|  |
| --- |
| Универзитет у крагујевцу  Факултет техничких наука |
| База података за праћење рада биоскопа „CINNEPLEXX BIG“ |
| Пројектни рад из предмета Програмирање база података |
|  |
|  |
|  |

Ментор: Студент:

Катарина Митровић, асистент *Петар Стефановић, 247/2016*

*Недељко Чкоњовић, 247/2016*

Чачак, 2019. године

**САДРЖАЈ**

[**1.** **Увод** 2](#_Toc20229242)

[**2.** **Анализа процеса и информационих токова** 2](#_Toc20229243)

[**3.** **Анализа релевантних носилаца докумената** 4](#_Toc20229244)

[**4.** **Модели објекта – везе** 7](#_Toc20229245)

[**5.** **Денормализација** 9](#_Toc20229246)

[**5.1.** **Денормализација 1** 9](#_Toc20229247)

[**5.1.1.** **Тригер 1** 11](#_Toc20229248)

[**5.1.2.** **Тригер 2** 11](#_Toc20229249)

[**5.1.3.** **Тригер 3** 11](#_Toc20229250)

[5.2. **Денормализација** 2 12](#_Toc20229251)

[**5.2.1.** **Тригер 1** 14](#_Toc20229252)

[**5.2.2.** **Тригер 2** 14](#_Toc20229253)

[**5.2.3.** **Тригер 3** 15](#_Toc20229254)

[**6.** **Оптимизација** 15](#_Toc20229255)

[**6.1.** **Оптимизација изведена на заснованим вредностима** 15](#_Toc20229256)

[**6.2.** **Индекси** 16](#_Toc20229257)

[**6.2.1.** **Индекс 1** 16](#_Toc20229258)

[**6.2.2.** **Индекс 2** 16](#_Toc20229259)

[**6.2.3.** **Индекс 3** 16](#_Toc20229260)

[**7.** **Релациона шема** 17](#_Toc20229261)

[**7.1.** **Кодови за креирање табела** 18](#_Toc20229262)

[**8.** **UML дијаграм случајева коришћења** 22](#_Toc20229263)

[**9.** **Услов за имплементацију** 22](#_Toc20229264)

[**10.** **Подаци за тестирање** 22](#_Toc20229265)

[**11.** **Опис апликације са упутством за коришћење** 36](#_Toc20229266)

[**11.1.** **Креирање апликације** 36](#_Toc20229267)

[**12.** **Закључак** 40](#_Toc20229268)

[**13.** **Литература** 41](#_Toc20229269)

1. **Увод**

Задатак нашег семинарског рада је представљање једног реалног система и креирање његове базе података, сто је у нашем слусају биоско. Циљ семинарског рада је упознавање са принципима креирања базе података и разликовање доброг и лошег дизајна базе података. Од дизајна базе података у великој мери зависи квалитет саме базе и њена функционалност након израде. Поред ових наведених ствки овај семинарски це нам помоћи да савладамо пројектовање ЕР дијаграма и код за креирање табела у SQL-у. Разлози због којих се креира база података су многобројни. Основни разлог за креирање базе података су сигурнији нацин за цување података, као и могуцност да се подаци лаксе филтрирају и обрађују.

“ CINNEPLEXX” представља ланац биоскопа у разним градовима. У сваком биоскопу постоји једна или више сала, ми ћемо у овом пројектном задатку одрдити базу за “ CINNEPLEXX BIG” који се налази у улици Јурија Гагарина 16-16А, локал 317 Београд . Клијент који жели да одгледа неки филм, може да резервише карту путем интернет или је купити на локацији самог биоскопа. Резервација се врши за одређени број седишта за пројекцију у одређено време и клијент може да резервише једну или више карата. Резервација се може извршити преко интернет или у самом биоскопу. Клијент такође може да откаже своју резервацију. Резервације важе до пола сата пре саме пројекције. Када клијент дође лично у биоскоп може да плати карте које је већ резервисао, а ако није резервисао може да купи карту уколико је остало слободних места. Радница у биоскопу има увид свих пројекција које се тренутно дају у том биоскопу. Биоскоп “ CINNEPLEXX BIG” има и администратора који је задужен за уређивање пројекција, тј. када се и где који филм приказује.

Пословни циљеви и разлози прављења базе података:

Онлајн продаја и резервација карата (повећање доступности наших садржаја).

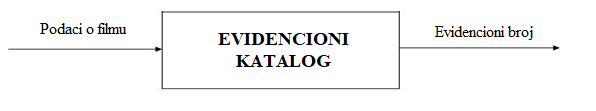
Повецање продаје карата и промовисање садрзаја.

Анализа афинитета наших гледалаца као и њихови омиљени жанрови.

1. **Анализа процеса и информационих токова**

Принцип рада овог биоскопа састоји се из више одређених процедура:

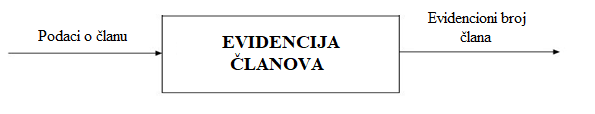
• Евиденције филма



Слика 1 Евиденциони каталог

Обухвата процесс од тренутка када филм стигне у биоскоп. Затим се врши унос податак о филму у постојећу базу. Подаци који се уносе су: назив филма, датум снимања, глумци, режисери (у зависности од броја режисера), жанр, сценаристи и јединствени број филма.

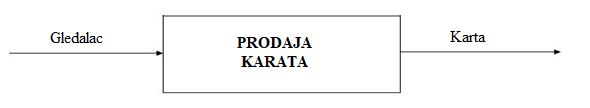
• Евиденција чланова



Слика 2 Евиденција чланова

Евиденција чланова се врши путем веб сајта или у самом биоскопу. Подаци које узимамо од чланова су:име, презиме, датум рођења, пол, омиљени жанрови. Разлози разлози за само евидентирање: Попусти које наши чланови добијају. Свеобухватна слика о жанровима које чланови преферирају, као и о њиховој старосној структури. Могућност обавеставања о премијерама нових филмова путем мејла. Највазни разлог је тај што само чланови нашег биоскопа могу онлајн резервисати карте.

• Продаја карата



Слика 3 Продаја карата

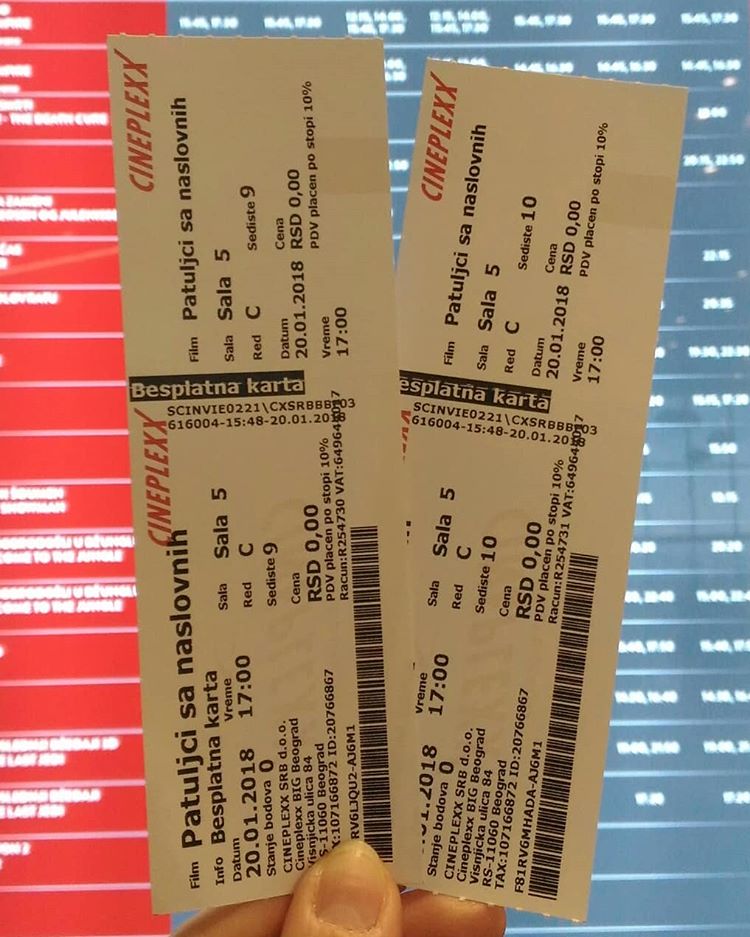
Продаја карата се врши у самом биоскопу, а резервација важи до тридесет минута пре почетка филма. Карту садржи цену, салу у којој се премијера одржава, ред и колону седиста, цену карте.

Анализа процеса за послов

1. **Анализа релевантних носилаца докумената**

• Карта за филм.

Карта за филм се купује у самом биоскопу. Садрзи податке о филму, времену и месту одржавањ, цени карте, салу и ред колону седишта.



Слика 4 Карта

• Лист за регистрацију корисника

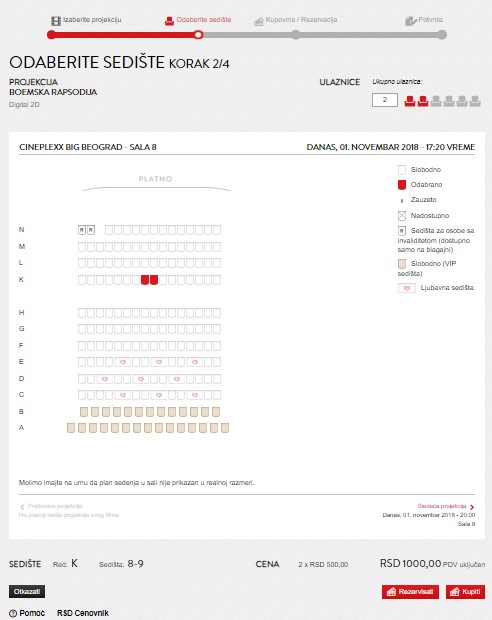
Мозе се попунити онлајн или у библиотеци, садржи пља као што су: надимак, мејл, име, презиме, име, презиме, датум рођења.



Слика 5 Регистрација

Онлајн резервација карата

Обухвата одабир филма и могућност да корисник сам изабере место у биоскопу на ком ће седети.



Слика 6 Примерак онлајн резервације

Примерак карте која се кориснику шаље на мејл приликом резервисања карте и са њом долази у биоскоп И купује праву карту.



Слика 7 Примерак резервисане карте

1. **Модели објекта – везе**

Kupuje

Kupljena

Važi

Ima

Ima

Obavlja

Ima

Ima

Ima

Važi

SALA

#R.br.

\*Broj mesta

\*Naziv

PRIKAZIVANJE

\*Datum i vreme

KARTA

#Br.kart

\*Vreme

\* Cena karte

\*Rezervacija

○Popust

RADNIK

#JMBG

\*Ime

\*Prezime

GLEDALAC

#R.br.

\*Član

oIme

○Prezime

○Pol

○Omiljeni žanr

GLUMAC

#ID

\*Ime

\*Prezime

FILM

#R.br

\*Naziv

\*Žanr

\*Reditelj

Ima

Prodaje

Prodata

ULOGA

#ID

\*Naziv uloge

Kupuje

Kupljena

Važi

Ima

Ima

Obavlja

Ima

Ima

Ima

Važi

SALA

#R.br.

\*Broj mesta

\*Naziv

PRIKAZIVANJE

\*Datum i vreme

KARTA

#Br.kart

\*Vreme

\* Cena karte

\*Rezervacija

○Popust

RADNIK

#JMBG

\*Ime

\*Prezime

GLEDALAC

#R.br.

\*Član

oIme

○Prezime

○Pol

○Omiljeni žanr

GLUMAC

#ID

\*Ime

\*Prezime

FILM

#R.br

\*Naziv

\*Žanr

\*Reditelj

Ima

Prodaje

Prodata

ULOGA

#ID

\*Naziv uloge

REDITELJ

#ID

\*Ime

\*Prezime

Ima

Ima

На дијаграму је извршена нормализација прве нормалне форме (1НФ). Прва нормална форма захтева да нема вишевредносних атрибута. Да би се уверили да је задовољена прва нормална форма, потребно је да се провери да ли атрибут има само једну вредност за сваку инстанцу ентитета.

Ентитет ФИЛМ није у првој нормалној форми, зато што атрибут редитељ може имати више вредности за једну инстанцу ентитета. Зато атрибут редитљ постаје ентитет РЕДИТЉ који је повезан са ентитетом ФИЛМ везом типа 1:М, као што је приказано на моделу.

1. **Денормализација**

Денормализација је поступак побољшања перфоманси базе података додавањем редундантних података, уз нарушавање нормалних форми.

Денормализација се врши над потпуно нормализованим концептуалним моделом.

Тригер се може дефинисати као процедурални код који се аутоматски извршава сваки пут када се деси дефинисани догађај над одређеном табелом или погледом.

* 1. **Денормализација 1**

На слици се може видети да је извршена денормализација у којој атрибут **\**Име***из ентитета **ГЛУМАЦ** прелази и у ентитет **УЛОГА**.

GLUMAC

#ID

\*Ime

\*Prezime

ULOGA

#ID

\*Naziv uloge

\*Ime glumca

GLUMCI

Pk ID

\*Ime

\*Prezime

ULOGE

Pk ID

\*Naziv\_uloge

\*Ime\_glumca

Fk1\* fil\_r\_b

Fk2\*glu\_id

У овој денормализацији коришћена је Pre-joining техника, и нарушена је трећа нормална форма (3НФ).Pre-joining је врста денормализације у којој се некључни атрибут из референциране релације додаје у референцирајућу релациу, преко спољног кључа у референцирајућој релацији или дела примарног кључа референцирајуће релације. Предност ове технике је операција спајања табела се изоставља.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Табела | Тип тригера | Колона | Потребан | Шта треба да се уради? | Редни број тригера |
| GLUMCI | INSERT |  | НЕ |  |  |
| UPDATE | Ime | ДА | Приликом ажурирања колоне **име** у табели **ГЛУМАЦ** покреће се тригер који ће ажурирати колону **име глумца** у табели **УЛОГА.** | 1 |
| DELETE |  | НЕ |  |  |
| ULOGE | INSERT |  | ДА | Ажурирање колоне **име** на основу вредности унете у колону **ИД** у табели **ГЛУМАЦ** | 3 |
| UPDATE | Glu\_id | ДА | Ажурирање колоне **име глумца** на основу измењене вредности **Глу\_ид**. | 3 |
| Ime\_glumca | ДА | Забрањено директо ажурирање ове колоне | 2 |
| DELETE |  | НЕ |  |  |

Табела 1 Табела спецификације тригера за денормализацију 1

ALTER TABLE uloge

ADD ime\_glumca VARCHAR2(50);

UPDATE uloge

SET ime\_glumca = (

SELECT ime

FROM glumci

WHERE uloge.glu\_id = glumci.id);

ALTER TABLE uloge

MODIFY ime\_glumca VARCHAR2(50) NOT NULL;

* + 1. **Тригер 1**

CREATE OR REPLACE TRIGGER glumci\_update

AFTER

UPDATE OF ime ON glumci

FOR EACH ROW

DECLARE

PRAGMA AUTONOMOUS\_TRANSACTION;

v\_new\_ime VARCHAR2(50);

v\_glumac\_id NUMBER(6,0);

BEGIN

v\_new\_ime := :NEW.ime;

v\_glumac\_id := :NEW.id;

EXECUTE IMMEDIATE 'ALTER TRIGGER uloge\_update DISABLE';

EXECUTE IMMEDIATE 'ALTER TRIGGER uloge\_insert\_update DISABLE';

UPDATE uloge

SET ime\_glumca = v\_new\_ime

WHERE glu\_id = v\_glumac\_id;

EXECUTE IMMEDIATE 'ALTER TRIGGER uloge\_update ENABLE';

EXECUTE IMMEDIATE 'ALTER TRIGGER uloge\_insert\_update ENABLE';

END;

* + 1. **Тригер 2**

CREATE OR REPLACE TRIGGER uloge\_update

BEFORE

UPDATE OF ime\_glumca ON uloge

FOR EACH ROW

BEGIN

RAISE\_APPLICATION\_ERROR(

NUM => -20002,

MSG => ’Naziv glumca se ne može direktno menjati’);

END;

* + 1. **Тригер 3**

CREATE OR REPLACE TRIGGER uloge\_insert\_update

BEFORE INSERT OR UPDATE ON uloge

FOR EACH ROW

DECLARE

PRAGMA AUTONOMOUS\_TRANSACTION;

v\_ime VARCHAR2(50);

BEGIN

EXECUTE IMMEDIATE 'ALTER TRIGGER uloge\_update DISABLE';

SELECT ime

INTO v\_ime

FROM glumci

WHERE id= :NEW.glu\_id;

:NEW.ime\_glumca:=v\_ime;

EXECUTE IMMEDIATE 'ALTER TRIGGER uloge\_update ENABLE';

END;

* 1. **Денормализација** 2

На слици се може видети да је извршена денормализација у којој атрибут **\**Назив***из ентитета **ФИЛМ** прелази и у ентитет **ПРИКАЗИВАЊЕ**. У овој денормализацији коришћена је Pre-joining техника, и нарушена је друга нормална форма (2НФ).

PRIKAZIVANJE

\*Datum i vreme

\*Naziv filma

FILM

#R.br

\*Naziv

\*Žanr

\*Reditelj

FILMOVI

Pk R\_br

\*Naziv

\*Zanr

\*Reditelj

PRIKAZIVANJA

Pk, Fk1 fil\_r\_br

\*datum\_i\_vreme

\*naziv\_filma

Pk, Fk2 sal\_r\_br

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Табела | Тип тригера | Колона | Потребан | Шта треба да се уради? | Редни број тригера |
| FILMOVI | INSERT |  | НЕ |  |  |
| UPDATE | naziv | ДА | Приликом ажурирања колоне **назив** у табели **ФИЛМОВИ** покреће се тригер који ће ажурирати колону **назив\_филма** у табели **ПРИКАЗИВАЊА** | 1 |
| DELETE |  | НЕ |  |  |
| PRIKAZIVANJA | INSERT |  | ДА | Ажурирање колоне **назив\_филма** на основу вредности унетеу колону **фил\_р\_бр** у табели  **ПРИКАЗИВАЊА** | 3 |
| UPDATE | Fil\_r\_br | ДА | Ажурирање колоне **назив\_филма** на основу вредности унете у колону **фил\_р\_бр** | 3 |
| Naziv\_filma | ДА | Забранити директно ажурирање ове колоне | 2 |
| DELETE |  | НЕ |  |  |

Табела 2 Табела спецификације тригера за денормализацију 2

ALTER TABLE prikazivanja

ADD naziv\_filma VARCHAR2(50);

UPDATE prikazivanja

SET naziv\_filma = (

SELECT naziv

FROM filmovi

WHERE prikazivanja.fil\_r\_br = filmovi.r\_br);

ALTER TABLE prikazivanja

MODIFY naziv\_filma VARCHAR2(50) NOT NULL;

* + 1. **Тригер 1**

CREATE OR REPLACE TRIGGER filmovi\_update\_naziv

AFTER

UPDATE OF naziv ON filmovi

FOR EACH ROW

DECLARE

v\_new\_naziv VARCHAR2(50);

v\_fil\_r\_br NUMBER(6,0);

BEGIN

v\_new\_naiv := :NEW.naziv;

v\_fil\_r\_br := :NEW.fil\_r\_br;

EXECUTE IMMEDIATE 'ALTER TRIGGER prikazivanja\_insert\_update DISABLE';

EXECUTE IMMEDIATE 'ALTER TRIGGER prikazivanja\_update\_naziv DISABLE';

UPDATE prikazivanja

SET naziv\_filma = v\_new\_naziv

WHERE fil\_r\_br = v\_fil\_r\_br;

EXECUTE IMMEDIATE 'ALTER TRIGGER prikazivanja\_insert\_update ENABLE';

EXECUTE IMMEDIATE 'ALTER TRIGGER prikazivanja\_update\_naziv ENABLE';

END;

* + 1. **Тригер 2**

CREATE OR REPLACE TRIGGER prikazivanja\_update\_naziv

BEFORE

UPDATE OF naziv\_filma ON prikazivanja

FOR EACH ROW

BEGIN

RAISE\_APPLICATION\_ERROR(

NUM => -20002,

MSG => 'Naziv filma se ne moze direktno menjati!');

END;

* + 1. **Тригер 3**

CREATE OR REPLACE TRIGGER prikazivanja\_insert\_update

BEFORE

INSERT OR UPDATE ON prikazivanja

FOR EACH ROW

DECLARE

PRAGMA AUTONOMOUS\_TRANSACTION;

v\_naziv VARCHAR2(50);

BEGIN

EXECUTE IMMEDIATE 'ALTER TRIGGER prikazivanja\_update\_naziv DISABLE';

SELECT naziv

INTO v\_naziv

FROM filmovi

WHERE r\_br = :NEW.fil\_r\_br;

:NEW.naziv\_filma := v\_naziv;

EXECUTE IMMEDIATE 'ALTER TRIGGER prikazivanja\_update\_naziv ENABLE';

END;

1. **Оптимизација**

Оптимизација базе података обухвата различите технике којима се желе побољшати перформансе рада базе података.

* 1. **Оптимизација изведена на заснованим вредностима**

Концептуални модел пре оптимизације:

SALA

#R.br.

\*Broj mesta

\*Naziv

PRIKAZIVANJE

\*Datum i vreme

Концептуални модел након оптимизације:

PRIKAZIVANJE

\*Datum i vreme

\*Sala

Вредности које је садржао ентитен САЛА ( број места и назив), сада се налазе у апликационом коду, при чему смо избегли спајање табела.

* 1. **Индекси**

Индекси представљају структуре података које омогућавају ефиксан приступ подацима који се налазе у бази података. То су објекти базе података који могу убрзати приступ редовима табеле. Побољшавају перфомансе претраживања и осигуравају јединственост.

* + 1. **Индекс 1**

Креирамо индекс за колону **назив** у табели **ФИЛМОВИ**. Често ће се претраживати називи филмова па ћемо овим индексом смањити време изврашања овог упита. Код за креирање индекса:

CREATE INDEX i\_naziv ON filmovi(naziv);

* + 1. **Индекс 2**

Креирамо индекс за колону **назив улоге** у табели **УЛОГЕ**. Често ће се претраживати назив улоге па ћемо овим индексом смањити време извршавања овог упита. Код за креирање индекса:

CREATE INDEX i\_uloga ON uloge(naziv\_uloge);

* + 1. **Индекс 3**

Креирамо индекс за колону **име** у табели **ГЛУМЦИ**. Често ће се претраживати име глумца па ћемо овим индексом смањити време извршавања овог упита. Код за креирање индекса:

CREATE INDEX i\_ime ON glumci(ime);

1. **Релациона шема**

FILMOVI

Pk R\_br

\*Naziv

\*Zanr

FK reditelj\_id

GLUMCI

Pk ID

\*Ime

\*Prezime

ULOGE

Pk ID

\*Naziv\_uloge

Fk1\* fil\_r\_b

Fk2\*glu\_id

KARTE

Pk Br\_karte

\*Vreme

\*Cena\_karte

\*Rezervacija

oPopust

Fk1 \* fil\_R\_br

Fk2 o sal\_R\_br

Fk3 o gle\_R\_br

Fk4 o rad\_JMBG

RADNICI

Pk JMBG

\*Ime

\*Prezime

GLEDAOCI

Pk R\_br

\*Clan

oIme

oPrezime

oPol

oOmiljeni\_zanr

PRIKAZIVANJA

Pk, Fk1 fil\_r\_br

\*datum\_i\_vreme

Pk, Fk2 sal\_r\_br

SALE

Pk R\_br

\*Broj\_mesta

\*Naziv

REDITELJI

#ID

\*Ime

\*Prezime

* 1. **Кодови за креирање табела**

**Код за креирање табеле ФИЛМОВИ**

CREATE TABLE filmovi

(R\_br NUMBER(30),

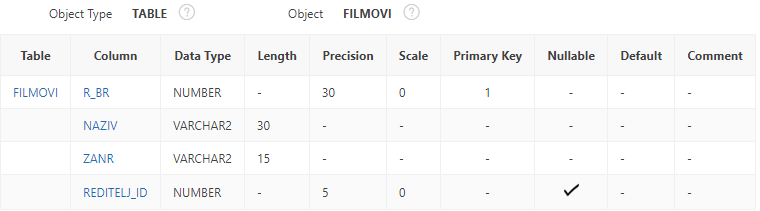
naziv VARCHAR2(30) NOT NULL,

zanr VARCHAR2(15) NOT NULL,

CONSTRAINT red\_id FOREIGN KEY (reditelj\_id)

REFERENCES reditelji(id),

CONSTRAINT fil\_R\_br\_pk PRIMARY KEY(R\_br));



Слика 8 Табела ФИЛМОВИ

**Код за креирање табеле ГЛУМЦИ**

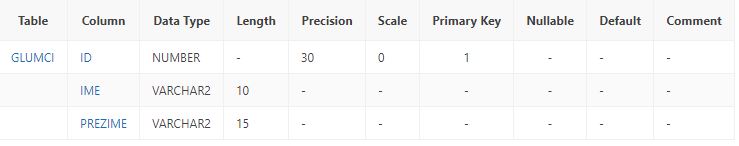
CREATE TABLE glumci

(id NUMBER(30),

ime VARCHAR2(10) NOT NULL,

prezime VARCHAR2(15) NOT NULL,

CONSTRAINT glu\_id\_pk PRIMARY KEY(id));



Слика 9 Табела ГЛУМЦИ

**Код за креирање табеле ГЛЕДАОЦИ**

CREATE TABLE gledaoci

(R\_br NUMBER(20),

clan VARCHAR2(2) NOT NULL,

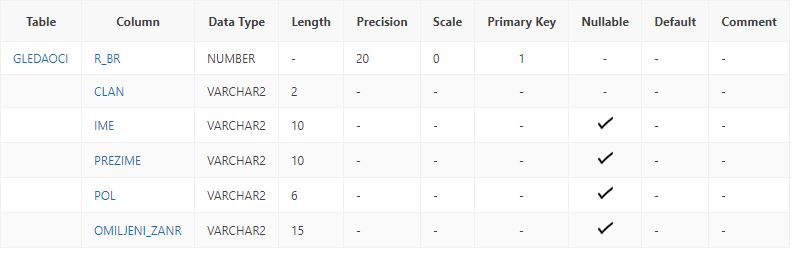
ime VARCHAR2(10),

prezime VARCHAR(10),

pol VARCHAR2(6),

omiljeni\_zanr VARCHAR2(15),

CONSTRAINT gle\_R\_br\_pk PRIMARY KEY(R\_br));



Слика 10 Табела ГЛЕДАОЦИ

**Код за креирање табеле РАДНИЦИ**

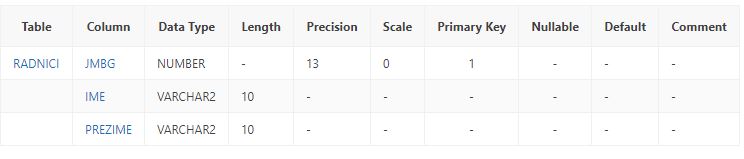
CREATE TABLE radnici

(JMBG NUMBER(13),

ime VARCHAR2(10) NOT NULL,

prezime VARCHAR(10) NOT NULL,

CONSTRAINT rad\_JMBG\_pk PRIMARY KEY(JMBG));



Слика 11 Табела РАДНИЦИ

**Код за креирање табеле САЛЕ**

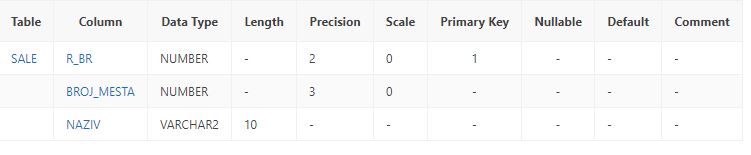
CREATE TABLE sale

(R\_br NUMBER(2),

broj\_mesta NUMBER(3) NOT NULL,

naziv VARCHAR2(10) NOT NULL,

CONSTRAINT sal\_R\_br\_pk PRIMARY KEY(R\_br));



Слика 12 Табела САЛЕ

**Код за креирање табеле УЛОГЕ**

CREATE TABLE uloge

(ID NUMBER(5),

naziv\_uloge VARCHAR2(20) NOT NULL,

fil\_R\_br NUMBER(30),

glu\_id NUMBER(30),

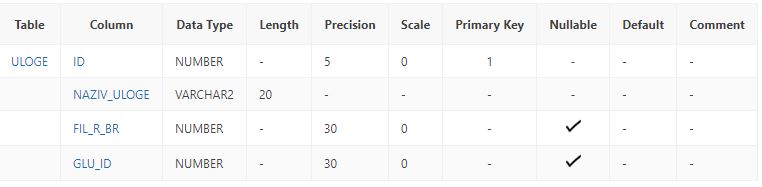
CONSTRAINT ulo\_id\_pk PRIMARY KEY(id),

CONSTRAINT ulo\_fil\_R\_br\_fk FOREIGN KEY(fil\_R\_br)

REFERENCES filmovi(R\_br),

CONSTRAINT ulo\_glu\_id\_fk2 FOREIGN KEY(glu\_id)

REFERENCES glumci(id));



Слика 13 Табела УЛОГЕ

**Код за креирање табеле КАРТЕ**

CREATE TABLE karte

(Br\_karte NUMBER(5),

vreme DATE NOT NULL,

cena\_karte NUMBER(4) NOT NULL,

rezervacija VARCHAR2(2) NOT NULL,

popust NUMBER(2),

fil\_R\_br NUMBER(30),

sal\_R\_br NUMBER(2),

gle\_R\_br NUMBER(20),

rad\_JMBG NUMBER(13),

CONSTRAINT kar\_Br\_karte\_pk PRIMARY KEY(Br\_karte),

CONSTRAINT kar\_fil\_R\_br\_fk FOREIGN KEY(fil\_R\_br)

REFERENCES filmovi(R\_br),

CONSTRAINT kar\_sal\_R\_br\_fk2 FOREIGN KEY(sal\_R\_br)

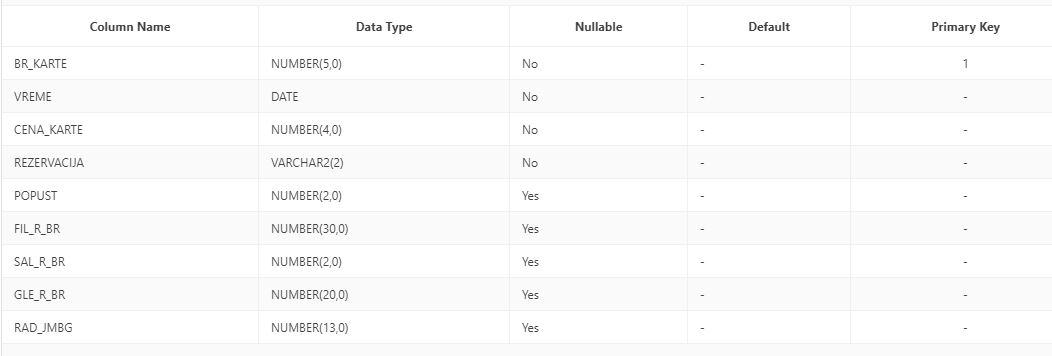
REFERENCES sale(R\_br),

CONSTRAINT kar\_gle\_R\_br\_fk3 FOREIGN KEY(gle\_R\_br)

REFERENCES gledaoci(R\_br),

CONSTRAINT kar\_rad\_JMBG\_fk4 FOREIGN KEY(rad\_JMBG)

REFERENCES radnici(JMBG));



Слика 14 Табела КАРТЕ

**Код за креирање табеле ПРИКАЗИВАЊА**

CREATE TABLE prikazivanja

(fil\_R\_br NUMBER(30),

datum\_i\_vreme DATE NOT NULL,

sal\_R\_br NUMBER(2),

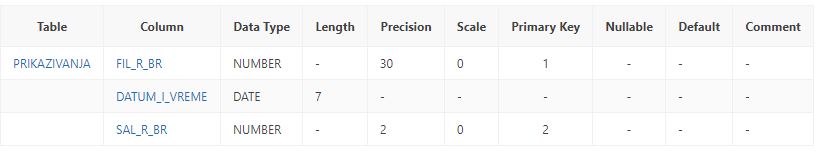
CONSTRAINT prikazivanja\_pk PRIMARY KEY(fil\_R\_br, sal\_R\_br),

CONSTRAINT fil\_R\_br\_fk FOREIGN KEY(fil\_R\_br)

REFERENCES filmovi(R\_br),

CONSTRAINT sal\_R\_br\_fk2 FOREIGN KEY(sal\_R\_br)

REFERENCES sale(R\_br));



Слика 15 Табела ПРИКАЗИВАЊА

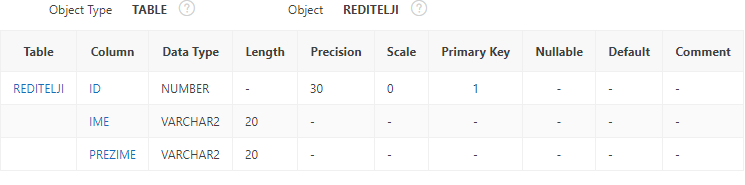
**Код за креирање табеле РЕДИТЕЉИ**

CREATE TABLE "REDITELJI"

( "ID" NUMBER(30,0),

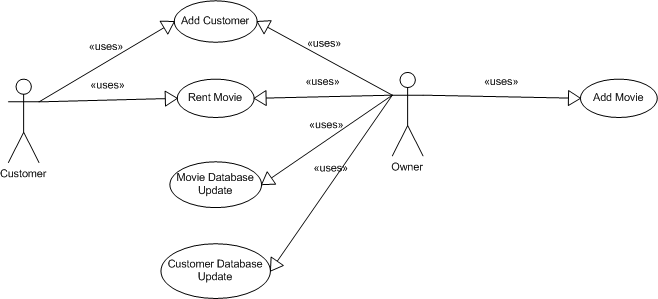
"IME" VARCHAR2(20) NOT NULL ENABLE,

"PREZIME" VARCHAR2(20) NOT NULL ENABLE,)



Слика 16 Табела РЕДИТЕЉИ

1. **UML дијаграм случајева коришћења**



Слика 17 УМЛ дијаграм

1. **Услов за имплементацију**

За израду комплетног пројектног рада, било је неопходно користити Oracle Apex и Microsoft Word.

У Apex-у је направњена база података SQL кодом, док је за апликацију било потребно користити чаробњака, који се налази у склопу самог Apex-a.

1. **Подаци за тестирање**

**Табела ФИЛМОВИ**

INSERT INTO filmovi

VALUES ('1', 'Južni vetar', 'triler');

INSERT INTO filmovi

VALUES ('2', 'Apsolutnih 100', 'drama');

INSERT INTO filmovi

VALUES ('3', 'Mrtav ladan', 'komedija');

INSERT INTO filmovi

VALUES('4', 'Lajanje na zvezde', 'komedija');

INSERT INTO filmovi

VALUES('5', 'Balkanski špijun', 'drama');

INSERT INTO filmovi

VALUES('6', 'Ko to tamo peva', 'komedija');

INSERT INTO filmovi

VALUES('7', 'Mi nismo anđeli 2', 'komedija');

INSERT INTO filmovi

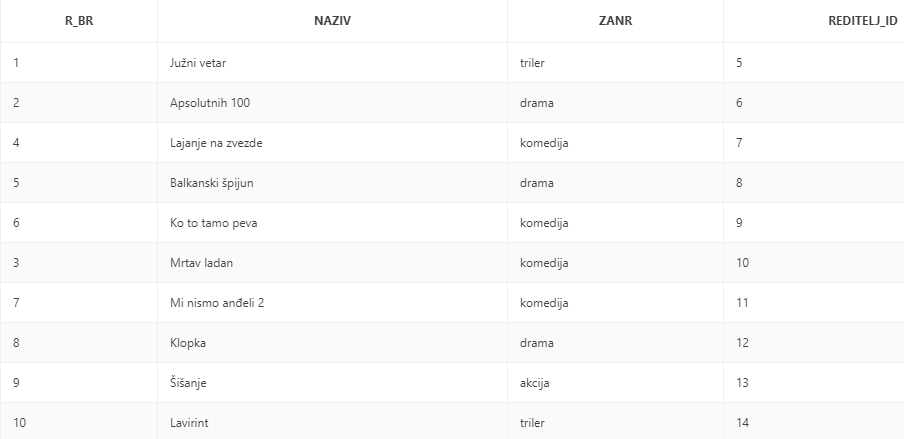
VALUES('8', 'Klopka', 'drama');

INSERT INTO filmovi

VALUES('9', 'Šišanje', 'akcija');

INSERT INTO filmovi

VALUES('10', 'Lavirint', 'triler');



Слика 18 Попуњена табела ФИЛМОВИ

**Табела ГЛУМЦИ**

INSERT INTO glumci

VALUES ('21', 'Miloš', 'Biković');

INSERT INTO glumci

VALUES ('56', 'Vuk', 'Kostić');

INSERT INTO glumci

VALUES ('31', 'Srđan', 'Todorović');

INSERT INTO glumci

VALUES ('89', 'Nataša', 'Šolak');

INSERT INTO glumci

VALUES ('196', 'Mira', 'Banjac');

INSERT INTO glumci

VALUES ('64', 'Dragan', 'Nikolić');

INSERT INTO glumci

VALUES ('259', 'Nikola', 'Kojo');

INSERT INTO glumci

VALUES ('659', 'Nebojša', 'Glogovac');

INSERT INTO glumci

VALUES ('69', 'Viktor', 'Savić');

INSERT INTO glumci

VALUES ('851', 'Branislav', 'Lekić');



Слика 19 Попуњена табела ГЛУМЦИ

**Табела ГЛЕДАОЦИ**

INSERT INTO gledaoci

VALUES ('48', 'Da', 'Milan', 'Pajović', 'muško', 'komedija');

INSERT INTO gledaoci

VALUES ('615', 'Ne', '', '', '', '');

INSERT INTO gledaoci

VALUES ('12', 'Da', 'Ivana', 'Janković', 'žensko', 'drama');

INSERT INTO gledaoci

VALUES ('18', 'Ne', '', '', '', '');

INSERT INTO gledaoci

VALUES ('99', 'Da', 'Dragan', 'Ivković', 'muško', 'akcija');

INSERT INTO gledaoci

VALUES ('151', 'Ne', '', '', '', '');

INSERT INTO gledaoci

VALUES ('325', 'Da', 'Marija', 'Terzić', 'žensko', 'triler');

INSERT INTO gledaoci

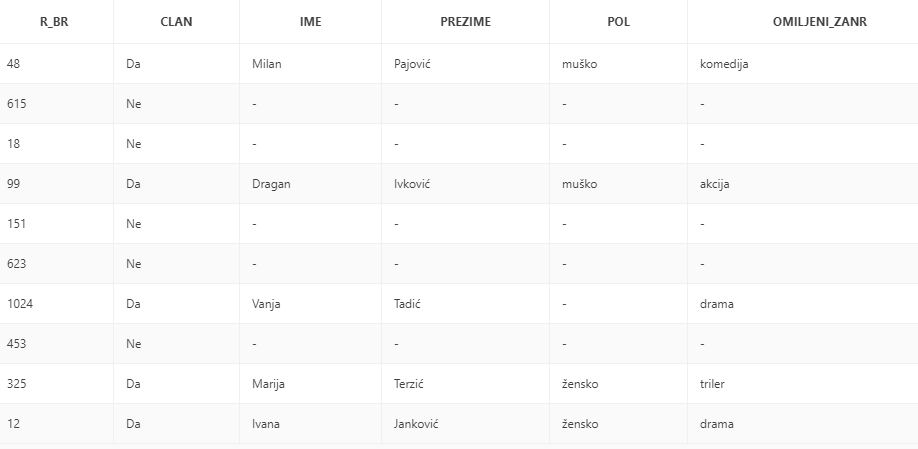
VALUES ('623', 'Ne', '', '', '', '');

INSERT INTO gledaoci

VALUES ('1024', 'Da', 'Vanja', 'Tadić', '', 'drama');

INSERT INTO gledaoci

VALUES ('453', 'Ne', '', '', '', '');



Слика 20 Попуњена табела ГЛЕДАОЦИ

**Табела РАДНИЦИ**

INSERT INTO radnici

VALUES ('1511978452565', 'Dragan', 'Marković');

INSERT INTO radnici

VALUES ('2109993426598', 'Marija', 'Savić');

INSERT INTO radnici

VALUES ('3012985421637', 'Ivan', 'Kovačić');

INSERT INTO radnici

VALUES ('2902984214585', 'Sanja', 'Ilić');

INSERT INTO radnici

VALUES ('0506986454758', 'Vladimir', 'Petković');

INSERT INTO radnici

VALUES ('0707976985847', 'Dragana', 'Stanić');

INSERT INTO radnici

VALUES ('0808975256597', 'Marko', 'Kostić');

INSERT INTO radnici

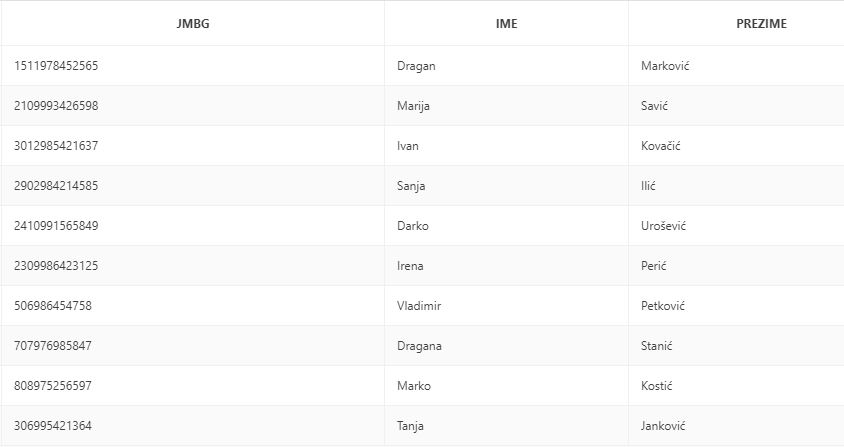
VALUES ('0306995421364', 'Tanja', 'Janković');

INSERT INTO radnici

VALUES ('2410991565849', 'Darko', 'Urošević');

INSERT INTO radnici

VALUES ('2309986423125', 'Irena', 'Perić');



Слика 21 Попуњена табела РАДНИЦИ

**Табела САЛЕ**

INSERT INTO sale

VALUES ('2', '150', 'Sala 1');

INSERT INTO sale

VALUES ('5', '200', 'Sala 2');

INSERT INTO sale

VALUES ('21', '100', 'Sala 3');

INSERT INTO sale

VALUES ('4', '70', 'Sala 4');

INSERT INTO sale

VALUES ('51', '50', 'Sala 5');

INSERT INTO sale

VALUES ('24', '150', 'Sala 6');

INSERT INTO sale

VALUES ('13', '250', 'Sala 7');

INSERT INTO sale

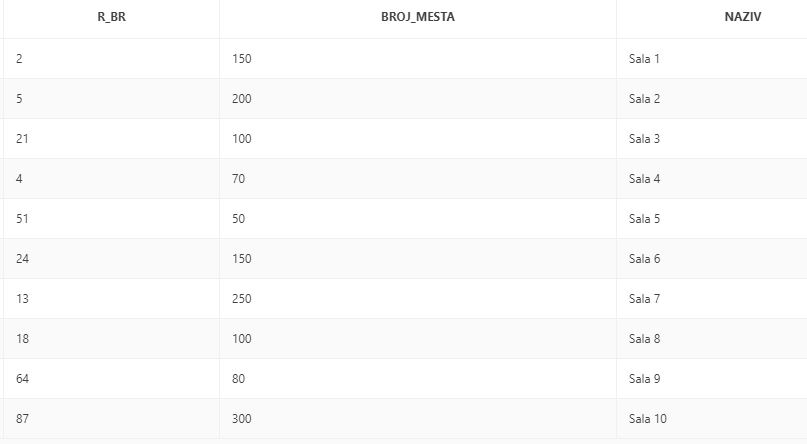
VALUES ('18', '100', 'Sala 8');

INSERT INTO sale

VALUES ('64', '80', 'Sala 9');

INSERT INTO sale

VALUES ('87', '300', 'Sala 10');



Слика 22 Попуњена табела САЛЕ

**Табела УЛОГЕ**

INSERT INTO uloge

VALUES ('5', 'Petar Maraš', '1', '21');

INSERT INTO uloge

VALUES ('21', 'Saša Gordić', '2', '56');

INSERT INTO uloge

VALUES ('13', 'Kiza', '3', '31');

INSERT INTO uloge

VALUES ('15', 'Danica Janković', '4', '89');

INSERT INTO uloge

VALUES ('96', 'Danica Čvorović', '5', '196');

INSERT INTO uloge

VALUES ('6', 'Šlager pevač', '6', '64');

INSERT INTO uloge

VALUES ('24', 'Nikola', '7', '259');

INSERT INTO uloge

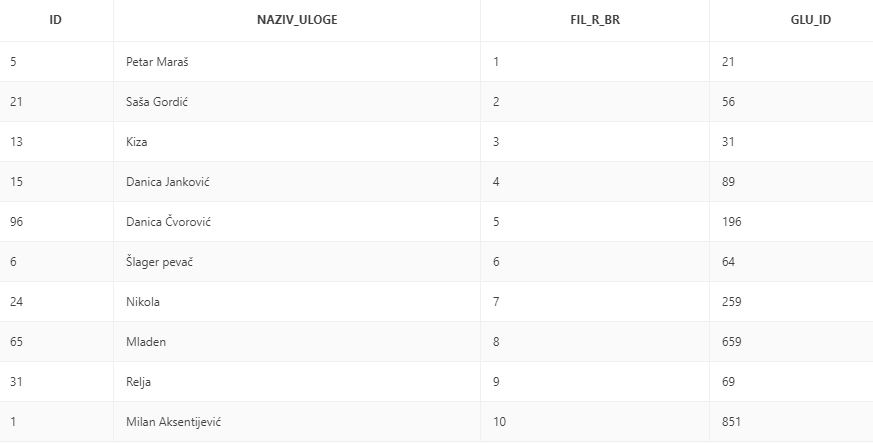
VALUES ('65', 'Mladen', '8', '659');

INSERT INTO uloge

VALUES ('31', 'Relja', '9', '69');

INSERT INTO uloge

VALUES ('1', 'Milan Aksentijević', '10', '851');



Слика 23 Попуњена табела УЛОГЕ

**Табела КАРТЕ**

INSERT INTO karte

VALUES ('51', '21', '09/23/2018', '250', 'Da', '50', '1', '2', '48', '1511978452565');

INSERT INTO karte

VALUES ('12', '34', '08/28/2018', '250', 'Ne', '', '2', '5', '615', '2109993426598');

INSERT INTO karte

VALUES ('65', '2', '07/21/2018', '250', 'Da', '50', '3', '21', '12', '3012985421637');

INSERT INTO karte

VALUES ('5', '56', '03/03/2018', '250', 'Ne', '', '4', '4', '18', '2902984214585');

INSERT INTO karte

VALUES ('13', '49', '12/29/2018', '250', 'Da', '50', '5', '51', '99', '0506986454758');

INSERT INTO karte

VALUES ('99', '132', '11/05/2018', '250', 'Ne', '', '6', '24', '151', '0707976985847');

INSERT INTO karte

VALUES ('2', '1', '06/13/2018', '250', 'Da', '50', '7', '13', '325', '0808975256597');

INSERT INTO karte

VALUES ('3', '16', '01/22/2018', '250', 'Ne', '', '8', '18', '623', '0306995421364');

INSERT INTO karte

VALUES ('69', '64', '02/21/2018', '250', 'Da', '50', '9', '64', '1024', '2410991565849');

INSERT INTO karte

VALUES ('14', '251', '09/11/2018', '250', 'Ne', '', '10', '87', '453', '2309986423125');



Слика 24 Попуњена табела КАРТЕ

**Табела ПРИКАЗИВАЊА**

INSERT INTO prikazivanja

VALUES ('1', '09/23/2018', '2');

INSERT INTO prikazivanja

VALUES ('2', '08/28/2018', '5');

INSERT INTO prikazivanja

VALUES ('3', '07/21/2018', '21');

INSERT INTO prikazivanja

VALUES ('4', '03/03/2018', '4');

INSERT INTO prikazivanja

VALUES ('5', '12/29/2018', '51');

INSERT INTO prikazivanja

VALUES ('6', '11/05/2018', '24');

INSERT INTO prikazivanja

VALUES ('7', '06/13/2018', '13');

INSERT INTO prikazivanja

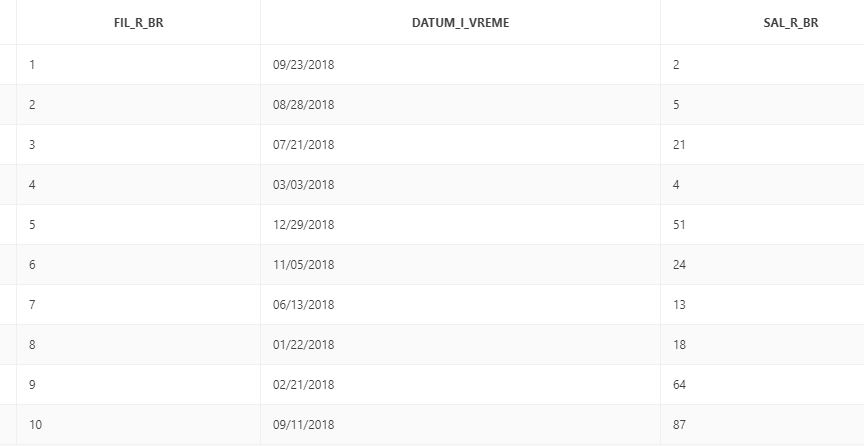
VALUES ('8', '01/22/2018', '18');

INSERT INTO prikazivanja

VALUES ('9', '02/21/2018', '64');

INSERT INTO prikazivanja

VALUES ('10', '09/11/2018', '87');



Слика 25 Попуњена табела ПРИКАЗИВАЊА

**Табела РЕДИТЕЉИ**

INSERT INTO reditelji

VALUES (’5’, ’Dušan’, ’Kovačević’);

INSERT INTO reditelji

VALUES (’6’, ’Srđan’, ’Golubović’);

INSERT INTO reditelji

VALUES (’7’, ’Milorad’, ’Milinković’);

INSERT INTO reditelji

VALUES (’8’, ’Zdravko’, ’Šotra’);

INSERT INTO reditelji

VALUES (’9’, ’Dušan’, ’Kovačević’);

INSERT INTO reditelji

VALUES (’10’, ’ Slobodan’, ’Šijan’);

INSERT INTO reditelji

VALUES (’11’, ’Srđan’, ’Dragojević’);

INSERT INTO reditelji

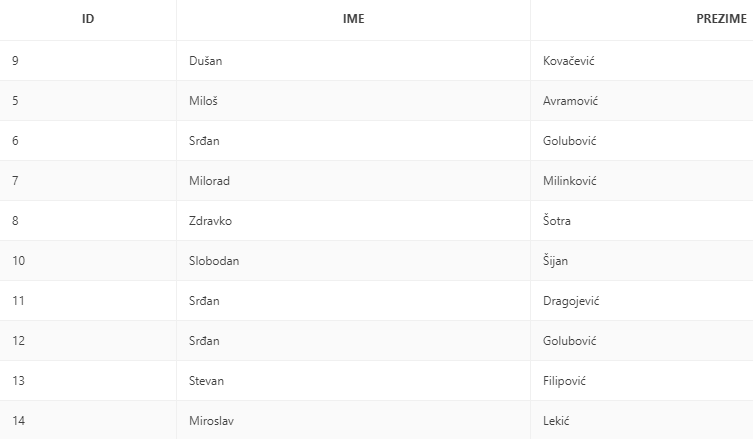
VALUES (’12’, ’ Srđan’, ’Golubović’);

INSERT INTO reditelji

VALUES (’13’, ’Stevan’, ’Filipović’);

INSERT INTO reditelji

VALUES (’14’, ’Miroslav’, ’Lekić’);



Слика 26 Попуњена табела РЕДИТЕЉИ

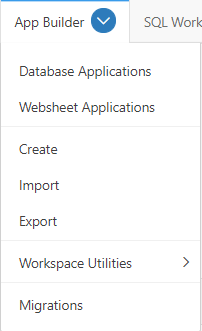
1. **Опис апликације са упутством за коришћење**

Како би се олакшао рад са клијентима, креирана је апликација, која ће помоћи запосленима да имају лакши и бољи приступ бази како би наши корисници добили најбољу и најбржу услугу. Апликација је креирана у Oracle Apex-u уз помоћ App Builder-a.

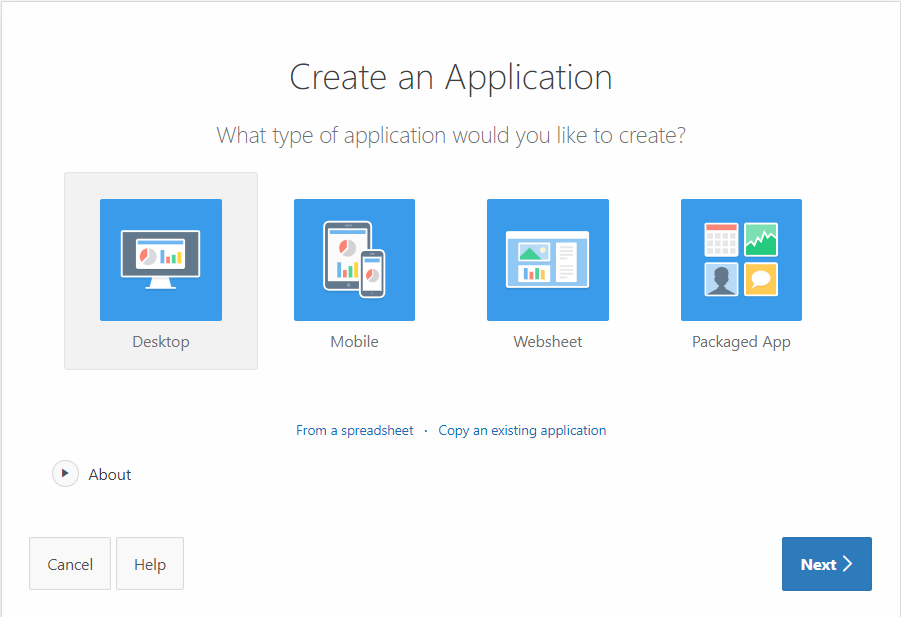
* 1. **Креирање апликације**

Апликација омогућава унос и измени постојећих вредности, као и увид у клијенте и њихову куповину.

Следе кораци за креирање апликације:

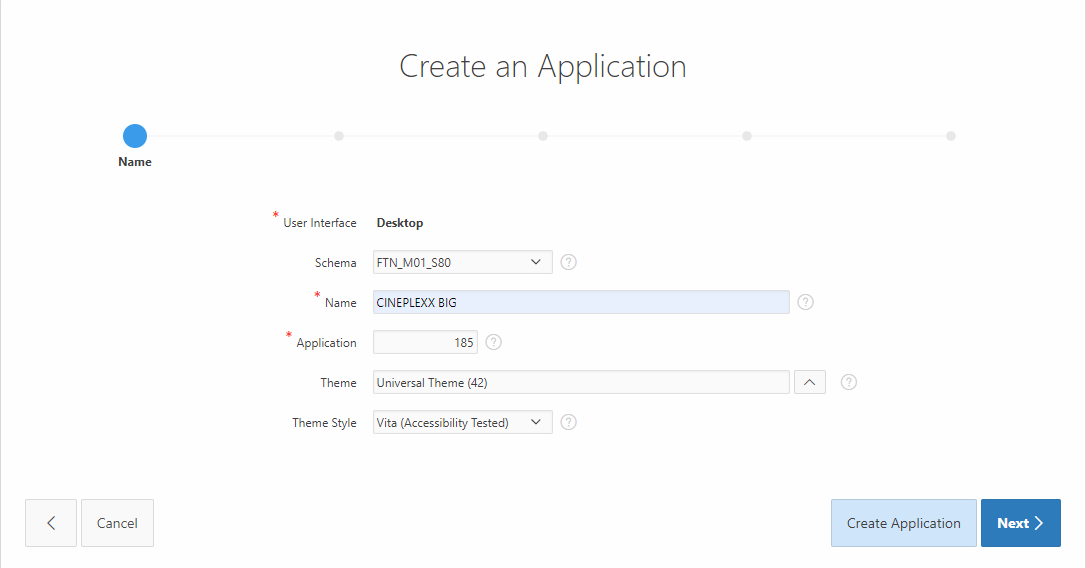
У App Builder-u је потребно изабрати Create, за креирање саме апликације.

Слика 27 App Builder

Потребно је изабрати тип апликације, која се жели креирати. Ова опликација ће бити Desktop.

Слика 28 Тип апликације

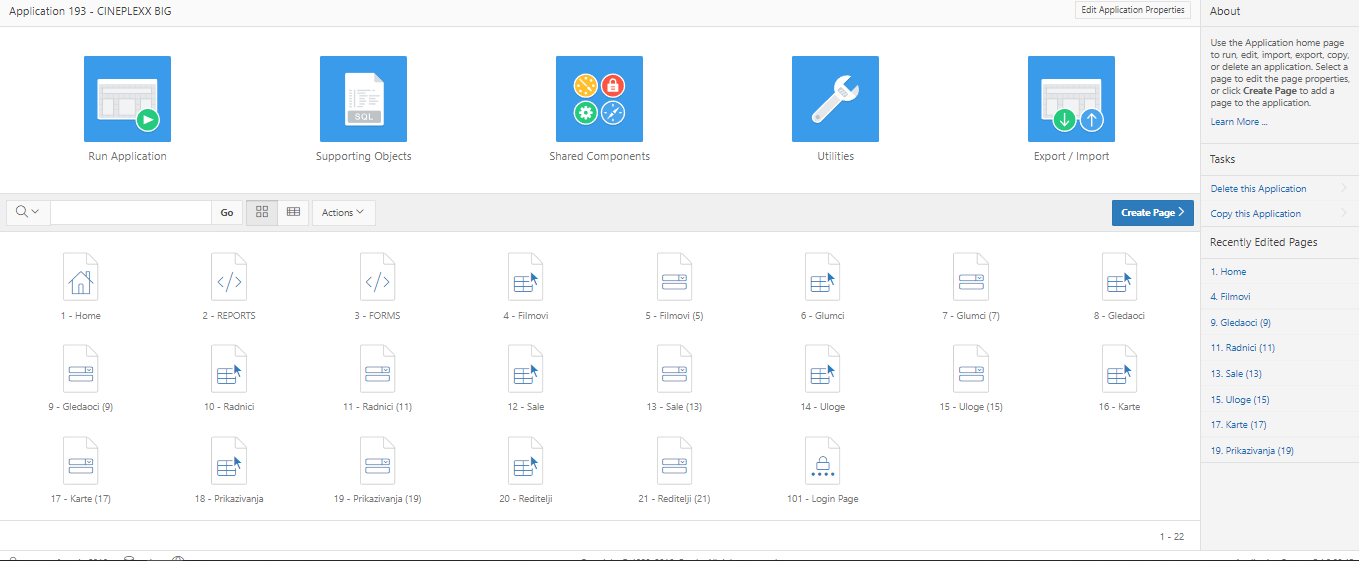
Потребно је дефинисати назив апликације, као и изабрати одговарајућу тему, а потом ићи на Next.



Слика 29 Креирање апликације

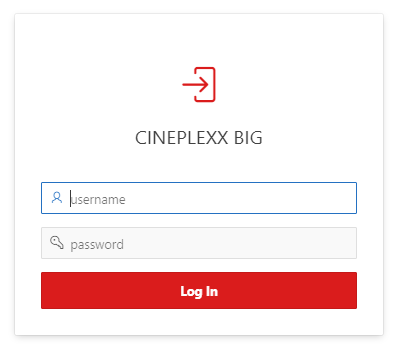
Након што смо попунили сва обавезна поља, добијамо апликацију, коју је потребно надоградити, тако што ћемо јој додавати нове странице и порадити на њеној функцоналоности.

Када се креира апликација, добијају се две странице: Home и Login.



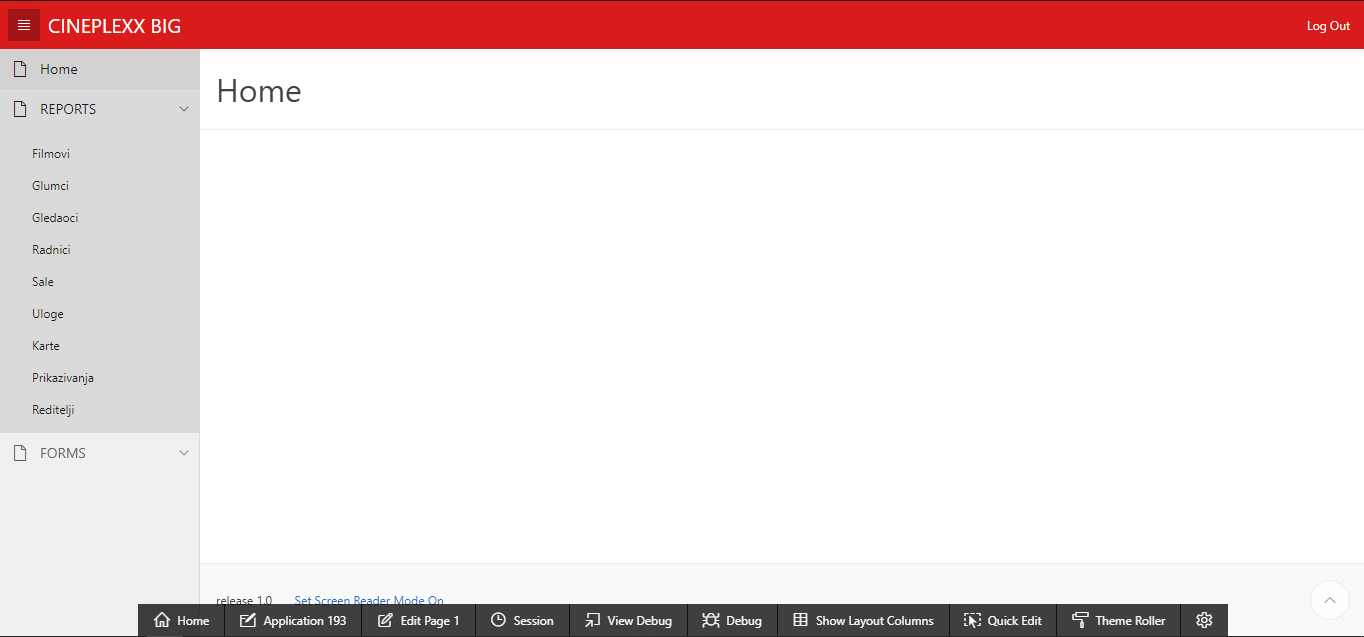
Слика 30 Приказ свих страница

Изглед Login стране са попуњеним корисничким именом и лозинком.



Слика 31 Login

Изглед Home странице у којој можемо изабрати преглед табела и где можемо унети податке у табеле уз помоћ форми.



Слика 32 Home

1. **Закључак**

На почетку семинарског рада је постављен детаљан опис рада биоскопа CINNEPLEXX BIG. У овом семинарском раду су коришћена средства (нормализација, денормализација, тригери, индекси и креирана апликација) како би се олакшао рад са базом података.

1. **Литература**
2. Катарина Митровић, *Курс програмирање база података*, <https://eucenje.ftn.kg.ac.rs/course/view.php?id=23>, 20.09.2019
3. Apex, <http://91.187.132.55:8081/apex>, 24.09.2019