**SVEUČILIŠTE U SPLITU**

**PRIRODOSLOVNO-MATEMATIČKI FAKULTET**

**SEMINARSKI RAD**

INFORMATIČKI PROJEKT IZ BAZA PODATAKA

WEB TRGOVINA

**NAZIV KOLEGIJA:** Informatički projekt iz baza podataka **STUDENT:** Rei Krstić

**PROFESORICE:** Antonela Prnjak i Tina Vicković **NAZIV STUDIJA:** Informatika 2

Sadržaj:

[1 UVOD 1](#_Toc163035612)

[1.1 Općenito 1](#_Toc163035613)

[1.2 Opis projekta 1](#_Toc163035614)

[2 ENTITETI 2](#_Toc163035615)

[2.1 Korisnik 2](#_Toc163035616)

[2.2 Kategorija 2](#_Toc163035617)

[2.3 Korisnik\_kategorija 3](#_Toc163035618)

[2.4 Spol 3](#_Toc163035619)

[2.5 Grad 3](#_Toc163035620)

[2.6 Država 3](#_Toc163035621)

[2.7 Kontinent 3](#_Toc163035622)

[2.8 Država\_Kontinent 3](#_Toc163035623)

[2.9 Proizvod 3](#_Toc163035624)

[2.10 Skladište 4](#_Toc163035625)

[2.11 Proizvod\_skladište 4](#_Toc163035626)

[2.12 Račun 4](#_Toc163035627)

[2.13 Račun\_proizvod 4](#_Toc163035628)

[2.14 Valuta 4](#_Toc163035629)

[2.15 Zaposlenik 4](#_Toc163035630)

[2.16 Funkcija\_zaposlenika 4](#_Toc163035631)

[3 RELACIJE 5](#_Toc163035632)

[3.1 Relacije 1:1 5](#_Toc163035633)

[3.2 Relacije 1:n 5](#_Toc163035634)

[3.3 Relacije m:n 6](#_Toc163035635)

[4 MODELI PODATAKA 7](#_Toc163035636)

[4.1 Konceptualni model 7](#_Toc163035637)

[4.2 Logički model 8](#_Toc163035638)

[4.3 Relacijski model 9](#_Toc163035639)

[5 KREIRANJE BAZE PODATAKA 12](#_Toc163035640)

[6 SQL UPITI 14](#_Toc163035641)

[6.1 „INSERT“ upit 14](#_Toc163035642)

[6.2 „UPDATE“ upit 14](#_Toc163035643)

[6.3 „DELETE“ upit 15](#_Toc163035644)

[6.4 „SELECT“ upit 15](#_Toc163035645)

[6.4.1 Prvi primjer „SELECT“ upita 15](#_Toc163035646)

[6.4.2 Drugi primjer „SELECT“ upita 16](#_Toc163035647)

[6.4.3 Treći primjer „SELECT“ upita 17](#_Toc163035648)

[6.4.4 Četvrti primjer „SELECT“ upita 17](#_Toc163035649)

[7 APEX 19](#_Toc163035650)

# UVOD

## Općenito

Web trgovine su nešto što dosta nas često koristi (Amazon, eBay, Njuškalo, Links itd.), te ne bismo mogli više uopće zamisliti život bez njih. I za to je upravo zaslužna njihova korisnost i jednostavnosti korištenja (u najviše slučajeva), ali važno je spomenuti kako ništa od tog ne bi bilo moguće bez njihove kompleksne, temeljite, detaljne, te, možda ono najvažnije, efikasne ***baze podataka***. S, često, milijunima razno raznih predmeta za prodaju u skladištima na različitim lokacijama, s mnoštvom korisnika i zaposlenika pa možda čak diljem svijeta, važno je voditi, skoro pa savršenu evidenciju svih „aspekata“ koji su važne za pravilno i profesionalno vođenje same te web trgovine.

## Opis projekta

Za ovaj konkretan projekt je potrebno osmisliti i realizirati što normaliziraniju bazu podataka za internacionalnu (i također interkontinentalnu) web trgovinu koje se bavi prodajom informatičke opreme (hardwarea), video igara i programske podrške (softwarea). Osim što se u bazi treba voditi evidencija o samoj robi, bitno je također da znamo sve što se treba znati o skladištima u kojima se ta roba nalazi, to jest primjerice, zaposlenici koji rade u pojedinom skladištu, na kojoj se lokaciji nalazi… Kad već pričamo o zaposlenicima, trebamo voditi detaljnu evidenciju o njima, njihovim osobnim podacima i njihovoj trenutnoj funkciji u poduzeću, od kad, pa do kad im je efektivan ugovor koji su potpisali itd. Možda i najvažniji dio vođenja ovakvog tipa baze podataka je detaljno i sigurno praćenje korisnika, jer bez korisnika ne bi ništa bilo moguće. Trebamo voditi detaljnu evidenciju o njihovim osobnim podacima (ime, prezime, OIB…), uključujući njihove podatke kreditne kartice, ako to oni požele, te o njihovim afinitetima za različite kategorije proizvoda nakon što svoje željene kategorije „poklikaju“ pri ulasku u web shop, i samim tim su im dane reklame na temelju tih afiniteta. Također im se bilježi i ukupan potrošen novac nakon čega bi, s dovoljno velikim brojem novca potrošenim, mogli dobiti određenu nagradu. Na kraju se naravno bilježi i (digitalni) račun koji se dobije nakon neke kupnje.

# ENTITETI

Entiteti su glavni aspekt baze podataka. Prikazani su u obliku tablica sastavljenih od više atributa koji sadrže sve podatke koji su nam potrebni u samoj bazi, te sve informacije koje naknadno sakupimo tu i spremamo u njihove jedinstvene retke, odnosno, n-torke.

Za ovaj model baze podataka bili su nam potrebni entiteti:

1. Korisnik
2. Kategorija
3. Korisnik\_kategorija
4. Spol
5. Grad
6. Država
7. Kontinent
8. Država\_kontinent
9. Proizvod
10. Skladište
11. Proizvod\_skladište
12. Račun
13. Račun\_proizvod
14. Valuta
15. Zaposlenik
16. Funkcija\_zaposlenika

## Korisnik

Jedan od važnijih „centralnih“ entiteta. Sadrži atribute koji bilježe bitne podatke kao što su ime, prezime, jedinstveni OIB, e-mail adresu, datum rođenja, te, eventualno, karticu registriranog korisnika.

## Kategorija

Ovaj se entitet odnosi na kategoriju proizvoda u skladištima, te (osim svog jedinstvenog primarnog ključa) samo sadrži svoj naziv.

## Korisnik\_kategorija

Ovaj entitet ćemo koristiti samo kao „most“ između entiteta Korisnik i Kategorija kako bismo pravilno realizirali many to many relaciju između tih entiteta, no više o relacijama poslije. Osim toga smo zapravo i pravili ovaj entitet kako bi u njemu zabilježili afinitete koje korisnik, kao što je prije rečeno, „poklika“ pri ulasku u stranicu web shopa i samim tim, posebne reklame koje će se mjesečno prikazivati korisniku. Također definiramo kompozitne primarne ključeve.

## Spol

U ovaj entitet samo navodimo podatke u atribute: Naziv spola i (opcionalno) oznaka spola.

## Grad

Entitet „Grad“ nam je bitan kako bismo evidentirali lokacije skladišta, korisnika i zaposlenika.

## Država

Entitet „Država“ također uključujemo u model jer uvijek ciljamo da nam baza podataka ima što manje redundancije, a da ima što višu normalizaciju.

## Kontinent

S obzirom na to da naša zadana web trgovina, za koju radimo bazu podataka, funkcionira na svjetskoj razini, također navodimo i entitet „Kontinent“.

## Država\_Kontinent

Kao i sa slučajem kod entiteta Korisnik\_kategorija, ovaj entitet koristimo kao način da ostvarimo vezu više na više. Stvorili smo ovu „tablicu“ jer, iako su rijetke, postoje države koje pripadaju više kontinenata.

## Proizvod

Jedan od važnijih entiteta. U njemu bilježimo podatke o proizvodima kao što su njihov naziv, cijena i opis.

## Skladište

U ovom entitetu evidentiramo podatke o lokaciji skladišta, te tko ondje uopće radi (u skladištima se rade i funkcije drugih poslova kao npr. tehnička podrška).

## Proizvod\_skladište

Pravljenjem ovog entiteta stvaramo još jednu m:n (više na više) vezu. U Proizvod\_skladište entitetu bilježimo količinu robe koje imamo u raznim skladištima.

## Račun

Još jedan važniji „centralni“ entitet. „Račun“ je digitalan, te se dobiva poslije kupnje. Račun se odnosi i na kupnju koja se upravo izvršila. Ovaj račun ima značajke „tipičnog računa“ kao što su primjerice datum kupnje i način plaćanja (tip CHAR duljine1 i označava se kao „P“ za pouzeće ili „K“ za karticu).

## Račun\_proizvod

U ovom entitetu, za razliku od drugih m:n entiteta, nemamo kompozitne primarne ključeve, već zaseban primarni ključ. Koristimo ovaj entitet da zabilježimo kupljene stavke na računu.

## Valuta

Ovim entitetom navodimo valute u atributu „Naziv\_valute“ (koje inače koristimo pri plaćanju računa).

## Zaposlenik

Isto među važnijim entitetima. Slično kao i za entitet „Korisnik“, navodimo temeljne podatke o zaposlenicima kao što su ime, prezime, OIB, e-mail, no također uz ovaj entitet vežemo neke druge entitete i atribute specifične zaposleniku.

## Funkcija\_zaposlenika

Pomoću ovog entiteta navodimo funkciju (posao) koju neki zaposlenik može imati, te ju poslije zapravo i vežemo sa entitetom „Zaposlenik“.

# RELACIJE

Relacije, odnosno veze, možemo zamisliti kao „linije“ koje povezuju neka dva entiteta.

Razlikujemo tri vrste relacija:

1. 1:1, odnosno jedan naprema jedan (eng. one to one)
2. 1:n, odnosno jedan naprema više (eng. one to many)
3. m:n, odnosno više naprema više (eng. many to many)

Također ih možemo „podesiti“ na način da izvor ili odredište budu „opcionalni“, međutim za potrebe ovog modela svaka je relacija podešena na način da je obavezna.

## Relacije 1:1

Zamislimo da imamo entitete e1 i e2. Kažemo da su ti entiteti u relaciji 1:1 ako jedan redak iz e1 pripada samo jednom retku iz e2 i obratno.

Primjer ovakve specifične relacije nije korišten u ovoj bazi podataka.

## Relacije 1:n

Zamislimo da imamo entitete e1 i e2. Kažemo da su ti entiteti u relaciji 1:n ako jedan redak iz e1 može pripasti više redaka iz e2, no obrat ne vrijedi.

Primjeri iz baze podataka:

* Jedne kategorije može biti više proizvoda, jedan proizvod može biti jedne kategorije.
* Jednog spola mogu biti više korisnika/zaposlenika, jedan korisnik/zaposlenik može imati jedan spol.
* Jedan korisnik/zaposlenik može živjeti u jednom gradu, u jednom gradu može živjeti više korisnika/zaposlenika.
* Jedan grad može biti u jednoj državi, u jednoj državi može biti više gradova.
* Jedan se račun može platiti jednom valutom, jednom valutom se može platiti više računa.
* Jedan zaposlenik radi jednu funkciju, jednu funkciju radi više zaposlenika.
* Jedan račun može platiti jedan korisnik, jedan korisnik može platiti više računa.
* Jedan zaposlenik može poslati više računa, jedan račun može poslati jedan zaposlenik.
* Jedan zaposlenik radi u jednom skladištu, u jednom skladištu radi više zaposlenika.
* Jedno skladište je locirano u jednom gradu, u jednom gradu može biti više skladišta.

## Relacije m:n

Zamislimo da imamo entitete e1 i e2. Kažemo da su ti entiteti u relaciji m:n ako jedan redak iz e1 može pripasti više redaka iz e2 i obratno.

Relacije m:n ne možemo koristiti kako bi direktno stavili u relaciju dva entiteta, već trebamo stvoriti novi entitet, te zatim povezati primarne ključeve s tim novim entitetom.

Primjeri iz baze podataka:

* Jednu kategoriju može odabrati više korisnika, jedan korisnik može odabrati više kategorija. Za ovo smo kreirali entitet „Korisnik\_kategorija“.
* Jedna država može biti u više kontinenata, jedan kontinent može imati više država. Za ovo smo kreirali entitet „Kontinent\_država“.
* Jedan račun može imati na sebi više proizvoda, jedan proizvod može biti na više računa. Za ovo smo kreirali entitet „Račun\_proizvod“.
* Jedan proizvod može biti u više skladišta, u jednom skladištu može biti više proizvoda. Za ovo smo kreirali entitet „Proizvod\_skladište“

# MODELI PODATAKA

## Konceptualni model

Konceptualni model baze podataka nam služi za analizu prvotnog zahtjeva i da skiciramo i razmišljamo o tome kako najbolje normalizirati model, a da sačuvamo sve podatke.

Primjer iz baze podataka:

Slika na kojoj se prikazuje tekst, rukopis, bilježnica, knjiga

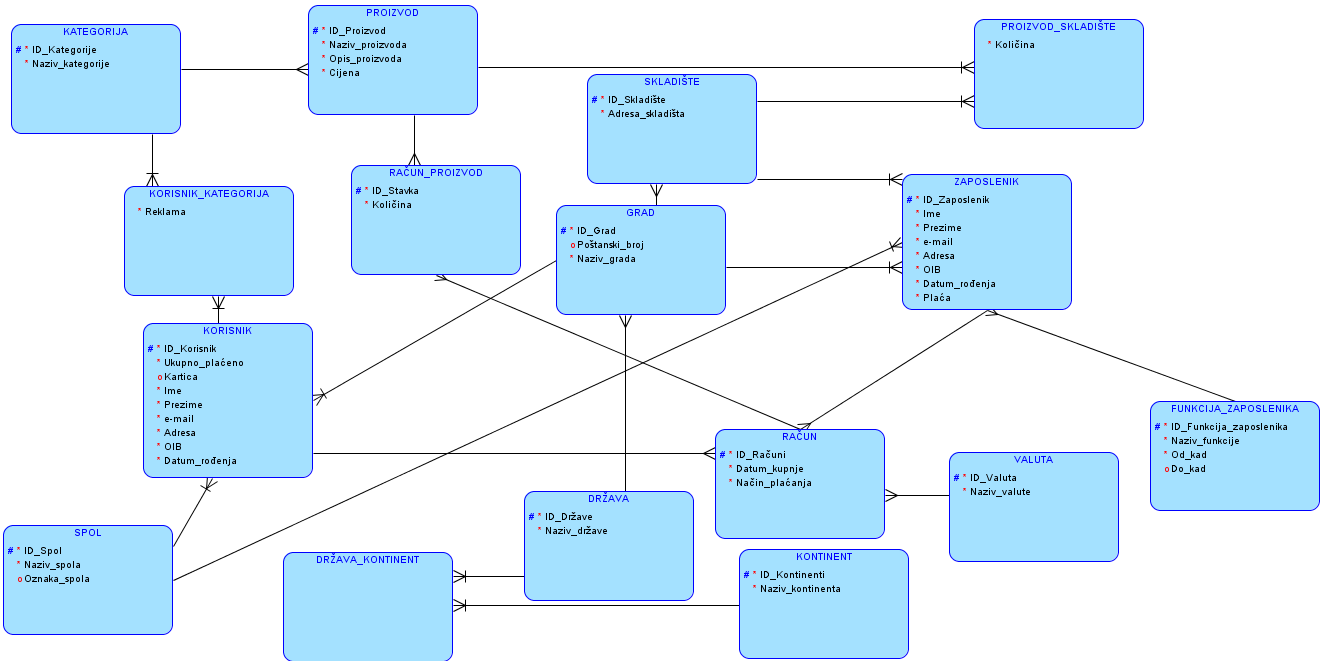
Opis je automatski generiran

Slika 1 - Konceputalni model baze podataka

## Logički model

U Oracle SQL Developer Data Modeler logičkom modelu baze podataka možemo vidjeti entitete, relacije, atribute, primarne, no ne i strane ključeve.

Primjer iz baze podataka:

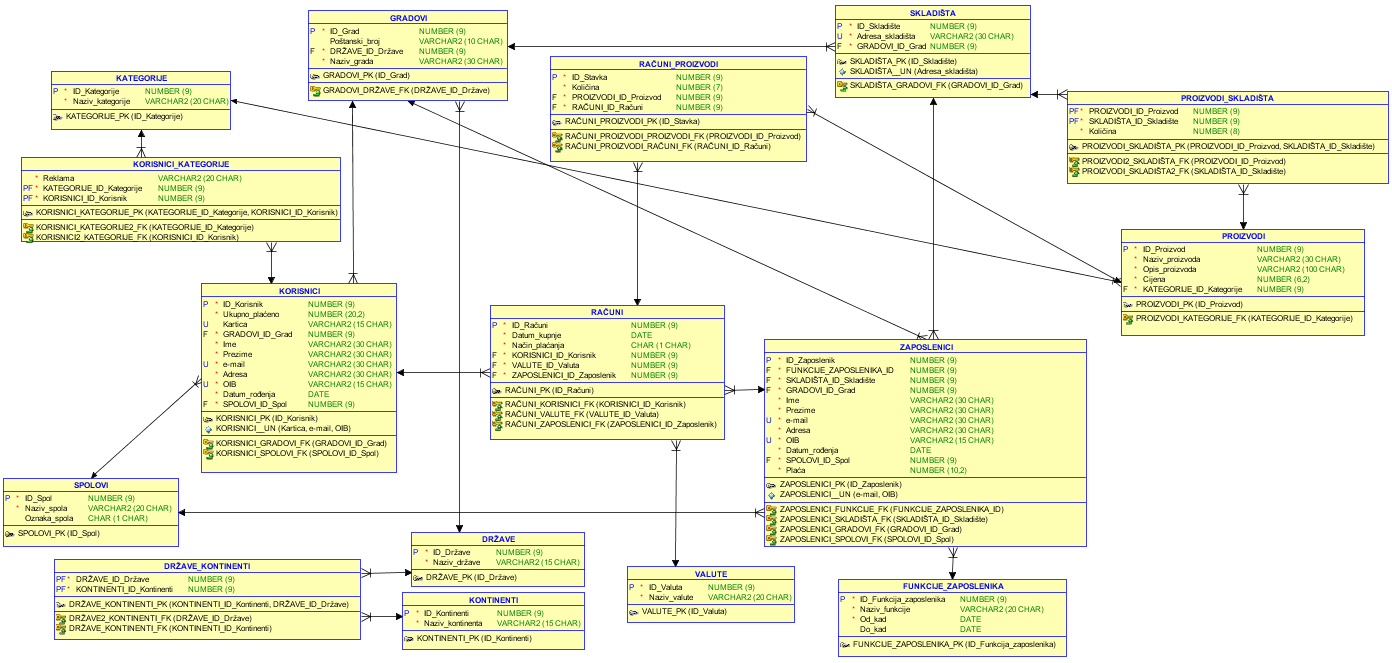


Slika 2 - Logički model baze podataka

## Relacijski model

Nakon što „engineeramo“ logički model, tada dobijemo relacijski model. U njemu možemo vidjet sve što smo vidjeli u logičkom modelu, pa i još više kao što su strani ključevi, tipovi podataka, te duljine atributa i ograničenja.

Primjer iz baze podataka:



Slika 3 - Relacijski model baze podataka

Poslije ovog koraka (kreiranja relacijskog modela) trebamo još u Data Modeleru kreirati „DDL“ datoteku koju možemo napokon i iskoristiti u samom Oracle SQL Developeru. No prije toga nam ostaje još jedan korak u relacijskom modelu: stvaranje sekvenci (eng. sequences) i okidača (eng. triggers).

To radimo na ovaj način:

1. Prvo uđemo u neki entitet dvoklikom i zatim uđemo u njegov primarni ključ, također dvoklikom:

Slika na kojoj se prikazuje tekst, snimka zaslona, softver, zaslon

Opis je automatski generiran

Slika 4 - Prvi korak generiranja sekvenci i okidača

1. Nakon što nam se otvori novi prozor, otiđemo na „tab“ General i označimo kvačicom opciju Auto Increment, kako bi se automatski povećavala vrijednost primarnog ključa:

Slika na kojoj se prikazuje tekst, snimka zaslona, softver, zaslon

Opis je automatski generiran

Slika 5 - Drugi korak generiranja sekvenci i okidača

1. Napokon, otiđemo na „tab“ Auto Increment, postavimo minimalnu vrijednost primarnog ključa na jedan, te damo ime sekvenci i okidaču:

Slika na kojoj se prikazuje tekst, snimka zaslona, softver, zaslon

Opis je automatski generiran

Slika 6 - Treći korak generiranja sekvenci i okidača

Kad smo ovo napravili za sve primarne ključeve, možemo na izbornoj traci kliknuti opciju „Generate DDL.“

Primjer iz DDL-a generiranog iz baze podataka – Web trgovina:

Slika na kojoj se prikazuje tekst, snimka zaslona, zaslon, softver

Opis je automatski generiran

Slika 7 - Dio DDL-a

# KREIRANJE BAZE PODATAKA

Sada dolazi vrijeme kad smo gotovi sa Data Modelerom, te napokon otvaramo sami Oracle SQL Developer i pravimo bazu podataka po odabranom modelu web trgovine.

Pri prvom pokretanju SQL Developera potrebno je napraviti novi „Connection“ kojeg zovemo SYSTEM. Ova veza ima najviše moguće ovlasti, te nam je samim time i nužna.

Prvo što poslije ovog trebamo napraviti jest napraviti novog korisnika u SYSTEM folderu „Other users.“ Kako bismo to napravili trebamo prvo napraviti novog korisnika, te važeću lozinku i zatim dati tom korisniku potrebne ovlasti. Ovo smo sve napravili u „SQL worksheetu“ od SYSTEM konekcije.

Slika na kojoj se prikazuje tekst, elektronika, snimka zaslona, softver

Opis je automatski generiran

Slika 8 - Kreiranje korisnika, njihove lozinke, te davanje ovlasti

Slika na kojoj se prikazuje tekst, snimka zaslona, Font

Opis je automatski generiran

Slika 9 - WEB\_TRGOVINA kao korisnik u SYSTEM konekciji

Nakon ovog koraka trebamo zapravo napraviti naš „connection“ WEB\_TRGOVINA, no ovaj put kroz prozor koji otvorimo klikom na zeleni plus u gornjem desnom kutu, te upisivanjem naziva „veze,“ korisničkog imena i lozinke.

Slika na kojoj se prikazuje tekst, snimka zaslona, zaslon, softver

Opis je automatski generiran

Slika 10 - prozor za kreiranje konekcije

Slika na kojoj se prikazuje tekst, snimka zaslona, Font, operacijski sustav

Opis je automatski generiran

Slika 11 - novokreirane konekcije

Lijepljenjem DDL datoteke (koju smo kreirali još u Data Modeleru) zapravo i kreiramo bazu podataka koja nam je potrebna.

Slika na kojoj se prikazuje tekst, snimka zaslona, softver, broj

Opis je automatski generiran

Slika 12 - kreirana baza podataka

# SQL UPITI

SQL upite koristimo kako bismo „manipulirali“ podacima u bazi podataka ili čak samom bazom podataka. Ovo možemo postići na različite načine, primjerice brisanjem ili ažuriranjem te baze ili podataka u njoj.

## „INSERT“ upit

„INSERT“ upit koristimo, logično, za „insertanje“, odnosno umetanje podataka u neki entitet (tablicu) baze podataka. Važno je zapamtiti da u zagrade poslije ključne riječi „VALUES“ upišemo podatke po redu kako idu atributi (stupci), osim ako „ručno“ ne izabiremo koje stupce želimo popuniti. Za primarni ključ koristimo sljedeću slobodnu vrijednost u sekvenci kako bismo postigli da svaki primarni ključ bude jedinstven i automatski generiran. Također, ako imamo unique constraint koji važi za neki stupac, tada ne smijemo ponavljati isti podatak u tom stupcu. Primjer: unošenje proizvoda u entitet „PROIZVOD.“

Slika na kojoj se prikazuje tekst, snimka zaslona, broj, Font

Opis je automatski generiran

Slika 13 - primjer unosa u entitet, te samog entiteta poslije unosa

## „UPDATE“ upit

Ovaj upit koristimo kako bismo ažurirali (postojeće) podatke u nekom entitetu baze podataka, no prvo je potrebno pronaći podatak (ili podatke) koje želimo promijeniti korištenjem ključne riječi „WHERE.“ Primjerice, ako smo unijeli zaposleniku krivu adresu, na ovaj ju način možemo promijeniti traženjem jedinstvenog ključa tog zaposlenika.

Slika na kojoj se prikazuje tekst, Font, snimka zaslona, bijelo

Opis je automatski generiran

Slika 14 - primjer ažuriranja retka u tablici „ZAPOSLENICI“

## „DELETE“ upit

„DELETE“ koristimo kada želimo izbrisati „n-torku“ (redak) ili „n-torke“ iz tablice. Također dohvaćamo željene (odnosno neželjene) podatke pomoću upita „WHERE“ kako bismo ih uopće mogli „obrisati.“ Na primjer, ako neki od korisnika web trgovine žele izbrisati svoj račun, tada ih mi možemo izbrisati i s naše strane iz entiteta „KORISNICI.“

Slika na kojoj se prikazuje tekst, Font, snimka zaslona, bijelo

Opis je automatski generiran

Slika 15 - primjer brisanja retka iz tablice „KORISNICI“

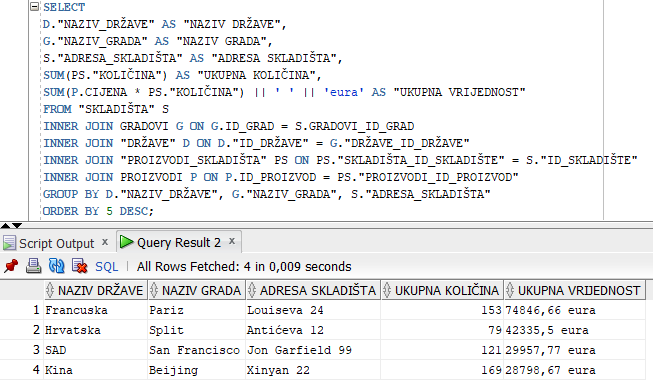
## „SELECT“ upit

Ovaj se upit se koristi za prikaz postojećih podataka u entitetu, te se njihovim pozivanjem prikazuje nula, jedna ili veći broj n-torki koje zatraži onaj koji zapravo i vrši „SELECT“ upit. Kao što je i prije spomenuto, možemo koristiti i „filtrirajući“ upit „WHERE“ u uniji sa „SELECT“ upitom kako bismo najefikasnije „selektirali“ željene podatke.

Postoje i razni drugi važni upiti i koncepti kao što su primjerice: „ORDER BY“ ako nam je bitan poredak prikazanih podataka, „JOIN“ za povezivanje više tablica, agregatne funkcije, odnosno ugrađene funkcije koje su nam na raspolaganju kao „COUNT()“ ili „SUM()“, podupiti, to jest upiti unutar upita itd. Ovim svim upitima i konceptima ćemo se i služiti u sljedećim primjerima „SELECT“ upita na našoj bazi podataka – Web trgovina.

### Prvi primjer „SELECT“ upita

Menadžer web trgovine traži izvještaj o pregledu inventara. Traže se podaci o ukupnoj vrijednosti (u eurima) i količini proizvoda u inventaru svakog pojedinog skladišta, te gdje se uopće nalazi to skladište (grad i država). Izvještaj, to jest tablica mora biti poredana, odnosno sortirana, silazno, po vrijednosti unutar skladišta.



Slika 16 - prvi primjer SQL upita i dobiven prikaz podataka

### Drugi primjer „SELECT“ upita

Web trgovina pokušava održati nagradnu igru koja se temelji na principu lutrije. Izvlače se kuglice sa različitim obilježjima, te oni korisnici koji zadovoljavaju sva ta obilježja dobivaju (još neodređenu) nagradu. Nakon izvlačenja kuglica, dobivena obilježja korisnika moraju biti sljedeća: korisnik ne smije biti označen na profilu kao muškarac, i njihovo ime i njihovo prezime moraju sadržavati slovo "n", trebali su ukupno potrošiti više od 100, a manje od 600 eura na web trgovini, te njihov OIB (odnosno jedinstven identifikator) mora sadržavati broj "78" i još mora vrijediti da su rođeni u prošlom tisućljeću. Traži se prikaz svih korisnika koji zadovoljavaju ova obilježja, točnije: njihovo ime i prezime, jedinstveni "OIB" tih korisnika, datum rođenja, njihov spol, te koliko su ukupno potrošili novca na web trgovini.

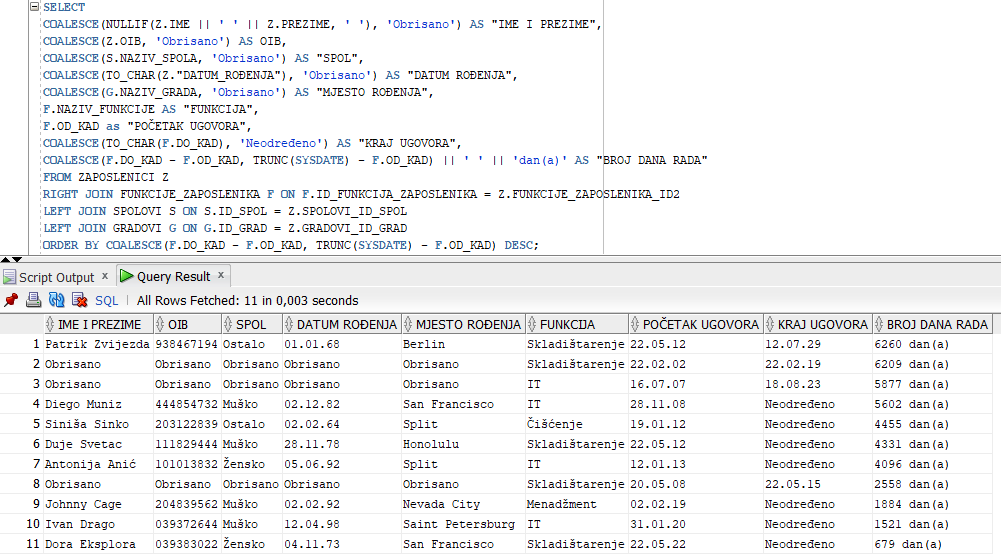
Slika na kojoj se prikazuje tekst, snimka zaslona, broj, Font

Opis je automatski generiran

Slika 17 - drugi primjer SQL upita i dobiven prikaz podataka

### Treći primjer „SELECT“ upita

Menadžera web trgovine zanimaju određene informacije vezane uz trajanje ugovora svakog trenutnog i/ili bivšeg zaposlenika. Znači, potreban nam je ispis (prikazanog kao entitet) na kojem ćemo saznati, kao prvo, o kojim se zaposlenicima radi, ako još uvijek rade(tj. traže se osnovni osobni podaci, tj. ime, prezime, OIB, spol, te datum i mjesto rođenja) ili upisati "Obrisano" ako više nemaju te podatke sadržane u bazi podataka, treba nam i njihova trenutna, odnosno bivša funkcija u radu za web trgovinu, te od kad, (pa i eventualno do kad) su izvršavali tu funkciju, s tim da ako im istek ugovora nije određen onda to trebamo i naznačiti, također odmah do tog podatka je željen taj isti podatak, no prikazan u formatu broja dana rada za web trgovinu radi lakšeg iščitavanja. Sve ove podatke, radi boljeg pregleda, sortirati silazno po broju dana rada za web trgovinu.



Slika 18 - treći primjer SQL upita i dobiven prikaz podataka

### Četvrti primjer „SELECT“ upita

Potrebna nam je detaljna analiza svih digitalnih računa koje smo izdali poslije 2018. godine. Trebamo prikazati prezime i ime (u formatu "prezime, ime") svakog izdavatelja računa i korisnika poslije te godine. Uz korisnika je potreban i njihov "OIB", odnosno jedinstven numerički identifikator. Također nam je bitno da točno saznamo kontinent i državu u kojoj se korisnik nalazi, te poštanski broj na koji su poslani ti proizvodi, ako je korisnik na računu uopće dao poštanski broj. Ako nije to trebamo u prikazu istaknuti. Potrebna nam je i informacija o tome koliko je kupac potrošio na pojedinu stavku računa, koliko su pojedinog proizvoda kupili, u koju kategoriju spadaju ti proizvodi, te iz kojeg su skladišta došli. Također, potreban nam je i datum same kupnje. Bitno je i na koji je način izvedena sama kupnja (pouzećem ili karticom), potrebno je ispisati broj kartice ako ju je korisnik dao. Prikaz sortirati po datumu kupnje, pa po prezimenima kupaca.

Slika na kojoj se prikazuje tekst, snimka zaslona, Font, dokument

Opis je automatski generiran

Slika 19 - četvrti primjer (SQL upit)

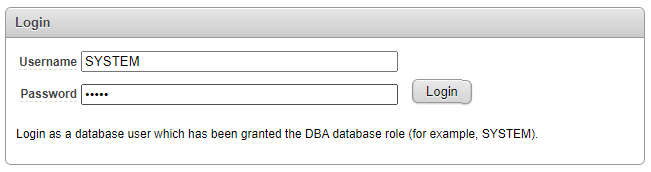


Slika 20 - četvrti primjer (dobiven prikaz podataka)

# APEX

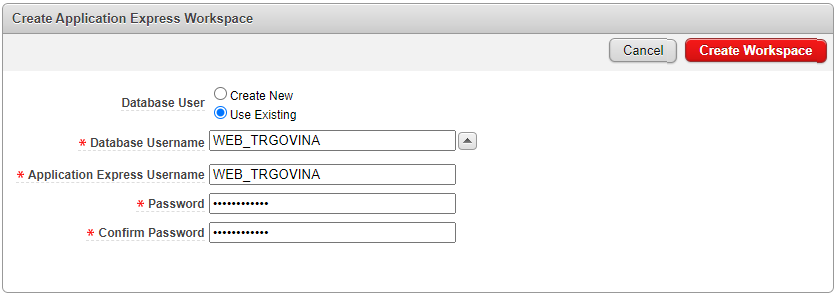
Kako bismo si olakšali korištenje i „manipuliranje“ našom bazom podataka, te kako bismo povećali njenu pristupačnost koristiti ćemo APEX. To je „web“ aplikacija sa jednostavnim i pristupačnim korisničkim (grafičkim) sučeljem, odnosno GUI-om (eng. Graphical User Interface). Ako trebamo dati nekome pristup iz Web trgovine pristup manipulaciji naše baze podataka, to ćemo najlakše učiniti ovdje, jer je APEX dosta „beginner-friendly“.

Prvo što trebamo napraviti je kad „otvorimo“ Oracle Database stranicu jest kliknuti tab APplication EXpress (APEX) što će nas dovesti na formu u koju prvo trebamo unijeti korisničke podatke SYSTEMA.



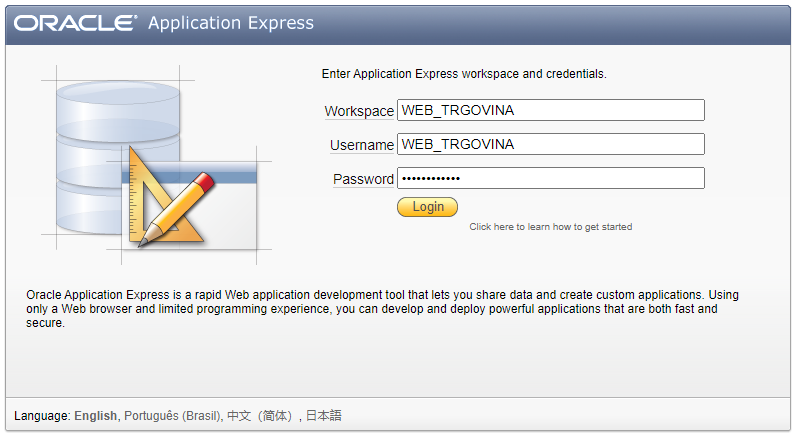
Slika 21 – prijava na SYSTEM račun

Sad dolazi vrijeme kad našu postojeću bazu podataka i njenu shemu moramo povezati sa APEX-om. Ovo radimo isto upisom korisničkih podataka u formu slično kao što smo radili i za SYSTEM (samo moramo označiti da koristimo postojeći račun).



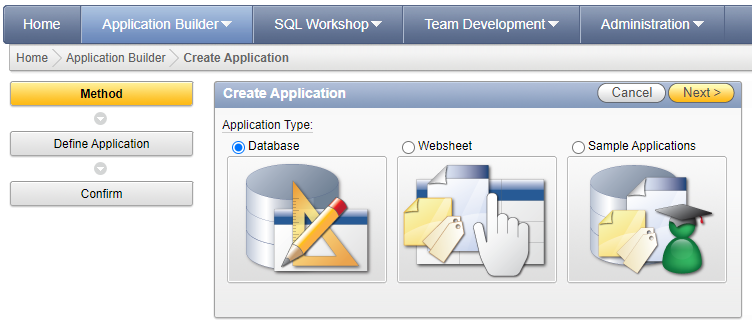
Slika 22 – kreiranje računa web trgovine u APEX

Ovdje smo se samo registrirali, poslije to se zapravo trebamo i prijaviti jer se to ne učini automatski poslije registracije.

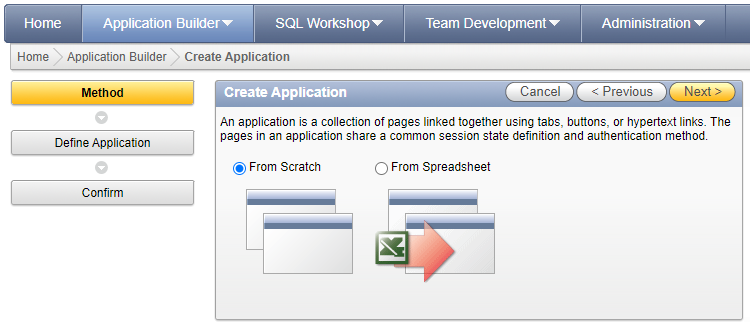


Slika 23 - prijava web trgovine na APEX

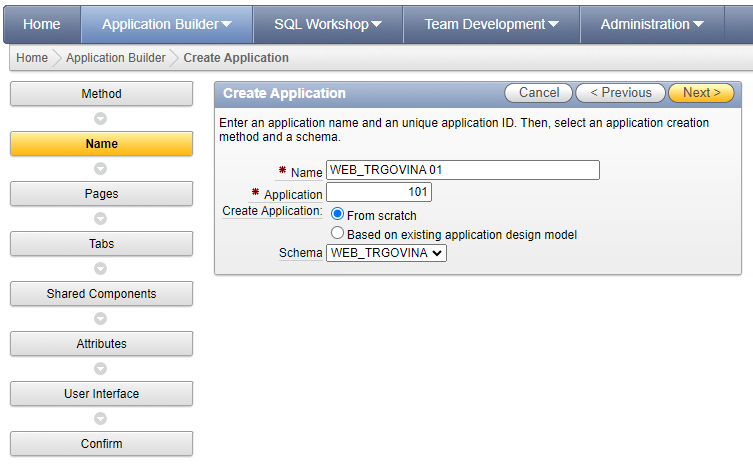
Sad možemo punim kapacitetom koristiti APEX. Idući korak jest napraviti „novu“ aplikaciju, odnosno bazu podataka. Ovo radimo putem Application Wizarda koji nam jako olakšava posao.



Slika 24 - izrada baze podataka

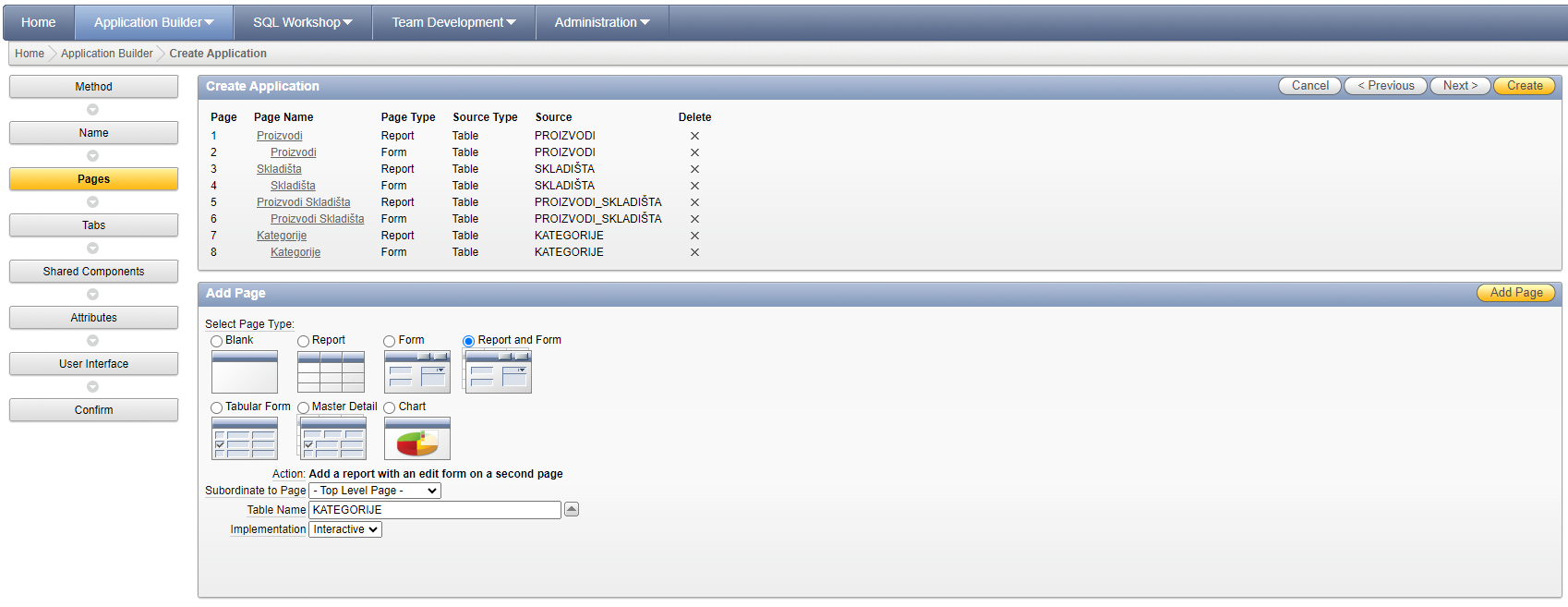


Slika 25 - zbog naših potreba radimo bazu otpočetka

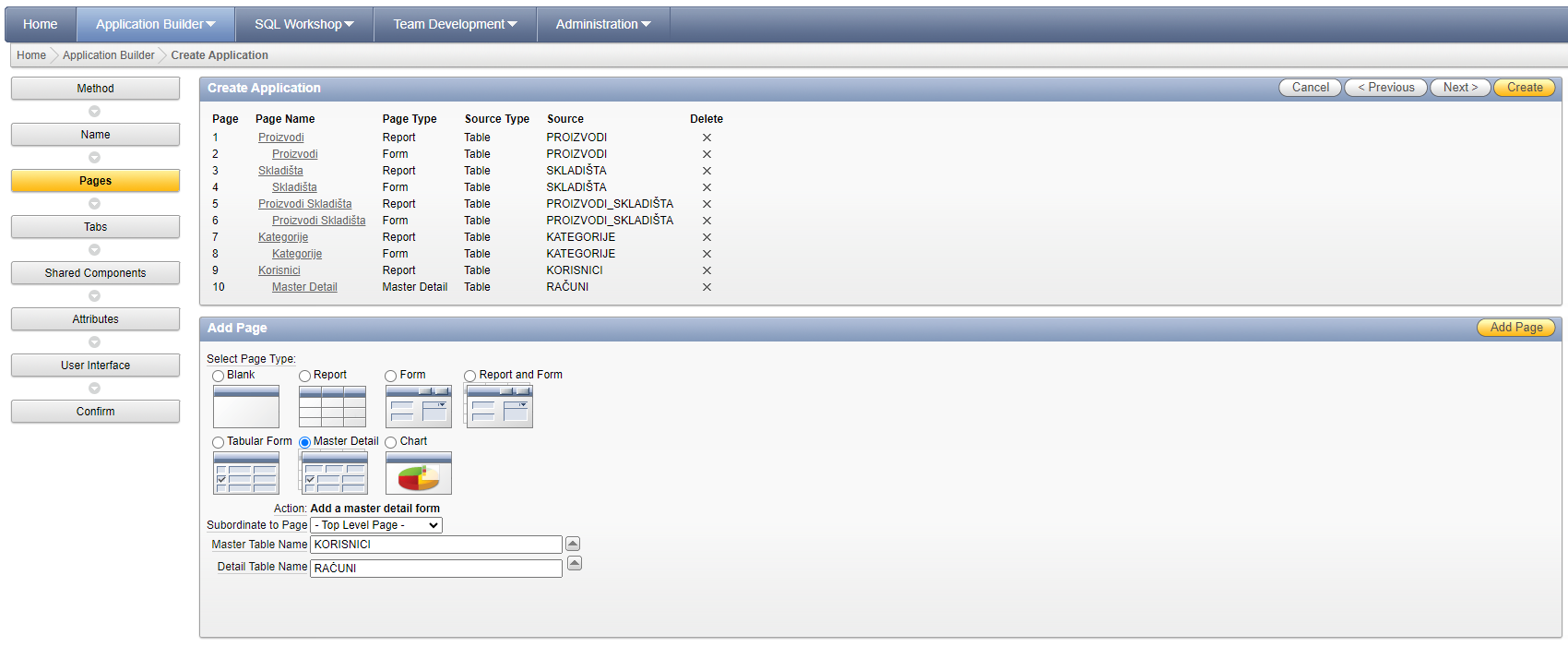


Slika 26 - dajemo ime aplikaciji

Nakon dodjeljivanja imena aplikaciji, odabrati ćemo tipove stranica za prikazivanje aplikacije. Za naše potrebe koristiti ćemo samo tipove: „Report and Form“ i „Master detail“. U „Report and Form“ tipu se prikazuju „osnovnije“ informacije. Kao primjer za taj tip ćemo staviti tablice: PROIZVODI, SKLADIŠTA, PROIZVODI\_SKLADIŠTA i KATEGORIJE. „Master detail“ stranice nam prikazuju informacije u tablicama na detaljniji način: Kao primjer za taj tip ćemo staviti tablice: KORISNICI i RAČUNI.

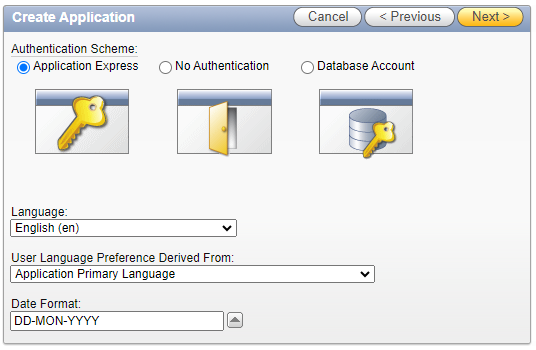


Slika 27 – „Report and Form“ primjeri



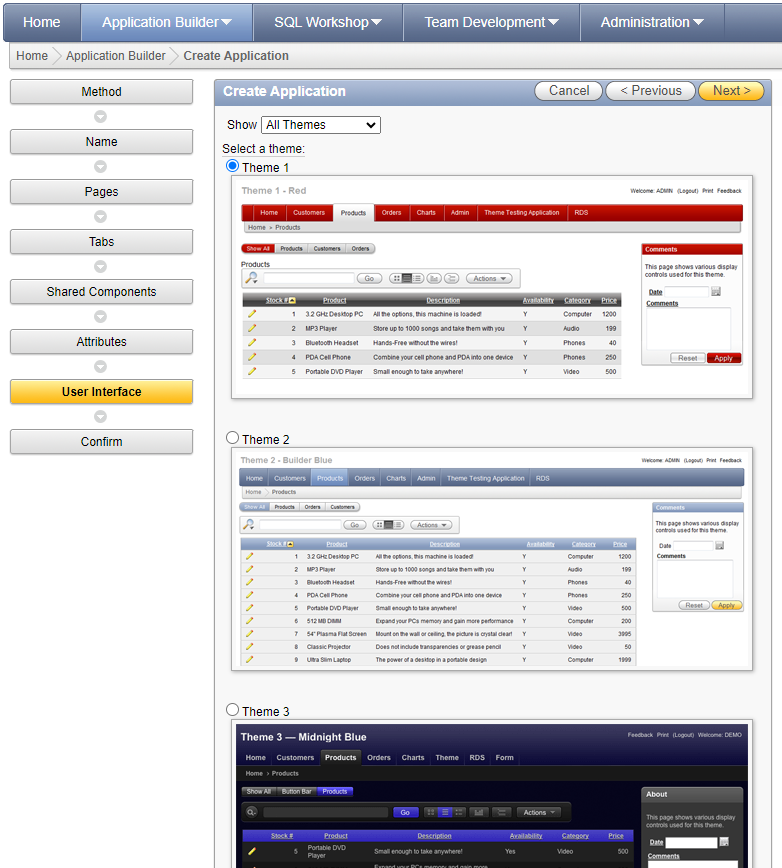
Slika 28 - dodani „Master detail“ primjeri

Zbog pristupačnosti možemo izabrati i format kojim bi nam se prikazali datumi.



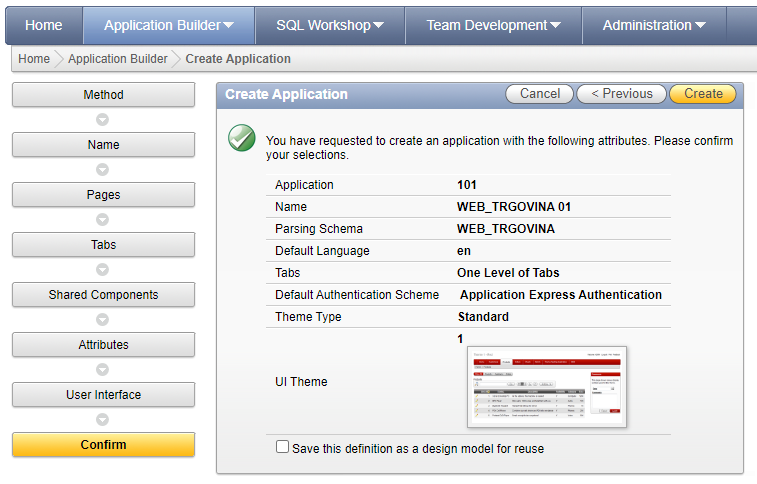
Slika 29 - prozor za biranje autentifikacije, jezika i formata datuma

Također možemo birati i „temu“ odnosno temeljni izgled i spektar boja stranice. Radi jednostavnosti ovog primjera ćemo odabrati najosnovniju temu, temu broj 1.



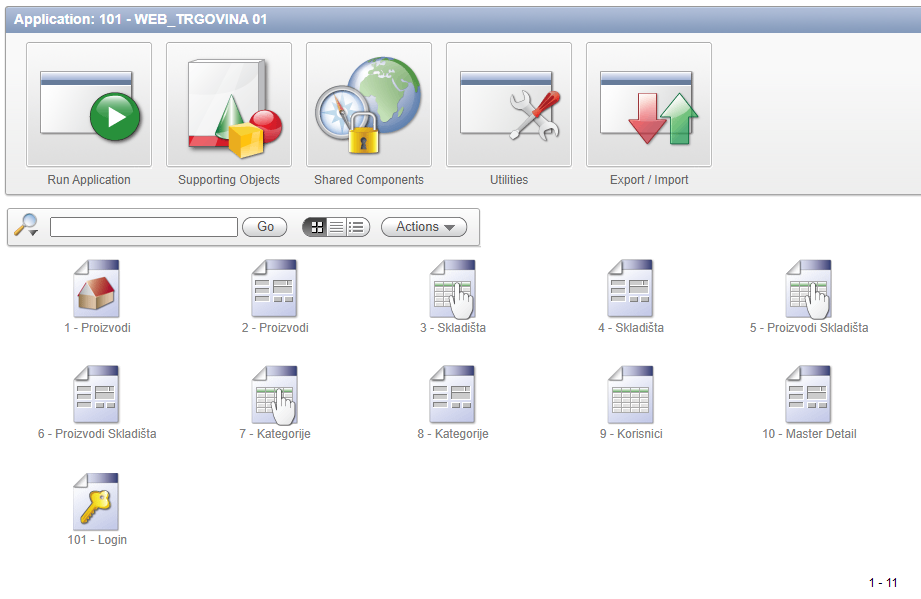
Slika 30 - biranje teme stranice

Poslije svih ovih koraka možemo kreirati željenu aplikaciju.

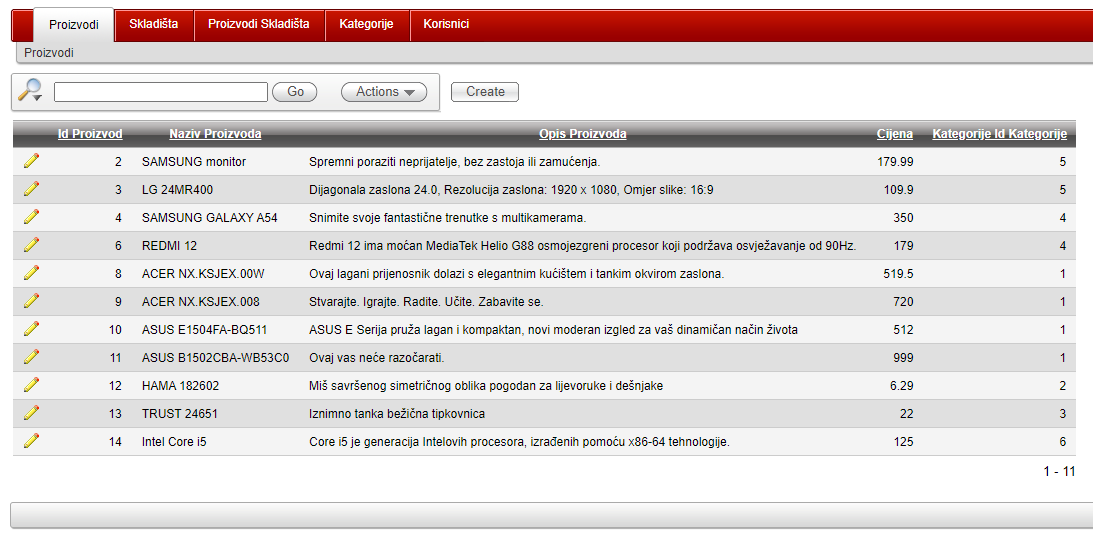


Slika 31 - prozor za kreiranje aplikacije

Sad kad se vratimo na Home tab možemo vidjeti našu aplikaciju. Kad kliknemo na button „Run application“ i nakon što upišemo korisničko ime i lozinku, tad možemo napokon vidjeti našu stranicu i također njome (u granicama) manipulirati.



Slika 32 – Home tab (možemo vidjeti našu novokreiranu aplikaciju)



Slika 33 - konačan izgled aplikacije

APEX nam je omogućio ovu stranicu za pristupačan vizualni prikaz naše baze podataka – Web trgovina. Kao što je prije rečeno, tu se mogu, ne samo vidjeti nego i mijenjati podaci. Primjerice, možemo uređivati n-torke (retke) klikom na olovku, filtrirati na različite načine, sortirati klikom na pojedini stupac i dodavati nove n-torke klikom na button „Create“.