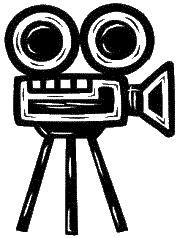
Imię i nazwisko: Jakub Kozubek

Kurs: Bazy danych gr.5

Rok akademicki: 2020/2021

Kierunek: Informatyka stosowana

Wydział: Wydział Nauk Ścisłych i Technicznych

Baza filmów

Aplikacja internetowa opierająca się na bazie filmów, aktorów i twórców

1. **Opis projektu**

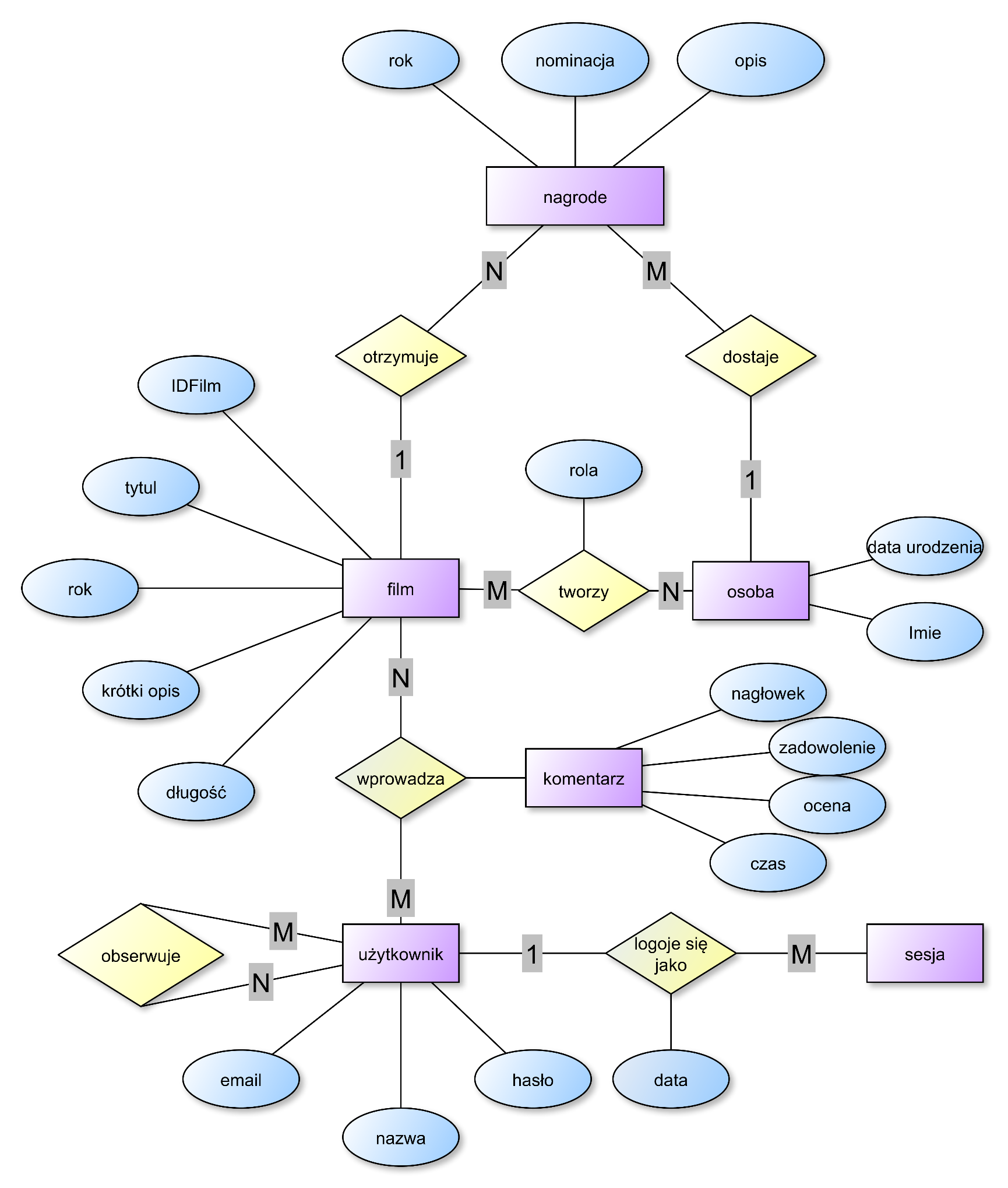
Aplikacja została przeznaczona na stronę internetową z dostępem dla każdego użytkownika urządzenia komputerowego lub smartfona posiadające aktywne połączenie z Internetem. Baza danych zawiera takie informacje jak: tytuł filmu, czas trwania filmu, krótki opis i inne. Baza posiada również informację o 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.

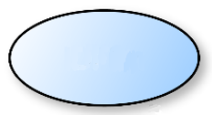
Użytkownicy strony internetowej nie posiadający jeszcze konta, mogą je utworzyć. Utworzenie konta polega na podaniu kilku informacji o sobie takich jak: imię, nazwisko, wiek, email, płeć oraz hasło do swojego konta.

Użytkownicy mają możliwość dodawania innych użytkowników do listy obserwowanych.

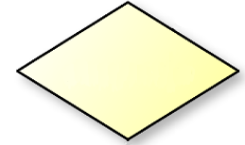
Baza prowadzi również informację o logowaniu się użytkowników, przechowuje informacje takie jak datę logowania.

Diagram związków encji





Opis



Kategorii

1

N

W

User\_idUser1

User\_idUser

date

User\_idUser

password

sex

age

email

name

idUser

id

mark

description

date

Comment\_idd

User\_idUser

Movie\_id

job

idGenre

Movie\_id

Person\_id

CreatorMovie\_id

photo

dateOfBirth

name

idPerson

description

idGenre

won

info

data

idGenre

idPerson

idMovie

genre

idGenre

Genre\_id

Movie\_id

poster

mark

length

description

relaseDate

Title

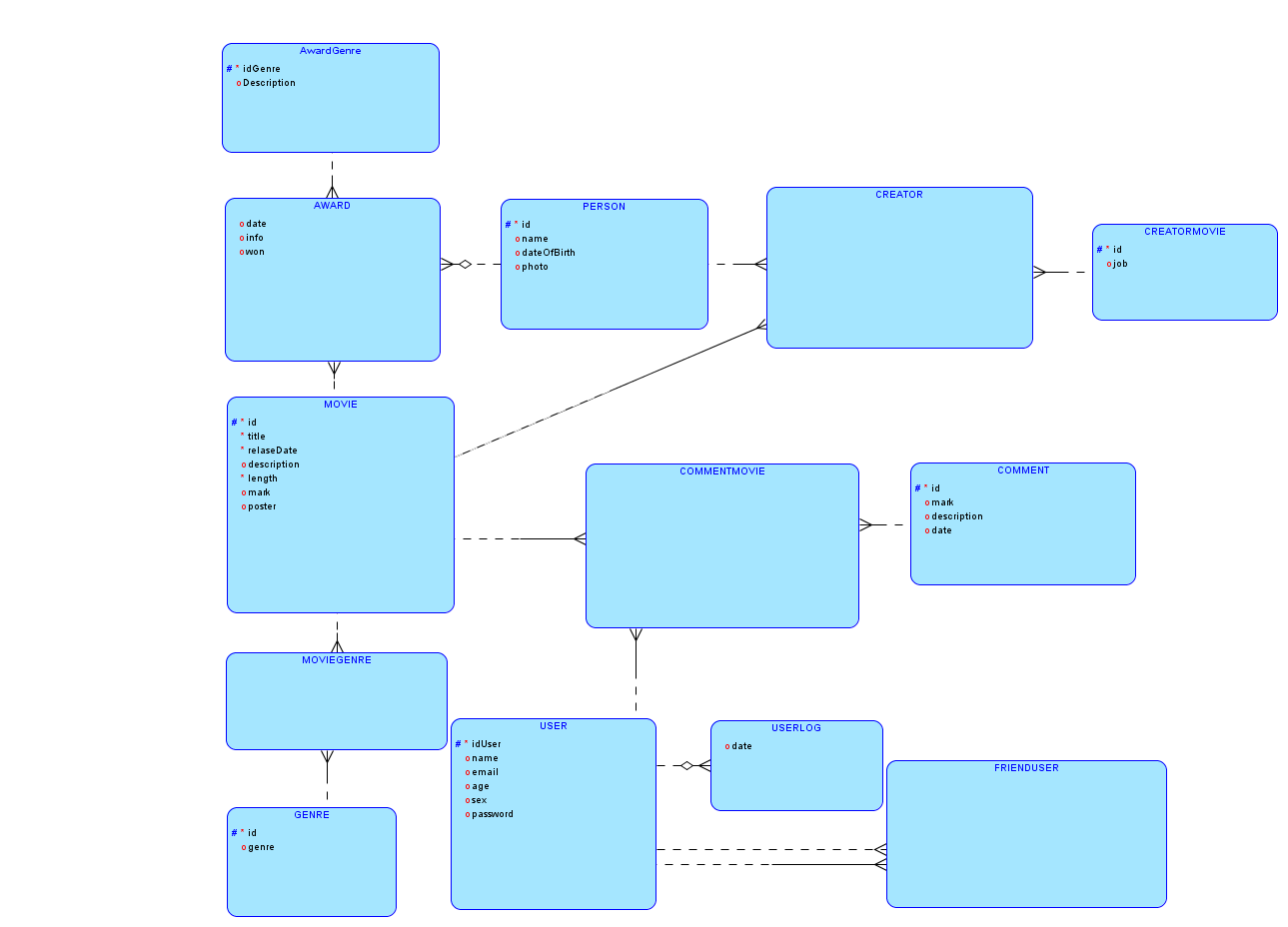
idMovie

1. Specyfikacje funkcjonalna:

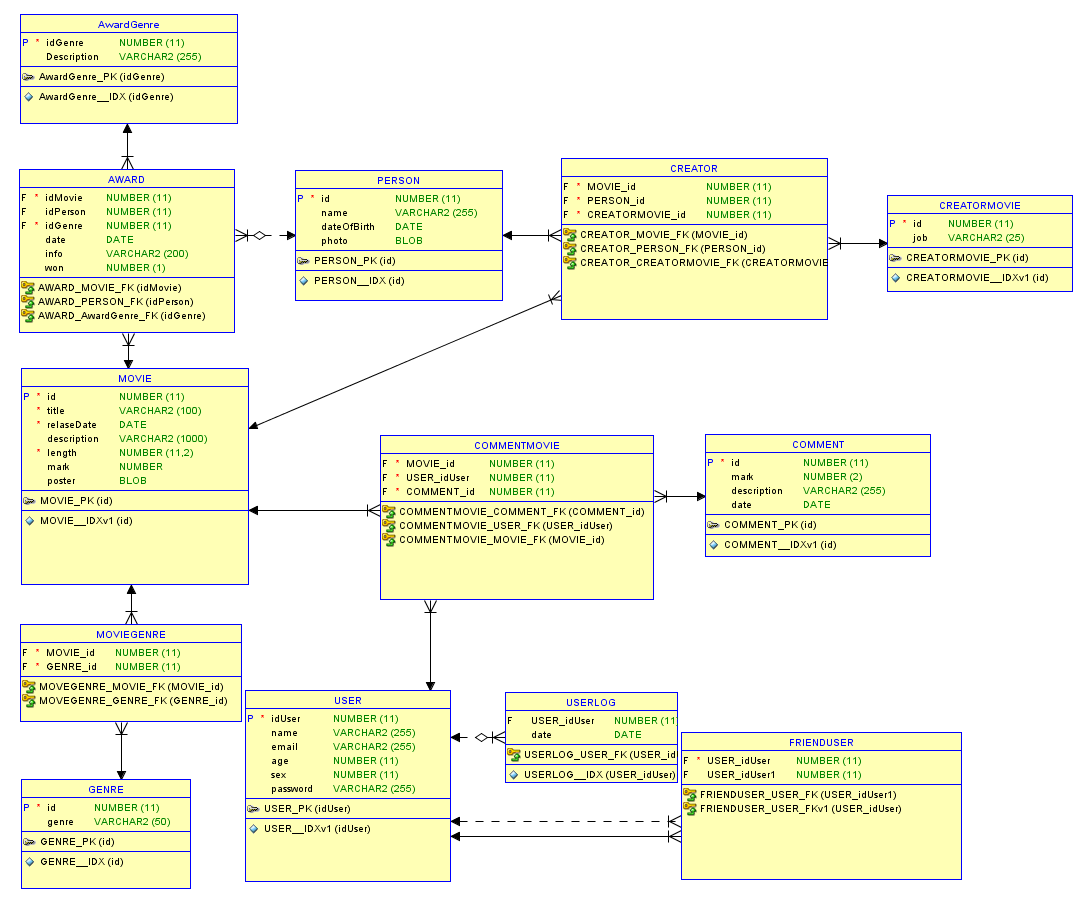
* Baza pozwala na wyświetlanie informacji o filmach, aktorach lub twórcach filmu z podstawowymi informacjami. Daje możliwość szybkiego wyszukania danego aktora, twórcy czy filmu
* Każdy użytkownik(zalogowany lub nie) ma dostęp do informacji o filmach, aktorach i twórcach filmu, oraz komentarzy zawartych 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.

1. 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

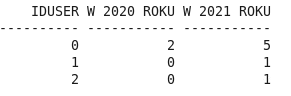
USERLOG,"USER"

WHERE

USERLOG.USER\_IDUSER = "USER".IDUSER

GROUP BY

IDUSER;



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

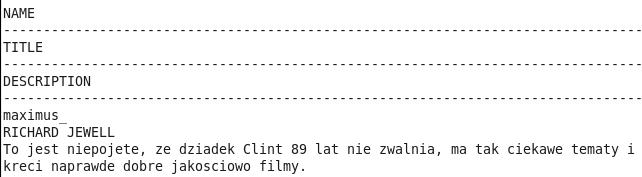
AND

"COMMENT"."ID" = COMMENTMOVIE.COMMENT\_ID

AND

MOVIE."ID" = commentmovie.movie\_id

AND

UPPER("USER"."NAME") = UPPER('&SELECTED\_NAME');

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;



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');

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

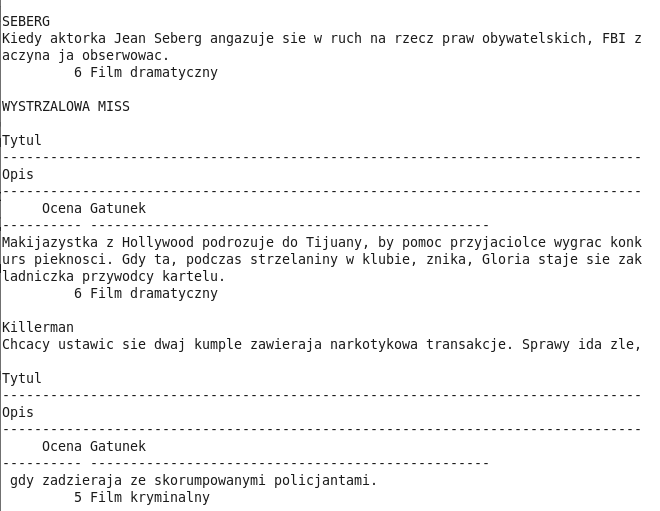
WHERE

MOVIEGENRE.MOVIE\_ID = MOVIE."ID"

AND

MOVIEGENRE.GENRE\_ID = GENRE."ID");

SELECT \* FROM MOVIE\_GENRE;



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

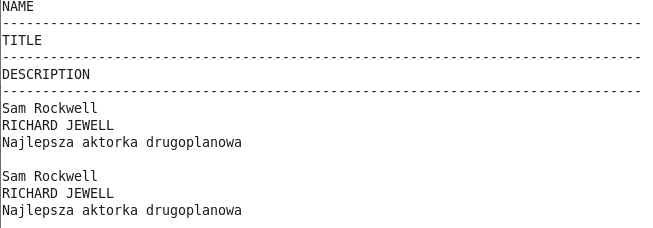
AWARD.IDGENRE = AWARDGENRE.IDGENRE

AND

AWARD.IDPERSON = PERSON."ID"

AND

AWARD.IDMOVIE = MOVIE."ID");



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.