**PRIRODOSLOVNO-MATEMATIČKI FAKULTET   
SVEUČILIŠTE U SPLITU**

SEMINARSKI RAD

**INFORMATIČKI PROJEKT IZ BAZA PODATAKA**

**BAZA PODATAKA FILMOVA**

Profesorica: Student:

Monika Mladenović Kristian Nenadović

Informatika

Split, 2020.

SADRŽAJ

[1 UVOD 1](#_Toc38232909)

[1.1 Općenito 1](#_Toc38232910)

[1.2 Opis problema 1](#_Toc38232911)

[2 Entiteti 2](#_Toc38232912)

[2.1 Osoba 2](#_Toc38232913)

[2.2 Film 2](#_Toc38232914)

[2.3 Spol 2](#_Toc38232915)

[2.4 Recenzija 2](#_Toc38232916)

[2.5 Glumac\_Film 2](#_Toc38232917)

[2.6 Scenarist\_Film 3](#_Toc38232918)

[2.7 Drzava 3](#_Toc38232919)

[2.8 Grad 3](#_Toc38232920)

[2.9 Lokacija\_Snimanja 3](#_Toc38232921)

[2.10 Klasifikacija 3](#_Toc38232922)

[2.11 Zanr 3](#_Toc38232923)

[2.12 Film\_Zanr 3](#_Toc38232924)

[2.13 Lista 3](#_Toc38232925)

[2.14 Lista\_Film 3](#_Toc38232926)

[2.15 Korisnik 4](#_Toc38232927)

[2.16 Ocjena\_Film 4](#_Toc38232928)

[3 RELACIJE 5](#_Toc38232929)

[3.1 Relacija jedan na jedan 5](#_Toc38232930)

[3.2 Relacija jedan na više 5](#_Toc38232931)

[3.3 Relacija više na više 5](#_Toc38232932)

[4 LOGIČKI MODEL 7](#_Toc38232933)

[5 RELACIJSKI MODEL 9](#_Toc38232934)

[6 IZRADA BAZE PODATAKA PREMA MODELU 11](#_Toc38232935)

[7 UPITI NA BAZU PODATAKA 15](#_Toc38232936)

[7.1 INSERT 15](#_Toc38232937)

[7.2 UPDATE 16](#_Toc38232938)

[7.3 DELETE 17](#_Toc38232939)

[7.4 SELECT 19](#_Toc38232940)

[7.5 ORDER BY 19](#_Toc38232941)

[8 ZADACI 20](#_Toc38232942)

[9 APEX 25](#_Toc38232943)

[9.1 Application wizard 26](#_Toc38232944)

# UVOD

## Općenito

Baze podataka o filmovima pružaju informacije vezane za filmove i sve o njima. Film je krajnji produkt velikog broja ljudi od kojih svatko ima određeno zanimanje pa tako za film vežemo glumce, redatelje, scenariste te ostalu ekipu u produkciji. Čest je slučaj da jedna osoba može imati više zanimanja u jednom filmu, tako primjerice jedna osoba može biti ujedno i redatelj i glumac. Ostale informacije koje korisnika mogu zanimati uz film su njegova radnja, žanrovi, lokacije na kojima je film snimljen, kada je održana premijera filma ili kritičke recenzije od strane poznatih kritičara.

## Opis problema

Za ovaj projekt bilo je potrebno osmisliti i realizirati bazu podataka koja može poslužiti za organizaciju i pretraživanje filmova kao što je npr. IMDb. Potrebno je omogućiti pregled filmova po odgovarajućim opcijama pretrage koje korisnik odabere (npr. po žanru, godini izlaska, glumcu...) te podacima o svakom filmu: godini izlaska, glumcima, redatelju, žanrovima itd.

Kako je cilj korisnicima pružiti što bolje iskustvo, dana im je mogućnost kreiranja vlastitih lista filmova po želji, koje kasnije mogu nadopunjavati i uređivati. Također, pozvani su pridonijeti zajednici kako bi pomogli ostalim budućim korisnicima pri odabiru filma opcijom ocjenjivanja bilo kojeg filma brojem od 1 do 10 te komentiranja i davanja vlastitog mišljenja.

# Entiteti

1. Osoba
2. Film
3. Spol
4. Recenzija
5. Glumac\_Film
6. Scenarist\_Film
7. Drzava
8. Grad
9. Lokacija\_Snimanja
10. Klasifikacija
11. Zanr
12. Film\_Zanr
13. Lista
14. Lista\_Film
15. Korisnik
16. Ocjena\_Film

## Osoba

Ovaj entitet namijenjen je za pohranu podataka o ljudima koji su dio ekipe filma. Dobivamo podatke kao što su **Ime** i ***Prezime***osobe, **Datum\_rodenja**, a **Spol** i **mjesto rođenja** preko stranih ključeva iz entiteta **Spol** i **Grad** . U slučaju ako je osoba umrla opcionalan je **Datum\_smrti**.

## Film

Središnji entitet **Film** sadrži informacije o najosnovnijim značajkama filma. Uz **Naslov** samog filma dodao sam i atribut **Opis\_filma** koji će sadržavati kratak opis radnje, te **Trajanje** koje je obično u minutama. Također je moguć uvid koliko je novaca uloženo u stvaranje filma preko atributa **Budzet**, **Datum\_izlaska** filma te njegovu prosječnu **Ocjenu**. Prema poslovnom pravilu da svaki film može imati samo jednog redatelja (a redatelj više filmova) postavljen je strani ključ iz entiteta **Osoba**, koji predstavlja podatak o redatelju. Isto tako preko stranog ključa dohvaćamo **Klasifikaciju** sadržaja filma.

## Spol

Navedeni su atributi Naziv i Oznaka spola.

## Recenzija

Film može biti ocjenjen od strane poznatih akademskih filmskih kritičara ili novinara pa stoga kompozitni ključ ovog entiteta čine ID-evi **Filma** i **Osobe** pa tako dobivamo mogućnost dohvaćanja informacija i o poznatom kritičaru. Svaka recenzija može imati **Ocjenu** od 1-5, **Komentar** kritičara te **Datum\_objave**.

## Glumac\_Film

U nekom filmu se nalaze glumci koji su tijekom života snimili više filmova pa preko kompozitnog ključa kojeg čine **ID\_Osoba** i **ID\_Film** vežemo glumca ( o kojem možemo dohvatiti informacije iz entiteta **Osoba**) za neki točno određeni film te kao atribut **Naziv\_lika** dobivamo ime uloge tj. ime karaktera kojeg je taj glumac glumio.

## Scenarist\_Film

U ovom entitetu **Osoba\_Id** i **Film\_Id** čine kompozitni ključ koji povezuje scenarista sa filmom. Za neki film može raditi više scenarista od kojih svaki može pisati poseban scenarij za određene likove pa za tu namjenu kategoriziranja služi atribut **Vrsta\_scenarija**. Primjerice, ako je film adaptacija knjige osim samog autora knjige, scenarij može pisati i neki drugi scenarist.

## Drzava

Ovaj entitet sadrži atribut **Naziv\_Drzave** koji se kasnije može dobiti ubacivanjem primarnog ključa **ID\_Drzava** kao strani ključ u entitet **Grad**. Može predstavljati državu rođenja osobe ili državu snimanja filma.

## Grad

U entitet **Grad** upisujemo **Naziv** i njegov **Poštanski\_broj** te ti podaci mogu služiti za određivanje mjesta rođenja neke osobe ili kao podatak o gradu u kojemu je snimljen film.

## Lokacija\_Snimanja

Ovo je entitet koji je nastao kao posljedični entitet iz veze više na više između **Grada** i **Filma** jer filmovi mogu biti snimljeni u različitim gradovima, a u nekom gradu može biti snimljeno više filmova. ID-evi ta dva entiteta **Grada** i **Filma** čine kompozitni ključ u ovom entitetu.

## Klasifikacija

Prema poslovnom pravilu da kriterij cenzure sadržaja filma vrijedi jednako za sve države, stvoren je entitet **Klasifikacija** u kojemu se nalaze atributi o punom **Nazivu klasifikacije** te njezinom skraćenom zapisu tj. **Oznaka**.

## Zanr

U ovom entitetu upisujemo **Naziv žanra** pojedinog filma.

## Film\_Zanr

Najčešće je svaki film kategoriziran s više različitih žanrova pa je ovo posljedični entitet koji je nastao iz veze **m:n** između entiteta **Film i Zanr** te njihovi primarni ključevi skupa čine kompozitni primarni ključ.

## Lista

Ovaj entitet predstavlja listu filmova koje korisnik može kreirati i uređivati. Proizvoljno može postaviti **Naziv\_liste** te postavku **Privatnosti** tj. hoće li lista biti vidljiva samo njemu ili i ostalim ljudima. Moguće je upisati **Datum\_kreiranja** te preko stranog ključa **ID\_Korisnik** povezujemo **Listu** s korisnikom.

## Lista\_Film

Kako je moguća situacija da se jedan film pojavljuje u više kreiranih lista drugih korisnika te se u listi nalazi uglavnom više filmova, proizlazi ovaj entitet kojemu su ključevi **ID\_Lista** i **ID\_Film** spojeni u kompozitni.

## Korisnik

Entitet **Korisnik** predstavlja klijenta koji je u sustav baze filmova prijavljen preko atributa **Email**, koji je uz **Ime** dovoljan podatak za mogućnost pretraživanja filmova, ocjenjivanja i komentiranja te kreiranja vlastitih lista filmova po želji. Po **Datum\_uclanjenja** evidentiramo kada je korisnikov profil kreiran.

## Ocjena\_Film

Svaki korisnik ima mogućnost ocjenjivanja filma **Ocjenom** od 1-10 te postavljanja **Komentara** uz koji bi stajao **Datum\_objave**.

# RELACIJE

Relacije (veze) odnosi su između 2 tablice koji pokazuju na koji način su entiteti međusobno povezani. Postoje 3 vrste relacija:

1. One-to-one ( jedan na jedan ) 1:1
2. One-to-many (jedan na više) 1:n
3. Many-to-many (više na više) m:n

## Relacija jedan na jedan

Svaka n-torka jednog entiteta može biti povezana samo s jednom n-torkom drugog entiteta.

Nije korišteno u izradi ove baze podataka.

## Relacija jedan na više

Najučestaliji tip relacije koji govori da jedan element iz entiteta A može biti povezan sa više elemenata iz entiteta B, a neki element iz B može biti povezan samo sa jednim elementom iz A.

* **Osoba** može biti jednog **spola**, ali jednom spolu možemo pridružiti više osoba.
* Jedna **država** može imati više **gradova**, ali jedan grad se nalazi u samo jednoj državi.
* U nekom gradu se može roditi više osoba, ali osoba ima samo jedno mjesto rođenja.
* Prema poslovnom pravilu vrijedi da film može imati samo jednog redatelja, a jedan redatelj može tijekom života režirati više filmova, pa iz entiteta **Osoba** pravimo vezu 1:n s **Filmom** .
* Svaki **korisnik** ovog sustava može kreirati više **lista** filmova ( npr. „najdraži trileri“, „horori“,…) , no svaka **lista** je točno određena jednim **korisnikom**.
* Prema poslovnom pravilu da postoji jedan sistem rangiranja sadržaja filma koji se vrednuje jednako za svaku državu znači da svaki film ima jednu jedinu **klasifikaciju**, dok neka klasifikacija može vrijediti za više **filmova**.

## Relacija više na više

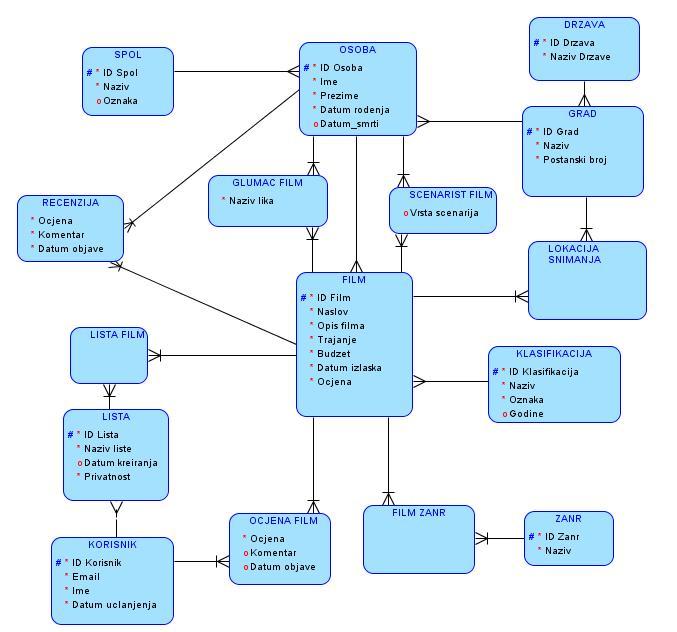
Jedan entitet može imati više elemenata iz drugog entiteta i obrnuto. Zbog nepoželjnosti ove relacije u bazama podataka, a s obzirom na to da se pojavljuje dosta često, rješava se uvođenjem novog (posljedičnog) entiteta čiji će primarni ključ u većini slučajeva biti sastavljen od primarnih ključeva osnovnih entiteta te kao takav činiti tzv. Kompozitni primarni ključ.

U mojoj bazi ovakve veze su se pojavile najviše puta:

* S obzirom da jedan **film** može biti snimljen u više različitih **gradova**, a neki **grad** može imati pogodnu lokaciju za snimanje na kojoj se stalno snimaju novi filmovi proizlazi entitet Lokacija\_**Snimanja** sa primarnim ključevima entiteta **Grad i Film**
* Poznato je da na filmu radi jako velik broj ljudi pa radi što veće normalizacije za vezu između **Osobe** i **Filma** stvoreni su entiteti **Glumac\_film** i **Scenarist\_film**. Obje tablice sadrže kompozitni ključ **Id\_Film** i **Id\_Osoba**, te one predstavljaju vrstu zanimanja pa tako stvaramo mogućnost proširivanja baze npr. dodavanjem ostale ekipe filma ( kamermani, producenti zvuka…). U entitetu **Glumac\_film** postavljen je atribut **Naziv\_lika** kojeg je taj glumac glumio u nekom filmu koji je vezan primarnim ključem. Scenarist je osoba koja je zaslužna za pisanje dijaloga, osobina likova ili cijele radnje filma, pa je u tu tablicu dodan atribut **Vrsta\_scenarija** radi optimalnog specificiranja vrste scenarija.
* Kako većina **filmova** sadrži više **žanrova** i obratno, nastaje posljedični entitet **Žanr\_Film** povezan kompozitnim ključem ID-eva iz **Filma i Zanr**.
* **Korisniku** smo omogućili da može ocjenjivati filmove po želji, a svaki **film** može biti ocjenjen od više korisnika kako bi se izračunala prosječna ocjena filma, pa proizlazi entitet **Ocjena\_Film**
* Entitet **Lista\_Film** nastaje jer neka kreirana **lista** od strane korisnika može sadržavati više različitih **filmova**, no jedan **film** se može pojavljivati i u drugim **listama filmova**
* S obzirom da su filmski kritičari osobe o kojima možda želimo znati više, kreirali smo ih preko entiteta **Osoba** te se oni nalaze (primarni ključevi) skupa sa **ID\_Film** u entitetu **Recenzija** te iz njega čitamo koju je **Ocjenu** i **Komentar** dao neki **kritičar** za neki **film**

# LOGIČKI MODEL

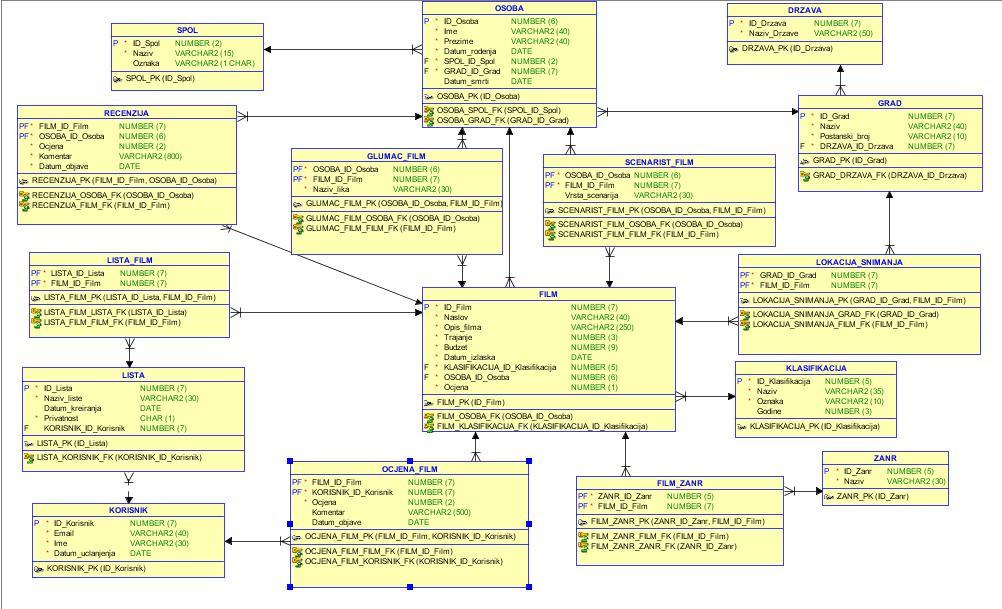
Logički model - ERM (Entity Relationship Model) uključuje sve entitete i odnose između njih. Prikazani su svi atributi za sve entitete te se određuje njihova optimalnost. Definira se opcionalnost i kardinalitet veza gdje su vidljivi primarni i/ili kompozitni ključevi, no ne i strani.



Slika 1 BP\_Filmova - logički model

# RELACIJSKI MODEL

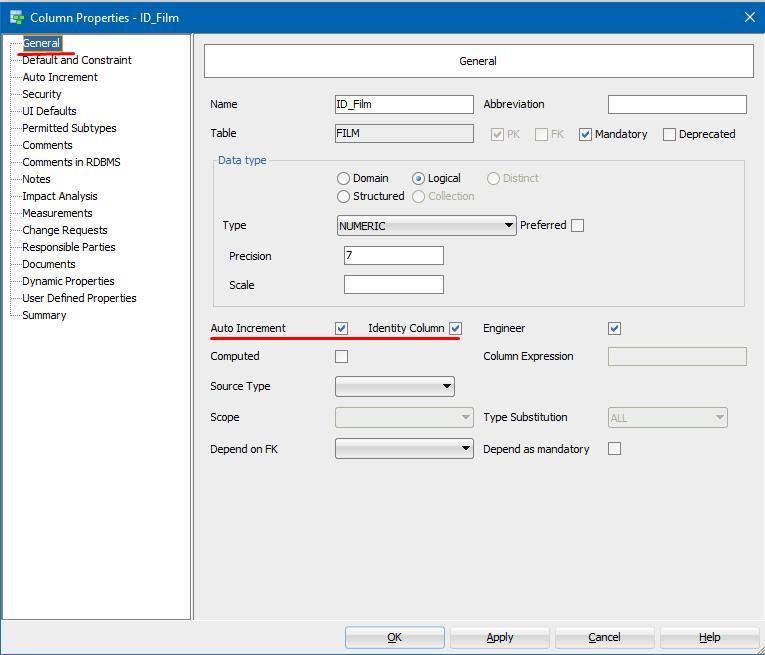
Nakon završenog logičkog modela izrađuje se relacijski model baze podataka. Iz ovog modela osim entiteta, relacija i atributa, možemo vidjeti i tipove podataka kao i strane ključeve u svim tablicama.



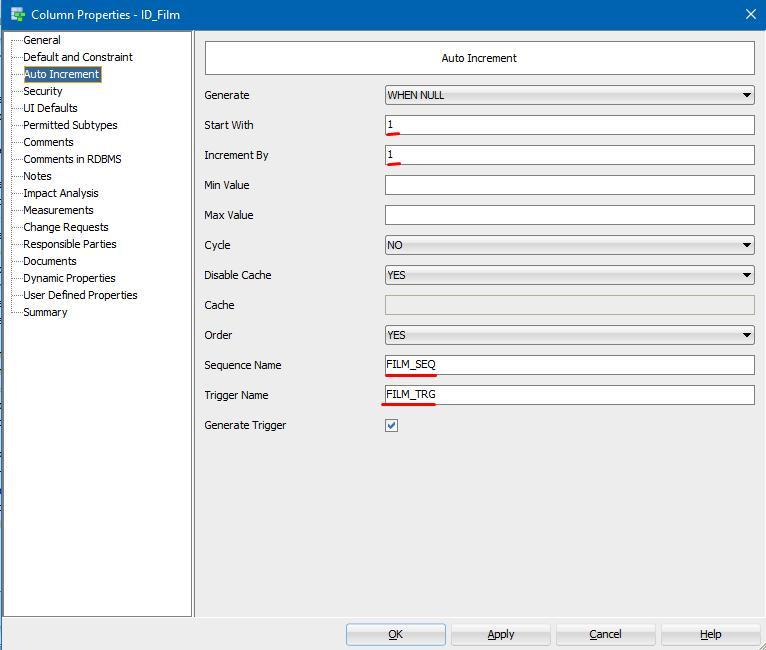
Slika 2 BP\_Filmova - Relacijski model

# IZRADA BAZE PODATAKA PREMA MODELU

Nakon što smo kreirali logički i relacijski model, potrebno je izraditi DDL te pomoću njega to sve prenijeti u SQL Developer gdje ćemo početi unositi podatke u bazu i izvršavati razne upite. Kako za Oracle bazu ne postoji ugrađeni tip podataka za primarni ključ, prije izrade DDL-a svim primarnim ključevima radimo sekvencu (sequence) i okidač (trigger) koji će uvećavati ID za svaki sljedeći unos.

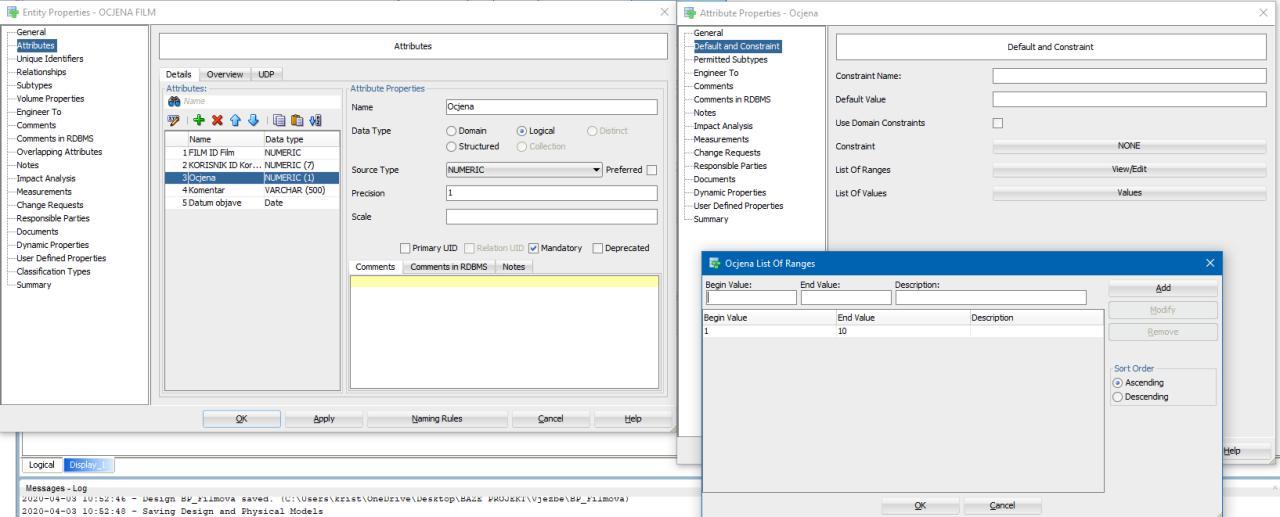


Slika 3 Označujemo Auto Increment i da je to Primarni ključ



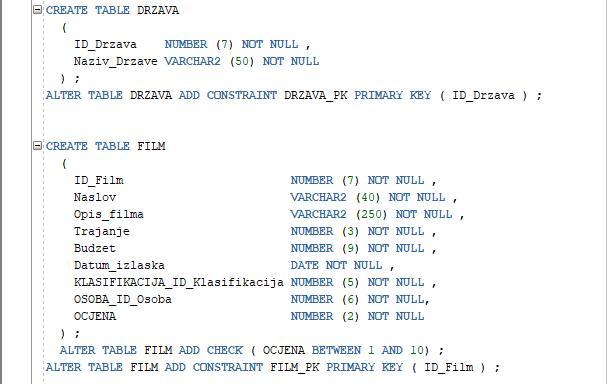
Slika 4 Izrada sekvenci i okidača

Kako bi osigurali da se za neki atribut npr. **Ocjena** ne može unijeti bilo koji broj, izrađujemo Constraint tj. ograničavanje unosa broja na 1-10.



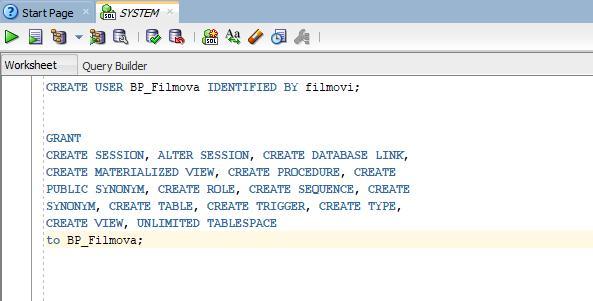
Slika 5 Postavljanje Ograničenja unosa broja

Nakon izrade sekvenci i okidača u relacijskom modelu, potrebno je izraditi DDL. Klikom na ikonu Generate DDL u SQL Modeleru, generiramo traženi DDL koji predstavlja kod za kreiranje svih entiteta, atributa, sekvenci i okidača naše baze podataka.



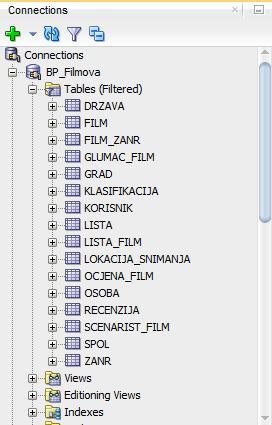
Slika 6 Primjer DDL-a: kreiranje tablica

Sada se prebacujemo u SQL Oracle developer gdje pravimo korisnika SYSTEM preko kojeg radimo svakog novog korisnika te njegovu šifru. Za moj slučaj to će biti korisnik BP\_Filmova kojem također dajemo neke najosnovnije ovlasti.



Slika 7 Kreiranje korisnika i definiranje ovlasti

Nakon kreiranja novog korisnika tj. nove konekcije, kopiramo sve iz generirane skripte DDL-a, zatim to zalijepimo i pokrenemo program. Osvježavanjem konekcije BP\_Filmova vidimo da smo uspješno napravili sve tablice.

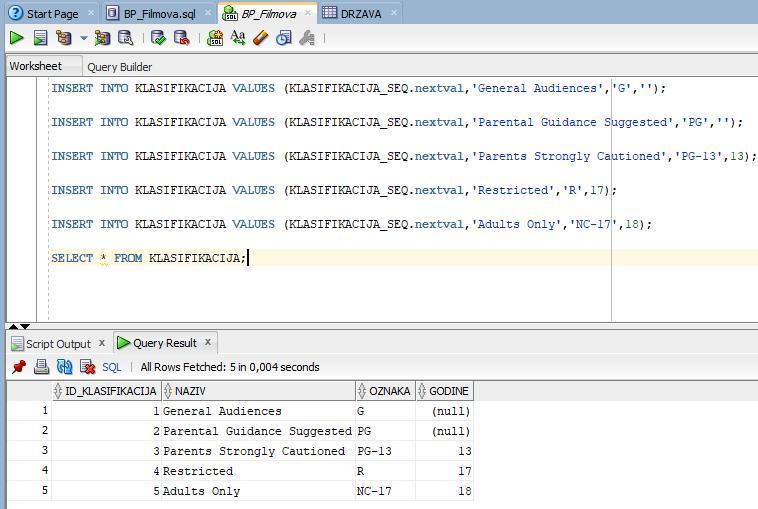


Slika 8 Tablice nakon izvršavanja DDL-a

# UPITI NA BAZU PODATAKA

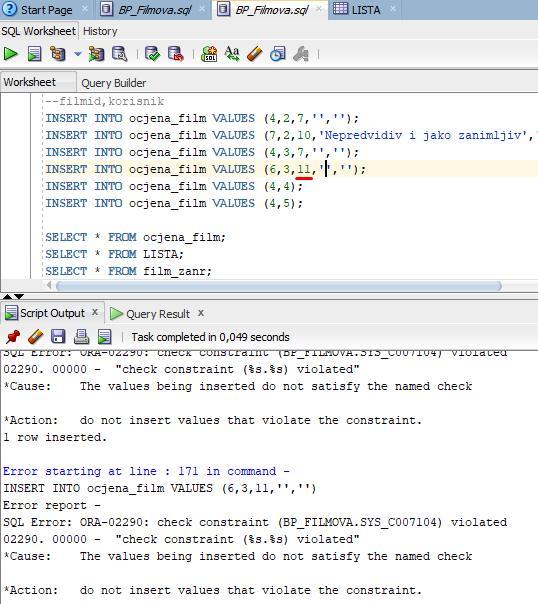
## INSERT

Insert je izraz koji koristimo kako bismo upisali podatke u tablicu. Važno je popunjavati po redu kako smo odredili atribute za svaku tablicu.



Slika 9 Insert podataka u tablice

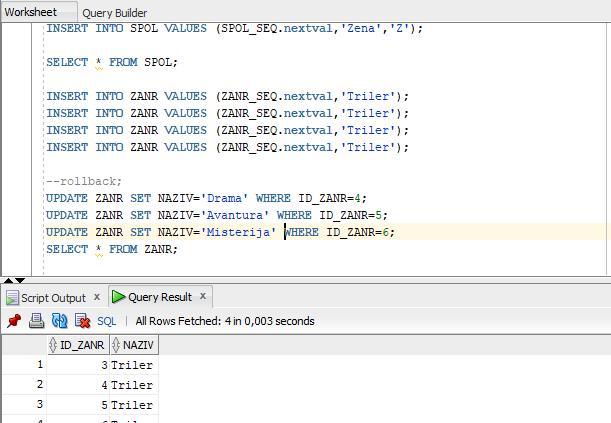
Ovdje je vidljiva nemogućnost unošenja brojeva koji ne pripadaju rangu kojeg smo naveli ranije (1-10) tako da sustav izbacuje grešku i javlja da smo prekršili ograničenje.



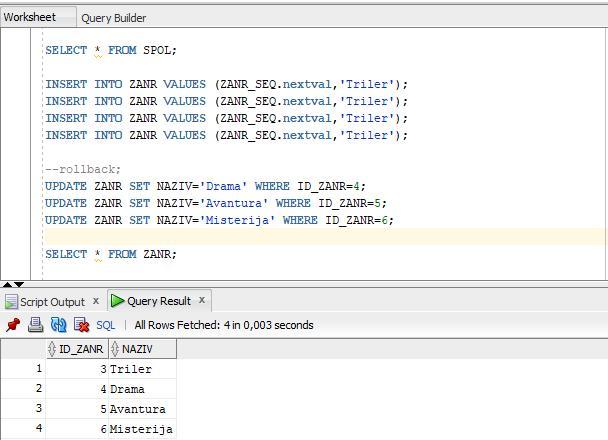
Slika 10 Prikaz djelovanja constrainta

## UPDATE

Upit Update koristimo kada želimo ažurirati podatke u bazi podataka.



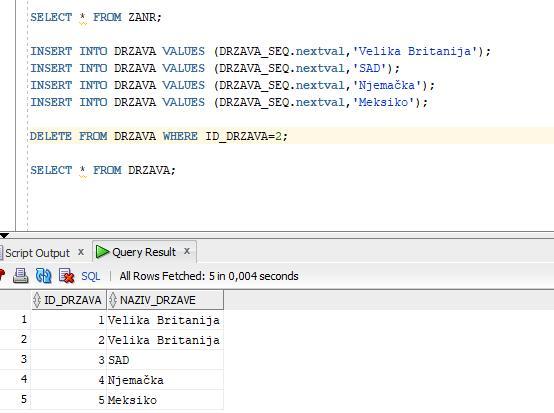
Slika 11 Prikaz selecta prije Update



Slika 12 Prikaz selecta nakon Update

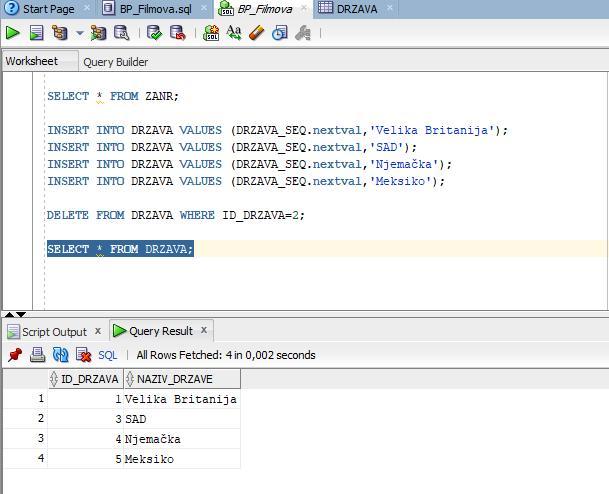
## DELETE

Ovaj upit koristimo kada želimo izbrisati neki podatak iz baze podataka. U ovom primjeru je slučajno unesen dupli redak međutim oni nisu potpuno isti jer nemaju isti ID.



Slika 13 Prikaz tablica prije brisanja

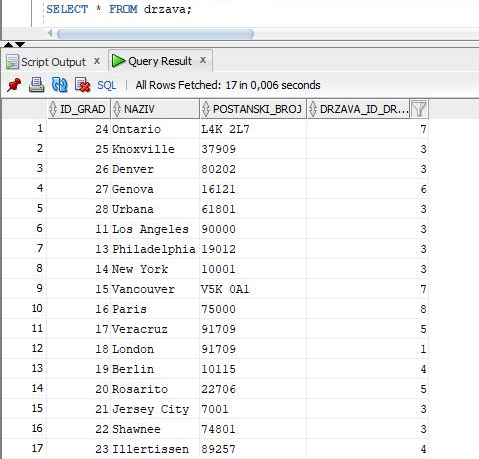
Uočimo kako se ostali ID-evi nisu promijenili već su ostali isti. Kada bismo nastavili unositi podatke sljedeći ID bi bio 6, a ne 2. To je iz razloga što smo već imali tablicu s tim ID-em, ali smo ju obrisali pa se ne smije više ponavljati taj broj.



Slika 14 Prikaz tablica nakon brisanja

## SELECT

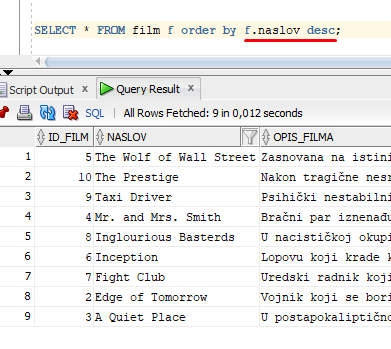
Select je upit u SQL jeziku koji se koristi za povrat odabranog skupa podataka iz jedne ili više tablica.



Slika 15 Jednostavni Select upit

## ORDER BY

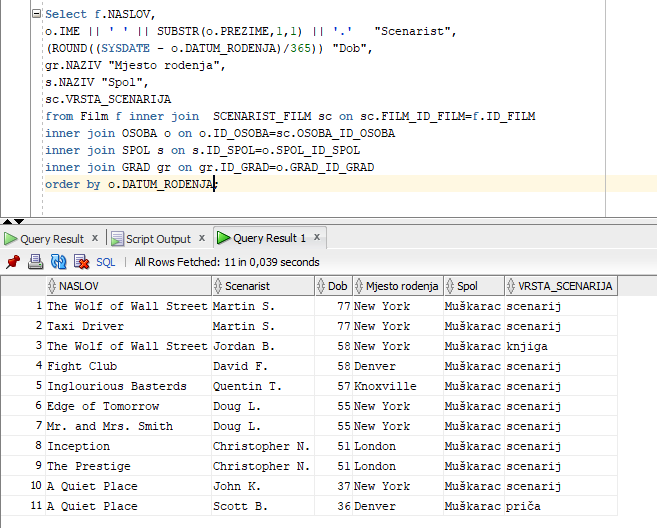
Kada želimo rezultate poredati po nekom kriteriju koristimo naredbu ORDER BY. Moguće je podatke poredati uzlazno (ascending) ili silazno (descening), po defaultu je uzlazno pa je dovoljno napisati samo „order by“.



Slika 16 Order by - silazno

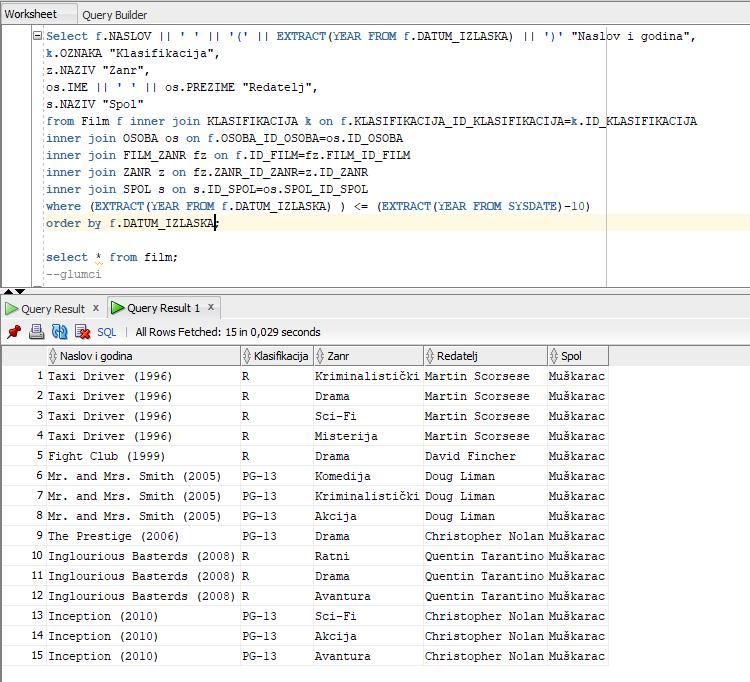
# ZADACI

1.Prikaži naslov filmova i scenariste s imenom i inicijalom prezimena, dob, mjesto rođenja, spol i vrstu scenarija. Neka budu organizirani od najstarijega prema najmlađem.



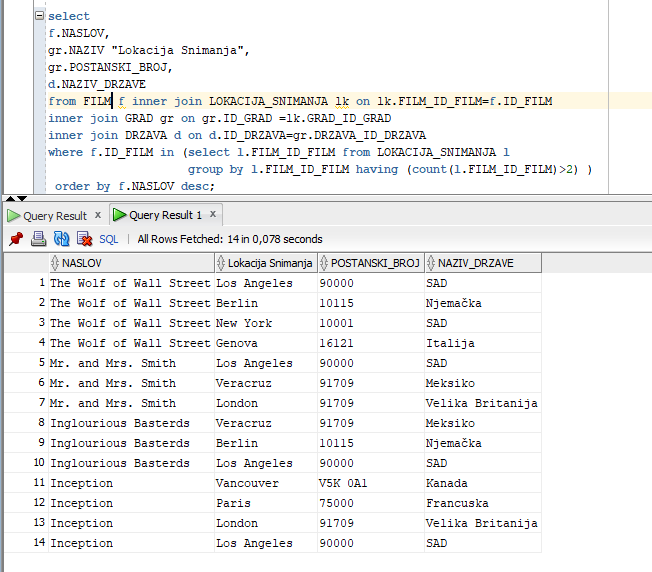
Slika 17 Zadatak 1

2.Ispiši naslov i godinu izlaska, klasifikaciju, sve žanrove tog filma, redatelja i njegov spol, za sve filmove koji su stariji od 10 godina, poredano od najstarijih prema najnovijim.



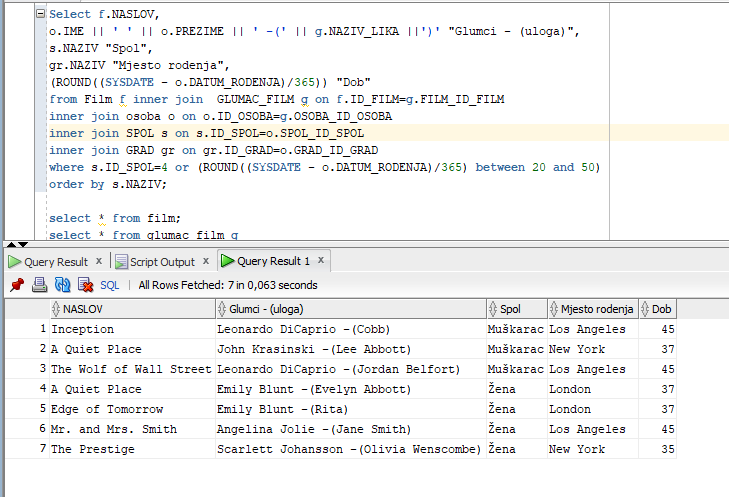
Slika 18 Zadatak 2

3. Prikaži naslove filmova te njihove lokacije snimanja skupa s državama, za sve filmove koji su snimani na više od 2 lokacije, poredano silazno prema naslovu filma.



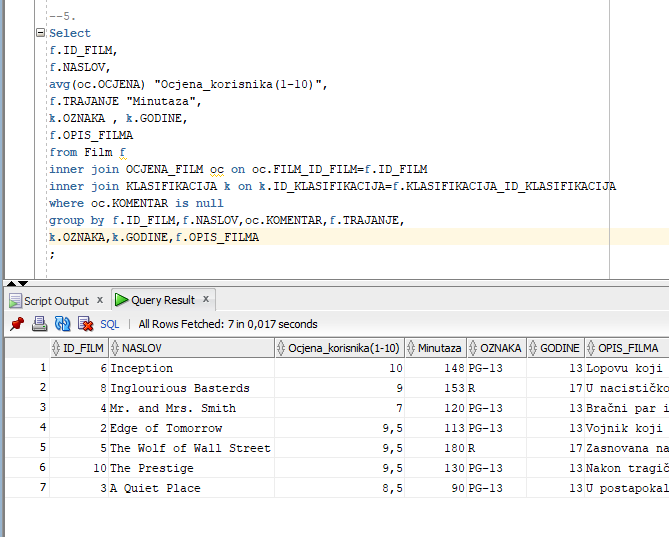
Slika 19 Zadatak 3

4.Prikaži sve glumce i njihove uloge u filmovima, te mjesto rođenja i dob. Glumci moraju biti stariji od 20 i mlađi od 50 godina ili su ženskog spola. Poredano po spolu.



Slika 20 Zadatak 4

5. Prikaži filmove te prosječne ocjene od korisnika koji nisu dali nikakav komentar, duljinu filma i klasifikaciju po godinama.

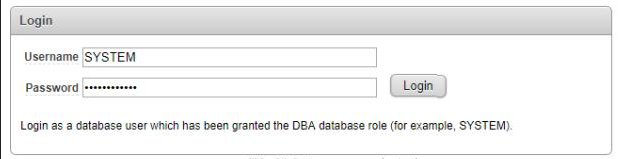


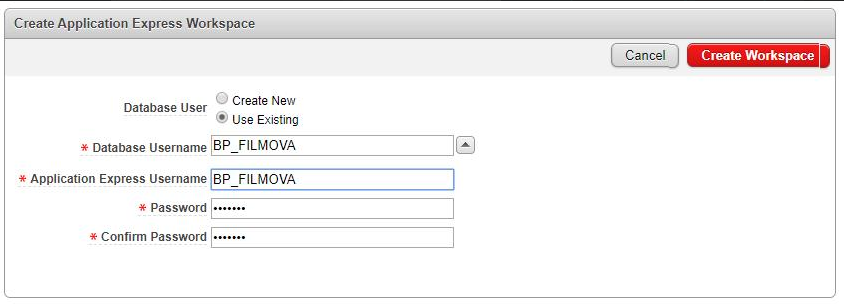
Slika 21 Zadatak 5

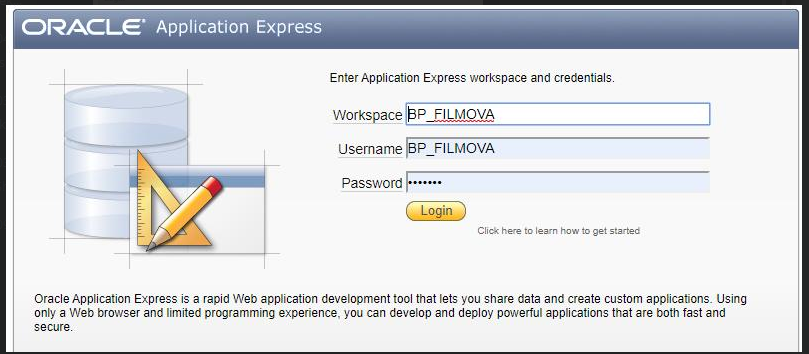
# APEX

Osim izvršavanja upita u SQL Developeru, dohvaćanje podataka iz baze, unošenje i prepravljanje istih, može se odrađivati i putem neke aplikacije. APEX se može koristiti za izradu složenih web aplikacija koje se mogu koristiti u većini modernih web preglednika.

Za ovaj projekt iz baza podataka, koristeći Apex napravit ćemo aplikaciju za prikazivanje podataka o filmovima, žanrovima, ljudima koji su radili na filmu te korisnicima te zamišljene baze. Najprije je potrebno prijaviti se na system korisnika, te onda napraviti novog korisnika za našu shemu.



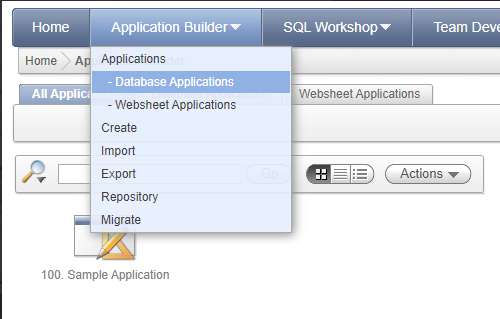


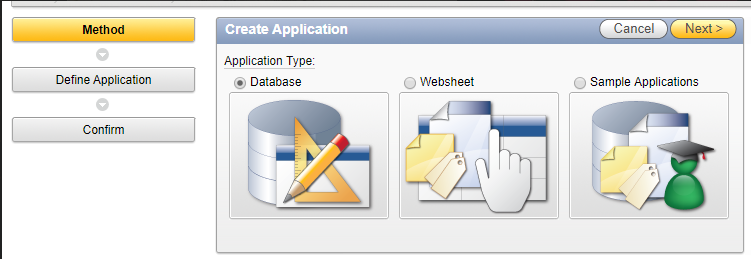


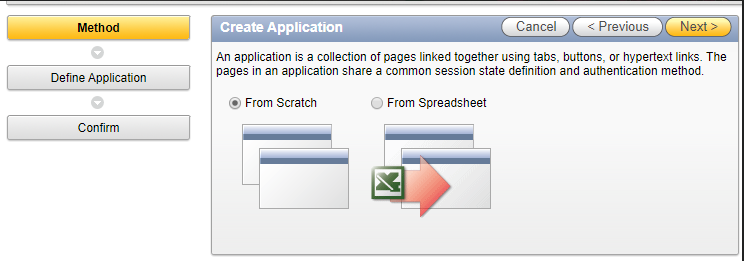
Slika 22,23 i 24 izrada i prijava korisnika

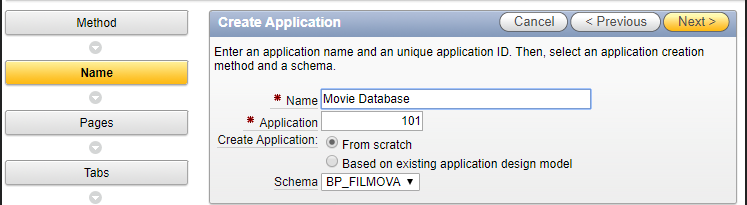
## Application wizard

Application wizard koristimo kako bi nizom jednostavnih koraka kreirali aplikaciju.



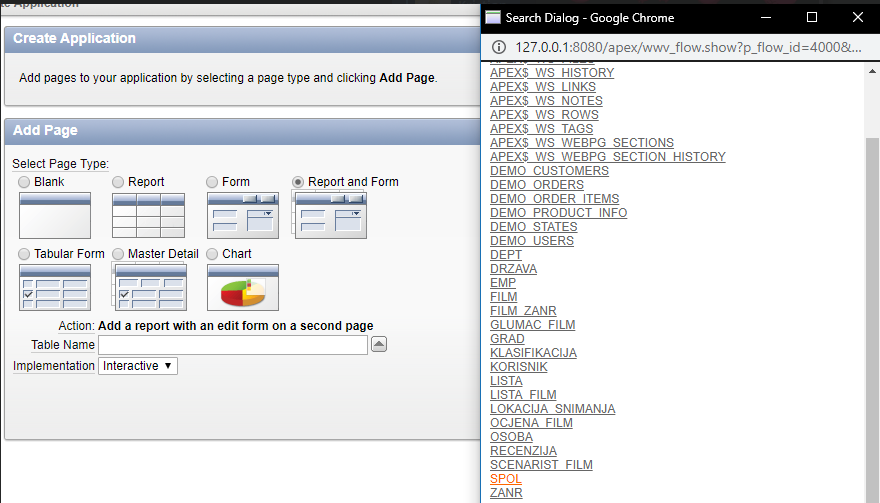


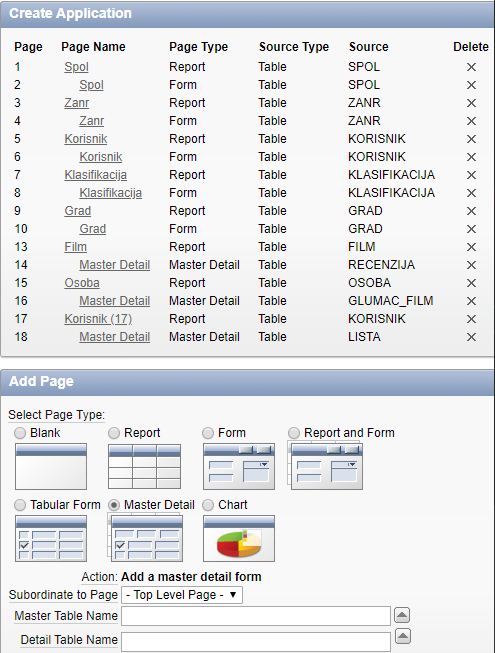




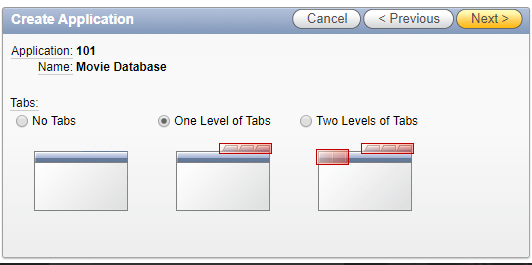
Slika 28 Odabir naziva aplikacije

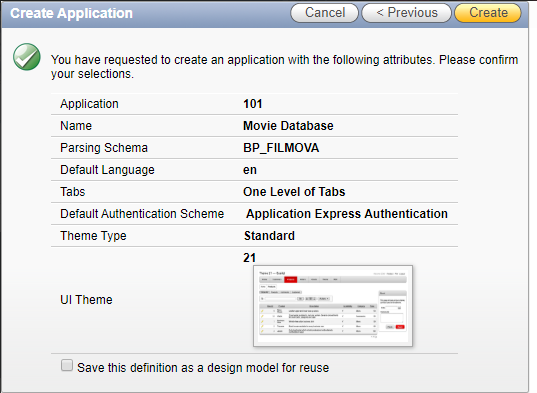
Nakon što smo izabrali ime aplikacije, dodajemo stranice tipa Report And Form i Master Detail koje će služiti za prikaz i uređivanje podataka naše baze. Obzirom da trenutno želimo pokazati kako aplikacija funkcionira, nećemo postaviti čitavu bazu neko samo neke njene dijelove. Za „Report and Form“ stranice odabrat ćemo entitete: Spol, Zanr, Korisnik, Klasifikacija, Grad, a za „Master Detail“: Film i Osoba.





Slika 29 i 30 Report and Form i Master detail

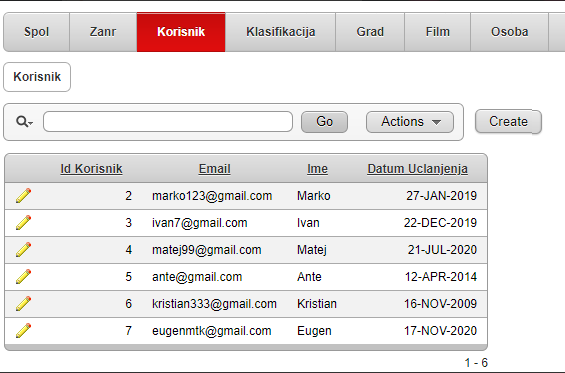


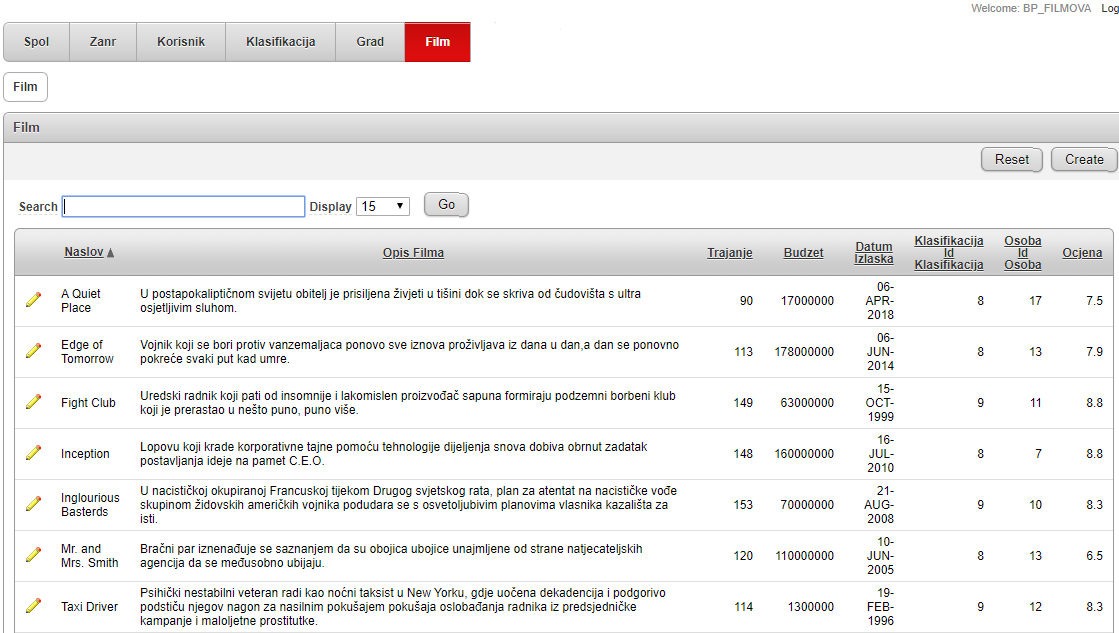


Slika 32 Zavrseno kreiranje aplikacije

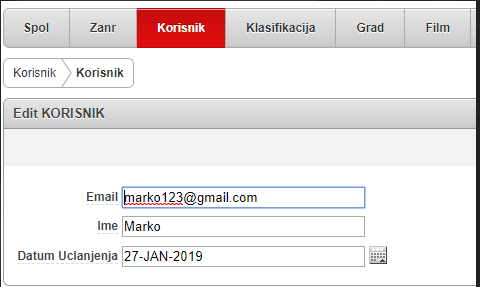


Slika 33 Primjer Report and Form

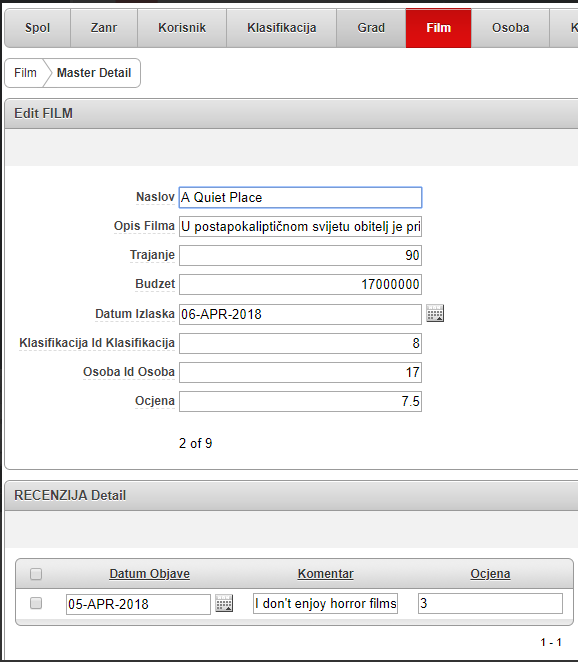




Slika 35 Primjer Master Detail



Slika 36 Primjer editiranja Report and Form stranice



Slika 37 Primjer editiranja Master Detail forme