ГУАП

КАФЕДРА № 41

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

Старший преподаватель		Б.К. Акопян
должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №4

ФИЗИЧЕСКАЯ РЕАЛИЗАЦИЯ РЕЛЯЦИОННОЙ БАЗЫ ДАННЫХ НА СЕРВЕРЕ

по курсу: БАЗЫ ДАННЫХ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №	4217		Д.М. Никитин
		подпись, дата	инициалы, фамилия

Санкт-Петербург 2024

1. Цель работы: физическая реализация реляционной БД на MySQLсервере.

2. Вариант: 24.

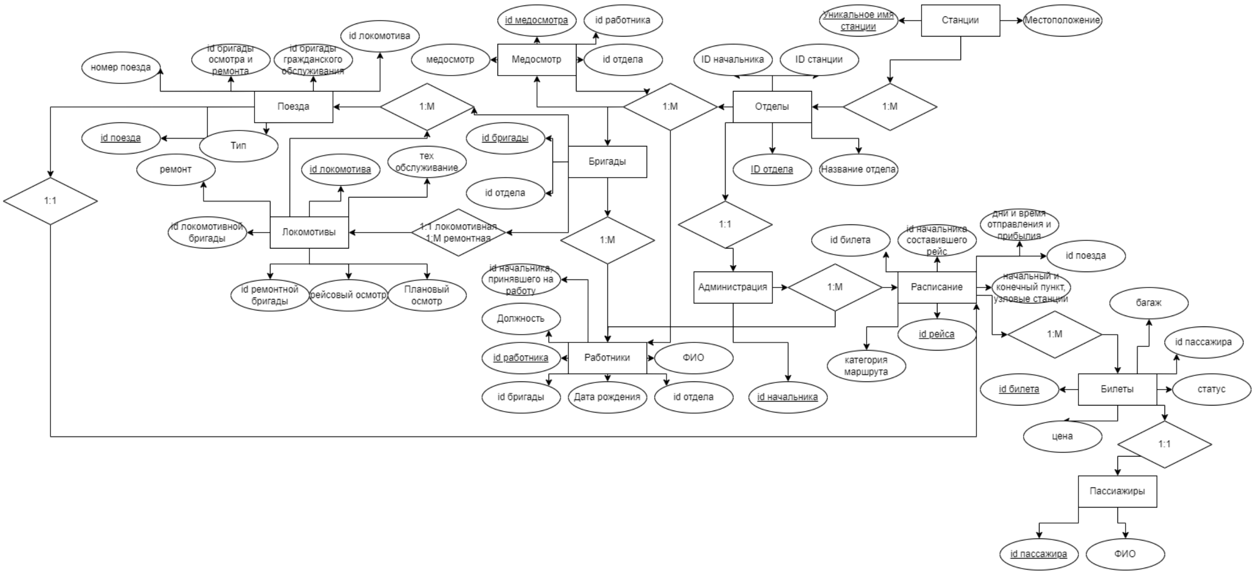


Рисунок 1 – Инфологическая схема

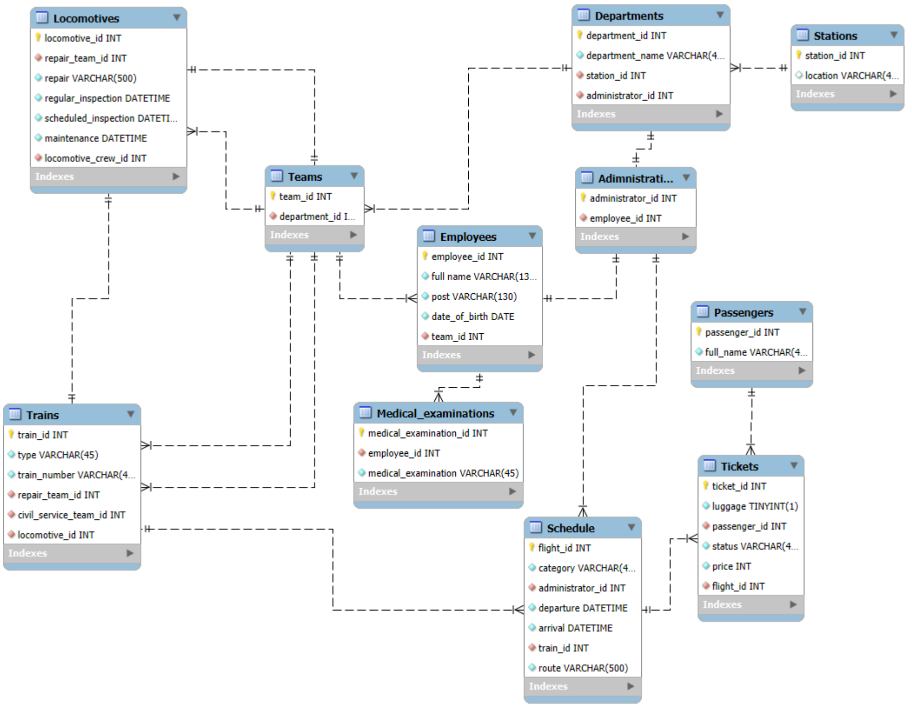


Рисунок 2 – Даталогическая схема

3. Процесс генерации программного кода:

Сначала открываем даталогическую модель данных в MySQL Workbench. После этого нажимаем File -> Export-> Forward Engineer SQL CREATE Script…

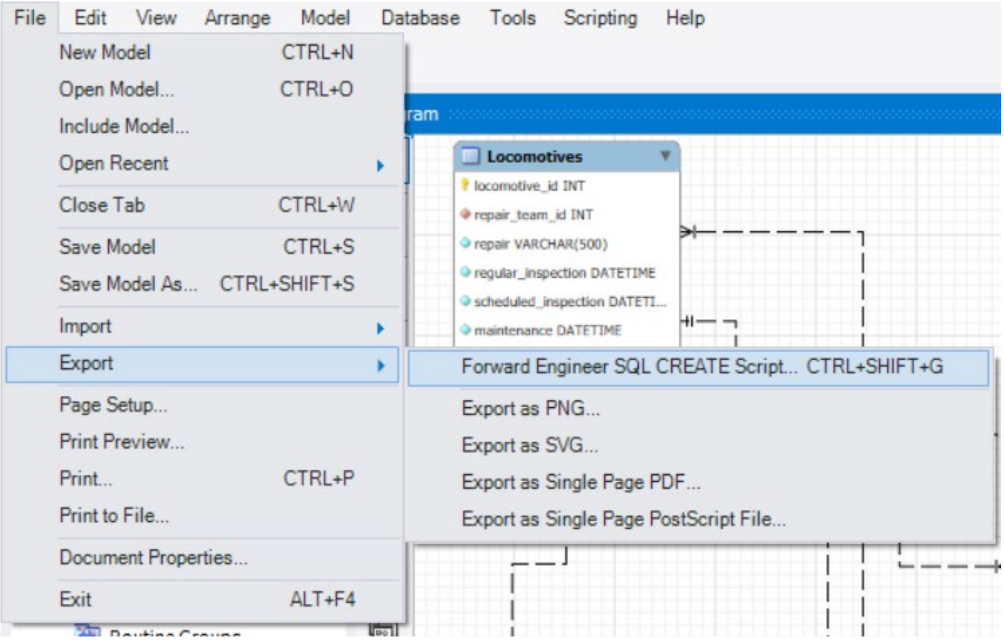
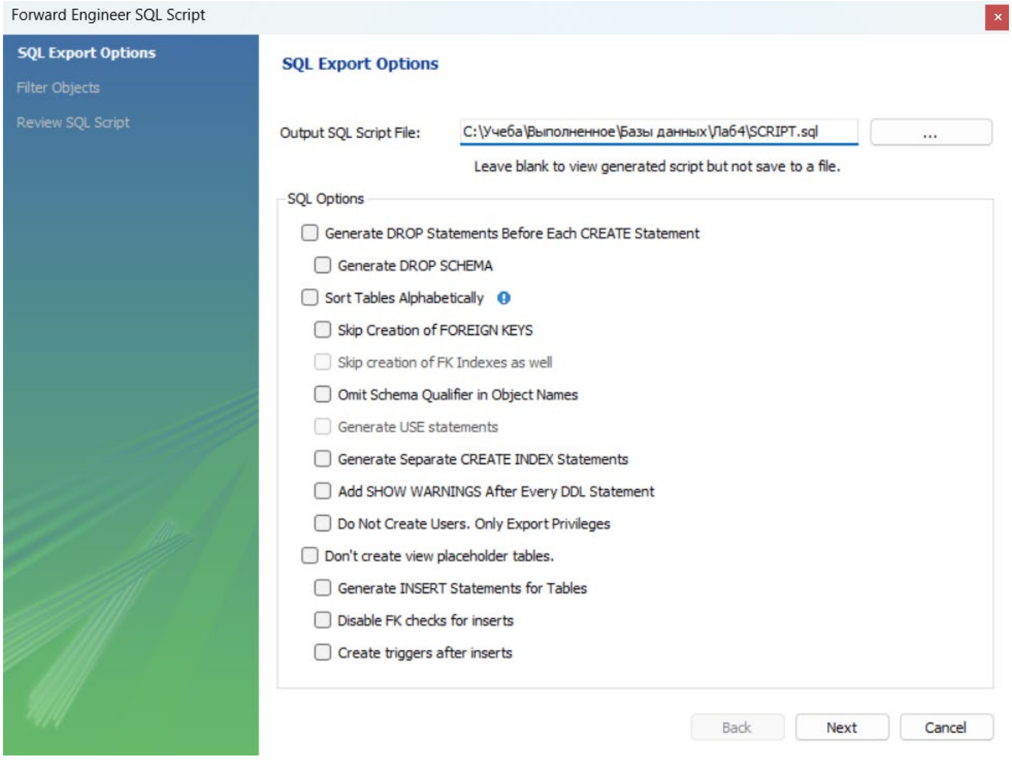


Рисунок 3 – Процесс создания скрипта

В выпадающем окне выбираем нужные параметры и путь.



SQL Export Options

Output SQL Script File: C:\Учеба\Выполненное\Базы данных\Лаб4\SCRIPT.sql

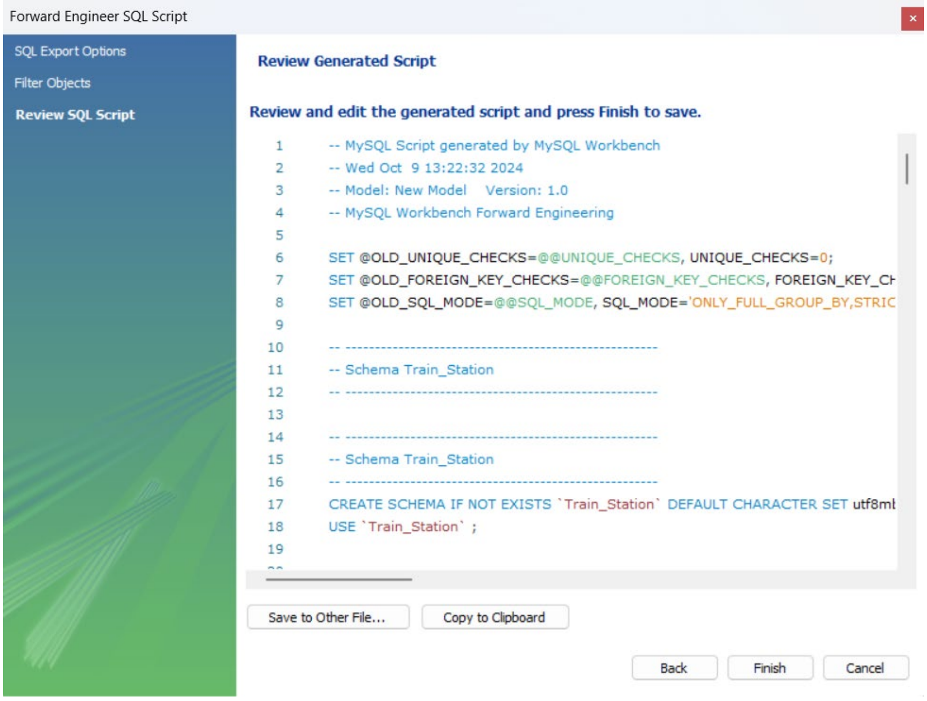
Leave blank to view generated script but not save to a file.

SQL Options

- □ Generate DROP Statements Before Each CREATE Statement
- □ Generate DROP SCHEMA
- □ Sort Tables Alphabetically
- □ Skip Creation of FOREIGN KEYS
- □ Skip creation of FK Indexes as well
- □ Omit Schema Qualifier in Object Names
- □ Generate USE statements
- □ Generate Separate CREATE INDEX Statements
- □ Add SHOW WARNINGS After Every DDL Statement
- □ Do Not Create Users. Only Export Privileges
- □ Don't create view placeholder tables.
- □ Generate INSERT Statements for Tables
- □ Disable FK checks for inserts
- □ Create triggers after inserts

Рисунок 4 – Процесс настройки создания скрипта

Получаем скрипт и жмём Finish, чтобы сохранить его с выбранными параметрами.



Review Generated Script

Review and edit the generated script and press Finish to save.

```
-- MySQL Script generated by MySQL Workbench
-- Wed Oct  9 13:22:32 2024
-- Model: New Model    Version: 1.0
-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CH
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='ONLY_FULL_GROUP_BY,STRIC

-- -----------------------------------------------------
-- Schema Train_Station
-- -----------------------------------------------------

-- -----------------------------------------------------
-- Schema Train_Station
-- -----------------------------------------------------
CREATE SCHEMA IF NOT EXISTS `Train_Station` DEFAULT CHARACTER SET utf8ml
USE `Train_Station` ;
```

Рисунок 5 – Итог генерации скрипта

4. Программные коды:

Далее будет представлен автоматически сгенерированный код для создания базы данных.

```
-- MySQL Script generated by MySQL Workbench
-- Thu Oct 10 23:17:53 2024
-- Model: New Model    Version: 1.0
-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET                 @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET                                       @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE
,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION'
;

-- -----------------------------------------------------
-- Schema Train_Station
-- -----------------------------------------------------

-- -----------------------------------------------------
-- Schema Train_Station
-- -----------------------------------------------------
CREATE SCHEMA IF NOT EXISTS `Train_Station` DEFAULT CHARACTER SET
utf8mb4 ;
USE `Train_Station` ;

-- -----------------------------------------------------
-- Table `Train_Station`.`Stations`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Train_Station`.`Stations` (
  `station_id` INT NOT NULL AUTO_INCREMENT,
  `location` VARCHAR(45) NULL,
  PRIMARY KEY (`station_id`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `Train_Station`.`Teams`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Train_Station`.`Teams` (
  `team_id` INT NOT NULL AUTO_INCREMENT,
  `department_id` INT NOT NULL,
  PRIMARY KEY (`team_id`),
  INDEX `department_id_idx` (`department_id` ASC) VISIBLE,
  CONSTRAINT `team_department_id`
    FOREIGN KEY (`department_id`)
    REFERENCES `Train_Station`.`Departments` (`department_id`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `Train_Station`.`Employees`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Train_Station`.`Employees` (
  `employee_id` INT NOT NULL AUTO_INCREMENT,
  `full_name` VARCHAR(130) NOT NULL,
  `post` VARCHAR(130) NOT NULL,
  `date_of_birth` DATE NOT NULL,
  `team_id` INT NOT NULL,
  PRIMARY KEY (`employee_id`),
  INDEX `team_id_idx` (`team_id` ASC) VISIBLE,
  CONSTRAINT `employees_team_id`
    FOREIGN KEY (`team_id`)
    REFERENCES `Train_Station`.`Teams` (`team_id`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `Train_Station`.`Administration`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Train_Station`.`Administration` (
  `administrator_id` INT NOT NULL AUTO_INCREMENT,
  `employee_id` INT NULL,
  PRIMARY KEY (`administrator_id`),
  INDEX `fk_Adimnistration_Employees1_idx` (`employee_id` ASC)
VISIBLE,
  CONSTRAINT `fk_Adimnistration_Employees1`
```

```
    FOREIGN KEY (`employee_id`)
    REFERENCES `Train_Station`.`Employees` (`employee_id`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `Train_Station`.`Departments`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Train_Station`.`Departments` (
  `department_id` INT NOT NULL AUTO_INCREMENT,
  `department_name` VARCHAR(45) NOT NULL,
  `station_id` INT NOT NULL,
  `administrator_id` INT NOT NULL,
  PRIMARY KEY (`department_id`),
  INDEX `station_id_idx` (`station_id` ASC) VISIBLE,
  INDEX  `fk_Departments_Adimnistration1_idx`  (`administrator_id`
ASC) VISIBLE,
  CONSTRAINT `department_station_id`
    FOREIGN KEY (`station_id`)
    REFERENCES `Train_Station`.`Stations` (`station_id`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `fk_Departments_Adimnistration1`
    FOREIGN KEY (`administrator_id`)
    REFERENCES                   `Train_Station`.`Administration`
(`administrator_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `Train_Station`.`Medical_examinations`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Train_Station`.`Medical_examinations`
(
  `medical_examination_id` INT NOT NULL AUTO_INCREMENT,
  `employee_id` INT NOT NULL,
  `medical_examination` VARCHAR(500) NOT NULL,
  PRIMARY KEY (`medical_examination_id`),
  INDEX `employee_id_idx` (`employee_id` ASC) VISIBLE,
  CONSTRAINT `medical_employee_id`
    FOREIGN KEY (`employee_id`)
    REFERENCES `Train_Station`.`Employees` (`employee_id`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `Train_Station`.`Locomotives`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Train_Station`.`Locomotives` (
  `locomotive_id` INT NOT NULL AUTO_INCREMENT,
  `repair_team_id` INT NOT NULL,
  `repair` VARCHAR(500) NOT NULL,
  `regular_inspection` DATETIME NOT NULL,
  `scheduled_inspection` DATETIME NOT NULL,
  `maintenance` DATETIME NOT NULL,
  `locomotive_crew_id` INT NOT NULL,
  PRIMARY KEY (`locomotive_id`),
  INDEX `team_id_idx` (`repair_team_id` ASC) VISIBLE,
  INDEX  `fk_Locomotives_Teams1_idx`  (`locomotive_crew_id`  ASC)
VISIBLE,
  CONSTRAINT `locomotive_team_id`
    FOREIGN KEY (`repair_team_id`)
    REFERENCES `Train_Station`.`Teams` (`team_id`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `fk_Locomotives_Teams1`
    FOREIGN KEY (`locomotive_crew_id`)
    REFERENCES `Train_Station`.`Teams` (`team_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `Train_Station`.`Trains`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Train_Station`.`Trains` (
  `train_id` INT NOT NULL AUTO_INCREMENT,
  `type` VARCHAR(45) NOT NULL,
  `train_number` VARCHAR(45) NOT NULL,
  `repair_team_id` INT NOT NULL,
  `civil_service_team_id` INT NOT NULL,
  `locomotive_id` INT NOT NULL,
  PRIMARY KEY (`train_id`),
  INDEX `repair_team_id_idx` (`repair_team_id` ASC) VISIBLE,
  INDEX   `fk_Trains_Locomotives1_idx`   (`locomotive_id`   ASC)
VISIBLE,
  INDEX `civil_service_team_id_idx` (`civil_service_team_id` ASC)
VISIBLE,
  CONSTRAINT `trains_repair_team_id`
    FOREIGN KEY (`repair_team_id`)
    REFERENCES `Train_Station`.`Teams` (`team_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Trains_Locomotives1`
    FOREIGN KEY (`locomotive_id`)
    REFERENCES `Train_Station`.`Locomotives` (`locomotive_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `trains_civil_service_team_id`
```

```
    FOREIGN KEY (`civil_service_team_id`)
    REFERENCES `Train_Station`.`Teams` (`team_id`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `Train_Station`.`Schedule`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Train_Station`.`Schedule` (
  `flight_id` INT NOT NULL AUTO_INCREMENT,
  `category` VARCHAR(45) NOT NULL,
  `administrator_id` INT NOT NULL,
  `departure` DATETIME NOT NULL,
  `arrival` DATETIME NOT NULL,
  `train_id` INT NOT NULL,
  `route` VARCHAR(500) NOT NULL,
  PRIMARY KEY (`flight_id`),
  INDEX `administrator_id_idx` (`administrator_id` ASC) VISIBLE,
  INDEX `train_id_idx` (`train_id` ASC) VISIBLE,
  CONSTRAINT `schedule_administrator_id`
    FOREIGN KEY (`administrator_id`)
    REFERENCES                   `Train_Station`.`Administration`
(`administrator_id`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `schedule_train_id`
    FOREIGN KEY (`train_id`)
    REFERENCES `Train_Station`.`Trains` (`train_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `Train_Station`.`Passengers`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Train_Station`.`Passengers` (
  `passenger_id` INT NOT NULL AUTO_INCREMENT,
  `full_name` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`passenger_id`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `Train_Station`.`Tickets`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Train_Station`.`Tickets` (
  `ticket_id` INT NOT NULL AUTO_INCREMENT,
  `luggage` TINYINT(1) NOT NULL,
  `passenger_id` INT NOT NULL,
  `status` VARCHAR(45) NOT NULL,
  `price` INT NOT NULL,
  `flight_id` INT NOT NULL,
  PRIMARY KEY (`ticket_id`),
  INDEX `flight_id_idx` (`flight_id` ASC) VISIBLE,
  INDEX `passenger_id_idx` (`passenger_id` ASC) VISIBLE,
  CONSTRAINT `tickets_flight_id`
    FOREIGN KEY (`flight_id`)
    REFERENCES `Train_Station`.`Schedule` (`flight_id`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `tickets_passenger_id`
    FOREIGN KEY (`passenger_id`)
    REFERENCES `Train_Station`.`Passengers` (`passenger_id`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;


SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

Далее представлен код для заполнения базы данных:

```
-- Вставка данных в таблицу Stations
INSERT INTO `Train_Station`.`Stations` (`location`) VALUES
('Новосибирск-Главный'),
('Бердск'),
('Обь');

INSERT INTO `Train_Station`.`Passengers` (`full_name`) VALUES
('Пассажир А'),
('Пассажир Б'),
('Пассажир В');

INSERT INTO `Train_Station`.`Administration` (`employee_id`)
VALUES
(NULL), (NULL), (NULL), (NULL), (NULL), (NULL), (NULL);

INSERT INTO `Train_Station`.`Departments` (`department_name`,
`station_id`, `administrator_id`) VALUES
('Водители', 1, 1),
('Диспетчеры', 1, 2),
('Ремонтники подвижного состава', 1, 3),
('Служба ремонта путей', 1, 4),
('Кассиры', 1, 5),
('Служба подготовки составов', 1, 6),
('Справочная служба', 1, 7);

INSERT INTO `Train_Station`.`Teams` (`department_id`) VALUES
(1), (2), (3), (4), (5), (6), (7);

INSERT INTO `Train_Station`.`Locomotives` (`repair_team_id`,
`repair`, `regular_inspection`, `scheduled_inspection`,
`maintenance`, `locomotive_crew_id`) VALUES
```

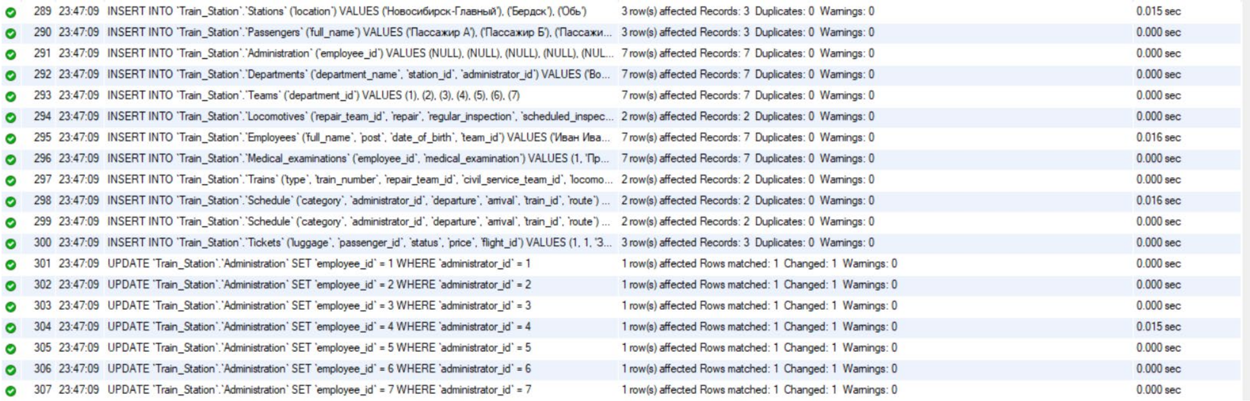
```
(3, 'Мелкий ремонт', '2024-10-01 08:00:00', '2024-10-15 08:00:00',
'2024-10-30 08:00:00', 1),
(3, 'Крупный ремонт', '2024-10-02 08:00:00', '2024-10-16
08:00:00', '2024-10-31 08:00:00', 2);

INSERT INTO `Train_Station`.`Employees` (`full_name`, `post`,
`date_of_birth`, `team_id`) VALUES
('Иван Иванов', 'Водитель', '1980-01-01', 1),
('Анна Петрова', 'Диспетчер', '1985-02-15', 2),
('Алексей Смирнов', 'Ремонтник', '1990-03-20', 3),
('Дмитрий Кузнецов', 'Путеец', '1995-04-25', 4),
('Екатерина Соколова', 'Кассир', '2000-05-30', 5),
('Мария Федорова', 'Служба подготовки составов', '2001-06-05', 6),
('Ольга Морозова', 'Справочная служба', '2000-07-10', 7);

INSERT INTO `Train_Station`.`Medical_examinations`
(`employee_id`, `medical_examination`) VALUES
(1, 'Пройден'),
(2, 'Пройден'),
(3, 'Пройден'),
(4, 'Пройден'),
(5, 'Пройден'),
(6, 'Пройден'),
(7, 'Предстоит пройти');

INSERT INTO `Train_Station`.`Trains` (`type`, `train_number`,
`repair_team_id`, `civil_service_team_id`, `locomotive_id`)
VALUES
('Скорый', '101', 3, 6, 1),
('Пассажирский', '102', 3, 6, 2);

INSERT INTO `Train_Station`.`Schedule` (`category`,
`administrator_id`, `departure`, `arrival`, `train_id`, `route`)
VALUES
('Внутренний', 1, '2024-10-10 08:00:00', '2024-10-10 12:00:00', 1,
'Новосибирск-Главный - Бердск'),
('Международный', 2, '2024-10-11 09:00:00', '2024-10-11 13:00:00',
2, 'Новосибирск-Главный - Обь');

INSERT INTO `Train_Station`.`Schedule` (`category`,
`administrator_id`, `departure`, `arrival`, `train_id`, `route`)
VALUES
('Внутренний', 1, '2024-10-10 08:00:00', '2024-10-10 12:00:00', 1,
'Новосибирск-Главный - Бердск'),
('Международный', 2, '2024-10-11 09:00:00', '2024-10-11 13:00:00',
2, 'Новосибирск-Главный - Обь');

INSERT INTO `Train_Station`.`Tickets` (`luggage`, `passenger_id`,
`status`, `price`, `flight_id`) VALUES
(1, 1, 'Забронирован', 50, 1),
(0, 2, 'Забронирован', 30, 2),
(1, 3, 'Отменен', 40, 1);
```

```
-- Обновляем оставшихся администраторов
UPDATE  `Train_Station`.`Administration`  SET  `employee_id`  =  1
WHERE `administrator_id` = 1;
UPDATE  `Train_Station`.`Administration`  SET  `employee_id`  =  2
WHERE `administrator_id` = 2;
UPDATE  `Train_Station`.`Administration`  SET  `employee_id`  =  3
WHERE `administrator_id` = 3;
UPDATE  `Train_Station`.`Administration`  SET  `employee_id`  =  4
WHERE `administrator_id` = 4;
UPDATE  `Train_Station`.`Administration`  SET  `employee_id`  =  5
WHERE `administrator_id` = 5;
UPDATE  `Train_Station`.`Administration`  SET  `employee_id`  =  6
WHERE `administrator_id` = 6;
UPDATE  `Train_Station`.`Administration`  SET  `employee_id`  =  7
WHERE `administrator_id` = 7;
```

Далее будет представлен код для вывода таблиц:

```
USE train_station;
SELECT * FROM stations;
SELECT SLEEP(1);
SELECT * FROM passengers;
SELECT SLEEP(1);
SELECT * FROM administration;
SELECT SLEEP(1);
SELECT * FROM departments;
SELECT SLEEP(1);
SELECT * FROM teams;
SELECT SLEEP(1);
SELECT * FROM locomotives;
SELECT SLEEP(1);
SELECT * FROM employees;
SELECT SLEEP(1);
SELECT * FROM Medical_examinations;
SELECT SLEEP(1);
SELECT * FROM Trains;
SELECT SLEEP(1);
SELECT * FROM tickets;
SELECT SLEEP(1);
SELECT * FROM schedule;
```

Заполнение базы данных прошло успешно.



```
✓ 289 23:47:09 INSERT INTO `Train_Station`.`Stations` (`location`) VALUES ('Новосибирск-Главный'), ('Бердск'), ('Обь')    3 row(s) affected Records: 3  Duplicates: 0  Warnings: 0    0.015 sec
✓ 290 23:47:09 INSERT INTO `Train_Station`.`Passengers` (`full_name`) VALUES ('Пассажир А'), ('Пассажир Б'), ('Пассажи...    3 row(s) affected Records: 3  Duplicates: 0  Warnings: 0    0.000 sec
✓ 291 23:47:09 INSERT INTO `Train_Station`.`Administration` (`employee_id`) VALUES (NULL), (NULL), (NULL), (NULL), (NUL...    7 row(s) affected Records: 7  Duplicates: 0  Warnings: 0    0.000 sec
✓ 292 23:47:09 INSERT INTO `Train_Station`.`Departments` (`department_name`, `station_id`, `administrator_id`) VALUES ('Bo...    7 row(s) affected Records: 7  Duplicates: 0  Warnings: 0    0.000 sec
✓ 293 23:47:09 INSERT INTO `Train_Station`.`Teams` (`department_id`) VALUES (1), (2), (3), (4), (5), (6), (7)    7 row(s) affected Records: 7  Duplicates: 0  Warnings: 0    0.000 sec
✓ 294 23:47:09 INSERT INTO `Train_Station`.`Locomotives` (`repair_team_id`, `repair`, `regular_inspection`, `scheduled_inspec...    2 row(s) affected Records: 2  Duplicates: 0  Warnings: 0    0.000 sec
✓ 295 23:47:09 INSERT INTO `Train_Station`.`Employees` (`full_name`, `post`, `date_of_birth`, `team_id`) VALUES ('Иван Ива...    7 row(s) affected Records: 7  Duplicates: 0  Warnings: 0    0.016 sec
✓ 296 23:47:09 INSERT INTO `Train_Station`.`Medical_examinations` (`employee_id`, `medical_examination`) VALUES (1, 'Пр...    7 row(s) affected Records: 7  Duplicates: 0  Warnings: 0    0.000 sec
✓ 297 23:47:09 INSERT INTO `Train_Station`.`Trains` (`type`, `train_number`, `repair_team_id`, `civil_service_team_id`, `locomo...    2 row(s) affected Records: 2  Duplicates: 0  Warnings: 0    0.000 sec
✓ 298 23:47:09 INSERT INTO `Train_Station`.`Schedule` (`category`, `administrator_id`, `departure`, `arrival`, `train_id`, `route`) ...    2 row(s) affected Records: 2  Duplicates: 0  Warnings: 0    0.016 sec
✓ 299 23:47:09 INSERT INTO `Train_Station`.`Schedule` (`category`, `administrator_id`, `departure`, `arrival`, `train_id`, `route`) ...    2 row(s) affected Records: 2  Duplicates: 0  Warnings: 0    0.000 sec
✓ 300 23:47:09 INSERT INTO `Train_Station`.`Tickets` (`luggage`, `passenger_id`, `status`, `price`, `flight_id`) VALUES (1, 1, '3...    3 row(s) affected Records: 3  Duplicates: 0  Warnings: 0    0.000 sec
✓ 301 23:47:09 UPDATE `Train_Station`.`Administration` SET `employee_id` = 1 WHERE `administrator_id` = 1    1 row(s) affected Rows matched: 1  Changed: 1  Warnings: 0    0.000 sec
✓ 302 23:47:09 UPDATE `Train_Station`.`Administration` SET `employee_id` = 2 WHERE `administrator_id` = 2    1 row(s) affected Rows matched: 1  Changed: 1  Warnings: 0    0.000 sec
✓ 303 23:47:09 UPDATE `Train_Station`.`Administration` SET `employee_id` = 3 WHERE `administrator_id` = 3    1 row(s) affected Rows matched: 1  Changed: 1  Warnings: 0    0.000 sec
✓ 304 23:47:09 UPDATE `Train_Station`.`Administration` SET `employee_id` = 4 WHERE `administrator_id` = 4    1 row(s) affected Rows matched: 1  Changed: 1  Warnings: 0    0.015 sec
✓ 305 23:47:09 UPDATE `Train_Station`.`Administration` SET `employee_id` = 5 WHERE `administrator_id` = 5    1 row(s) affected Rows matched: 1  Changed: 1  Warnings: 0    0.000 sec
✓ 306 23:47:09 UPDATE `Train_Station`.`Administration` SET `employee_id` = 6 WHERE `administrator_id` = 6    1 row(s) affected Rows matched: 1  Changed: 1  Warnings: 0    0.000 sec
✓ 307 23:47:09 UPDATE `Train_Station`.`Administration` SET `employee_id` = 7 WHERE `administrator_id` = 7    1 row(s) affected Rows matched: 1  Changed: 1  Warnings: 0    0.000 sec
```

Рисунок 6 – Результат работы кода

5. Вывод:

В ходе выполнения лабораторной работы № 4 я освоил процесс физической реализации реляционной базы данных на MySQL сервере, включая создание таблиц, установление связей между ними и заполнение данными. Это позволило мне углубить понимание структуры баз данных и научиться правильно выбирать типы данных, а также формировать SQL-запросы для вставки и выборки информации.

В процессе работы я столкнулся с проблемами, связанными с ограничениями внешних ключей, что требовало внимательного подхода к порядку выполнения операций. Для решения этих задач я использовал временные значения NULL и постепенно заполнял таблицы, что помогло мне лучше понять логику работы реляционных баз данных и научиться эффективно управлять данными.