Imię i nazwisko: Jakub Kozubek

Kurs: Bazy danych gr.5

Rok akademicki: 2020/2021

Kierunek: Informatyka stosowana

Wydział: Wydział Nauk Ścisłych i Technicznych

Aplikacja internetowa opierająca się na bazie filmów, aktorów i twórców

1. Opis projektu

Aplikacja została przeznaczona na stronę internetową z dostępem dla każdego użytkownika urządzenia komputerowego lub smartfona posiadające aktywne połączenie z Internetem. Baza danych zawiera takie informacje jak: tytuł filmu, czas trwania filmu, krótki opis i inne. Baza posiada również informację o obsadzie filmowej jak i twórców filmów. Każdy aktor jak i twórca jest ściśle powiązany z filmami w których grali lub byli jego twórcami.

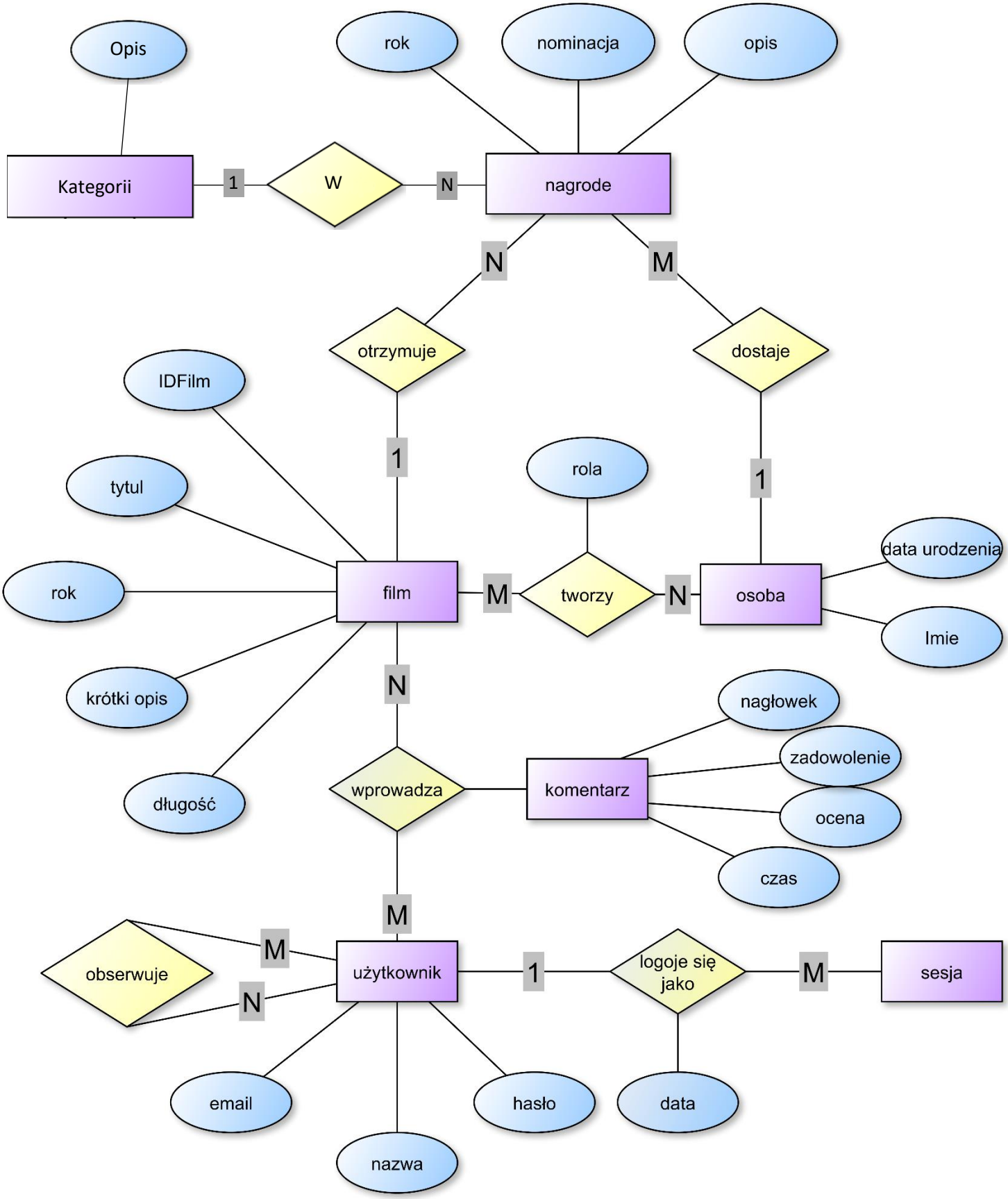
Każdy film może być oceniany przez zalogowanych użytkowników do danej strony internetowej. Oceny są w postaci komentarzy, to znaczy krótkich informacji tekstowych na temat filmu.

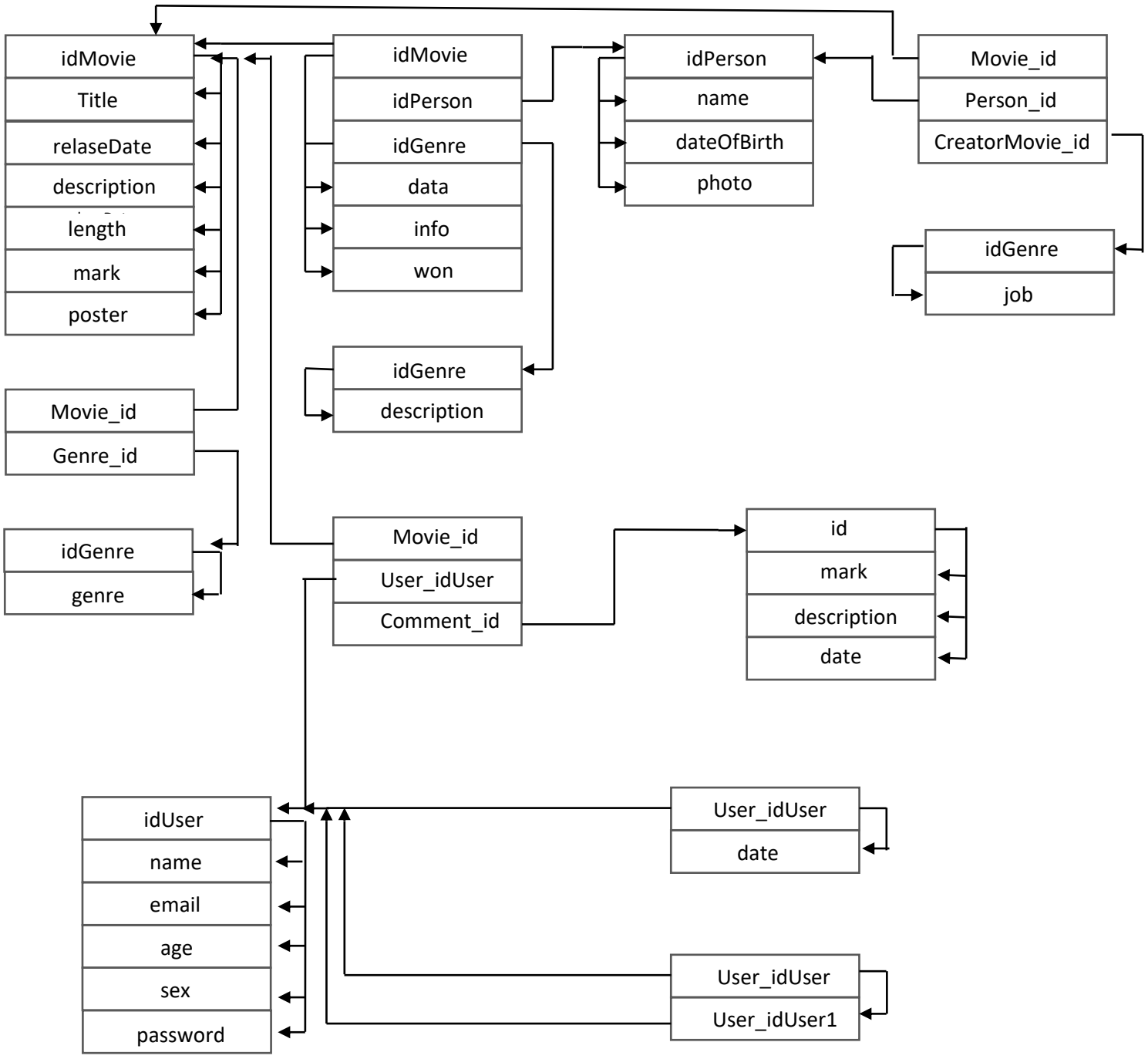
Użytkownicy strony internetowej nie posiadający jeszcze konta, mogą je utworzyć. Utworzenie konta polega na podaniu kilku informacji o sobie takich jak: imię, nazwisko, wiek, email, płeć oraz hasło do swojego konta.

Użytkownicy mają możliwość dodawania innych użytkowników do listy obserwowanych.

Baza prowadzi również informację o logowaniu się użytkowników, przechowuje informacje takie jak datę logowania.

Diagram związków encji

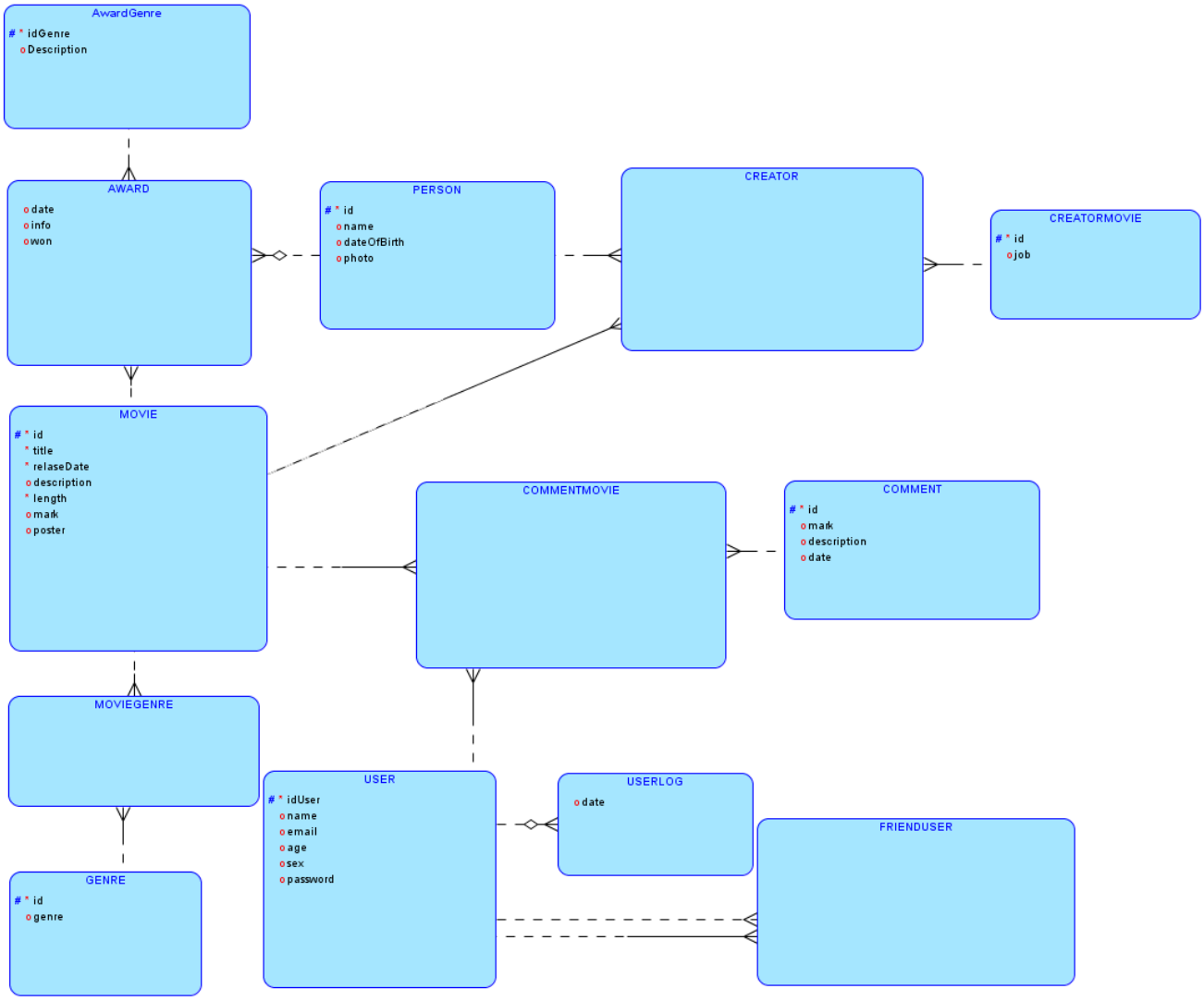



idMovie
Title
relaseDate
description
length
mark
poster
idMovie
idPerson
idGenre
data
info
won
idPerson
name
dateOfBirth
photo
Movie_id
Person_id
CreatorMovie_id
idGenre
job
idGenre
description
Movie_id
Genre_id
idGenre
genre
Movie_id
User_idUser
Comment_id
id
mark
description
date
idUser
name
email
age
sex
password
User_idUser
date
User_idUser
User_idUser1

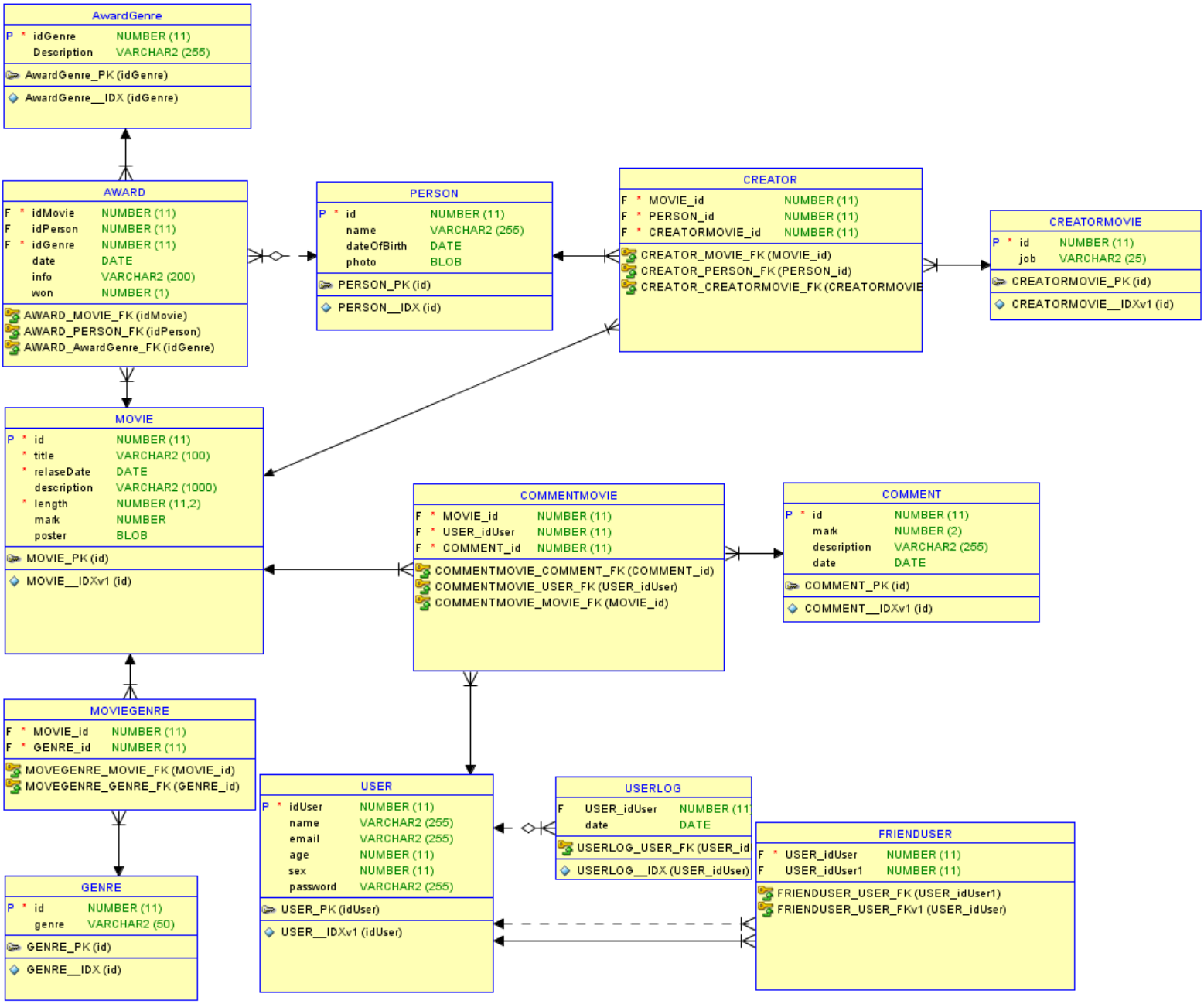
2. Specyfikacje funkcjonalna:
    - Baza pozwala na wyświetlanie informacji o filmach, aktorach lub twórcach filmu z podstawowymi informacjami. Daje możliwość szybkiego wyszukania danego aktora, twórcy czy filmu
    - Każdy użytkownik(zalogowany lub nie) ma dostęp do informacji o filmach, aktorach i twórcach filmu, oraz komentarzy zawartych pod filmem.
    - Użytkownik zalogowany dostaje możliwość dodawania komentarzy pod filmami, szukania i dodawania innych użytkowników do grona znajomych, przeglądania aktywności innych użytkowników.
    - Użytkownicy nie posiadający jeszcze konta mają możliwość stworzenia go.
3. Grupa docelowa osób:

    Grupą odbiorców aplikacji internetowej będę osoby chcące dowiedzieć się podstawowych informacji o filmie lub osobach tworzących film. Wiek nie jest istotny, konieczne jest jednak podstawowa umiejętność korzystania z komputera czy urządzenia mobilnego z dostępem do Internetu.

Model logiczny bazy danych



Model relacyjny bazy danych



Skrypt języka SQL (Fragment)

Cały skrypt ze względu na swoją ogromną strukturę zostanie skompresowany do zewnętrznego pliku.

```
CREATE TABLE award (
   idmovie   NUMBER(11) NOT NULL,
   idperson  NUMBER(11),
   idgenre   NUMBER(11) NOT NULL,
   "date"    DATE,
   info      VARCHAR2(200),
   won       NUMBER(1)
);

ALTER TABLE award
   ADD CHECK ( won BETWEEN 0 AND 1
            OR won IN ( 0, 1 ) );

COMMENT ON COLUMN award.idmovie IS
   'idMovie wskazuje na konkretny film, który otrzymał nagrodę lub otrzymał nominację';

COMMENT ON COLUMN award.idperson IS
   'idPerson wskazuje na konkretną osobę, która otrzymała nagrodę lub otrzymała
nominację';

COMMENT ON COLUMN award."date" IS
   'Data otrzymania nominacji/nagrody';

COMMENT ON COLUMN award.info IS
   'Opis nominacja/wygrana oraz jaka kategoria';

COMMENT ON COLUMN award.won IS
   'Jeśli 0 film otrzymał nominację
Jeśli 1 film otrzymał wygraną';

CREATE TABLE awardgenre (
   idgenre      NUMBER(11) NOT NULL,
   description  VARCHAR2(255)
);

COMMENT ON COLUMN awardgenre.idgenre IS
   'idGenre oznacza jednoznacznie na daną kategorię w jakiej film uzyskał nagrode';

COMMENT ON COLUMN awardgenre.description IS
   'Dłuższy opis kategorii w jakiej film otrzymał nagrodę';

CREATE INDEX awardgenre__idx ON
   awardgenre (
      idgenre
   ASC );
```

```
ALTER TABLE awardgenre ADD CONSTRAINT awardgenre_pk PRIMARY KEY ( idgenre );

CREATE TABLE "COMMENT" (
   id          NUMBER(11) NOT NULL,
   mark        NUMBER(2),
   description  VARCHAR2(255),
   "date"      DATE
);

COMMENT ON COLUMN "COMMENT".id IS
   'Id wskazujące jednoznacznie na konkretny komentarz użytkownia ';

COMMENT ON COLUMN "COMMENT".mark IS
   'Ocena użytkownika w skali od 1 do 10';

COMMENT ON COLUMN "COMMENT".description IS
   'Opis słowny komentarza';

COMMENT ON COLUMN "COMMENT"."date" IS
   'Data wystawienia komentarza przez użytkownika';

CREATE UNIQUE INDEX comment__idxv1 ON
   "COMMENT" (
      id
   ASC );

ALTER TABLE "COMMENT" ADD CONSTRAINT comment_pk PRIMARY KEY ( id );

CREATE TABLE commentmovie (
   movie_id     NUMBER(11) NOT NULL,
   user_iduser  NUMBER(11) NOT NULL,
   comment_id   NUMBER(11) NOT NULL
);

CREATE TABLE creator (
   movie_id        NUMBER(11) NOT NULL,
   person_id       NUMBER(11) NOT NULL,
   creatormovie_id  NUMBER(11) NOT NULL
);

COMMENT ON COLUMN creator.creatormovie_id IS
   '0 = aktor, 1 = twórca(np: dziękowiec, reżyser)';

CREATE TABLE creatormovie (
   id   NUMBER(11) NOT NULL,
   job  VARCHAR2(25)
);

COMMENT ON COLUMN creatormovie.id IS
   'Id pracy jaka istnieje przy tworzeniu filmu';
```

```
COMMENT ON COLUMN creatormovie.job IS
   'Opis pracy jaką można wykonywać przy tworzeniu filmu';

CREATE UNIQUE INDEX creatormovie__idxv1 ON
   creatormovie (
      id
   ASC );

ALTER TABLE creatormovie ADD CONSTRAINT creatormovie_pk PRIMARY KEY ( id );

CREATE TABLE frienduser (
   user_iduser   NUMBER(11) NOT NULL,
   user_iduser1  NUMBER(11)
);

CREATE TABLE genre (
   id     NUMBER(11) NOT NULL,
   genre  VARCHAR2(50)
);

COMMENT ON COLUMN genre.id IS
   'Id generuje się automatycznie przez wyzwalacz';

COMMENT ON COLUMN genre.genre IS
   'Opis gatunku filmu np akcji, fantastyczny itp.';

CREATE INDEX genre__idx ON
   genre (
      id
   ASC );

ALTER TABLE genre ADD CONSTRAINT genre_pk PRIMARY KEY ( id );

CREATE TABLE movie (
   id          NUMBER(11) NOT NULL,
   title       VARCHAR2(100) NOT NULL,
   relasedate   DATE NOT NULL,
   description  VARCHAR2(1000),
   length      NUMBER(11, 2) NOT NULL,
   mark        NUMBER,
   poster      BLOB
);

COMMENT ON COLUMN movie.id IS
   'Id filmu przypisywana automatycznie przez wyzwalacz ';

COMMENT ON COLUMN movie.title IS
   'Tytuł filmu';

COMMENT ON COLUMN movie.relasedate IS
   'Data wypuszczenia filmu do kin';
```

```
COMMENT ON COLUMN movie.description IS
   'Opis filmu';

COMMENT ON COLUMN movie.length IS
   'Długość filmu w minutach';

COMMENT ON COLUMN movie.mark IS
   'Ocena filmu pobierana z zewnętrznej aplikacji';

COMMENT ON COLUMN movie.poster IS
   'Plakat filmu';

CREATE UNIQUE INDEX movie__idxv1 ON
   movie (
      id
   ASC );

ALTER TABLE movie ADD CONSTRAINT movie_pk PRIMARY KEY ( id );

CREATE TABLE moviegenre (
   movie_id  NUMBER(11) NOT NULL,
   genre_id  NUMBER(11) NOT NULL
);

COMMENT ON COLUMN moviegenre.movie_id IS
   'IdMovie wskazuje na konkretny film';

COMMENT ON COLUMN moviegenre.genre_id IS
   'IdGenre wskazuje na konkretny gatunek. Jeden film może mieć wiele gatunków';

CREATE TABLE person (
   id          NUMBER(11) NOT NULL,
   name        VARCHAR2(255),
   dateofbirth  DATE,
   photo       BLOB
);

COMMENT ON COLUMN person.id IS
   'Id identyfikuje jednoznacznie konkretną osobę, która tworzyła w jakikolwiek stopniu film';

COMMENT ON COLUMN person.photo IS
   'Zdjęcie osoby, która tworzyła film. Zdjęcie nie jest wymagane';

CREATE UNIQUE INDEX person__idx ON
   person (
      id
   ASC );

ALTER TABLE person ADD CONSTRAINT person_pk PRIMARY KEY ( id );

CREATE TABLE "USER" (
   iduser    NUMBER(11) NOT NULL,
```

```
   name     VARCHAR2(255),
   email    VARCHAR2(255),
   age      NUMBER(11),
   sex      NUMBER(11),
   password  VARCHAR2(255)
);

ALTER TABLE "USER"
   ADD CHECK ( sex IN ( 0, 1 ) );

COMMENT ON COLUMN "USER".iduser IS
   'IdUser wskazujące jednoznacznie na konkretnego użytkownika aplikacji';

COMMENT ON COLUMN "USER".name IS
   'Nazwa wybrana przez użytkownika';

COMMENT ON COLUMN "USER".sex IS
   '0 = mężczyzna a 1 = kobieta';

CREATE INDEX user__idxv1 ON
   "USER" (
      iduser
   ASC );

ALTER TABLE "USER" ADD CONSTRAINT user_pk PRIMARY KEY ( iduser );

CREATE TABLE userlog (
   user_iduser  NUMBER(11),
   "date"      DATE
);

COMMENT ON COLUMN userlog.user_iduser IS
   'IdUser wskazujący na użytkownika. Wszystkie logowania będą zapisywane w tej tabeli';

COMMENT ON COLUMN userlog."date" IS
   'Data logowania użytkownika';

CREATE INDEX userlog__idx ON
   userlog (
      user_iduser
   ASC );

ALTER TABLE award
   ADD CONSTRAINT award_awardgenre_fk FOREIGN KEY ( idgenre )
      REFERENCES awardgenre ( idgenre );

ALTER TABLE award
   ADD CONSTRAINT award_movie_fk FOREIGN KEY ( idmovie )
      REFERENCES movie ( id );

ALTER TABLE award
   ADD CONSTRAINT award_person_fk FOREIGN KEY ( idperson )
```

```
    REFERENCES person ( id );

ALTER TABLE commentmovie
  ADD CONSTRAINT commentmovie_comment_fk FOREIGN KEY ( comment_id )
    REFERENCES "COMMENT" ( id );

ALTER TABLE commentmovie
  ADD CONSTRAINT commentmovie_movie_fk FOREIGN KEY ( movie_id )
    REFERENCES movie ( id );

ALTER TABLE commentmovie
  ADD CONSTRAINT commentmovie_user_fk FOREIGN KEY ( user_iduser )
    REFERENCES "USER" ( iduser );

ALTER TABLE creator
  ADD CONSTRAINT creator_creatormovie_fk FOREIGN KEY ( creatormovie_id )
    REFERENCES creatormovie ( id );

ALTER TABLE creator
  ADD CONSTRAINT creator_movie_fk FOREIGN KEY ( movie_id )
    REFERENCES movie ( id );

ALTER TABLE creator
  ADD CONSTRAINT creator_person_fk FOREIGN KEY ( person_id )
    REFERENCES person ( id );

ALTER TABLE frienduser
  ADD CONSTRAINT frienduser_user_fk FOREIGN KEY ( user_iduser1 )
    REFERENCES "USER" ( iduser );

ALTER TABLE frienduser
  ADD CONSTRAINT frienduser_user_fkv1 FOREIGN KEY ( user_iduser )
    REFERENCES "USER" ( iduser );

ALTER TABLE moviegenre
  ADD CONSTRAINT movegenre_genre_fk FOREIGN KEY ( genre_id )
    REFERENCES genre ( id );

ALTER TABLE moviegenre
  ADD CONSTRAINT movegenre_movie_fk FOREIGN KEY ( movie_id )
    REFERENCES movie ( id );

ALTER TABLE userlog
  ADD CONSTRAINT userlog_user_fk FOREIGN KEY ( user_iduser )
    REFERENCES "USER" ( iduser );

CREATE OR REPLACE TRIGGER fknto_award BEFORE
  UPDATE OF idperson ON award
  FOR EACH ROW
BEGIN
  IF :old.idperson IS NOT NULL THEN
```

```
      raise_application_error(-20225, 'Non Transferable FK constraint AWARD_PERSON_FK on
table AWARD is violated');
   END IF;
END;
/

CREATE OR REPLACE TRIGGER fknto_userlog BEFORE
   UPDATE OF user_iduser ON userlog
   FOR EACH ROW
BEGIN
   IF :old.user_iduser IS NOT NULL THEN
      raise_application_error(-20225, 'Non Transferable FK constraint USERLOG_USER_FK on
table USERLOG is violated');
   END IF;
END;
/

CREATE SEQUENCE comment_id_seq START WITH 0 MINVALUE 0 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER comment_id_trg BEFORE
   INSERT ON "COMMENT"
   FOR EACH ROW
   WHEN ( new.id IS NULL )
BEGIN
   :new.id := comment_id_seq.nextval;
END;
/

CREATE SEQUENCE creatormovie_id_seq START WITH 0 MINVALUE 0 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER creatormovie_id_trg BEFORE
   INSERT ON creatormovie
   FOR EACH ROW
   WHEN ( new.id IS NULL )
BEGIN
   :new.id := creatormovie_id_seq.nextval;
END;
/

CREATE SEQUENCE genre_id_seq START WITH 0 MINVALUE 0 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER genre_id_trg BEFORE
   INSERT ON genre
   FOR EACH ROW
   WHEN ( new.id IS NULL )
BEGIN
   :new.id := genre_id_seq.nextval;
END;
/

CREATE SEQUENCE movie_id_seq START WITH 0 MINVALUE 0 NOCACHE ORDER;
```

```
CREATE OR REPLACE TRIGGER movie_id_trg BEFORE
   INSERT ON movie
   FOR EACH ROW
   WHEN ( new.id IS NULL )
BEGIN
   :new.id := movie_id_seq.nextval;
END;
/

CREATE SEQUENCE moviegenre_movie_id_seq START WITH 0 MINVALUE 0 NOCACHE
ORDER;

CREATE OR REPLACE TRIGGER moviegenre_movie_id_trg BEFORE
   INSERT ON moviegenre
   FOR EACH ROW
   WHEN ( new.movie_id IS NULL )
BEGIN
   :new.movie_id := moviegenre_movie_id_seq.nextval;
END;
/

CREATE SEQUENCE person_id_seq START WITH 0 MINVALUE 0 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER person_id_trg BEFORE
   INSERT ON person
   FOR EACH ROW
   WHEN ( new.id IS NULL )
BEGIN
   :new.id := person_id_seq.nextval;
END;
/

CREATE SEQUENCE user_iduser_seq START WITH 0 MINVALUE 0 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER user_iduser_trg BEFORE
   INSERT ON "USER"
   FOR EACH ROW
   WHEN ( new.iduser IS NULL )
BEGIN
   :new.iduser := user_iduser_seq.nextval;
END;
/
```

Przykładowe selekty

1.Wyświetlanie aktywności użytkowników z poprzednich 2 lat:

```
SELECT
   IDUSER,
   (SELECT COUNT(*) from USERLOG WHERE "date" BETWEEN TO_DATE('01/01/2020','DD/MM/YYYY') and
to_date('01/01/2021','DD/MM/YYYY') AND IDUSER = userlog.user_iduser ) "W 2020 ROKU",
   (SELECT COUNT(*) from USERLOG WHERE "date" BETWEEN TO_DATE('01/01/2021','DD/MM/YYYY') and
to_date('01/01/2022','DD/MM/YYYY') AND IDUSER = userlog.user_iduser ) "W 2021 ROKU"
FROM
   USERLOG,"USER"
WHERE
   USERLOG.USER_IDUSER = "USER".IDUSER
GROUP BY
   IDUSER;
```

IDUSER	W 2020 ROKU	W 2021 ROKU
0	2	5
1	0	1
2	0	1

2.Wybór konkretnego użytkownika i wyświetlenie jego wszystkich komentarzy

```
SELECT
   "USER"."NAME",
   MOVIE.TITLE,
   "COMMENT"."DESCRIPTION"
FROM
   "COMMENT",COMMENTMOVIE,MOVIE,"USER"
WHERE
   "USER".IDUSER = COMMENTMOVIE.USER_IDUSER
AND
   "COMMENT"."ID" = COMMENTMOVIE.COMMENT_ID
AND
   MOVIE."ID" = commentmovie.movie_id
AND
   UPPER("USER"."NAME") = UPPER('&SELECTED_NAME');
```

```
NAME
--------------------------------------------------------------------------------
TITLE
--------------------------------------------------------------------------------
DESCRIPTION
--------------------------------------------------------------------------------
maximus_
RICHARD JEWELL
To jest niepojete, ze dziadek Clint 89 lat nie zwalnia, ma tak ciekawe tematy i
kreci naprawde dobre jakosciowo filmy.
```

3.Wyświetla osoby, które otrzymały nagrody wybrane przez nas

```
SELECT
   PERSON."NAME",
   MOVIE.TITLE,
   AWARD.INFO
FROM
   AWARD,CREATOR,CREATORMOVIE,MOVIE,PERSON
WHERE
   AWARD.IDMOVIE = MOVIE."ID"
AND
   AWARD.IDPERSON = PERSON."ID"
AND
   creator.person_id = PERSON."ID"
AND
   creator.creatormovie_id = CREATORMOVIE."ID"
AND
   UPPER(CREATORMOVIE."JOB") = UPPER('&SELECT_JOB')
AND
   AWARD.IDPERSON IS NOT NULL;
```

```
NAME
--------------------------------------------------------------------------------
TITLE
--------------------------------------------------------------------------------
INFO
--------------------------------------------------------------------------------
Jamie Bell I
ROCKETMAN
Zloty Glob

Sam Rockwell
RICHARD JEWELL
Oscar

NAME
--------------------------------------------------------------------------------
TITLE
--------------------------------------------------------------------------------
INFO
--------------------------------------------------------------------------------

Sam Rockwell
RICHARD JEWELL
Zloty Glob
```

4.Wypisanie liczby komentarzy pod konkretnym tytułem filmu

```
SELECT
   COUNT(*)
FROM
   MOVIE,COMMENTMOVIE
WHERE
   MOVIE."ID" = COMMENTMOVIE.MOVIE_ID
AND
   UPPER(MOVIE.TITLE) = UPPER('&SELECT_TITLE');
```

```
  COUNT(*)
----------
         1
```

5.Wypisanie gatunków i tytułów filmów, które znajdują się w bazie:

```
CREATE OR REPLACE VIEW Movie_Genre as(
SELECT
   MOVIE.TITLE "Tytul",
   MOVIE."DESCRIPTION" "Opis",
   MOVIE."MARK" "Ocena",
   GENRE.GENRE "Gatunek"
FROM
   MOVIE,GENRE,MOVIEGENRE
WHERE
   MOVIEGENRE.MOVIE_ID = MOVIE."ID"
AND
   MOVIEGENRE.GENRE_ID = GENRE."ID");

SELECT * FROM MOVIE_GENRE;
```

```
SEBERG
Kiedy aktorka Jean Seberg angazuje sie w ruch na rzecz praw obywatelskich, FBI z
aczyna ja obserwowac.
         6 Film dramatyczny

WYSTRZALOWA MISS

Tytul
--------------------------------------------------------------------------------
Opis
--------------------------------------------------------------------------------
     Ocena Gatunek
---------- --------------------------------------------------
Makijazystka z Hollywood podrozuje do Tijuany, by pomoc przyjaciolce wygrac konk
urs pieknosci. Gdy ta, podczas strzelaniny w klubie, znika, Gloria staje sie zak
ladniczka przywodcy kartelu.
         6 Film dramatyczny

Killerman
Chcacy ustawic sie dwaj kumple zawieraja narkotykowa transakcje. Sprawy ida zle,

Tytul
--------------------------------------------------------------------------------
Opis
--------------------------------------------------------------------------------
     Ocena Gatunek
---------- --------------------------------------------------
 gdy zadzieraja ze skorumpowanymi policjantami.
         5 Film kryminalny
```

6.Wypisanie za co aktor otrzymał nagrodę – aktor jest wybierany przez nas

```
CREATE OR REPLACE VIEW PERSON_WIN AS(
SELECT
   PERSON."NAME",
   MOVIE.TITLE,
   AWARDGENRE."DESCRIPTION"
FROM
   AWARD,AWARDGENRE,MOVIE,PERSON
WHERE
   AWARD.IDGENRE = AWARDGENRE.IDGENRE
AND
   AWARD.IDPERSON = PERSON."ID"
AND
   AWARD.IDMOVIE = MOVIE."ID");
```

```
NAME
--------------------------------------------------------------------------------
TITLE
--------------------------------------------------------------------------------
DESCRIPTION
--------------------------------------------------------------------------------
Sam Rockwell
RICHARD JEWELL
Najlepsza aktorka drugoplanowa

Sam Rockwell
RICHARD JEWELL
Najlepsza aktorka drugoplanowa
```

Źródła danych

Wszystkie informacje o filmach aktorach oraz użytkownikach zostały pobrane z strony internetowej pod adresem: www.filmweb.pl.

Informacje o nagrodach(Oskarach, nominacjach) zostały pobrane z strony: https://www.aggdata.com/awards/oscar.