

Министерство науки и высшего образования РФ
Государственное образовательное учреждение высшего
профессионального образования
Белгородский Государственный Технологический Университет им
В. Г. Шухова

Кафедра программного обеспечения вычислительной техники и
автоматизированных систем.

Лабораторная работа №4 по дисциплине
«Технологии web-программирования»

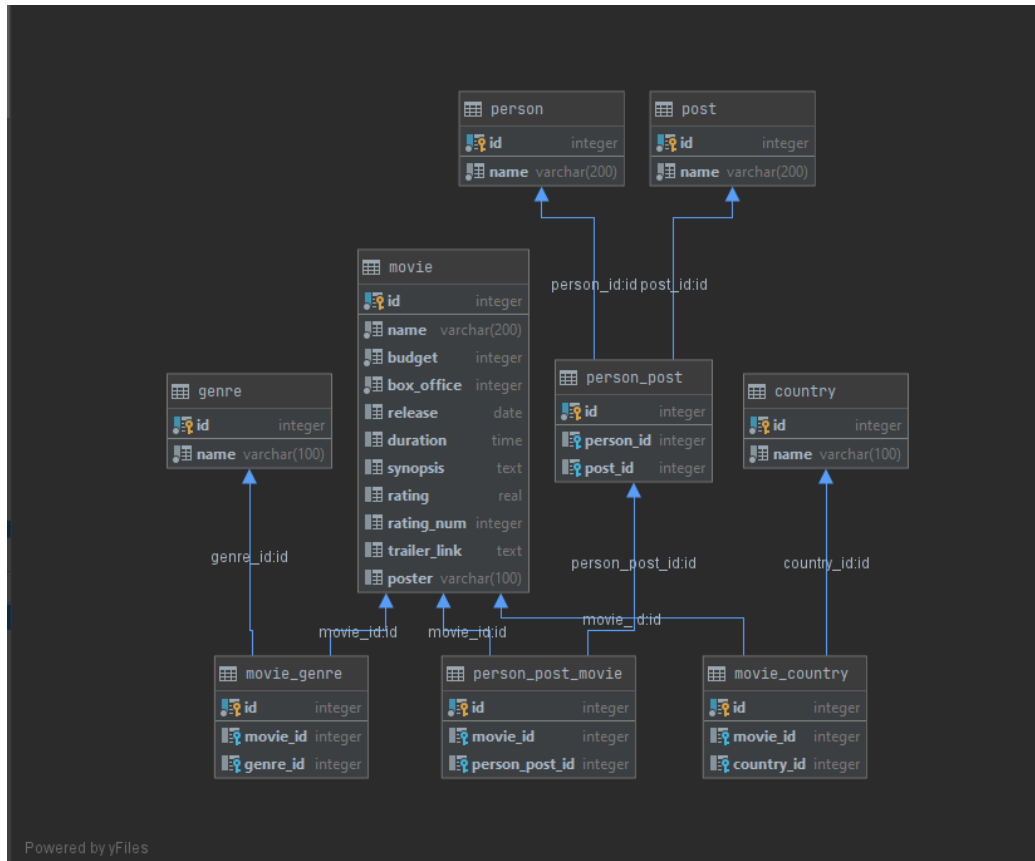
Разработка и проектирование базы данных.

Выполнил:
студент группы ПВ-42
Волобуев И.К.
Проверил:
Картамышев С.В.

Белгород 2020 г.

Цель работы: изучить основы взаимодействия web-приложения с базой данных. Спроектировать базу данных для хранения информации приложения (страницы, пользователи и т.п.).

В качестве СУБД для проекта был сделан выбор в пользу postgresql. Схема базы данных:



По данным сущностям были построены следующие модели в программной среде:

Movie.java

```
@Entity(name = "movie")
@Data
@FieldDefaults(level = AccessLevel.PRIVATE)
public class Movie {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    Long id;

    @NotNull
    String name;

    @NotNull
    Integer budget;

    @NotNull
    Integer box_office;
```

```

@NotNull
@Temporal(TemporalType.DATE)
Date release;

@NotNull
Time duration;

@NotNull
String synopsis;

@NotNull
Double rating;

@NotNull
Integer rating_num;

@NotNull
String trailer_link;

@NotNull
String poster;

@ManyToMany(fetch = FetchType.EAGER)
@JoinTable(
    name = "movie_country",
    joinColumns = @JoinColumn(name = "movie_id", referencedColumnName
= "id"),
    inverseJoinColumns = @JoinColumn(name = "country_id",
referencedColumnName = "id")
)
Set<Country> countries;

@ManyToMany(fetch = FetchType.EAGER)
@JoinTable(
    name = "movie_genre",
    joinColumns = @JoinColumn(name = "movie_id", referencedColumnName
= "id"),
    inverseJoinColumns = @JoinColumn(name = "genre_id",
referencedColumnName = "id")
)
Set<Genre> genres;

@ManyToMany(fetch = FetchType.EAGER)
@JoinTable
(
    name = "person_post_movie",
    joinColumns = @JoinColumn(name = "movie_id",
referencedColumnName = "id"),
    inverseJoinColumns = @JoinColumn(name = "person_post_id",
referencedColumnName = "id")
)
List<PersonPost> personPosts;

```

Country.java

```

@Entity(name = "country")
@Getter
@Setter
@FieldDefaults(level = AccessLevel.PRIVATE)
public class Country {

```

```

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    Long id;

    @NotNull
    String name;

    @ManyToMany
    Set<Movie> movies;
}

```

Genre.java

```

@Entity(name = "genre")
@Getter
@Setter
@FieldDefaults(level = AccessLevel.PRIVATE)
public class Genre {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    Long id;

    @NotNull
    String name;

    @ManyToMany
    Set<Movie> movies;
}

```

PersonPost.java

```

@Entity(name = "person_post")
@Getter
@Setter
@FieldDefaults(level = AccessLevel.PRIVATE)
public class PersonPost {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    Long id;

    @ManyToOne(fetch = FetchType.EAGER)
    @JoinColumn(name = "person_id")
    Person person;

    @ManyToOne(fetch = FetchType.EAGER)
    @JoinColumn(name = "post_id")
    Post post;

    @ManyToMany(mappedBy = "personPosts")
    Set<Movie> movies;
}

```

Person.java

```

@Entity(name = "person")
@Data

```

```
@FieldDefaults(level = AccessLevel.PRIVATE)
public class Person {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    Long id;

    String name;

    @OneToMany(mappedBy = "person")
    Set<PersonPost> personPosts;

}
```

Post.java

```
@Entity(name = "post")
@Data
@FieldDefaults(level = AccessLevel.PRIVATE)
public class Post {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    Long id;

    @NotNull
    String name;

    @OneToMany(mappedBy = "post")
    Set<PersonPost> personPosts;

}
```