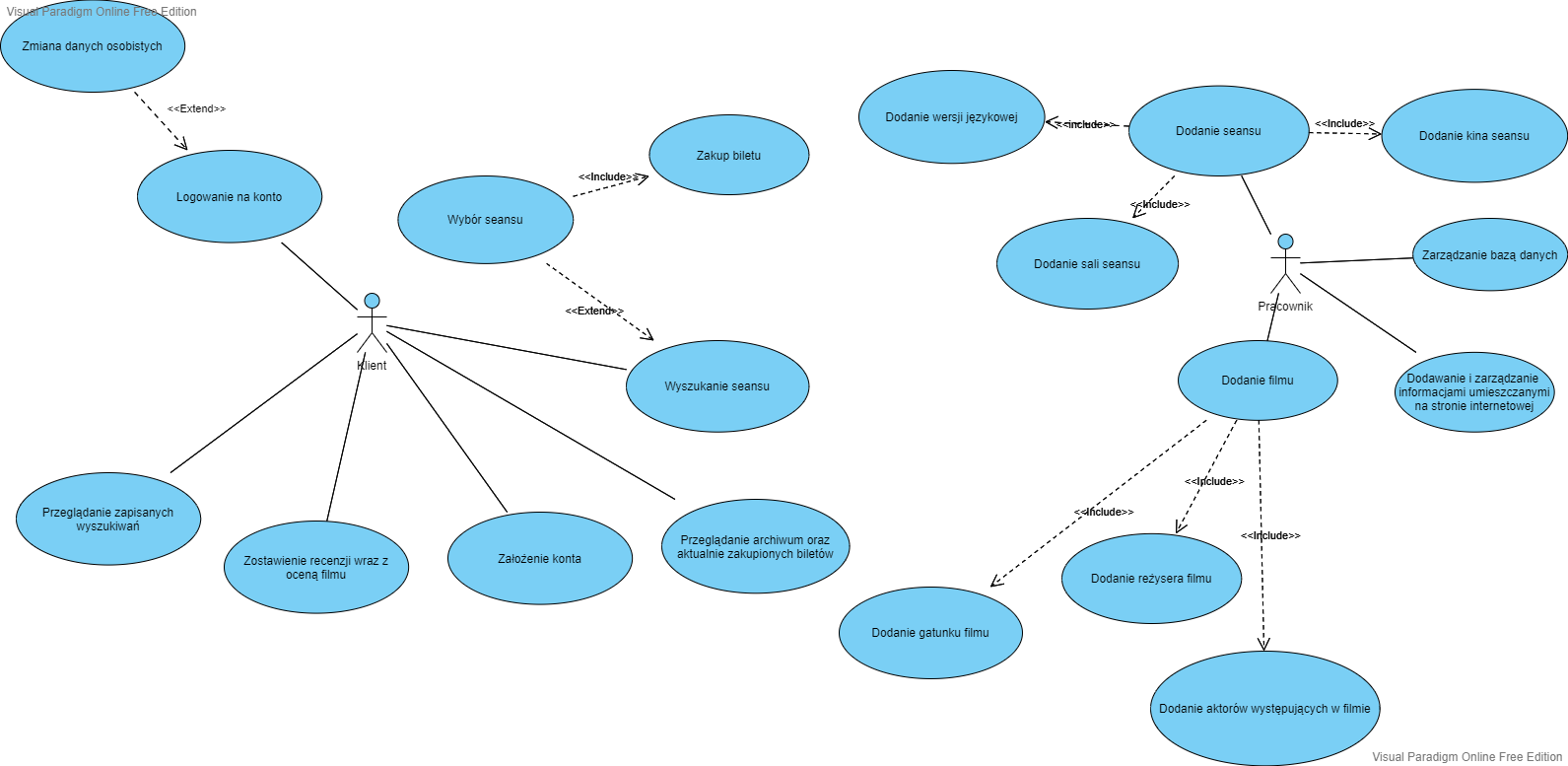
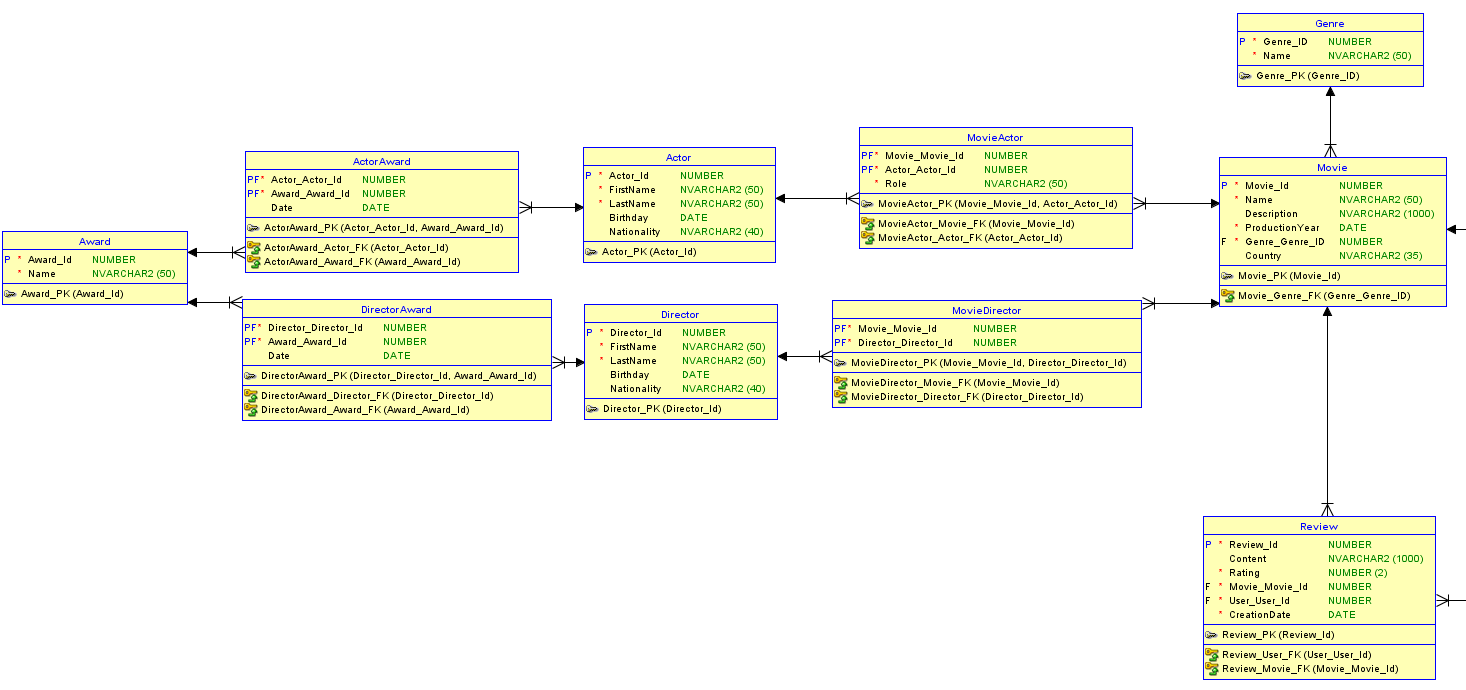
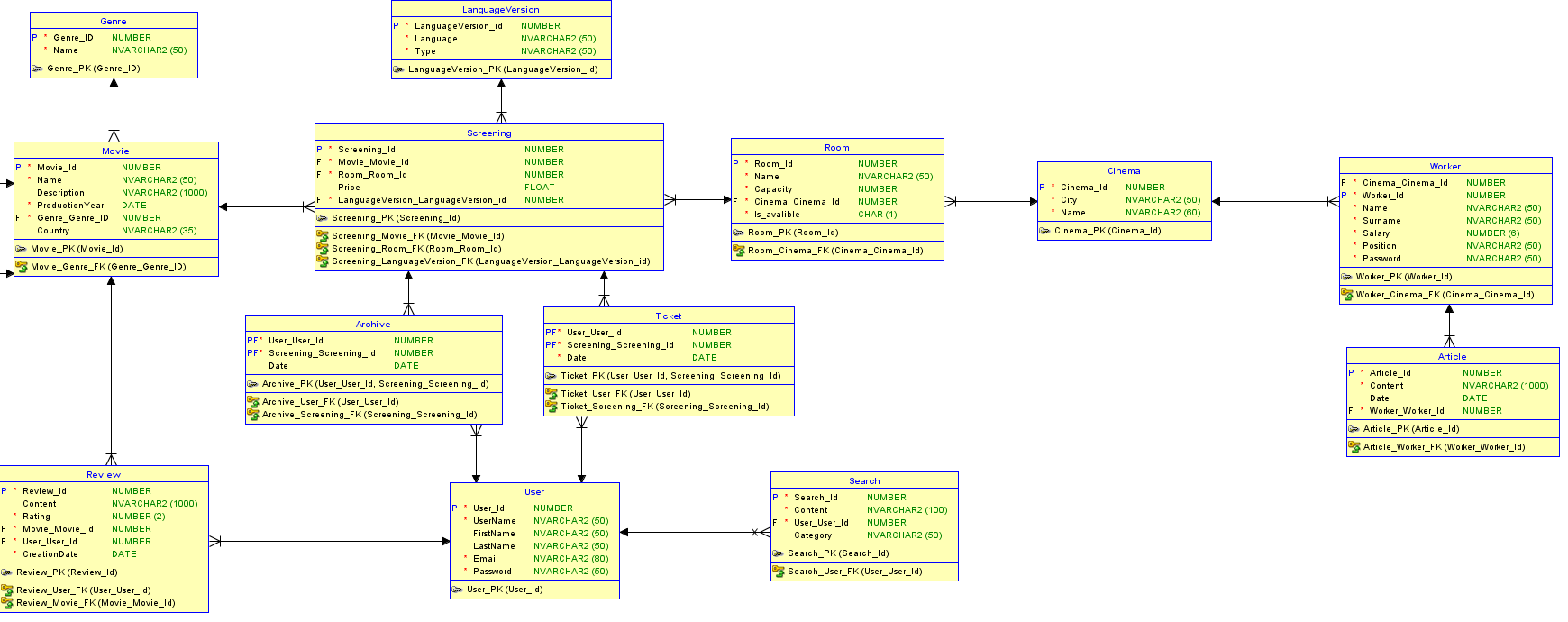
Opis Problemu

Schemat bazy danych będzie przedstawiał sposób przechowywania danych zarządzanych przez Kino. Z systemu będą mogli korzystać klienci oraz pracownicy co będzie odbywać się poprzez stronę internetową. System przechowuje dane w tabelach m.in. dane aktorów w tabeli *Actor*, użytkowników w tabeli *User,* czy filmów w tabeli *Movie*.

Diagram Przypadków

Diagram ER



Tabela Actor –

* Actor\_Id - klucz główny tabeli **Actor**, unikatowe ID(typ:liczba całkowita)
* FirstName – imię aktora, pole wymagane (typ:napis)
* LastName – nazwisko aktora, pole wymagane (typ:napis)
* Birthday – data urodzin aktora, pole niewymagane (typ:data)
* Nationality – narodowość aktora, pole niewymagane (typ:napis)

Tabela ActorAward –

* Actor\_Actor\_Id - klucz obcy z tabeli **Actor**, służy identyfikacji który aktor otrzymał nagrodę(typ:liczba całkowita)
* Award\_Award\_Id – klucz obcy z tabeli **Award**, służy identyfikacji którą nagrodę otrzymał aktor (typ:liczba całkowita)
* Date – data przyznania nagrody, pole niewymagane (typ:data)

Tabela Archive –

* User\_User\_Id - klucz obcy z tabeli **User**, służy identyfikacji użytkownika który obejrzał film(typ:liczba całkowita)
* Screening \_Screening\_Id – klucz obcy z tabeli **Screening**, służy identyfikacji obejrzanego filmu(typ:liczba całkowita)
* Date – data obejrzenia danego filmu przez użytkownika, pole niewymagane (typ:data)

Tabela Article –

* Article\_Id - klucz główny tabeli **Article**, unikatowe ID(typ:liczba całkowita)
* Content – opisu filmu, pole wymagane (typ:napis)
* Date – data napisania opisu, pole niewymagane (typ:data)
* Worker\_Worker\_Id - klucz obcy z tabeli **Worker**, służy identyfikacji pracownika który wykonał opis (typ:liczba całkowita)

Tabela Award –

* Award\_Id - klucz główny tabeli **Award**, unikatowe ID(typ:liczba całkowita)
* Name – nazwa nagrody, pole wymagane (typ:napis)

Tabela Cinema –

* Cinema\_Id - klucz główny tabeli **Cinema**, unikatowe ID(typ:liczba całkowita)
* City – w którym mieście znajduje się kino, pole wymagane (typ:napis)
* Name – nazwa kina, pole wymagane (typ:napis)

Tabela Director –

* Director\_Id - klucz główny tabeli **Director**, unikatowe ID(typ:liczba całkowita)
* FirstName – imię reżysera, pole wymagane (typ:napis)
* LastName – nazwisko reżysera, pole wymagane (typ:napis)
* Birthday – data urodzin reżysera, pole niewymagane (typ:data)
* Nationality – narodowość reżysera, pole niewymagane (typ:napis)

Tabela DirectorAward –

* Director\_Director\_Id - klucz obcy z tabeli **Director**, służy identyfikacji który reżyser otrzymał nagrodę(typ:liczba całkowita)
* Award\_Award\_Id – klucz obcy z tabeli **Award**, służy identyfikacji którą nagrodę otrzymał reżyser (typ:liczba całkowita)
* Date – data przyznania nagrody, pole niewymagane (typ:data)

Tabela Genre –

* Genre\_Id - klucz główny tabeli **Genre**, unikatowe ID(typ:liczba całkowita)
* Name – nazwa gatunku filmu, pole wymagane (typ:napis)

Tabela LanguageVersion –

* LanguageVersion\_Id - klucz główny tabeli **LanguageVersion**, unikatowe ID(typ:liczba całkowita)
* Name – język wersji językowej, pole wymagane (typ:napis)
* Type – rodzaj wersji, pole wymagane (typ:napis)

Tabela Movie –

* Movie\_Id - klucz główny tabeli **Movie**, unikatowe ID(typ:liczba całkowita)
* Name – nazwa filmu, pole wymagane (typ:napis)
* Description – opis filmu, pole niewymagane (typ:napis)
* ProductionYear – rok wyprodukowania filmu, pole wymagane (typ:data)
* Genre\_Genre\_Id – klucz obcy z tabeli **Genre**, służy identyfikacji do którego gatunku należy film (typ:liczba całkowita)
* Country – w jakim kraju film został wyprodukowany, pole niewymagane (typ:napis)

Tabela MovieActor –

* Movie\_Movie\_Id - klucz obcy z tabeli **Movie**, służy identyfikacji w jakim filmie grał aktor (typ:liczba całkowita)
* Actor\_Actor\_Id – klucz obcy z tabeli **Actor**, służy identyfikacji który aktor grał w filmie (typ:liczba całkowita)
* Role – rola grana przez aktora, pole wymagane (typ:napis)

Tabela MovieDirector –

* Movie\_Movie\_Id - klucz obcy z tabeli **Movie**, służy identyfikacji w jaki film stworzył reżyser (typ:liczba całkowita)
* Director\_Director\_Id – klucz obcy z tabeli **Director**, służy identyfikacji który reżyser stworzył film (typ:liczba całkowita)

Tabela Review –

* Review\_Id - klucz główny tabeli **Review**, unikatowe ID(typ:liczba całkowita)
* Content – treść recenzji, pole niewymagane (typ:napis)
* Rating – ocena filmu, pole wymagane (typ: liczba całkowita)
* Movie\_Movie\_Id – klucz obcy z tabeli **Movie**, służy identyfikacji który film został oceniony (typ:liczba całkowita)
* User\_User\_Id – klucz obcy z tabeli **User**, służy identyfikacji który użytkownik wystawił recenzję (typ:liczba całkowita)
* CreationDate – data wystawienia recenzji, pole wymagane (typ:data)

Tabela Room –

* Room\_Id - klucz główny tabeli **Room**, unikatowe ID(typ:liczba całkowita)
* Content – nazwa pomieszczenia, pole wymagane (typ:napis)
* Capacity – pojemność pomieszczenia, pole wymagane (typ: liczba całkowita)
* Cinema\_Cinema\_Id – klucz obcy z tabeli **Cinema**, służy identyfikacji w którym kinie znajduje się pomieszczenie (typ:liczba całkowita)
* Is\_avaliable – czy dany pokój nie został już zarezerwowany, pole wymagane (typ: znak)

Tabela Screening –

* Screening\_Id - klucz główny tabeli **Screening**, unikatowe ID(typ:liczba całkowita)
* Movie\_Movie\_Id – klucz obcy z tabeli **Movie**, służy identyfikacji który film zostanie wyświetlony (typ:liczba całkowita)
* Room\_Room\_Id – klucz obcy z tabeli **Room**, służy identyfikacji w którym pomieszczeniu odbędzie się seans (typ:liczba całkowita)
* Price – koszt zakupu wejścia na seans, pole niewymagane (typ: liczba zmiennoprzecinkowa)
* LanguageVersion\_LanguageVersion\_Id – klucz obcy z tabeli **LanguageVersion**, służy identyfikacji wersji językowej użytej w czasie seansu (typ:liczba całkowita)

Tabela Search –

* Search\_Id - klucz główny tabeli **Search**, unikatowe ID(typ:liczba całkowita)
* Content – wyszukiwana treść, pole wymagane (typ:napis)
* User\_User\_Id – klucz obcy z tabeli **User**, który użytkownik wyszukuje (typ:liczba całkowita)
* Category – kategoria w zakresie której prowadzone jest wyszukiwanie, pole niewymagane (typ: liczba całkowita)

Tabela Ticket –

* Ticket\_Ticket\_Id - klucz obcy z tabeli **Ticket**, służy identyfikacji użytkownika który kupił bilet(typ:liczba całkowita)
* Screening\_Screening\_Id – klucz obcy z tabeli **Screening**, służy identyfikacji na jaki film dokonano zakupu (typ:liczba całkowita)
* Date – data zakupu, pole wymagane (typ:data)

Tabela User –

* User\_Id - klucz główny tabeli **User**, unikatowe ID(typ:liczba całkowita)
* User Name – nazwa użytkownika, pole wymagane (typ:napis)
* FirstName – imię użytkownika, pole niewymagane (typ:napis)
* LastName – nazwisko użytkownika, pole niewymagane (typ:napis)
* Email – adres email użytkownika, pole wymagane (typ:napis)
* Password – hasło użytkownika, pole wymagane (typ:napis)

Tabela Worker –

* Cinema\_Cinema\_Id – klucz obcy z tabeli **Cinema**, służy identyfikacji w którym kinie pracuje dana osoba (typ:liczba całkowita)
* Worker\_Id - klucz główny tabeli **Worker**, unikatowe ID(typ:liczba całkowita)
* Name – imie pracownika, pole wymagane (typ:napis)
* Surname – nazwisko pracownika, pole wymagane (typ:napis)
* Salary – pensja pracownika, pole wymagane (typ:liczba całkowita)
* Position – stanowisko pracy pracownika, pole wymagane (typ:napis)
* Email – hasło pracownika, pole wymagane (typ:napis)

Implementacja bazy danych

Wybrany został serwer bazy danych Oracle

Skrypt tworzący bazę danych:

CREATE TABLE actor(

actor\_id NUMBER NOT NULL,

firstname NVARCHAR2(50) NOT NULL,

lastname NVARCHAR2(50) NOT NULL,

birthday DATE,

nationality NVARCHAR2(40)

);

ALTER TABLE actor ADD CONSTRAINT actor\_pk PRIMARY KEY ( actor\_id );

CREATE TABLE actoraward (

actor\_actor\_id NUMBER NOT NULL,

award\_award\_id NUMBER NOT NULL,

"Date" DATE

);

ALTER TABLE actoraward ADD CONSTRAINT actoraward\_pk PRIMARY KEY ( actor\_actor\_id,

award\_award\_id );

CREATE TABLE archive (

user\_user\_id NUMBER NOT NULL,

screening\_screening\_id NUMBER NOT NULL,

"Date" DATE

);

ALTER TABLE archive ADD CONSTRAINT archive\_pk PRIMARY KEY ( user\_user\_id,

screening\_screening\_id );

CREATE TABLE article (

article\_id NUMBER NOT NULL,

content NVARCHAR2(1000) NOT NULL,

"Date" DATE,

worker\_worker\_id NUMBER NOT NULL

);

ALTER TABLE article ADD CONSTRAINT article\_pk PRIMARY KEY ( article\_id );

CREATE TABLE award (

award\_id NUMBER NOT NULL,

name NVARCHAR2(50) NOT NULL

);

ALTER TABLE award ADD CONSTRAINT award\_pk PRIMARY KEY ( award\_id );

CREATE TABLE cinema (

cinema\_id NUMBER NOT NULL,

city NVARCHAR2(50) NOT NULL,

name NVARCHAR2(60) NOT NULL

);

ALTER TABLE cinema ADD CONSTRAINT cinema\_pk PRIMARY KEY ( cinema\_id );

CREATE TABLE director (

director\_id NUMBER NOT NULL,

firstname NVARCHAR2(50) NOT NULL,

lastname NVARCHAR2(50) NOT NULL,

birthday DATE,

nationality NVARCHAR2(40)

);

ALTER TABLE director ADD CONSTRAINT director\_pk PRIMARY KEY ( director\_id );

CREATE TABLE directoraward (

director\_director\_id NUMBER NOT NULL,

award\_award\_id NUMBER NOT NULL,

"Date" DATE

);

ALTER TABLE directoraward ADD CONSTRAINT directoraward\_pk PRIMARY KEY ( director\_director\_id,

award\_award\_id );

CREATE TABLE genre (

genre\_id NUMBER NOT NULL,

name NVARCHAR2(50) NOT NULL

);

ALTER TABLE genre ADD CONSTRAINT genre\_pk PRIMARY KEY ( genre\_id );

CREATE TABLE languageversion (

lv\_id NUMBER NOT NULL,

language NVARCHAR2(50) NOT NULL,

type NVARCHAR2(50) NOT NULL

);

ALTER TABLE languageversion ADD CONSTRAINT languageversion\_pk PRIMARY KEY ( lv\_id );

CREATE TABLE movie (

movie\_id NUMBER NOT NULL,

name NVARCHAR2(50) NOT NULL,

description NVARCHAR2(1000),

productionyear DATE NOT NULL,

genre\_genre\_id NUMBER NOT NULL,

country NVARCHAR2(35)

);

ALTER TABLE movie ADD CONSTRAINT movie\_pk PRIMARY KEY ( movie\_id );

CREATE TABLE movieactor (

movie\_movie\_id NUMBER NOT NULL,

actor\_actor\_id NUMBER NOT NULL,

role NVARCHAR2(50) NOT NULL

);

ALTER TABLE movieactor ADD CONSTRAINT movieactor\_pk PRIMARY KEY ( movie\_movie\_id,

actor\_actor\_id );

CREATE TABLE moviedirector (

movie\_movie\_id NUMBER NOT NULL,

director\_director\_id NUMBER NOT NULL

);

ALTER TABLE moviedirector ADD CONSTRAINT moviedirector\_pk PRIMARY KEY ( movie\_movie\_id,

director\_director\_id );

CREATE TABLE review (

review\_id NUMBER NOT NULL,

content NVARCHAR2(1000),

rating NUMBER(2) NOT NULL,

movie\_movie\_id NUMBER NOT NULL,

user\_user\_id NUMBER NOT NULL,

creationdate DATE NOT NULL

);

ALTER TABLE review ADD CONSTRAINT review\_pk PRIMARY KEY ( review\_id );

CREATE TABLE room (

room\_id NUMBER NOT NULL,

name NVARCHAR2(50) NOT NULL,

capacity NUMBER NOT NULL,

cinema\_cinema\_id NUMBER NOT NULL,

is\_avalible CHAR(1) NOT NULL

);

ALTER TABLE room ADD CONSTRAINT room\_pk PRIMARY KEY ( room\_id );

CREATE TABLE screening (

screening\_id NUMBER NOT NULL,

movie\_movie\_id NUMBER NOT NULL,

room\_room\_id NUMBER NOT NULL,

price FLOAT,

languageversion\_lv\_id NUMBER NOT NULL

);

ALTER TABLE screening ADD CONSTRAINT screening\_pk PRIMARY KEY ( screening\_id );

CREATE TABLE search (

search\_id NUMBER NOT NULL,

content NVARCHAR2(100) NOT NULL,

user\_user\_id NUMBER NOT NULL,

category NVARCHAR2(50)

);

ALTER TABLE search ADD CONSTRAINT search\_pk PRIMARY KEY ( search\_id );

CREATE TABLE ticket (

user\_user\_id NUMBER NOT NULL,

screening\_screening\_id NUMBER NOT NULL,

"Date" DATE NOT NULL

);

ALTER TABLE ticket ADD CONSTRAINT ticket\_pk PRIMARY KEY ( user\_user\_id,

screening\_screening\_id );

CREATE TABLE "User" (

user\_id NUMBER NOT NULL,

username NVARCHAR2(50) NOT NULL,

firstname NVARCHAR2(50),

lastname NVARCHAR2(50),

email NVARCHAR2(80) NOT NULL,

password NVARCHAR2(50) NOT NULL

);

ALTER TABLE "User" ADD CONSTRAINT user\_pk PRIMARY KEY ( user\_id );

CREATE TABLE worker (

cinema\_cinema\_id NUMBER NOT NULL,

worker\_id NUMBER NOT NULL,

name NVARCHAR2(50) NOT NULL,

surname NVARCHAR2(50) NOT NULL,

salary NUMBER(6) NOT NULL,

position NVARCHAR2(50) NOT NULL,

password NVARCHAR2(50) NOT NULL

);

ALTER TABLE worker ADD CONSTRAINT worker\_pk PRIMARY KEY ( worker\_id );

ALTER TABLE actoraward

ADD CONSTRAINT actoraward\_actor\_fk FOREIGN KEY ( actor\_actor\_id )

REFERENCES actor ( actor\_id );

ALTER TABLE actoraward

ADD CONSTRAINT actoraward\_award\_fk FOREIGN KEY ( award\_award\_id )

REFERENCES award ( award\_id );

ALTER TABLE archive

ADD CONSTRAINT archive\_screening\_fk FOREIGN KEY ( screening\_screening\_id )

REFERENCES screening ( screening\_id );

ALTER TABLE archive

ADD CONSTRAINT archive\_user\_fk FOREIGN KEY ( user\_user\_id )

REFERENCES "User" ( user\_id );

ALTER TABLE article

ADD CONSTRAINT article\_worker\_fk FOREIGN KEY ( worker\_worker\_id )

REFERENCES worker ( worker\_id );

ALTER TABLE directoraward

ADD CONSTRAINT directoraward\_award\_fk FOREIGN KEY ( award\_award\_id )

REFERENCES award ( award\_id );

ALTER TABLE directoraward

ADD CONSTRAINT directoraward\_director\_fk FOREIGN KEY ( director\_director\_id )

REFERENCES director ( director\_id );

ALTER TABLE movie

ADD CONSTRAINT movie\_genre\_fk FOREIGN KEY ( genre\_genre\_id )

REFERENCES genre ( genre\_id );

ALTER TABLE movieactor

ADD CONSTRAINT movieactor\_actor\_fk FOREIGN KEY ( actor\_actor\_id )

REFERENCES actor ( actor\_id );

ALTER TABLE movieactor

ADD CONSTRAINT movieactor\_movie\_fk FOREIGN KEY ( movie\_movie\_id )

REFERENCES movie ( movie\_id );

ALTER TABLE moviedirector

ADD CONSTRAINT moviedirector\_director\_fk FOREIGN KEY ( director\_director\_id )

REFERENCES director ( director\_id );

ALTER TABLE moviedirector

ADD CONSTRAINT moviedirector\_movie\_fk FOREIGN KEY ( movie\_movie\_id )

REFERENCES movie ( movie\_id );

ALTER TABLE review

ADD CONSTRAINT review\_movie\_fk FOREIGN KEY ( movie\_movie\_id )

REFERENCES movie ( movie\_id );

ALTER TABLE review

ADD CONSTRAINT review\_user\_fk FOREIGN KEY ( user\_user\_id )

REFERENCES "User" ( user\_id );

ALTER TABLE room

ADD CONSTRAINT room\_cinema\_fk FOREIGN KEY ( cinema\_cinema\_id )

REFERENCES cinema ( cinema\_id );

ALTER TABLE screening

ADD CONSTRAINT screening\_languageversion\_fk FOREIGN KEY ( languageversion\_lv\_id )

REFERENCES languageversion ( lv\_id );

ALTER TABLE screening

ADD CONSTRAINT screening\_movie\_fk FOREIGN KEY ( movie\_movie\_id )

REFERENCES movie ( movie\_id );

ALTER TABLE screening

ADD CONSTRAINT screening\_room\_fk FOREIGN KEY ( room\_room\_id )

REFERENCES room ( room\_id );

ALTER TABLE search

ADD CONSTRAINT search\_user\_fk FOREIGN KEY ( user\_user\_id )

REFERENCES "User" ( user\_id )

ON DELETE CASCADE;

ALTER TABLE ticket

ADD CONSTRAINT ticket\_screening\_fk FOREIGN KEY ( screening\_screening\_id )

REFERENCES screening ( screening\_id );

ALTER TABLE ticket

ADD CONSTRAINT ticket\_user\_fk FOREIGN KEY ( user\_user\_id )

REFERENCES "User" ( user\_id );

ALTER TABLE worker

ADD CONSTRAINT worker\_cinema\_fk FOREIGN KEY ( cinema\_cinema\_id )

REFERENCES cinema ( cinema\_id );

Mapowanie klas