) za manipulaciju te strane. Toj novoj strani (formularu) će se pristupiti preko postojeće i to preko dugmeta kojeg ću naknadno kreirati.

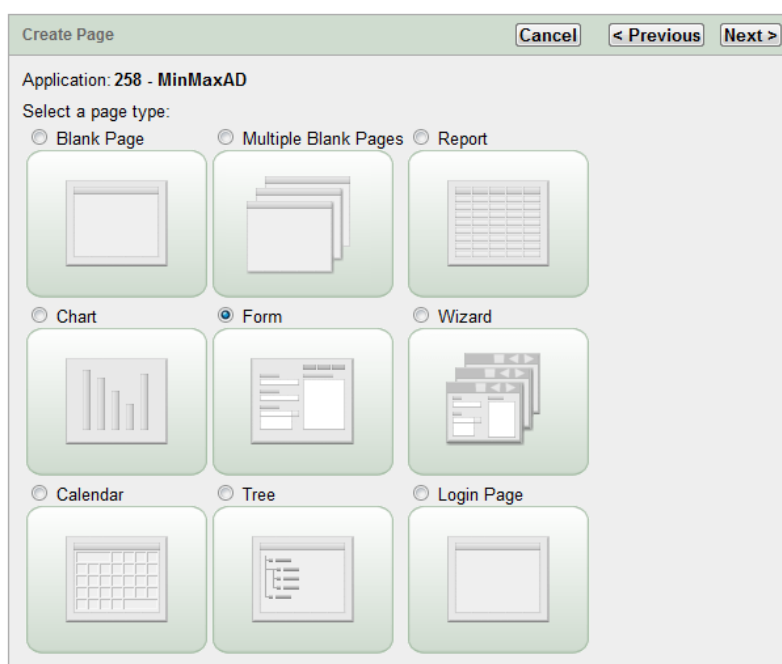
U nastavku dokumentacije o implementaciji, pokazaću primer kreiranja formulara i povezivanje sa matičnom stranom preko dugmeta (Button).

Iz početne stranice "Application Builder-a" bira se opcija "Create Page" (Slika 5.12.)



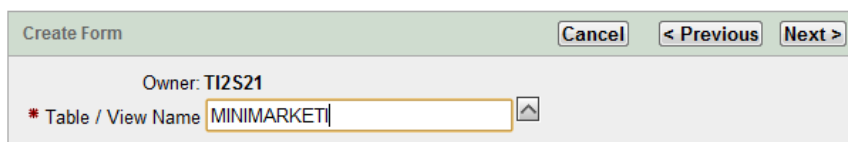
Slika 5.12. Izbor opcije "Create Page"

Nakon izbora ove opcije, otvoriće se formular za kreiranje nove stranice. Na prvoj strani formulara biram "Form" kao tip stranice koji nameravam da kreiram (Slika 5.13.)



Slika 5.13. Izbor tipa stranice

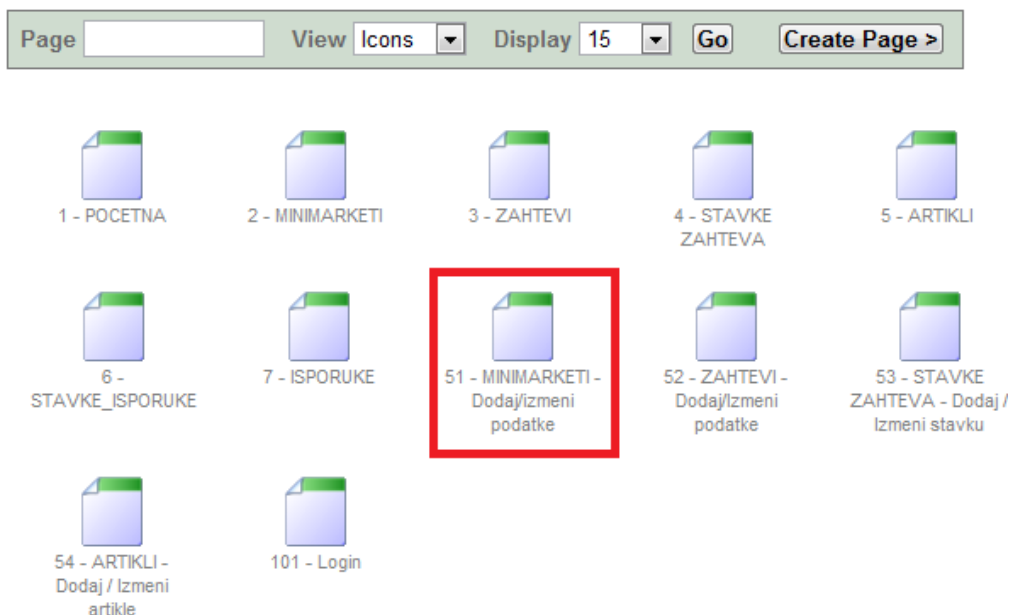
U sledećim zahtevima "Create Page" procesa biram "Form on a Table or View" i kucam ime tabele na koju će se nova stranica (formular). U ovom primeru to je tabela MINIMARKETI. (Slika 5.14.)



Slika 5.14. Izbor tabele na koju se odnosi formular

Ono što sledi je unos rednog broja stranice, dodeljivanje imena, izbor kartica, podešavanje primarnog ključa tabele (u ovom slučaju to je "Id"), izbor kolona nad kojima će se vršiti manipulacija (u ovom slučaju to su sve kolone) podešavanje redirekcije, itd. Stranici sam dodelio redni broj 51, a nazvao je "MINIMARKETI – Dodaj / Izmeni podatke". Nakon svih unosa, potvrđujem kreiranje stranice klikom na dugme "Create".

Stranica se sada pojavila u spisku kreiranih stranica, a klikom na nju mogu izvršiti dodatna podešavanja.



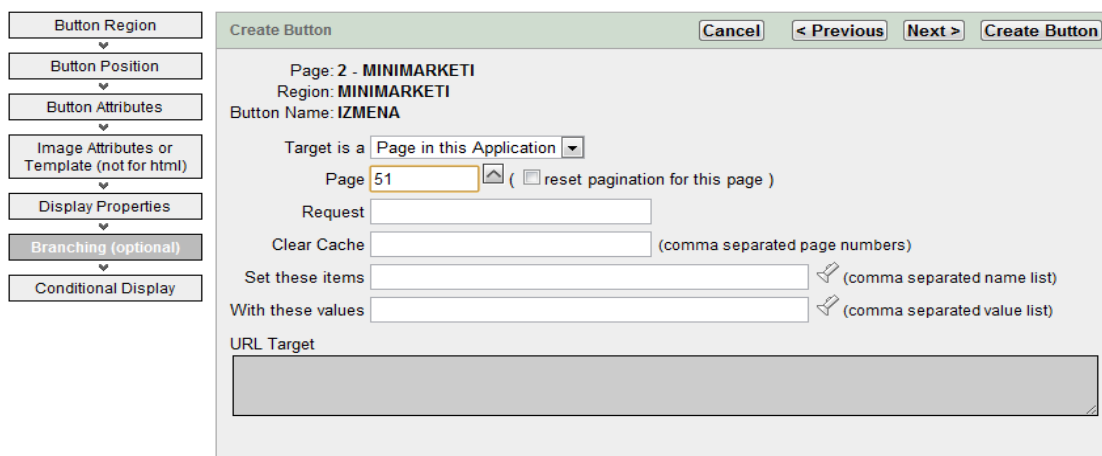
Slika 5.15. Novokreirana stranica u spisku stranica

Sada ću na stranici "2 – MINIMARKETI" da kreiram dugme preko koga će korisnik aplikacije pristupati kreiranom formularu. Najpre ću u spisku stranica kliknuti na stranicu 2, a zatim u podešavanjima stranice u okviru sekcije "Buttons" dodaću novo dugme (Slika 5.16.)



Slika 5.16. Dodavanje dugmeta na stranicu

Otvoriće se čarobnjak za kreiranje dugmeta čijim praćenjem kreiramo dugme. Neki od važnih stavki koje će čarobnjak tražiti su ime dugmeta, pozicija na stranici, a najbitnija stavka je stranica na koju dugme "vodi" (redirektuje) . U polje "Page" upisujem broj stranice koja sadrži formular za izmenu stranice, dakle 51. (Slika 5.17.)



Slika 5.17. Izbor stranice na koju dugme "vodi"

Evo i izgleda stranice sa izveštajem za tabelu MINIMARKETI i upravo kreiranog dugmeta.

The screenshot shows a web interface for 'MINIMARKETI'. At the top, there are tabs for 'POCETNA' and 'MINIMARKETI'. Below the tabs, there is a search bar with a 'Search' label, a 'Display' dropdown set to '15', and a 'Go' button. A table with columns 'Id', 'Ime', 'Adresa', and 'Grad' contains five rows of data. Below the table, there is a 'Spread Sheet' indicator showing '1 - 5'. A button labeled 'Dodaj / Izmeni podatke' is highlighted with a red rectangular box.

Id	Ime	Adresa	Grad
1	Minimarket 1	B.Vasovica 1	Cacak
2	Minimarket 2	K.Miloša 2	Beograd
3	Minimarket 3	S.Stepanovica 3	Niš
4	Minimarket 4	L. Teleckog 4	Novi Sad
5	Minimarket 5	K. Mihaila 5	Kragujevac

Slika 5.18. Stranica sa izveštajem i dugme koje vodi na formular za izmenu podataka

Ovim smo povezali stranicu koja sadrži izveštaj o tabeli MINIMARKETI povezali preko dugmeta sa stranicom koja sadrži formular za manipulaciju podataka iz te tabele. Na isti način ću kreirati formulare koji se odnose na ostale tabele iz baze podataka a zatim ih pomoću "dugmića" povezati sa odgovarajućim izveštajima.

Nakon kreiranja ostalih formulara i povezivanja sa stranim sa izveštajima, odlučio sam se da svim formularima dodam i pregled izveštaja na koji se odnose. Na taj način će korisnik, kada iz stranice sa izveštaje klikne na dugme "Dodaj / Izmeni podatke", dobiti formular za manipulaciju podataka a pri tom će mu ostati i izveštaj koji prikazuje sadržaj tabele.

Pored toga, svakoj stranici sa formularom sam dodao i uprošćeni sistem pomoći koji će ukazivati na obavezna polja i objasniti akcije svakog dugmeta formulara (Slika 5.19.).

The screenshot shows a web interface for 'MINIMARKETI - Dodaj/izmeni podatke'. It features a form with fields for 'Id', 'Ime', 'Adresa', and 'Grad'. Below the form are buttons for 'Nazad', 'Obriši unos', 'Potvrdi promene', and 'Dodaj unos'. To the right of the form is a 'Pomoc:' section with a 'SISTEM POMOĆI' heading, containing instructions for the form fields and buttons. Below the form and help section is a table labeled 'Tabela MINIMARKETI' with columns 'Id', 'Ime', 'Adresa', and 'Grad', containing the same five rows of data as in Slika 5.18. The table is labeled 'PREGLED TABELE'.

**FORMULAR**

**SISTEM POMOĆI**

**DUGMAD ZA MANIPULACIJU PODACIMA**

**PREGLED TABELE**

Id	Ime	Adresa	Grad
1	Minimarket 1	B.Vasovica 1	Cacak
2	Minimarket 2	K.Miloša 2	Beograd
3	Minimarket 3	S.Stepanovica 3	Niš
4	Minimarket 4	L. Teleckog 4	Novi Sad
5	Minimarket 5	K. Mihaila 5	Kragujevac

Slika 5.19. Unapređena stranica sa formularom (uz prikaz sekcija)

Sada sam dodao i novu liniju sa karticama za brži pristup formularima koji se odnose na ostale stranice. Kartice koje isključivo prikazuju izveštaje su obeležene velikim slovima, a kartice koje prikazuju izveštaje sa formularima za manipulaciju podacima su obeležene malim slovima.

Ovim sam završio osnovne elemente aplikacije, a to su pregled i manipulacija podacima u tabelama. Na neki način, i ovim urađenim je ispoštovan poslovni zahtev, a to je jednostavan pregled zahteva koje šalju minimarketi, svih stavki tih zahteva kao i pregled isporuka koji se odnose na te zahteve.

Međutim, pokušao sam da budem preduseljiv i kreiram dodatne izveštaje koji bi mogli da budu od koristi naručiocu ili korisnicima aplikacije. Šta ako korisnik aplikacije bude želeo da prati koji minimarket šalje najviše zahteva? A koji najmanje? Ili koji minimarket traži najveće količine artikala? Logično bi bilo da minimarketi u većim gradovima bolje posluju ali ipak bi precizni izveštaji bili više nego dobrodošli.

Zašto kreirati dodatne izveštaje kada to od nas nije traženo? Odgovor je jednostavan. Velika je verovatnoća da će ti dodatni izveštaji pomoći poslovanju preduzeća. Naručilac projekta bi mogao shvatiti da je, ukoliko se dvoumio, ipak izabrao dobrog projektanta (programera) za bazu podataka, koji će svojim samoinicijativnim delovanjem unaprediti poslovanje. Kasnije bi to mogla postati i svojevrsna reklama i sl. Svi ti dodatni izveštaji ne iziskuju preterano mnogo truda ali odaju sliku profesionalnog i kreativnog programera, koji su uvek na ceni. Iz tih razloga je kreiran i sistem pomoći koji može da bude od velike pomoći korisniku koji nema puno iskustva u radu sa aplikacijama. Pokušao sam da se postavim u "kožu" korisnika čija su iskustva u radu sa bazama podataka mala ili pak nikakva. Takvom korisniku i najmanja smernica može biti dragocena.

Elem, da se vratim u implementaciju. Te "samoinicijativne" izveštaje moraću da kreiram spajanjem tabela jer sadrže kolone koje se nalaze u različitim tabelama.

Npr. ukoliko želim da kreiram izveštaj koji prikazuje koliko koji minimarket izdaje zahteva, moraću da spojim podatke iz dve tabele: MINIMARKETI i ZAHTEVI. Kao dodatak ovom izveštaju postavljam i ukupan broj svih poslanih zahteva. Evo SQL kôdova:

```
SELECT id AS "Id minimarketa",
COUNT(rbr_zahteva) AS "Poslatih zahteva"
FROM minimarketi JOIN zahtevi
ON minimarketi.id = zahtevi.id_minimarketa
GROUP BY id;
```

```
SELECT COUNT(rbr_zahteva)
AS "Ukupno poslanih zahteva"
FROM zahtevi;
```

Poslati zahtevi

Id minimarketa	Poslatih zahteva
1	2
2	3
4	5
3	1
1 - 4	

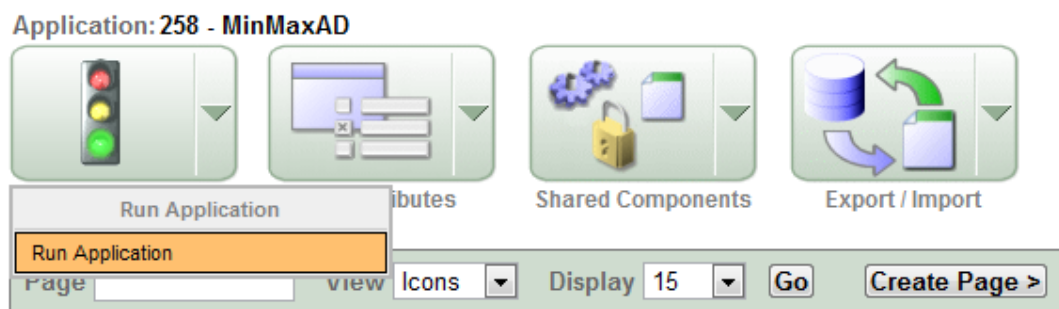
Slika 5.20. "Spojena" tabela

Kreirao sam i izveštaj koji prikazuje sve otpremnice koji pokazuje gotovo ceo ciklus od zahteva do isporuke. Za ovaj izveštaj sam koristio APEX-ov čarobnjak za spajanje tabela, jer sam tako smanjio mogućnost greške koja bi se mogla javiti kao posledica spajanja više tabela.



## 5.4. Pokretanje aplikacije

Nakon što je proces implementacije uspešno završen i nakon što su izveštaji i formulari testirani, aplikacija je spremna za rad. Aplikacija se pokreće izborom aplikacije u Application Builder-u (u ovom slučaju to je aplikacija MinMaxAD), klikom na meni Run Application, pa na opciju Run Application (Slika 5.21.).



Slika 5.21. Pokretanje aplikacije

Nakon pokretanja i uspešno obavljene autentifikacije, aplikacija nam prikazuje svoju početnu stranu (Slika 5.22.). Početnu stranu sam uredio tako da sa leve strane sadrži informacije o autoru, kratka uputstva i kratak informacioni blok, dok je desna strana rezervisana za dodatne izveštaje. Na vrhu se nalazi glavni meni preko kojeg se brzo prustupa stranicama sa izveštajima.



Slika 5.22. Izgled početne stranice aplikacije MinMaxAD

## 6. Zaključak

Polazeći od misije, ciljeva i informacionih potreba korisnika, kreirana je baza podataka i aplikacija "MinMaxAD" da bi bila omogućena kvalitetnija evidencija zahteva koje dostavljaju minimarket i isporuke koje se na njih odnose. Pored osnovnih informacionih potreba, kreirane su i neke dodatne koje bi mogle pomoći boljem poslovanju preduzeća.

Baza podataka je zasnovana na stabilnom i precizno definisanom modelu, pa se novi poslovni zahtevi mogu sa lakoćom implementirati u kreirani sistem. Po želji naručioca, aplikacija za pregled i manipulaciju podacima iz baze podataka se može postaviti i na Internet, te se može omogućiti udaljeni pristup aplikaciji.