**PRIRODOSLOVNO-MATEMATIČKI FAKULTET**

**SVEUČILIŠTE U SPLITU**

SEMINARSKI RAD

**INFORMATIČKI PROJEKT IZ BAZA PODATAKA**

**NETFLIX**

Profesorica: Studentica:

Monika Mladenović Matea Vučemilović-Vranjić

Split, 2023.

Sadržaj

[**1.** **UVOD** 1](#_Toc132142496)

[**2.** **MODEL PODATAKA** 2](#_Toc132142497)

[**2.1.** **ENTITETI** 2](#_Toc132142498)

[**2.2.** **RELACIJE** 4](#_Toc132142499)

[**2.3.** **KONCEPTUALNI MODEL** 6](#_Toc132142500)

[**2.4.** **LOGIČKI MODEL** 7](#_Toc132142501)

[**2.5.** **RELACIJSKI MODEL** 8](#_Toc132142502)

[**2.6.** **DDL** 9](#_Toc132142503)

[**3.** **BAZA PODATAKA** 12](#_Toc132142504)

[**3.1.** **IZRADA KORISNIKA I DODJELA PRAVA** 12](#_Toc132142505)

[**3.2.** **IZRADA VEZE NA NOVOG KORISNIKA** 13](#_Toc132142506)

[**3.3.** **IZVRŠAVANJE DDL-A** 14](#_Toc132142507)

[**3.4.** **OSNOVNI UPITI** 15](#_Toc132142508)

[**3.4.1.** **INSERT** 15](#_Toc132142509)

[**3.4.2.** **UPDATE** 17](#_Toc132142510)

[**3.4.3.** **DELETE** 18](#_Toc132142511)

[**3.4.4.** **SELECT** 19](#_Toc132142512)

[**3.4.5.** **ORDER BY** 20](#_Toc132142513)

[**4.** **ZADACI** 21](#_Toc132142514)

# **UVOD**

Baza podataka je način organiziranja i pretraživanja uzajamno povezanih podataka. Služi nam za bolju dostupnost i razvrstavanje podataka. Ona može sadržavati podatke u raznim oblicima, od jednostavnog tekstu do kompleksnih struktura. Najčešće pohranjuju podatke o osobama, proizvodima, narudžbama… Računalna baza podataka jest spremnik objekata. Kroz ovaj projekt kreirat ću i baviti se bazom podataka za aplikaciju za gledanje filmova i serija (kao što je npr. Netflix). Ona nam pruža uvid u sve filmove i serije koje su dostupne za gledanje kao i njihove glumce, redatelje, godinu nastanka i slične informacije o kojima ću više pričati u nastavku. Također imamo saznanja o ljudima koji koriste aplikaciju i njihovim profilima i članarinama te vidimo koje filmove i serije oni gledaju, koji plan programa koriste i slično. Za kreiranje baze podataka koristimo tablice (tj. njihove entitete i atribute te veze kojima ih spajamo tj. koje predstavljaju odnose među njima.

# **MODEL PODATAKA**

## **ENTITETI**

Entitet je nešto o čemu želimo spremati podatke i nešto što je u stanju postojati ili ne postojati, a može se identificirati. Entitet može biti stvar ili biće (npr. kuća, fakultet, student, roba, ispit...). Također može biti i neki događaj ili pojava (npr. polaganje ispita, nogometna utakmica...). Svaki entitet opisan je atributima. U vidu teme svojega projekta koristila sam sljedeće entitete:

1. Film - Sadrži podatke o filmu koji je dostupan za gledanje. Imamo atribute Naziv, Godina, Trajanje i Opis.
2. Serija - Analogno atributu Film. Dodan je još i atribut Broj\_Sezona.
3. Žanr - Navodi sve žanrove koje filmovi i serije mogu imati.
4. Serija\_ Žanr
5. Film\_ Žanr
6. Sezona - Podaci o sezonama vezanima za seriju. Navodimo sljedeće atribute: Naziv, Godina, Opis i Broj\_Epizoda.
7. Epizoda - Informacije o pojedinoj epizodi svake sezone. Atributi su Naziv, Trajanje i Opis.
8. Glumac - Ime i Prezime glumca/glumice. S time da je prezime neobavezan podatak ako neka osoba ima npr. umjetničko 1 ime).
9. Glumac\_Film
10. Glumac\_Epizoda
11. Redatelj – Analogno kao za glumca.
12. Država - koristi samo atribut naziv i govori u kojoj državi je nastao film ili serija te iz koje države je korisnik.
13. Korisnik - Osoba koja kreira svoj profil za gledanje personalizirane liste filmova i serija. za atribut imamo njezino ime.
14. Film\_Korisnik
15. Epizoda\_ Korisnik
16. Prijevod - Prijevod koji korisnik može koristiti i ne mora prilikom gledanja filma ili serije.
17. Članarina - Račun kojega jedan od korisnika stvara, a može ga koristiti više osoba. Potrebni su nam e-mail i broj telefona kao atributi.
18. Plan - Predstavlja plan članarine (npr. Premium, Regular...). Za atribute ima Naziv i Cijenu samoga plana.
19. Račun - Račun kojega dobiva korisnik svaki mjesec. Kao atributi potrebni su nam Broj\_ Računa i Datum.
20. Kartica - Predstavlja podatke o kartici kojom korisnik plaća svoju mjesečnu članarinu. Zato za atribute imamo Broj\_Kartice, CCV i Datum\_ Isteka kartice.

## **RELACIJE**

Relacije predstavljaju odnose između dvije tablice tj. način na koji su entiteti međusobno povezani. Imamo tri vrste relacija:

1. **Jedan na jedan** (1:1) – Svaka n-torka jednog entiteta može biti povezana samo s jednom n-torkom drugog entiteta. Ovu vrstu relacije nisam koristila u ovome projektu.
2. **Jedan na više** (1:n) – Govori da jedan element 1. entiteta može biti povezan sa više elemenata iz 2. entiteta. Međutim, neki element iz 2. entiteta može biti povezan samo sa jednim elementom iz 1.entiteta. ovo je najčešći tip relacije. Relacije jedan na više koje sam koristila u okviru svojeg modela:

* Jedna serija može imati više sezona. Međutim, jedna sezona može biti samo u jednoj seriji.
* Analogno jedna sezona može imati više epizoda, a jedna epizoda može biti samo u jednoj sezoni.
* Iz jedne države može biti više serija i filmova, a jedna serija ili film mogu biti samo iz jedne države.
* Jedan redatelj može režirati više serija i filmova, a serija ili film imaju samo jednoga redatelja(tako je navedenu u aplikaciji po kojoj radim ovaj projekt).
* Jedna članarina može sadržavati više korisnika, ali jedan korisnik može biti u jednoj članarini.
* Jedna kartica može plaćati više članarina. Međutim jedna članarina se plaća jednom karticom.
* Svaka članarina ima jedan plan koji koristi, ali se jedan plan može ponavljati u više članarina.
* Na jednom računu nalazi se jedan plan, ali se taj plan može ponavljati na više računa.
* Jedan prijevod može se nalaziti u više gledanja filmova i serija, ali u jednom gledanju možemo postaviti samo jedan prijevod. Također možemo nešto gledati i bez prijevoda pa zato ta relacija ima svojstvo „Source optional“.

1. **Više na više** (n:m) – Kada jedan entitet može imati više elemenata iz drugog entiteta i obrnuto. Ovakva vrsta relacije nije baš poželjna u bazama podataka pa to rješavamo uvođenjem novog entiteta koji se sastoji od kompozitnog primarnog ključa i dodatnih atributa ako je potrebno. Relacije više na više koje sam koristila u svojoj bazi i novi entiteti koje sam dodala zbog toga:

* Jedna serija ili film mogu spadati u više žanrova i jedan žanr može imati više serija i filmova te su zbog toga dodani entiteti Zanr\_Serija i Zanr\_Film.
* Jedan glumac može glumiti u više serija i filmova tj. jedna serija ili film mogu imati više glumaca pa sam dodala entitete Glumac\_Epizoda i Glumac\_Film.
* Jedan korisnik može gledati više serija i filmova i jedan film ili seriju mogu gledati više osoba pa su zato tu entiteti Film\_Korisnik i Epizoda\_Korisnik.

## **KONCEPTUALNI MODEL**

Predstavlja moju ideju modela na papiru. To je prvi korak pri izradi baze podataka.

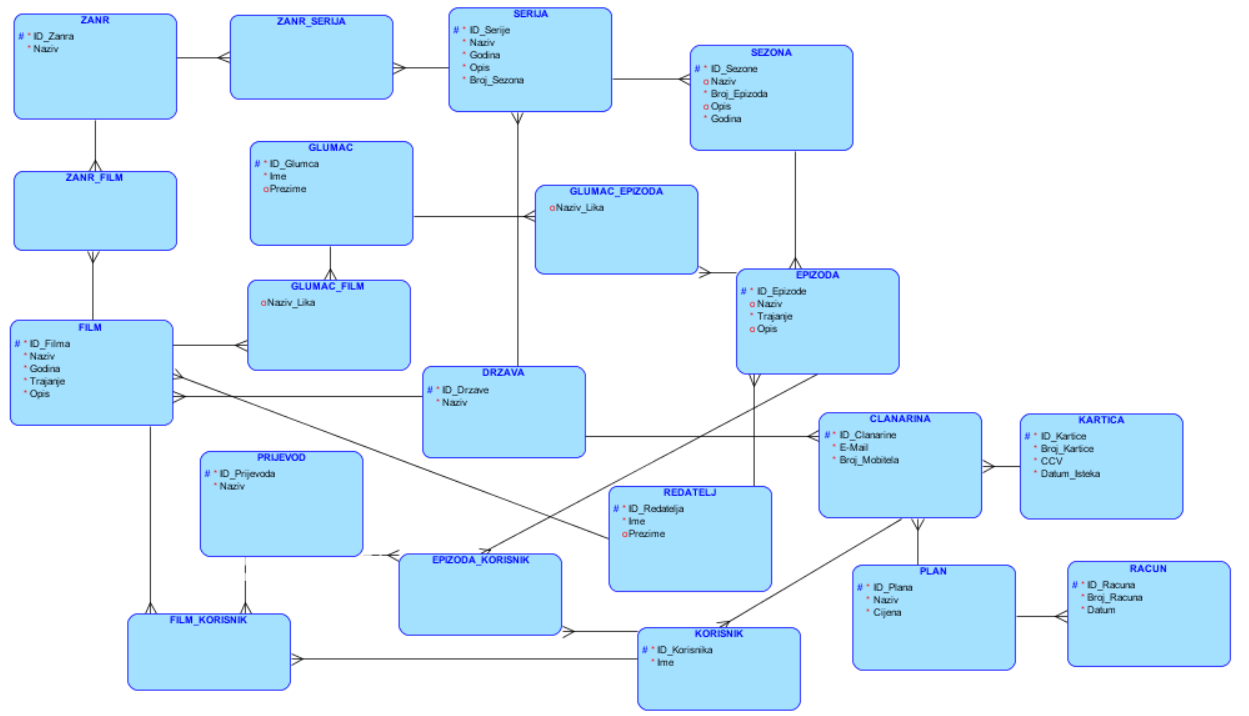
Timeline

Description automatically generated

Slika 1. Konceptualni model

## **LOGIČKI MODEL**

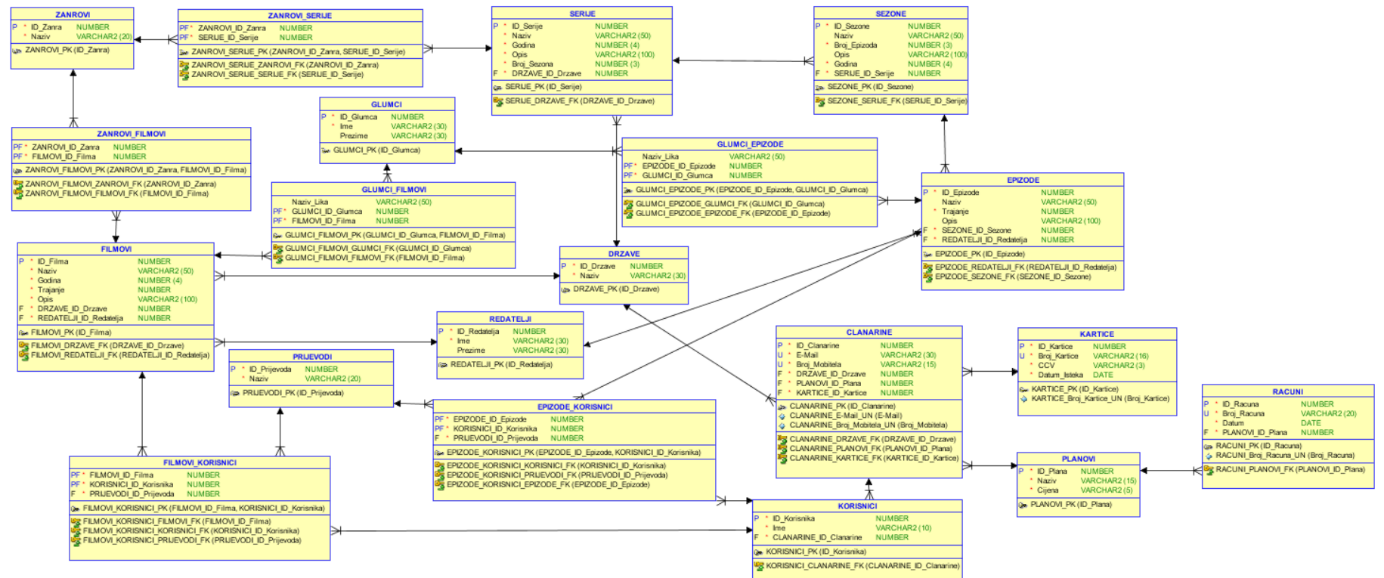
Entiteti baze podataka koji su povezani određenim relacijama predstavljeni su logičkim modelom. Vidljivi su nam atributi, primarni ključevi i relacije među entitetima.



Slika 2. Logički model

## **RELACIJSKI MODEL**

Nakon izrađenog logičkog modela automatski možemo izraditi i relacijski model. On se sastoji od tablica i relacija među njima. Također možemo vidjeti i tipove podataka s ograničenjima i strane ključeve.



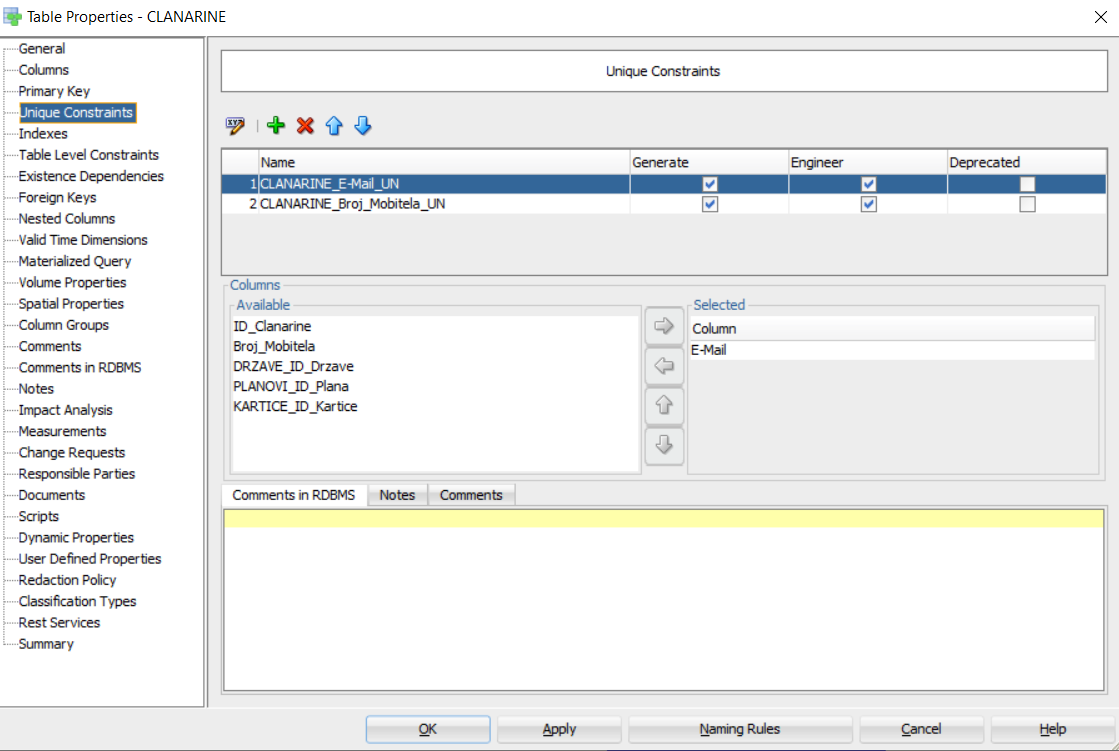
Slika 3. Relacijski model

## **DDL**

Sljedeće što želimo je zapravo prenijeti ovaj model u SQL Developer kako bismo mogli ispuniti tablice podacima i raditi upite. Kako bismo to mogli, izrađujemo DDL datoteku.

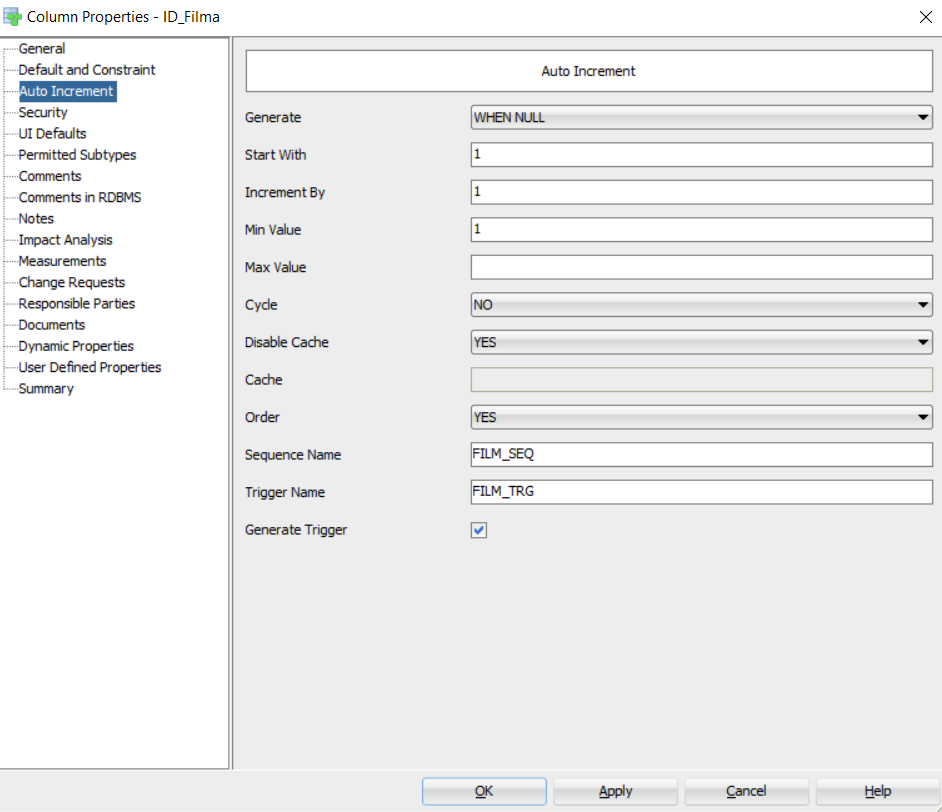
Prije same izrade DDL-a radimo još neke izmjene u relacijskom modelu odnosno još u fazi Data Modeler-a. Kako bismo za neki atribut osigurali da se ne može unijeti bilo koja vrijednost, postavljamo mu donju i/ili gornju granicu, odnosno izrađujemo ograničavanje unosa(**constraint**). Ovakvu vrstu ograničenja nisam koristila u svome projektu.

Međutim, kada želimo osigurati jedinstvenost nekog atributa koji nije primarni ključ koristimo **Unique Constraint** što možemo vidjeti na slici na primjeru e-maila i broja mobitela koji moraju biti jedinstveni za svaku članarinu. Također i atributi Broj\_Kartice i Broj\_Racuna imaju to svojstvo.



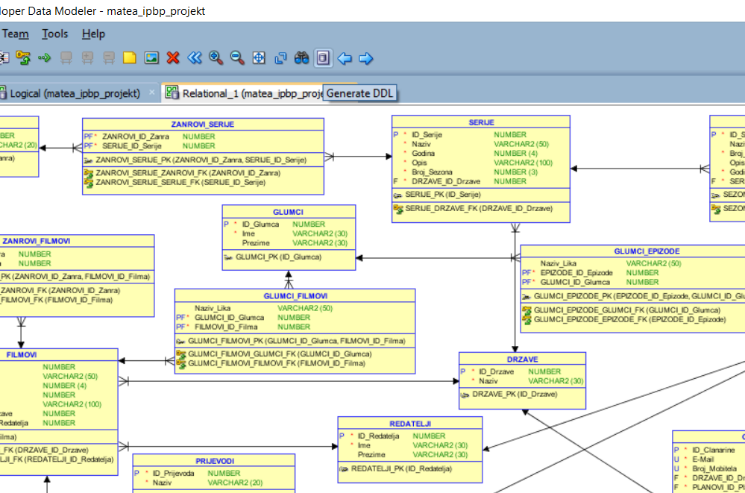
Slika 4. Unique Constraint

Zadnje što ćemo napraviti prije izrade DDL-a su sekvenca(**sequence**) i okidač(**trigger**) za svaki od primarnih ključeva budući da za Oracle bazu ne postoji ugrađeni tip podataka za primarni ključ. Sekvenca omogućava povećanje ID-a za svaki sljedeći unos, a okidač će nam omogućiti obavljanje akcija koje su povezane sa specifičnom akcijom koju smo pokrenuli. Upisujemo i minimalnu vrijednost od koje započinjemo.

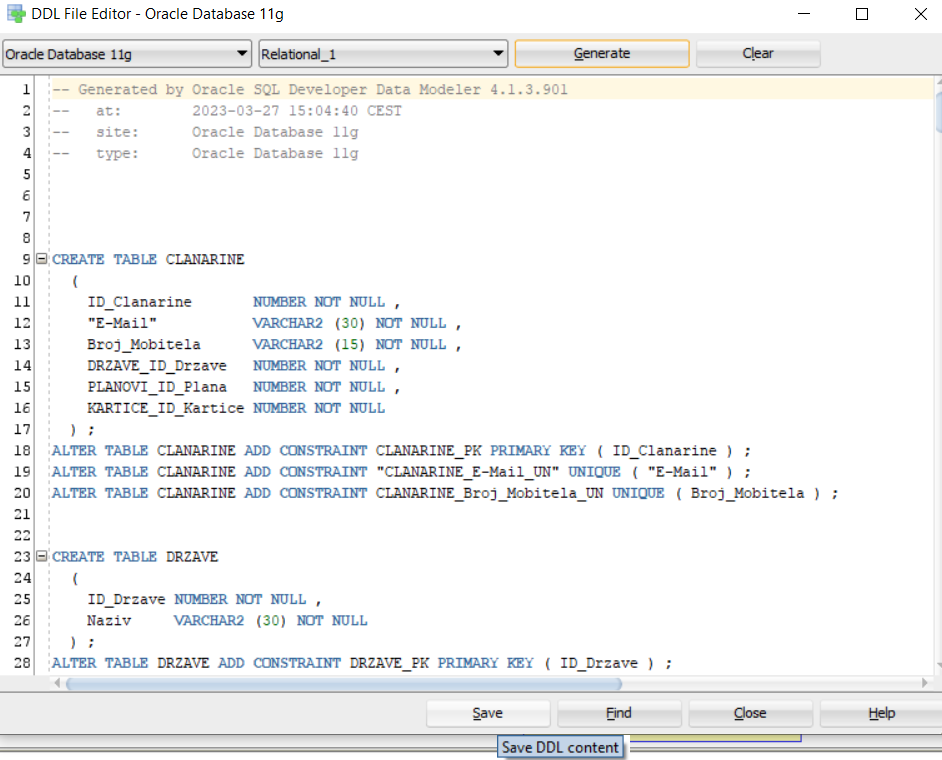


Slika 5. Sekvenca i okidač

Za kraj, DDL datoteku (datoteku koja sadrži naredbe koje se koriste za opisivanje strukture baze podataka) kreiramo klikom na ikonu **Generate DDL** u Data Modeleru kao što vidimo na slici.



Slika 6. Generiranje DDL-a

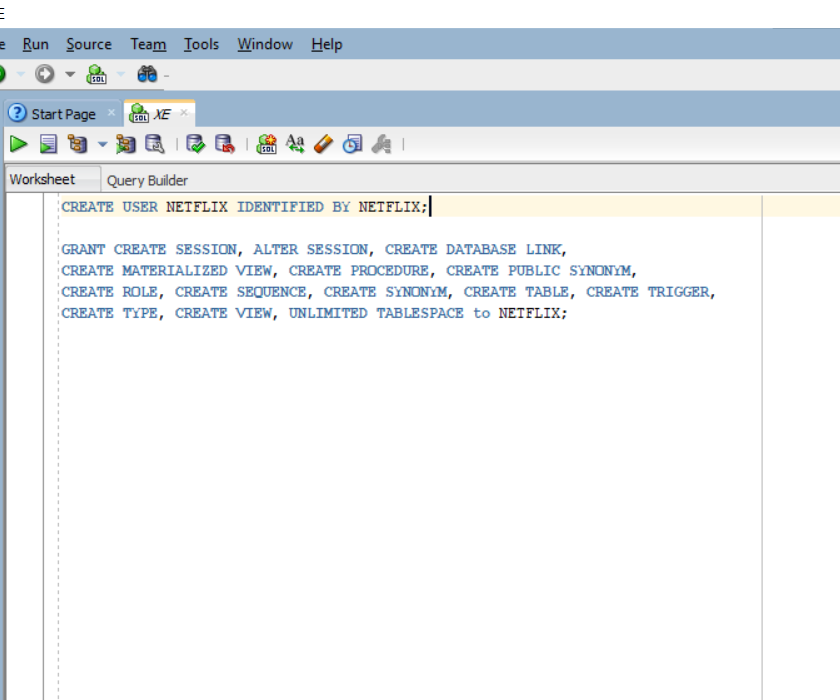


Slika 7. Dio sadržaja DDL-a

# **BAZA PODATAKA**

## **IZRADA KORISNIKA I DODJELA PRAVA**

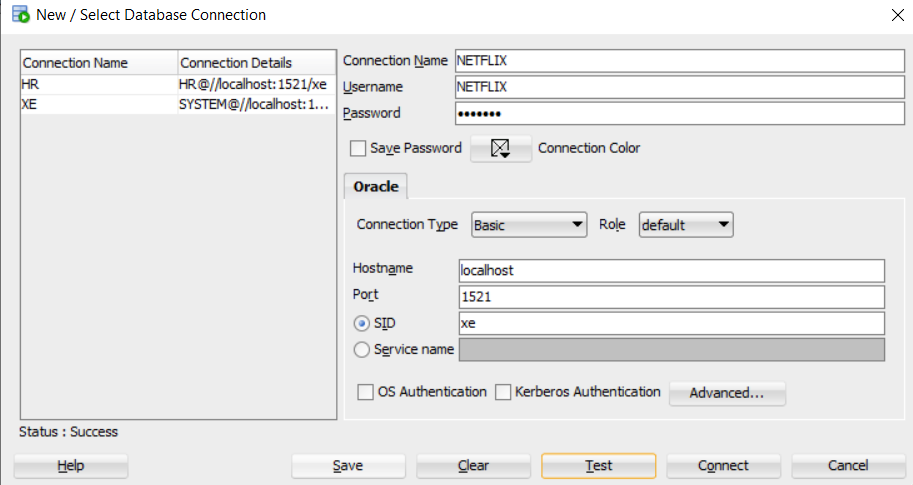
Nakon što smo završili sa Data Modelerom prebacujemo se u SQL Developer gdje ćemo se prijaviti preko korisnika SYSTEM preko kojega ćemo napraviti novog korisnika naredbom CREATE USER. Tek izrađeni korisnik nema nikakva prava, a možemo mu ih dodijeliti naredbom GRANT.



Slika 8. Kreiranje korisnika i dodjela prava

## **IZRADA VEZE NA NOVOG KORISNIKA**

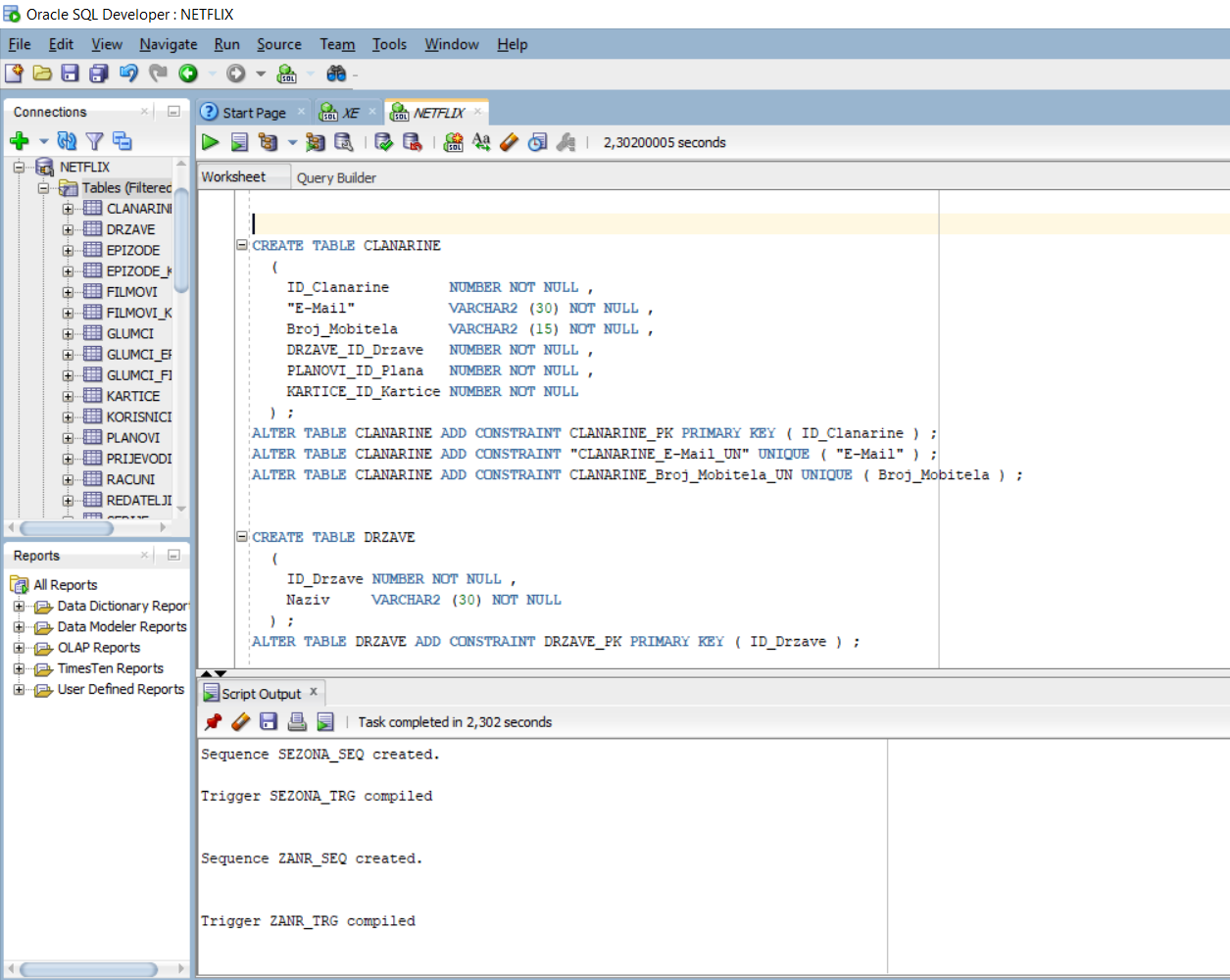
Zatim, klikom na New Connection ikonu dodajemo vezu na novog korisnika i upisujemo potrebne podatke.



Slika 9. Izrada veze na novog korisnika

## **IZVRŠAVANJE DDL-A**

U toj novoj konekciji izvršit ćemo našu DDL datoteku. Odnosno kopirati njezin sadržaj, zalijepiti ga i pokrenuti program.

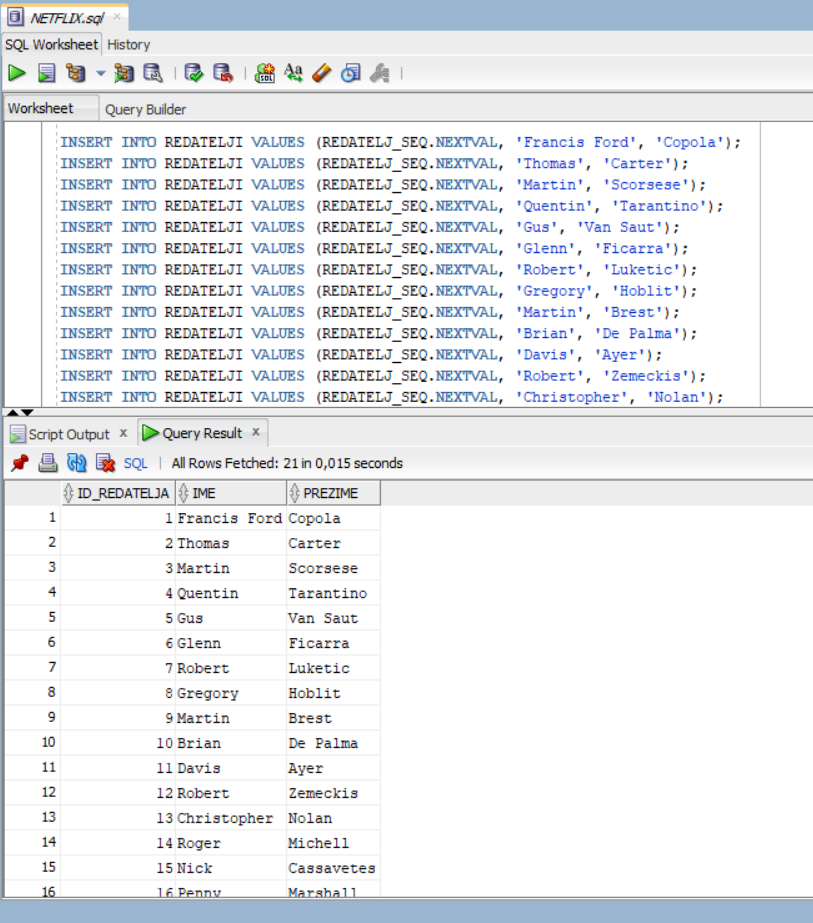


Slika 10. Izgled nakon uspješnog izvršavanja DDL-a

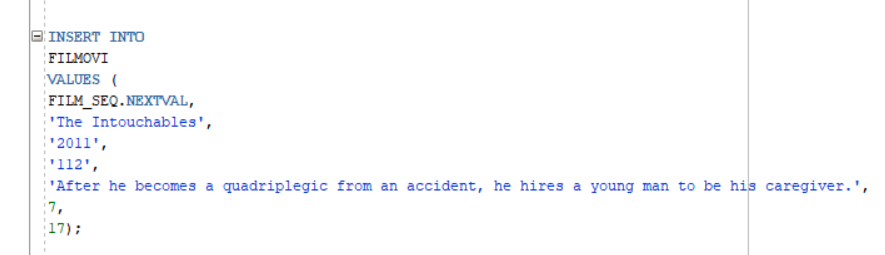
## **OSNOVNI UPITI**

### **INSERT**

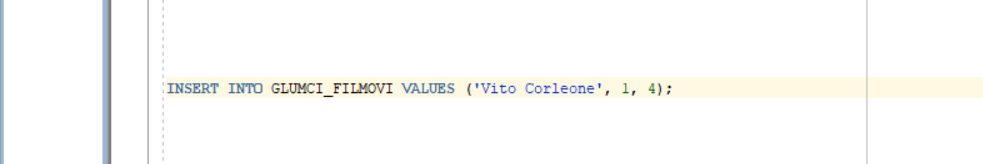
Insert je izraz koji koristimo kako bismo upisali podatke u tablicu. Važno je popunjavati po redu tj. počinjemo od „vanjskih“ tablica. U slučaju povezivanja preko stranog ključa ne koristimo navodnike u upisu.



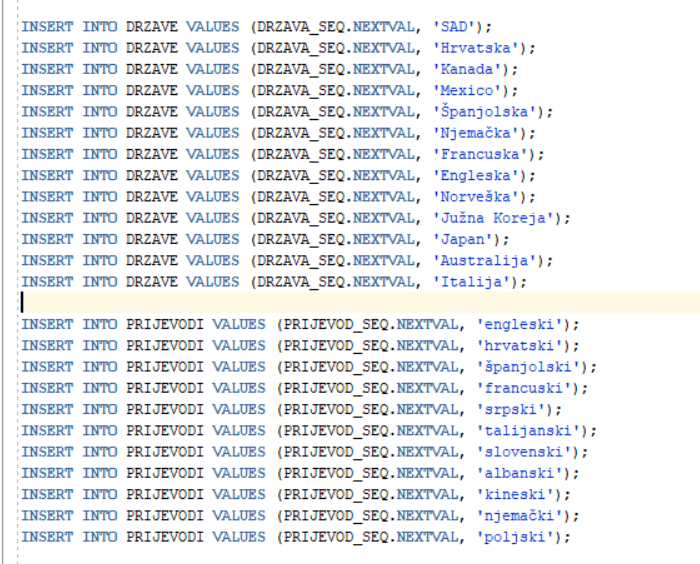
Slika 11. Primjer INSERT upita



Slika 12. Primjer INSERT upita preko stranih ključeva



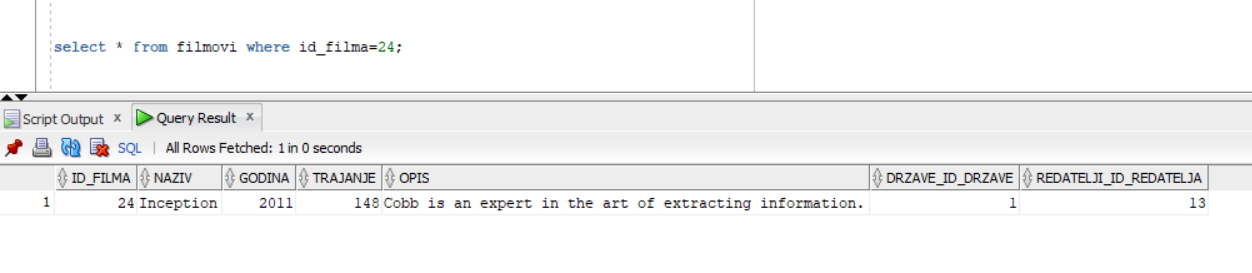
Slika . Primjer INSERT upita preko stranih ključeva



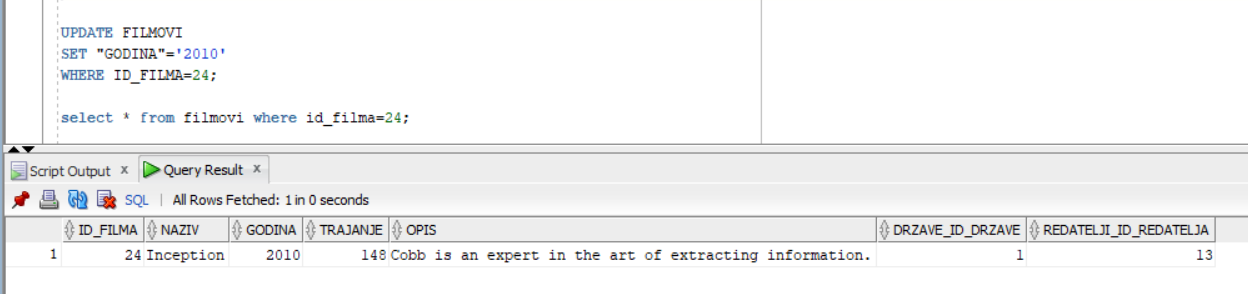
Slika 14. Dodavanje država i prijevoda

### **UPDATE**

Ovaj upit koristimo kada želimo ažurirati podatke u bazi podataka. Na slikama vidimo primjer krivo unesene godine za film, te način ispravka.



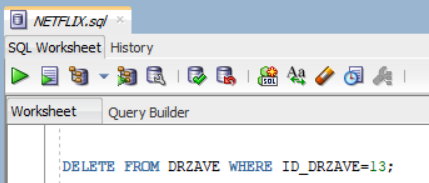
Slika 15. Podaci prije UPDATE upita



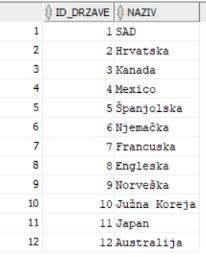
Slika 16. Primjer nakon UPDATE upita

### **DELETE**

Delete (izbriši) upit koristimo kada želimo izbrisati neki podatak iz baze podataka. U primjeru vidimo dva puta unesenu istu državu pa ćemo jednu izbrisati.



Slika 17. Primjer DELETE upita

Slika 18. Podaci prije DELETE upita Slika 19. Podaci nakon DELETE upita

### **SELECT**

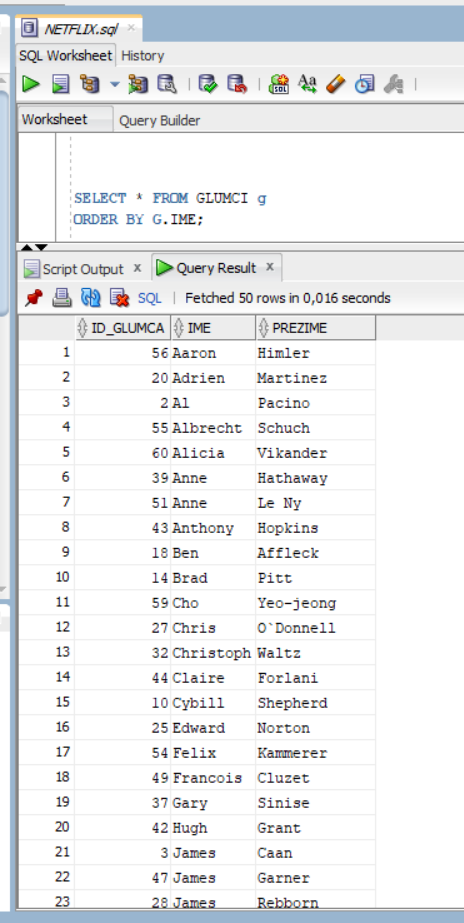
Select je upit u SQL jeziku koji se koristi za povrat odabranog skupa podataka iz jedne ili više tablica.



Slika 20. Primjer jednostavnog SELECT upita

### **ORDER BY**

Kada želimo rezultate poredati po nekom kriteriju koristimo naredbu order by. Pri sortiranju možemo birati želimo li sortirati uzlazno ili silazno.

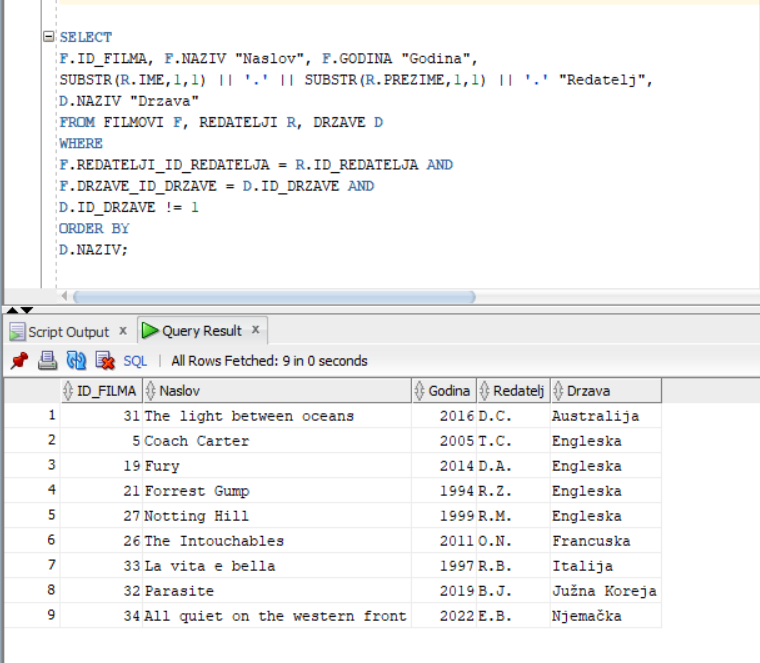


Slika 21. Primjer upita ORDER BY

# **ZADACI**

1.ZADATAK

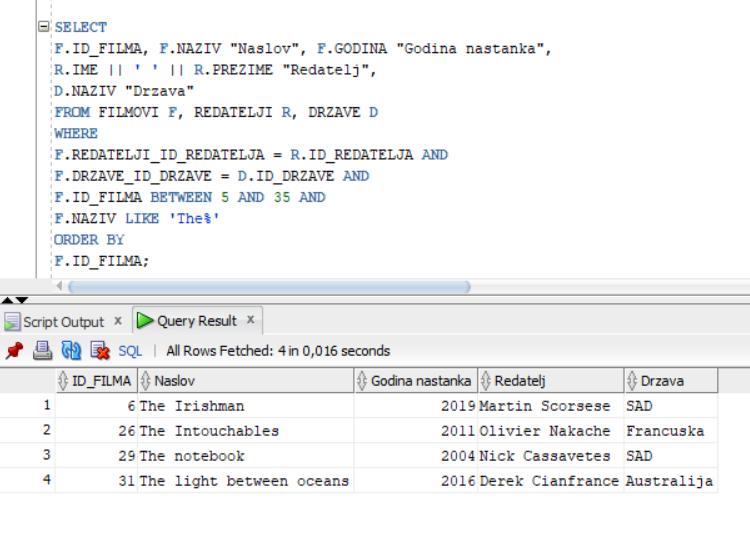
Prikaži sve filmove koji nisu iz SAD-a i inicijale njihovih redatelja u jednom polju. Ispiši iz koje su države i godinu nastanka te poredaj po nazivu države.



Slika . 1.zadatak

2.ZADATAK

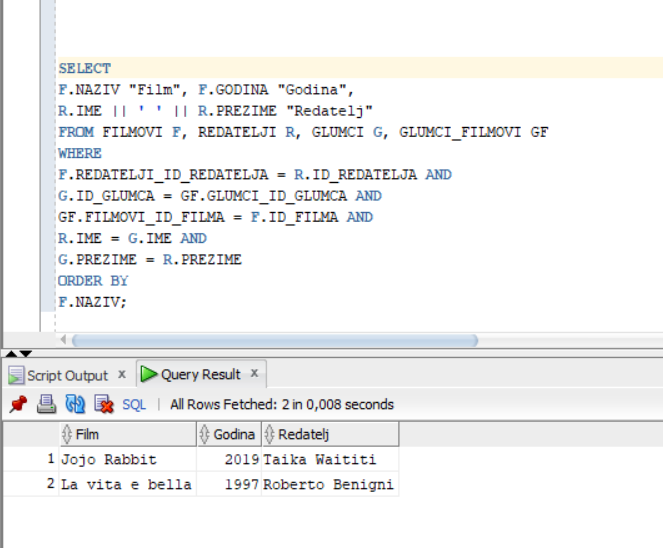
Prikaži sve filmove (ID\_Filma, naziv, godinu nastanka, redatelja u jednom polju i državu) kojima je ID\_Filma između 5 i 30, a naziv filma počinje s riječi „The“ te ih poredaj po ID-u filma.



Slika . 2.zadatak

3.ZADATAK

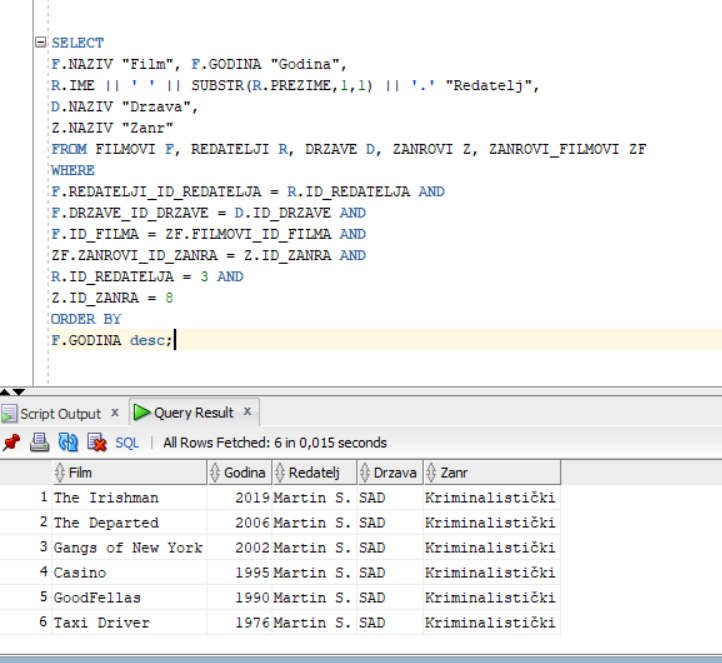
Prikaži sve filmove (naziv, godinu, redatelja) u kojima redatelj također glumi u filmu. Ime i prezime redatelja napišite u jednom polju. Poredajte ih po nazivu filma.



Slika . 3.zadatak

4.ZADATAK

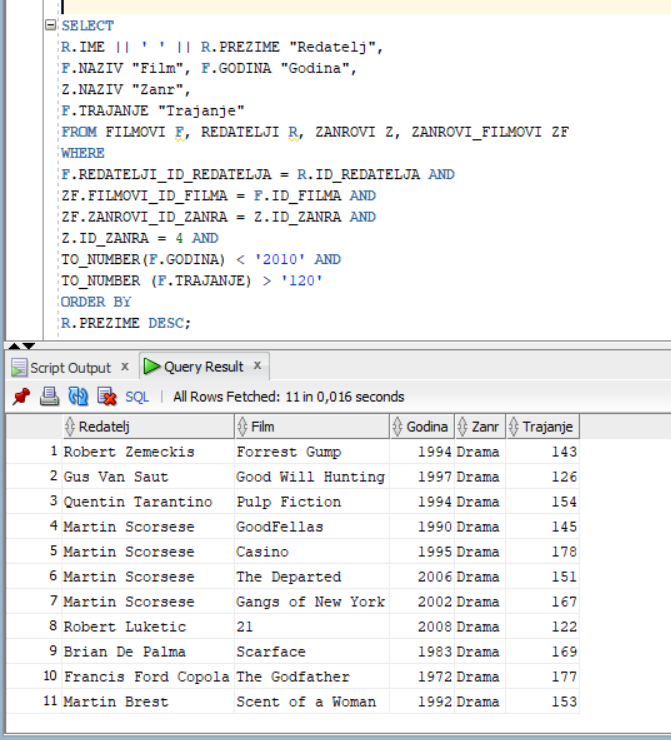
Prikaži sve filmove (naziv, godinu, redatelja, i žanr) žanra kriminalistički film kojima je redatelj Martin Scorsese te ih poredaj silazno po godini izlaska filma. Redatelju napišite ime i inicijal prezimena u jednom polju.



Slika . 4.zadatak

5.ZADATAK

Prikaži imena i prezimena redatelja (u jednom polju) koji su režirali drame prije 2010. godine koje traju više od dva sata i ispiši nazive tih filmova te godinu nastanka, žanr i trajanje i poredaj ih po prezimenu redatelja silazno.



Slika . 5.zadatak