

# POLITECHNIKA WROCŁAWSKA

## WYDZIAŁ ELEKTRONIKI

---

KIERUNEK: Informatyka (INF)  
SPECJALNOŚĆ: Inżynieria systemów informatycznych (INS)

### **Projekt z rozproszonych i obiektowych systemów baz danych**

Rozproszony system bazodanowy przeznaczony do  
obsługi kina

AUTOR:  
Radosław Taborski - 209347  
Piotr Konieczny - 209174

PROWADZĄCY PROJEKT:  
dr inż. Robert Wójcik

OCENA PROJEKTU:

# Spis tabel

# Spis rysunków

3.1	Model konceptualny węzła rozproszonej bazy danych wykonany w programie Microsoft Visio . . . . .	8
3.2	Model fizyczny węzła rozproszonej bazy danych wykonany w programie Microsoft Visio . . . . .	9
5.1	Diagram przypadków użycia . . . . .	13

# Spis treści

<b>Spis rysunków</b>	<b>3</b>
<b>Spis tabel</b>	<b>4</b>
<b>1 Wstęp</b>	<b>5</b>
1.1 Cele projektu . . . . .	5
1.2 Założenia projektowe . . . . .	5
1.3 Zakres projektu . . . . .	5
<b>2 Replikacja w systemie baz danych MySQL</b>	<b>7</b>
2.1 Pojęcie replikacji i podstawowe informacje . . . . .	7
2.2 Replikacja master-slave . . . . .	7
2.3 Testowanie mechanizmów replikacji . . . . .	7
<b>3 Model konceptualny i fizyczny baz danych</b>	<b>8</b>
3.1 Model konceptualny . . . . .	8
3.2 Model fizyczny . . . . .	9
<b>4 Implementacja baz danych w środowisku MySQL</b>	<b>11</b>
4.1 Realizacja bazy danych . . . . .	11
4.2 Wykorzystanie mechanizmów replikacji master-slave . . . . .	11
<b>5 Projekt i implementacja aplikacji klienckiej</b>	<b>12</b>
5.1 Funkcje aplikacji - diagram przypadków użycia . . . . .	12
5.2 Realizacja wybranych funkcjonalności . . . . .	13
<b>6 Wdrożenie i testowanie aplikacji</b>	<b>14</b>
<b>7 Podsumowanie</b>	<b>15</b>
<b>Literatura</b>	<b>16</b>

# Rozdział 1

## Wstęp

### 1.1 Cele projektu

Celem projektu jest stworzenie systemu wspomagającego obsługę kina w oparciu o rozproszoną i obiektową bazę danych. System będzie umożliwiać zarządzanie kinem – z wykorzystaniem relacyjnych baz danych replikujących między sobą dane. W pojedynczym węźle bazy danych zawarte będą tabele opisujące między innymi – seanse filmowe, przydział ich do poszczególnych sal kinowych. Aplikacja będzie umożliwiać ponadto tworzenie nowych wpisów w zależności od rodzaju użytkownika obsługującego program. Pracownik kina będzie wprowadzać nowe seanse do bazy; podczas bezpośredniej sprzedaży biletów będzie również wykreślał miejsca na sali już zajęte – miejsca zawarte na biletach, poszukiwanie rezerwacji wykonanej na konkretną osobę (po imieniu lub nazwisku, czy też numerze rezerwacji). Użytkownik(Klient) będzie mógł rezerwować konkretne miejsce na określony seans.

### 1.2 Założenia projektowe

Projekt został wykonany przy użyciu MySQL 5.7. Rozproszoność systemu oparta została o dockery, na których skonfigurowane zostały węzły zarówno slave jak i master. W trakcie realizacji projektu zostały wykorzystane mechanizmy replikacji master-slave oraz master-slave z opóźnieniem. Do wykonania projektu bazy danych wykorzystane zostało narzędzie Microsoft Visio. Zarządzanie bazą danych odbywało się z poziomu narzędzia zwanego phpMyAdmin. Aplikacja kliencka została wykonana w technologii webowej, z wykorzystaniem platformy programistycznej Angular2 oraz języka programowania TypeScript. Komunikacja między bazą danych a aplikacją kliencką zapewnia api restowe napisane w języku PHP.

### 1.3 Zakres projektu

Zakres projektu dotyczy zaprojektowania i implementacji rozproszonego systemu bazodanowego dla kina. Projekt składa się z kilku etapów.

- Określenie wymagań funkcjonalnych aplikacji bazodanowej
- Testowanie mechanizmów replikacji oraz rozpraszania danych
- Opracowanie modelu koncepcyjnego i fizycznego bazy danych
- Implementacja bazy danych, procedur i widoków

- 
- Projektowanie i implementacja aplikacji klienckiej
  - Wdrożenie i testowanie aplikacji klienckiej

# Rozdział 2

## Replikacja w systemie baz danych MySQL

### 2.1 Pojęcie replikacji i podstawowe informacje

### 2.2 Replikacja master-slave

### 2.3 Testowanie mechanizmów replikacji

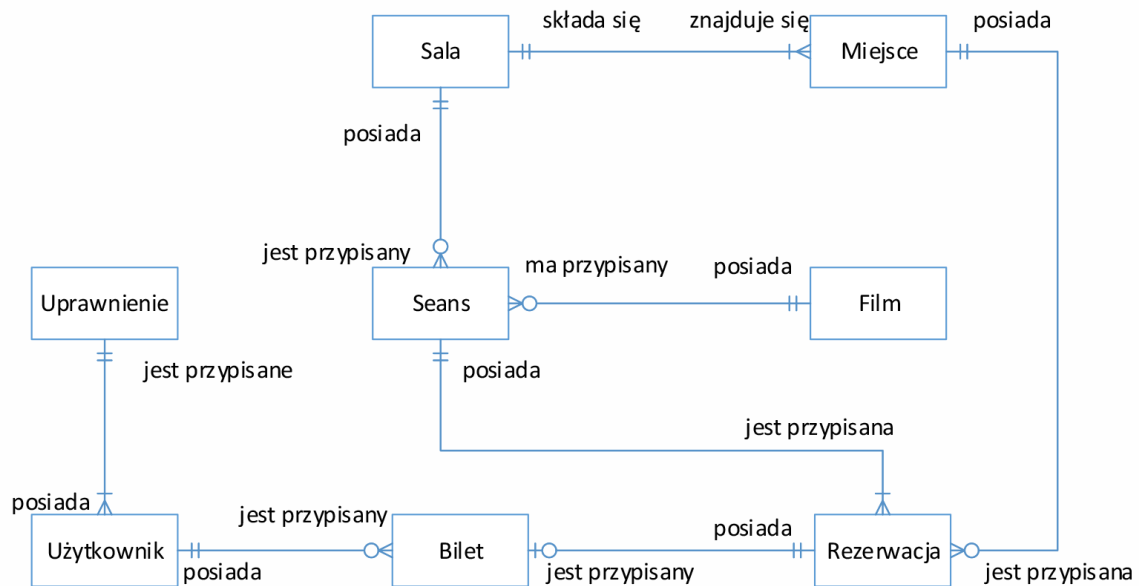
W projekcie użyta zostanie konfiguracja Master-Slave. Poniżej wypunktowane zostały wnioski z przeprowadzonych testów replikacji bazodanowej MySQL.

- Po skonfigurowaniu replikacji wymagane jest utworzenie bazy danych *slave*, która posiada tą samą strukturę co *master*
- Dane zawarte w bazie *master* nie zostaną automatycznie skopiowane do bazy *slave* po skonfigurowaniu replikacji. Należy ręcznie zsynchronizować dane w tabelach.
- W przypadku wyłączenia bazy danych *slave* i modyfikacji bazy *master* baza *slave* zostanie zsynchronizowana po ponownym podłączeniu do sieci.
- *Slave* – odczyt; *master* – zapis, modyfikacja, usuwanie. Gdy *slave* jest wyłączony zapytania GET są wysyłane do innego *slave*, a w ostateczności *mastera*. Gdy jest wyłączony *master* można jedynie odczytywać dane z serwera. Natomiast na ten czas jakakolwiek modyfikacja danych jest niemożliwa.
- Od wersji *MySQL 5.7* możliwa jest replikacja Master-Slave z opóźnieniem. Domyślnie master natychmiastowo wysyła bin-log do węzłów typu *slave*, jednak możliwe jest celowe wprowadzenie opóźnienia, np. w celu ochrony bazy danych przed poleceniem DROP, który wykonany na *masterze*, usunie również bazę/ tabele na standardowych węzłach *slave*. Odpowiednio duże opóźnienie daje możliwość na reakcję ze strony admina, tak aby w razie konieczności ocalić opóźniony węzeł.

## Rozdział 3

# Model konceptualny i fizyczny baz danych

### 3.1 Model konceptualny

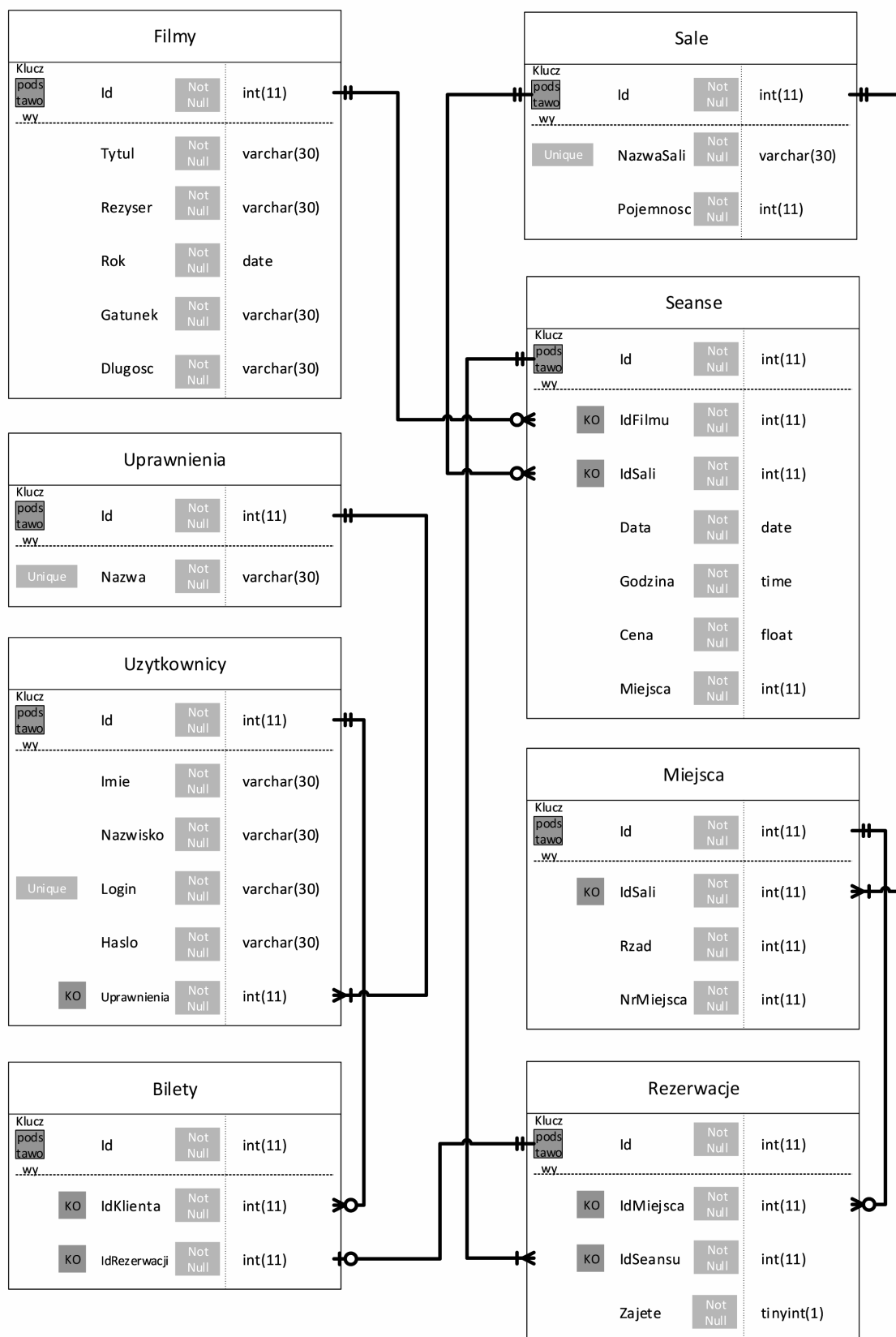


Rysunek 3.1 Model konceptualny węzła rozproszonej bazy danych wykonany w programie Microsoft Visio

W bazie danych dominują relacje typu jeden do wielu. Na każdej sali może odbywać się wiele seansów, natomiast każdy seans posiada tylko jeden film i jedną salę. Każde miejsce również ma przypisaną jedną konkretną salę. Na jedno miejsce może przypadać wiele rezerwacji, w zależności od seansu. Również każdy użytkownik może mieć wiele kupionych biletów lub nie mieć ich wcale. Oraz istnieje wielu użytkowników o tych samych uprawnieniach. Jedyną relacją, która nie jest jeden do wielu, to powiązanie między tabelą *Bilety* i *Rezerwacje*. Na każdą rezerwację może być maksymalnie jeden bilet, który odpowiada jednej rezerwacji, rozumianej tutaj jako miejsce na konkretny seans, które może być zajęte lub też nie.



## 3.2 Model fizyczny



Rysunek 3.2 Model fizyczny węzła rozproszonej bazy danych wykonany w programie Microsoft Visio

**Tabele:**

- *Filmy* - przechowuje najważniejsze informacje o filmach, takie jak ich tytuł, nazwiska reżyserów, daty premiery, gatunek oraz czas trwania, żadna z tych wartości nie może pozostać pusta;
- *Sale* - jedynymi potrzebnymi w projekcie parametrami charakteryzującymi sale są jej unikalna nazwa i pojemność ukazująca liczbę dostępnych miejsc siedzących;
- *Seanse* - każdy seans ma przypisany film oraz salę. Dodatkowo przechowuje takie informacje jak data i godzina wyświetlenia seansu, cenę oraz pozostałą liczbę wolnych miejsc;
- *Miejsca* - w tej tabeli przechowywane są wszystkie miejsca siedzące dostępne w kinie. Każde miejsce znajduje się w swojej określonej sali. Każde miejsce dodatkowo ma też numer i rząd w którym się znajduje na sali;
- *Rezerwacje* - jest to spis wszystkich miejsc na wszystkie dostępne seanse. Dodatkowo przechowywana jest wartość zero-jedynkowa odpowiadająca za stan czy miejsce jest już zajęte;
- *Uprawnienia* - w projekcie przewidziane są uprawnienia dwojakiego rodzaju: uprawnienia administratora i użytkownika. Informacja ta ma kluczowe znaczenie podczas logowania do systemu i wyświetlania w nim dostępnych funkcjonalności;
- *Użytkownicy* - każdy użytkownik jest zobligowany podczas procesu rejestracji do podania takich informacji o sobie jak imię i nazwisko, oraz podania hasła i unikalnego loginu przez który będzie się logował i to właśnie te dane są przechowywane w tej tabeli. Dodatkowo również do każdego użytkownika dodawane są jego uprawnienia: administratora lub zwykłego użytkownika;
- *Bilety* - każdy bilet jest przypisany do konkretnego użytkownika i do konkretnego rekordu z tabeli *Rezerwacje*.

## **Rozdział 4**

# **Implementacja baz danych w środowisku MySQL**

### **4.1 Realizacja bazy danych**

### **4.2 Wykorzystanie mechanizmów replikacji master-slave**

# Rozdział 5

## Projekt i implementacja aplikacji klienckiej

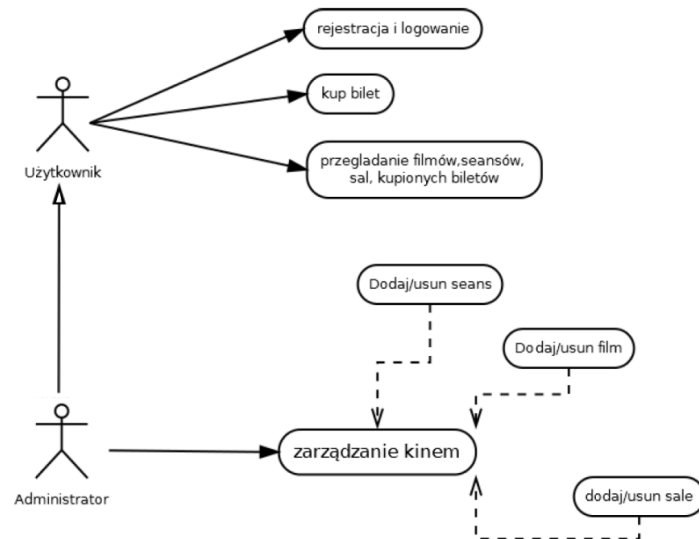
### 5.1 Funkcje aplikacji - diagram przypadków użycia

Użytkownik:

- tworzenie nowego konta (podanie loginu, hasła itp.);
- logowanie;
- przeglądanie filmów, seansów, kupionych biletów;
- kupowanie biletów.

Administrator:

- te same funkcjonalności co użytkownik;
- dodawanie/usuwanie/edytowanie seansów;
- dodawanie/usuwanie/edytowanie filmów;
- dodawanie/usuwanie/edytowanie dostępnych sal.



Rysunek 5.1 Diagram przypadków użycia

## 5.2 Realizacja wybranych funkcjonalności

## **Rozdział 6**

# **Wdrożenie i testowanie aplikacji**

aaa

# Rozdział 7

## Podsumowanie

aaa

# Literatura

- [1] Thomson L., Welling L., *PHP i MySQL. Tworzenie stron WWW*, Helion, Gliwice, 2001.
- [2] Strona internetowa: <http://wazniak.mimuw.edu.pl/index.php> - systemy rozproszone, zaawansowane systemy baz danych, dostęp: 22-11-2017.
- [3] Meloni J. C., *PHP-programowanie*, RM, Warszawa, 2001.
- [4] Knopczyński P., Talarczyk M., *Duplikacja i replikacja MySQL*, dostęp: 22-11-2017