

POLITECHNIKA KRAKOWSKA im. T. KOŚCIUSZKI

WYDZIAŁ INŻYNIERII ELEKTRYCZNEJ I KOMPUTEROWEJ

Projekt z przedmiotu:

**Systemy Baz Danych**

**Ciepiela Adrian**

**Góra Patryk**

**Iwanowicz Michał**

**Semestr 5**

**Rok akademicki 2016/2017**

Grupa 31i

**Projekt nadzorował:**

Mgr Krzysztof Czajkowski

Kraków, styczeń 2017

1. **Założenia projektowe**

Głównym celem projektu jest utworzenie bazy danych opartej o technologię relacyjnych baz danych firmy Oracle. Język użyty do implementacji bazy danych to OracleSQL oraz PL/SQL. Utwożona baza danych będzie wystarczająca aby umożliwić skuteczną pracę aplikacji do obsługi sieci kin, rozwinąć ją o dodatkowe funkcjonalności, dane. Baza danych zawierać będzie informacje potrzebne do pracy kasjera jak również menadżera kina. Produkt będzie także służył jako archiwum danych, które może być wykorzystane do rozszerzenia możliwości bazy.

Do implementacji bazy danych dodawany jest opis, który pomaga używać bazę, zrozumieć jej strukturę, a także umożliwić jej dalsze rozwijanie przez programistów.

Dokumentacja przystosowana jest do użytku zarówno przez programistów jak   
i użytkowników.

Baza przechowywać będzie informacje na temat:

* oddziałów kina,
* zatrudnionych pracowników,
* klientów,
* poszczególnych sal w kinie,
* obecnie granych filmów,
* archiwum granych filmów,
* rodzajów biletów,
* sprzedanych biletów.

Dzięki bazie aplikacja będzie w stanie oferować następnujące funkcje:

Funkcje obsługiwane przez kasjera:

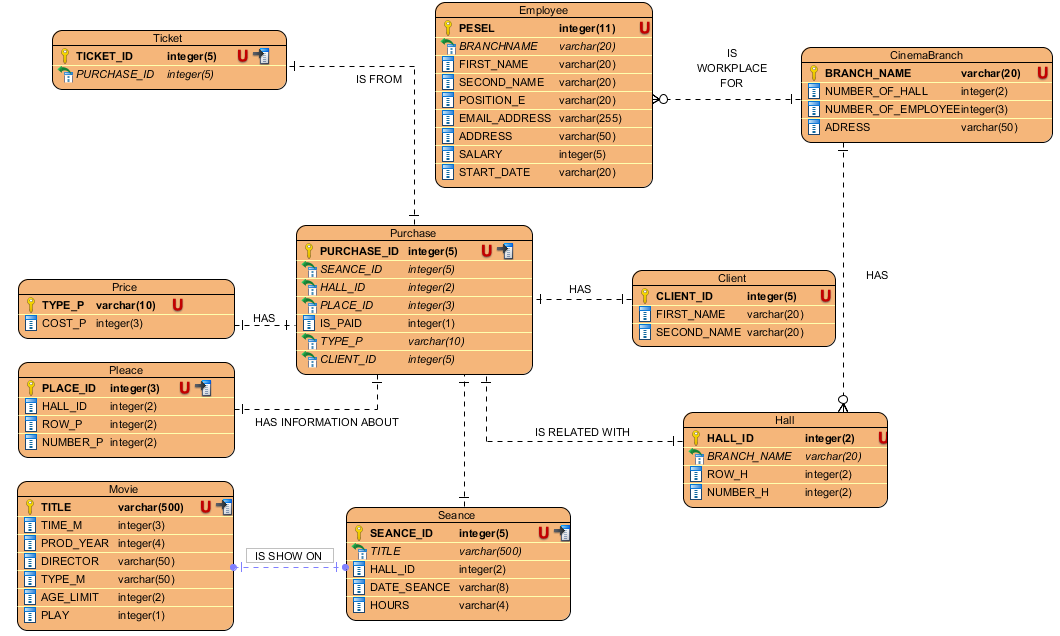
* wydawanie biletów,
* zwrot biletów,
* wymiana biletu,
* rezerwacja miejsc,
* wyświetlanie informacji o cenach i zniżkach,
* wyświetlanie repertuaru.

Funkcje obsługiwane przez menadżera:

* zatrudnianie pracowników,
* zwalnianie pracowników,
* dodawanie filmów do bazy,
* wprowadzanie filmów do repertuaru,
* usuwanie filmów z repertuaru,
* ustalanie w jakich salach będzie wyświetlany dany film,
* modyfikacja cen biletów,
* wyświetlanie informacji o pracownikach.

Na kluczowe elementy bazy – takie które mogą zaważać na szybkości działania aplikacji nałożone będą indeksy. Te specjalne struktury danych umożliwiają zwiększenie prędkości wykonywania operacji na tabelach w których zostały użyte. W przypadku tej bazy indeksy założone zostaną na następujące elementy:

* TICKET\_ID,
* SEANCE\_ID,
* MOVIE\_ID,
* PURCHASE\_ID,
* PLACE\_ID.

1. Diagram ERD
2. Opis relacji, diagram

**Ticket:**

Relacja ta jest odzwierciedleniem biletu w kinie.

*TICKET\_ID* – samogenerujący się, unikalny numer biletu,klucz główny, założony jest na nim indeks

*PURCHASE\_ID* – klucz obcy, unikalny numer zakupu, realizuje powiązanie pomiędzy biletem i zakupem

**Employee:**

Relacja ta jest odzwierciedleniem pracownika kina.

*PESEL* – pesel pracownika, unikalny klucz główny

*BRANCHNAME* – klucz obcy, nazwa oddziału kina, w którym pracuje dany pracownik

*FIRST\_NAME* – imię pracownika

*SECOND\_NAME* – nazwisko pracownika

*POSITION\_E* – stanowisko na którym zatrudniony jest pracownik

*EMAIL\_ADDRESS* – adres email pracownika

*ADDRESS* – adres zamieszkania pracownika

*SALARY* – wynagrodzenie pracownika

*STARD\_DATE* – data zatrudnienia pracownika

**Client:**

Relacja reprezentuje klienta kina.

*CLIENT\_ID* – klucz główny, unikalny numer klienta

*FIRST\_NAME* – imię klienta

*SECOND\_NAME* – nazwisko klienta

**Purchase:**

Relacja reprezentuje zarezerwowany lub kupiony bilet.

*PURCHASE\_ID* – klucz główny, unikalny numer zakupu  
*SEANCE\_ID* – klucz obcy, numer seansu  
*HALL\_ID* – klucz obcy, numer sali  
*PLACE\_ID* – klucz obcy, numer miejsca w Sali

*IS\_PAID* – pole określające czy została już uiszczona opłata za bilet  
*TYPE\_P* – klucz obcy, rodzaj ceny (np. ulgowy, normalny)  
*CLIENT\_ID* – klucz obcy, numer klienta

**Price:**

Relacja opisuje rodzaje ceny (np. ulgowy, normalny).

*TYPE\_P* – klucz główny, rodzaj ceny

*COST\_P* – koszt danego rodzaju biletu

**CinemaBranch:**

Tabela ta opisuje oddział kina.

*BRANCH\_NAME* – nazwa oddziału kina, jest unikalnym kluczem głównym

*NUMBER\_OF\_HALL* – ilość sal w oddziale

*NUMBER\_OD\_EMPLOYEE* –ilość pracowników pracujących w danym oddziale

*ADRESS* – dokładny adres oddziału

**Hall:**

Zawiera informacje o poszczególnych salach w kinie.

*HALL\_ID* – unikalny numer sali w całej sieci kin, jest kluczem głównym tabeli

*BRANCH\_NAME* – klucz obcy pokazujący, w którym oddziale banku znajduje się sala

*ROW\_H* – ilość rzędów w sali

*NUMBER\_H* – ilość miejsc w rzędzie

**Seance:**

Relacja zawierające informacje o seansach.

*SEANCE\_ID* – numer seansu, unikalny klucz główny, na którym założony jest indeks

*TITLE* – tytuł filmu wyświetlanego podczas seansu

*HALL\_ID* –numer sali przypisany do seansu

*DATE\_SEANCE* – data seansu

*HOURS* – godzina seansu

**Movie:**

Relacja reprezentująca film, dane o nim

*TITLE* – tytuł filmu, który jest jego kluczem głównym i założony jest na niego indeks

*TIME\_M* – czas trwania filmu

*PROD\_YEAR* – rok produkcji filmu

*DIRECTOR* – reżyser

*TYPE\_M* – gatunek filmowy

*AGE\_LIMIT* – ograniczenie wiekowe

*PLAY* - informacja o tym, czy film jest aktualnie grany czy nie (1 – grany, 0 - nie)

**Place:**

Tabela odpowiadająca za opis każdego miejsca w kinie

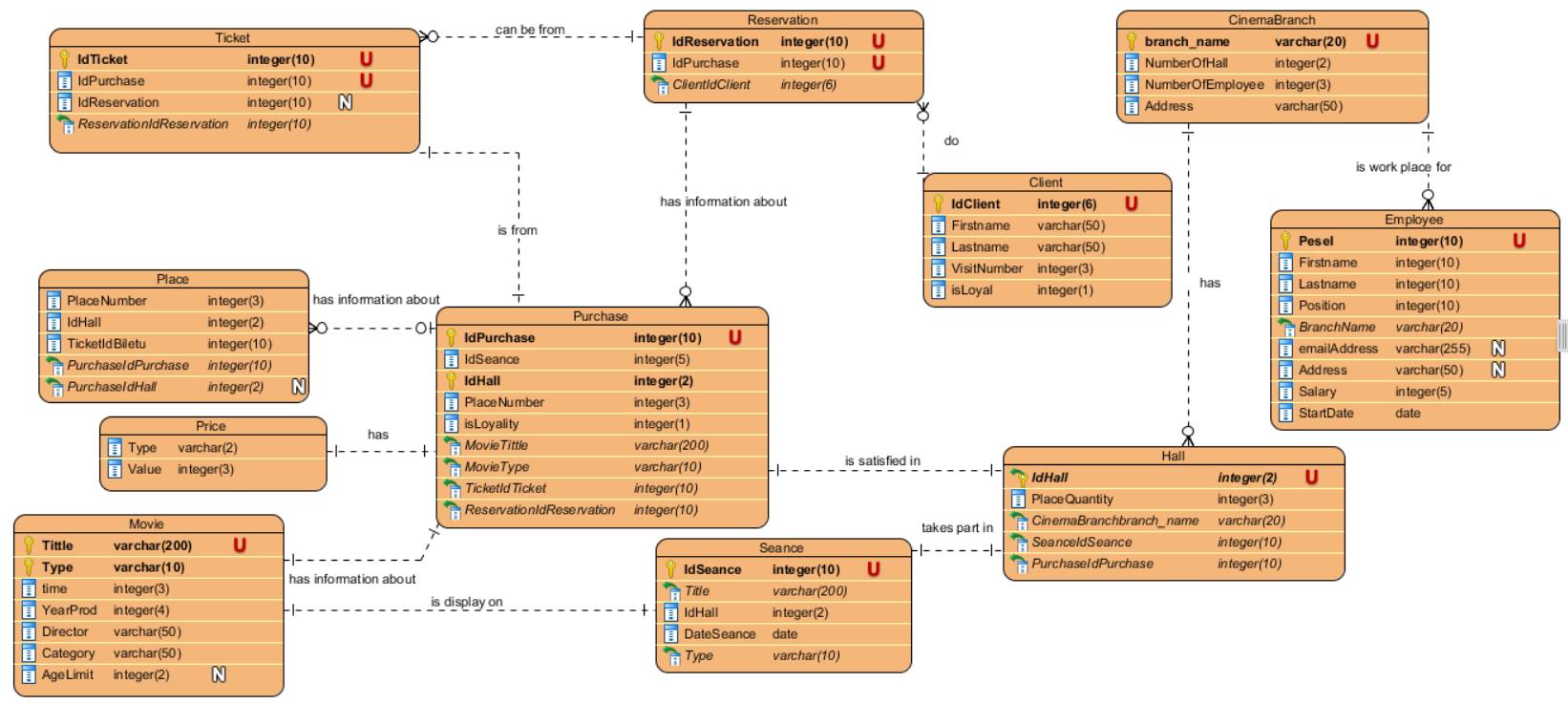
*PLACE\_ID* – numer krzesła/miejsca w kinie, unikalny klucz główny

*HALL\_ID* – numer sali, w której jest krzesło

*ROW\_P* – rząd, w którym jest krzesło

*NUMBER\_P* – miejsce w rzędzie, w którym jest krzesło

1. Wcześniejsze wersje projektu



Schemat przedstawia wcześniejszą wersję bazy, która zawierała kilka nieprzemyślanych do końca rozwiązań:

* informacje się zapętlały,
* relacja RESERVATION była niepotrzebna,
* w relacji PURCHASE i TICKET były nie potrzebne dane.

Schemat ten był za mało zoptymalizowany, przez co w procedurach pojawiały się zapętlenia dostępu do danych.

Na tym etapie chcieliśmy tworzyć funkcje naprzemiennie z procedurami, jednak stwierdziliśmy, że dla naszego problemu procedury są w pełni wystarczające.

W obecnym projekcie procedury tworzone są łatwiej, bez niepotrzebnych zawiłości, przez co są wykonywane szybciej i bardziej intuicyjne.

Na tym etapie dane były zadeklarowane jako elementarne, tak aby spełniona była pierwsza postać normalna.

Tablea PURCHASE nie była zgodna z 2 postacią normalną ponieważ dane np IdHall, MovieTittle, MovieType można było przewidzieć na podstawie idSeance.

Przez to, że druga postać nie była spełniona, trzecia także nie mogła być spełniona.

Na obecnym etapie projektu trzecia postać jest spełniona co implikuje poprawność pierwszej i drugiej.

1. Kod SQL tworzący struktury bazodanowe

Opis poszczególnych relacji oraz ich atrybutów znajduje się powyżej w punkcie 3.

CREATE TABLE CINEMABRANCH (

BRANCH\_NAME VARCHAR(20) CONSTRAINT PK\_BRANCH\_NAME PRIMARY KEY NOT NULL,

NUMBER\_OF\_HALL NUMBER(2) NOT NULL,

NUMBER\_OF\_EMPLOYEE NUMBER(3) NOT NULL,

ADDRESS VARCHAR(50) NOT NULL

);

CREATE TABLE CLIENT (

CLIENT\_ID NUMBER(5) CONSTRAINT PK\_CLIENT\_ID PRIMARY KEY NOT NULL,

FIRST\_NAME VARCHAR(20) NOT NULL,

SECOND\_NAME VARCHAR(20) NOT NULL,

IS\_LOYAL NUMBER(1) NOT NULL

);

CREATE TABLE EMPLOYEE (

PESEL NUMBER(11) CONSTRAINT PK\_PESEL PRIMARY KEY NOT NULL,

BRANCH\_NAME VARCHAR(20),

FIRST\_NAME VARCHAR(20) NOT NULL,

SECOND\_NAME VARCHAR(20) NOT NULL,

POSITION\_E VARCHAR2(20) NOT NULL,

EMAIL\_ADDRESS VARCHAR2(255) NOT NULL,

ADDRESS VARCHAR2(50) NOT NULL,

SALARY NUMBER(5) NOT NULL,

START\_DATE VARCHAR2(8) NOT NULL

);

CREATE TABLE HALL (

HALL\_ID NUMBER(2) CONSTRAINT PK\_HALL\_ID PRIMARY KEY NOT NULL,

BRANCH\_NAME VARCHAR2(20) NOT NULL,

ROW\_H NUMBER(2) NOT NULL,

NUMBER\_H NUMBER(2) NOT NULL

);

CREATE TABLE MOVIE (

TITLE VARCHAR2(500) CONSTRAINT PK\_TITLE PRIMARY KEY NOT NULL,

TIME\_M NUMBER(3) NOT NULL,

PROD\_YEAR NUMBER(4) NOT NULL,

DIRECTOR VARCHAR2(50) NOT NULL,

TYPE\_M VARCHAR2(50) NOT NULL,

AGE\_LIMIT NUMBER(2) NOT NULL,

PLAY NUMBER(1) NOT NULL

);

CREATE TABLE PLEACE (

PLACE\_ID NUMBER(3) CONSTRAINT PK\_PLACE\_ID PRIMARY KEY NOT NULL,

HALL\_ID NUMBER(2) NOT NULL,

ROW\_P NUMBER(2) NOT NULL,

NUMBER\_P NUMBER(2) NOT NULL

);

CREATE TABLE PRICE (

TYPE\_P VARCHAR2(10) CONSTRAINT PK\_TYPE\_P PRIMARY KEY NOT NULL,

COST\_P NUMBER(3) NOT NULL

);

CREATE TABLE PURCHASE (

PURCHASE\_ID NUMBER(5) CONSTRAINT PK\_PURCHASE\_ID PRIMARY KEY,

SEANCE\_ID NUMBER(5) NOT NULL,

HALL\_ID NUMBER(2) NOT NULL,

PLACE\_ID NUMBER(3) NOT NULL,

IS\_PAID NUMBER(1) NOT NULL,

TYPE\_P VARCHAR2(10) NOT NULL

);

CREATE TABLE SEANCE (

SEANCE\_ID NUMBER(5) CONSTRAINT PK\_SEANCE\_ID PRIMARY KEY NOT NULL,

TITLE VARCHAR2(500) NOT NULL,

HALL\_ID NUMBER(2) NOT NULL,

DATE\_SEANCE VARCHAR2(8) NOT NULL,

HOURS VARCHAR2(4) NOT NULL

);

CREATE TABLE TICKET (

TICKET\_ID NUMBER(5) CONSTRAINT PK\_TICKET\_ID PRIMARY KEY NOT NULL,

PURCHASE\_ID NUMBER(5) NOT NULL

);