|  |
| --- |
| ­­­­­­­­­­­­­­ |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА - Российский технологический университет»**  **РТУ МИРЭА** |

Институт Искусственного Интеллекта

Кафедра Вычислительной Техники (ПИ)

**ОТЧЁТ ПО РАБОТЕ В СЕМЕСТРЕ**

по дисциплине

«Разработка баз данных»

|  |  |
| --- | --- |
| Выполнил студент группы  ИВБО-06-20 | Чурилов А.К. |
| Принял преподаватель кафедры ПИ | Благовещенский В.Г. |
| Практическая работа выполнена | «\_\_»\_\_\_\_\_\_\_2022 г. |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_2022 г. |

Москва 2022 г.

**Содержание**

[Логическая модель базы данных 3](#_Toc117959296)

[Физическая модель базы данных 3](#_Toc117959297)

[Создание базы данных через консоль 4](#_Toc117959298)

[Заполнение базы данных 11](#_Toc117959299)

[Создание запросов на получение данных 13](#_Toc117959300)

[Хранимые процедуры, функции и триггеры 15](#_Toc117959301)

[Вывод 18](#_Toc117959302)

[Список используемой литературы 18](#_Toc117959303)

Логическая модель базы данных

Составим логическую модель базы данных.

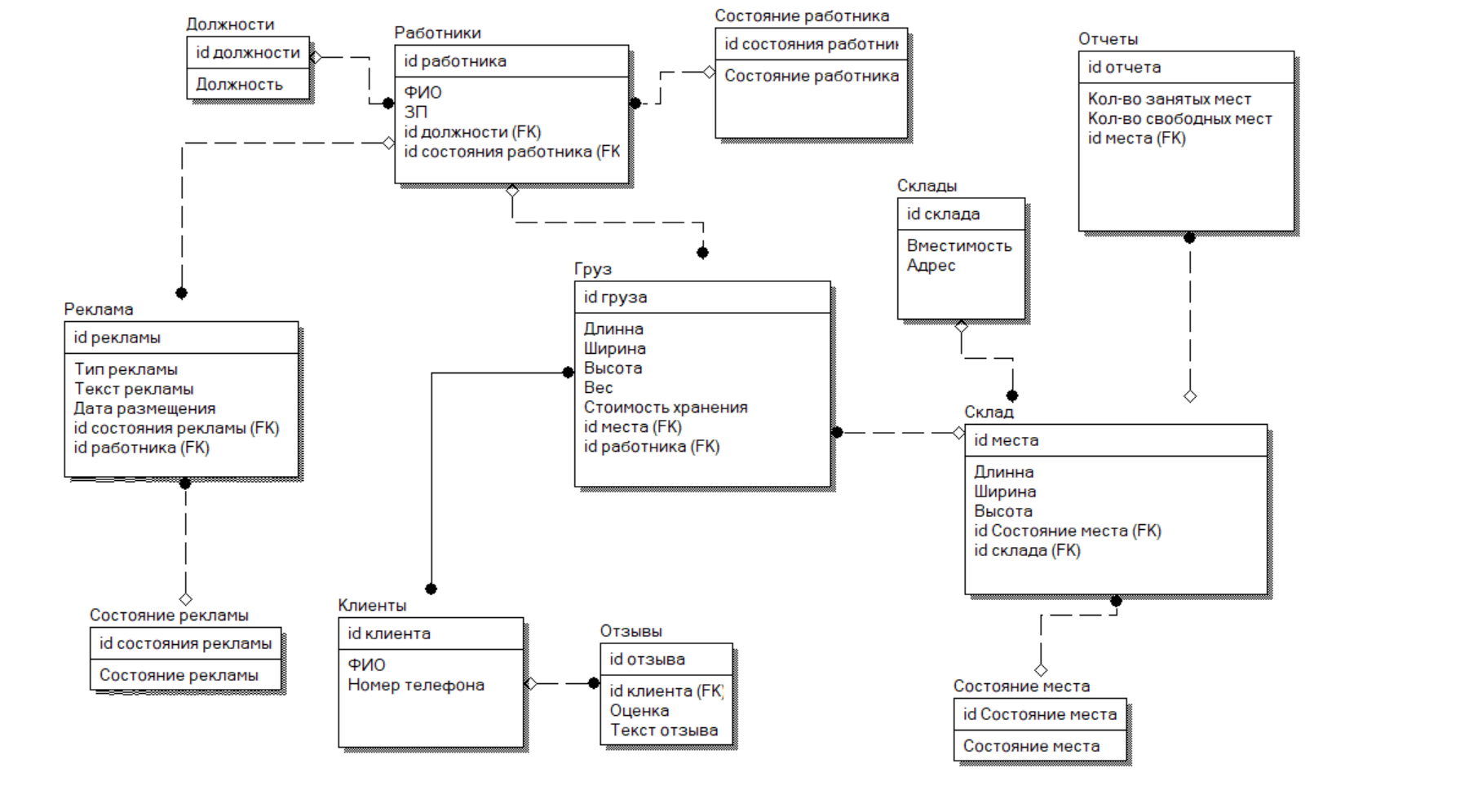


Рис. 1 – Логическая модель базы данных

Физическая модель базы данных

Составим физическую модель.

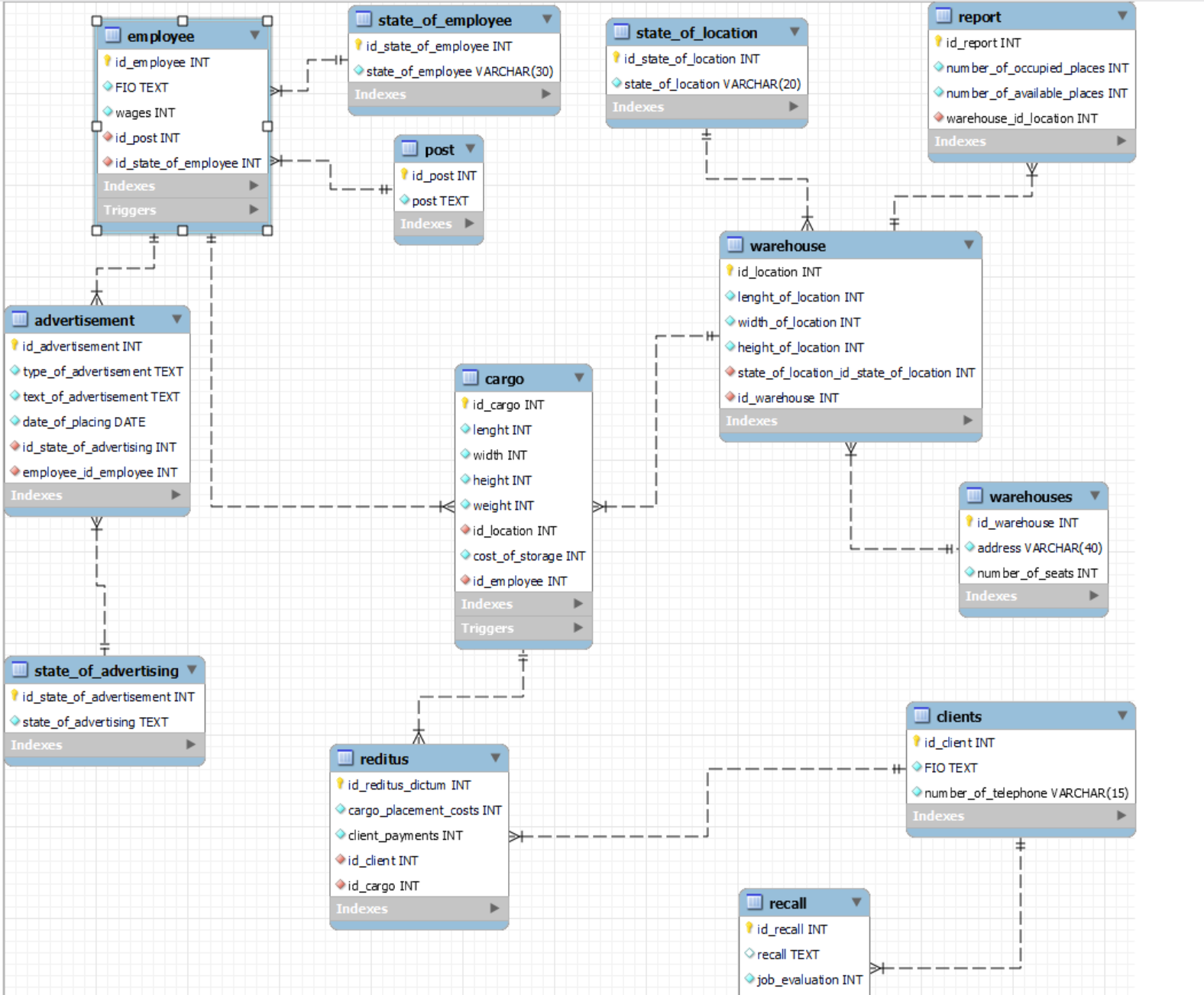


Рис.2 – Физическая модель базы данных

Создание базы данных через консоль

Листинг 1 – Скрипт SQL для создания таблиц

|  |
| --- |
| CREATE TABLE IF NOT EXISTS `warehouse`.`state\_of\_advertising` (  `id\_state\_of\_advertisement` INT NOT NULL AUTO\_INCREMENT,  `state\_of\_advertising` TEXT NOT NULL,  PRIMARY KEY (`id\_state\_of\_advertisement`));  CREATE TABLE IF NOT EXISTS `warehouse`.`post` (  `id\_post` INT NOT NULL AUTO\_INCREMENT,  `post` TEXT NOT NULL,  PRIMARY KEY (`id\_post`));  CREATE TABLE IF NOT EXISTS `warehouse`.`state\_of\_employee` (  `id\_state\_of\_employee` INT NOT NULL AUTO\_INCREMENT,  `state\_of\_employee` VARCHAR(30) NOT NULL,  PRIMARY KEY (`id\_state\_of\_employee`));  CREATE TABLE IF NOT EXISTS `warehouse`.`employee` (  `id\_employee` INT NOT NULL AUTO\_INCREMENT,  `FIO` TEXT NOT NULL,  `wages` INT NOT NULL,  `id\_post` INT NOT NULL,  `id\_state\_of\_employee` INT NOT NULL,  PRIMARY KEY (`id\_employee`),  FOREIGN KEY (`id\_post`)  REFERENCES `warehouse`.`post` (`id\_post`),  FOREIGN KEY (`id\_state\_of\_employee`)  REFERENCES `warehouse`.`state\_of\_employee` (`id\_state\_of\_employee`));  CREATE TABLE IF NOT EXISTS `warehouse`.`advertisement` (  `id\_advertisement` INT NOT NULL AUTO\_INCREMENT,  `type\_of\_advertisement` TEXT NOT NULL,  `text\_of\_advertisement` TEXT NOT NULL,  `date\_of\_placing` DATE NOT NULL,  `id\_state\_of\_advertising` INT NOT NULL,  `employee\_id\_employee` INT NOT NULL,  PRIMARY KEY (`id\_advertisement`),  FOREIGN KEY (`id\_state\_of\_advertising`)  REFERENCES `warehouse`.`state\_of\_advertising` (`id\_state\_of\_advertisement`),  FOREIGN KEY (`employee\_id\_employee`)  REFERENCES `warehouse`.`employee` (`id\_employee`));  CREATE TABLE IF NOT EXISTS `warehouse`.`state\_of\_location` (  `id\_state\_of\_location` INT NOT NULL AUTO\_INCREMENT,  `state\_of\_location` VARCHAR(20) NOT NULL,  PRIMARY KEY (`id\_state\_of\_location`));  CREATE TABLE IF NOT EXISTS `warehouse`.`warehouses` (  `id\_warehouse` INT NOT NULL AUTO\_INCREMENT,  `address` VARCHAR(40) NOT NULL,  `number\_of\_seats` INT NOT NULL,  PRIMARY KEY (`id\_warehouse`));  CREATE TABLE IF NOT EXISTS `warehouse`.`warehouse` (  `id\_location` INT NOT NULL AUTO\_INCREMENT,  `lenght\_of\_location` INT NOT NULL,  `width\_of\_location` INT NOT NULL,  `height\_of\_location` INT NOT NULL,  `state\_of\_location\_id\_state\_of\_location` INT NOT NULL,  `id\_warehouse` INT NOT NULL,  PRIMARY KEY (`id\_location`),  FOREIGN KEY (`state\_of\_location\_id\_state\_of\_location`)  REFERENCES `warehouse`.`state\_of\_location` (`id\_state\_of\_location`)  FOREIGN KEY (`id\_warehouse`)  REFERENCES `warehouse`.`warehouses` (`id\_warehouse`));  CREATE TABLE IF NOT EXISTS `warehouse`.`cargo` (  `id\_cargo` INT NOT NULL AUTO\_INCREMENT,  `lenght` INT NOT NULL,  `width` INT NOT NULL,  `height` INT NOT NULL,  `weight` INT NOT NULL,  `id\_location` INT NOT NULL,  `cost\_of\_storage` INT NOT NULL,  `id\_employee` INT NOT NULL,  PRIMARY KEY (`id\_cargo`),  FOREIGN KEY (`id\_employee`)  REFERENCES `warehouse`.`employee` (`id\_employee`),  FOREIGN KEY (`id\_location`)  REFERENCES `warehouse`.`warehouse` (`id\_location`));  CREATE TABLE IF NOT EXISTS `warehouse`.`clients` (  `id\_client` INT NOT NULL AUTO\_INCREMENT,  `FIO` TEXT NOT NULL,  `number\_of\_telephone` VARCHAR(15) NOT NULL,  PRIMARY KEY (`id\_client`));  CREATE TABLE IF NOT EXISTS `warehouse`.`recall` (  `id\_recall` INT NOT NULL AUTO\_INCREMENT,  `recall` TEXT NULL DEFAULT NULL,  `job\_evaluation` INT NOT NULL,  `id\_сlient` INT NOT NULL,  PRIMARY KEY (`id\_recall`),  FOREIGN KEY (`id\_сlient`)  REFERENCES `warehouse`.`clients` (`id\_client`));  CREATE TABLE IF NOT EXISTS `warehouse`.`reditus` (  `id\_reditus\_dictum` INT NOT NULL AUTO\_INCREMENT,  `cargo\_placement\_costs` INT NOT NULL,  `client\_payments` INT NOT NULL,  `id\_client` INT NOT NULL,  `id\_cargo` INT NOT NULL,  PRIMARY KEY (`id\_reditus\_dictum`),  FOREIGN KEY (`id\_client`)  REFERENCES `warehouse`.`clients` (`id\_client`),  FOREIGN KEY (`id\_cargo`)  REFERENCES `warehouse`.`cargo` (`id\_cargo`))  CREATE TABLE IF NOT EXISTS `warehouse`.`report` (  `id\_report` INT NOT NULL AUTO\_INCREMENT,  `number\_of\_occupied\_places` INT NOT NULL,  `number\_of\_available\_places` INT NOT NULL,  `warehouse\_id\_location` INT NOT NULL,  PRIMARY KEY (`id\_report`),  FOREIGN KEY (`warehouse\_id\_location`)  REFERENCES `warehouse`.`warehouse` (`id\_location`)); |

Посмотрим на созданные таблицы.

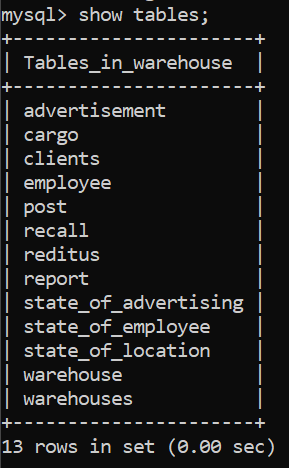


Рис. 3 – Созданные таблицы

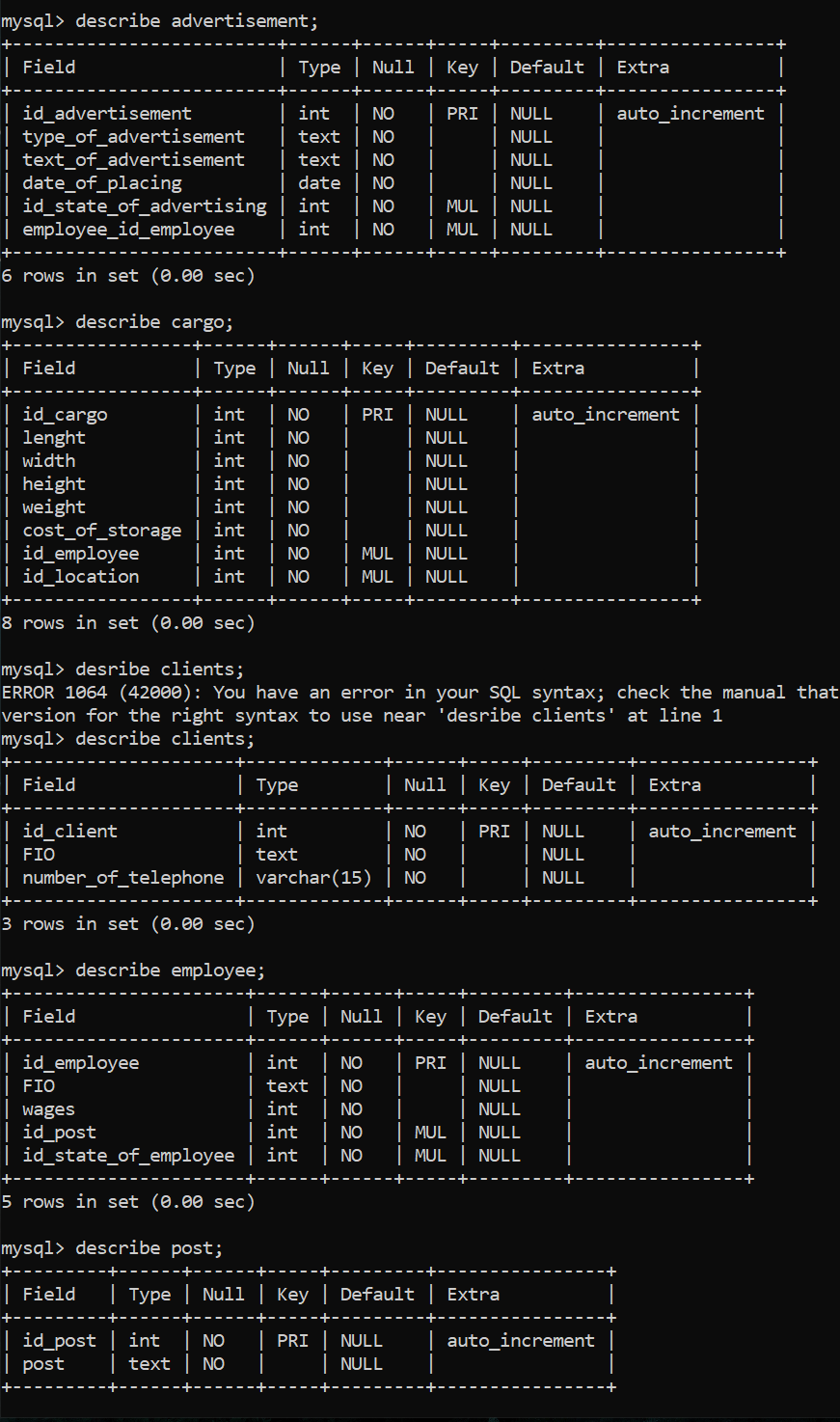


Рис. 4 – Описание созданных таблиц

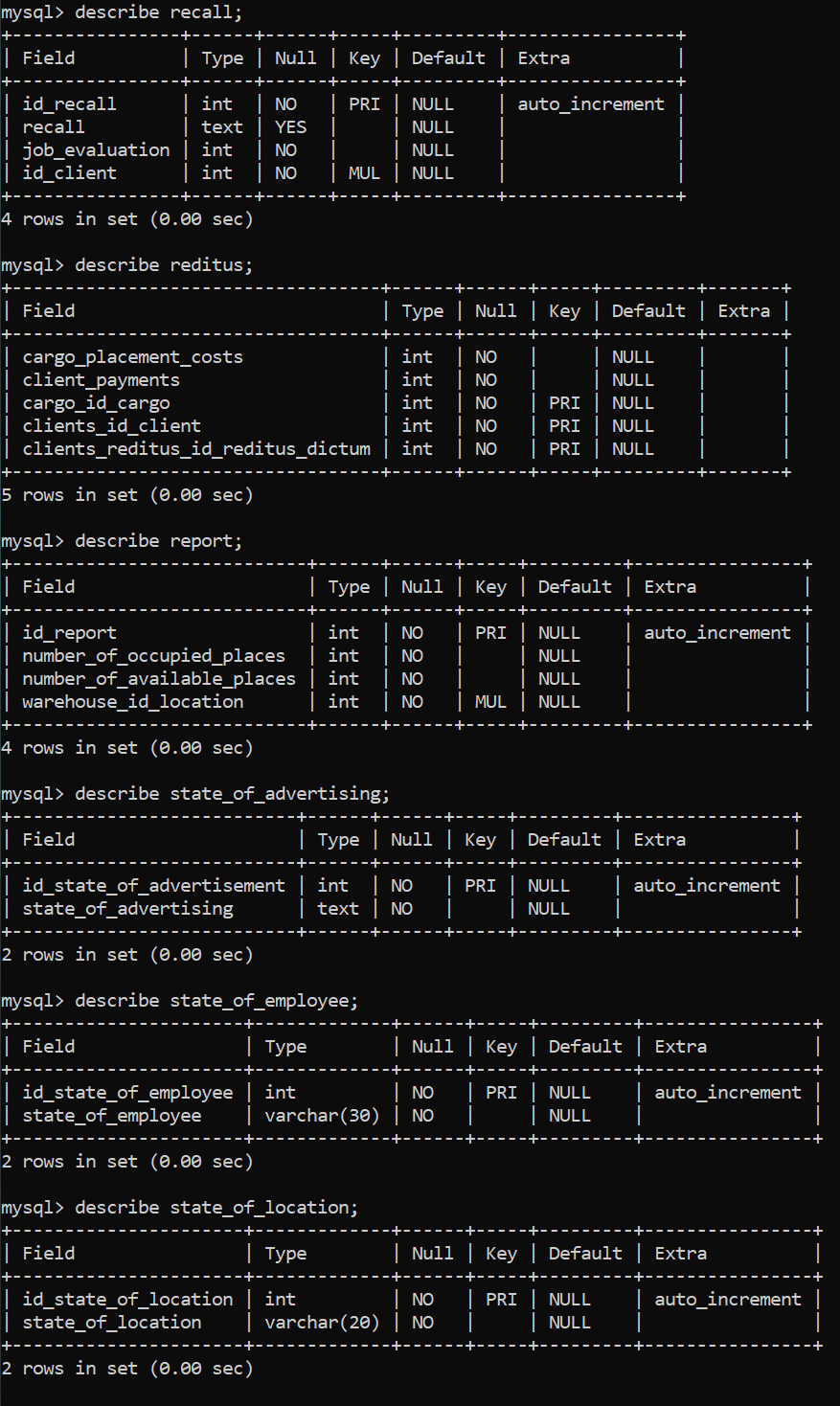


Рис. 5 – Описание созданных таблиц

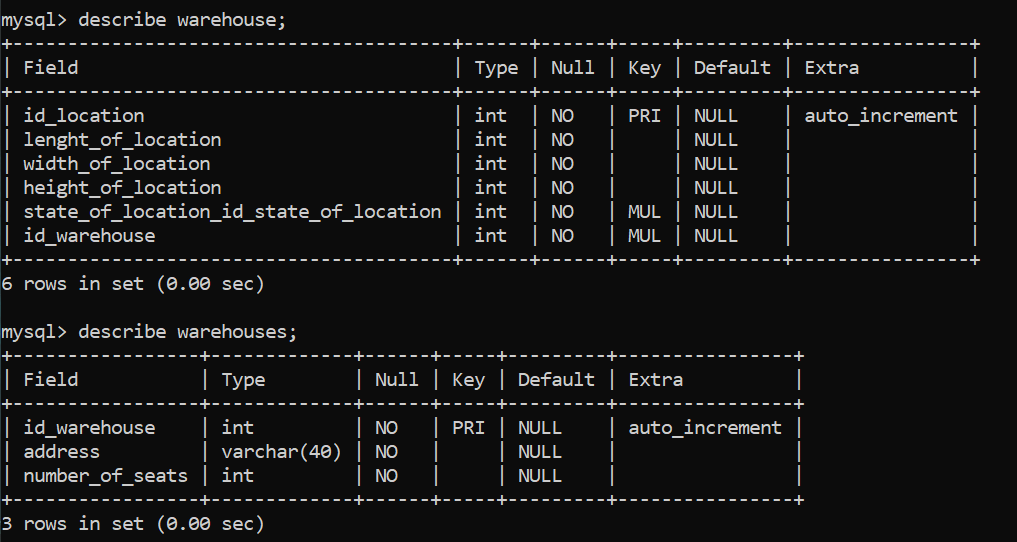


Рис. 6 – Описание созданных таблиц

Заполнение базы данных

Заполним созданную базу данных, собственно, данными.

Листинг 2 – Скрипт SQL для заполнения базы данных

|  |
| --- |
| INSERT INTO `advertisement` (`id\_advertisement`,`type\_of\_advertisement`,`text\_of\_advertisement`,`date\_of\_placing`,`id\_state\_of\_advertising`,`employee\_id\_employee`) VALUES (1,'листовка','не дорого','25.06.2022',2,1);  INSERT INTO `advertisement` (`id\_advertisement`,`type\_of\_advertisement`,`text\_of\_advertisement`,`date\_of\_placing`,`id\_state\_of\_advertising`,`employee\_id\_employee`) VALUES (2,'билборд','лучшие условия','15.02.2022',2,1);  INSERT INTO `advertisement` (`id\_advertisement`,`type\_of\_advertisement`,`text\_of\_advertisement`,`date\_of\_placing`,`id\_state\_of\_advertising`,`employee\_id\_employee`) VALUES (3,'листовка','дешево','16.08.2020',3,1);  INSERT INTO `cargo` (`id\_cargo`,`lenght`,`width`,`height`,`weight`,`cost\_of\_storage`,`id\_employee`) VALUES (1,600,500,400,1,1000,2);  INSERT INTO `cargo` (`id\_cargo`,`lenght`,`width`,`height`,`weight`,`cost\_of\_storage`,`id\_employee`) VALUES (2,200,200,100,2,500,2);  INSERT INTO `cargo` (`id\_cargo`,`lenght`,`width`,`height`,`weight`,`cost\_of\_storage`,`id\_employee`) VALUES (3,900,850,900,5,1000,2);  INSERT INTO `cargo` (`id\_cargo`,`lenght`,`width`,`height`,`weight`,`cost\_of\_storage`,`id\_employee`) VALUES (4,900,850,900,5,1000,2);  INSERT INTO `clients` (`id\_client`,`FIO`,`number\_of\_telephone`) VALUES (1,'Марченко Ю.П.','89149875562');  INSERT INTO `clients` (`id\_client`,`FIO`,`number\_of\_telephone`) VALUES (2,'Коршунов А.С.','89658456632');  INSERT INTO `clients` (`id\_client`,`FIO`,`number\_of\_telephone`) VALUES (3,'Фролов В.К.', 89149786650);  INSERT INTO `employee` (`id\_employee`,`FIO`,`wages`,`id\_post`,`id\_state\_of\_employee`) VALUES (1,'Москаленко А.А.',50000,1,1);  INSERT INTO `employee` (`id\_employee`,`FIO`,`wages`,`id\_post`,`id\_state\_of\_employee`) VALUES (2,'Семёнов В.Д. ',60000,3,1);  INSERT INTO `employee` (`id\_employee`,`FIO`,`wages`,`id\_post`,`id\_state\_of\_employee`) VALUES (3,'Волков В.С.',45000,2,2);  INSERT INTO `employee` (`FIO`,`wages`,`id\_post`,`id\_state\_of\_employee`) VALUES ('Володин В.Д',50000,1,1);  INSERT INTO `employee` (`FIO`,`wages`,`id\_post`,`id\_state\_of\_employee`) VALUES ('Самолов К.Л ',60000,3,1);  INSERT INTO `employee` (`FIO`,`wages`,`id\_post`,`id\_state\_of\_employee`) VALUES ('Карпов Н.А.',45000,2,2);  INSERT INTO `post` (`id\_post`,`post`) VALUES (1,'Менеджер по продвижению');  INSERT INTO `post` (`id\_post`,`post`) VALUES (2,'Управляющий');  INSERT INTO `post` (`id\_post`,`post`) VALUES (3,'Грузчик');  INSERT INTO `recall` (`id\_recall`,`recall`,`job\_evaluation`,`id\_сlient`) VALUES (1,'good',5,1);  INSERT INTO `recall` (`id\_recall`,`recall`,`job\_evaluation`,`id\_сlient`) VALUES (2,'отлично',5,2);  INSERT INTO `recall` (`id\_recall`,`recall`,`job\_evaluation`,`id\_сlient`) VALUES (3,'могло быть и лучше, го все равно хорошо',4,3);  INSERT INTO `reditus` (`cargo\_placement\_costs`,`client\_payments`,`cargo\_id\_cargo`,`clients\_id\_client`,`clients\_reditus\_id\_reditus\_dictum`) VALUES (500,900,1,1,1);  INSERT INTO `reditus` (`cargo\_placement\_costs`,`client\_payments`,`cargo\_id\_cargo`,`clients\_id\_client`,`clients\_reditus\_id\_reditus\_dictum`) VALUES (600,1200,2,1,2);  INSERT INTO `reditus` (`cargo\_placement\_costs`,`client\_payments`,`cargo\_id\_cargo`,`clients\_id\_client`,`clients\_reditus\_id\_reditus\_dictum`) VALUES (900,2000,3,2,NULL);  INSERT INTO `report` (cargo\_id\_cargo,warehouses\_id\_warehouse,warehouse\_id\_location) VALUES (1,1,3);  INSERT INTO `state\_of\_advertising` (`id\_state\_of\_advertisement`,`state\_of\_advertising`) VALUES (1,'в разработке');  INSERT INTO `state\_of\_advertising` (`id\_state\_of\_advertisement`,`state\_of\_advertising`) VALUES (2,'опубликована');  INSERT INTO `state\_of\_advertising` (`id\_state\_of\_advertisement`,`state\_of\_advertising`) VALUES (3,'снята с производства');  INSERT INTO `state\_of\_employee` (`id\_state\_of\_employee`,`state\_of\_employee`) VALUES (1,'работает');  INSERT INTO `state\_of\_employee` (`id\_state\_of\_employee`,`state\_of\_employee`) VALUES (2,'в отпуске');  INSERT INTO `state\_of\_employee` (`id\_state\_of\_employee`,`state\_of\_employee`) VALUES (3,'на больничном');  INSERT INTO `state\_of\_location` (`id\_state\_of\_location`,`state\_of\_location`) VALUES (1,'свободно');  INSERT INTO `state\_of\_location` (`id\_state\_of\_location`,`state\_of\_location`) VALUES (2,'занято');  INSERT INTO `warehouse` (`lenght\_of\_location`,`width\_of\_location`,`height\_of\_location`,`state\_of\_location\_id\_state\_of\_location`, id\_warehouse) VALUES (1000,1000,1000,2,1);  INSERT INTO `warehouse` (`lenght\_of\_location`,`width\_of\_location`,`height\_of\_location`,`state\_of\_location\_id\_state\_of\_location`, id\_warehouse) VALUES (1000,1000,1000,2,1);  INSERT INTO `warehouse` (`lenght\_of\_location`,`width\_of\_location`,`height\_of\_location`,`state\_of\_location\_id\_state\_of\_location`, id\_warehouse) VALUES (1200,1000,1000,2,1);  INSERT INTO `warehouses` (`id\_warehouse`,`address`,`number\_of\_seats`) VALUES (1,'г. Москва ул Победы 23а',50); |

Создание запросов на получение данных

Напишем несколько простых запросов к базе данных используя конструкцию SELECT.

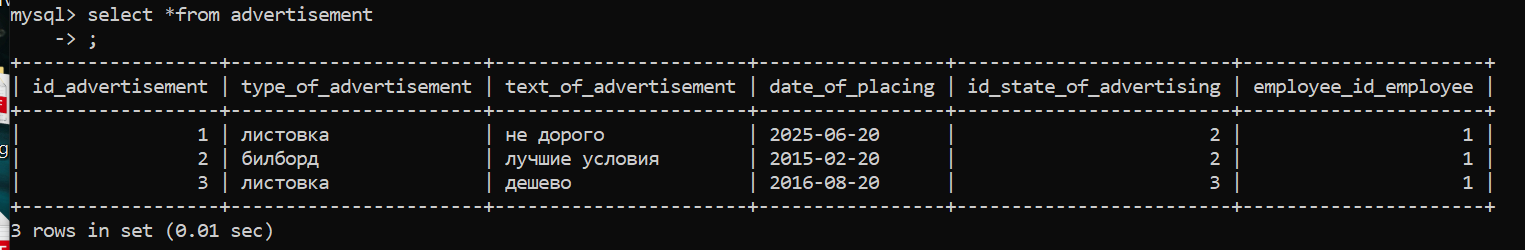


Рис. 7 – Запрос к таблице

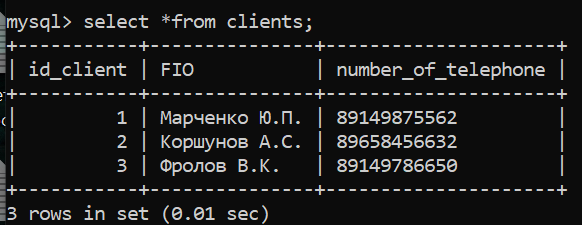


Рис. 8 – Запрос к таблице

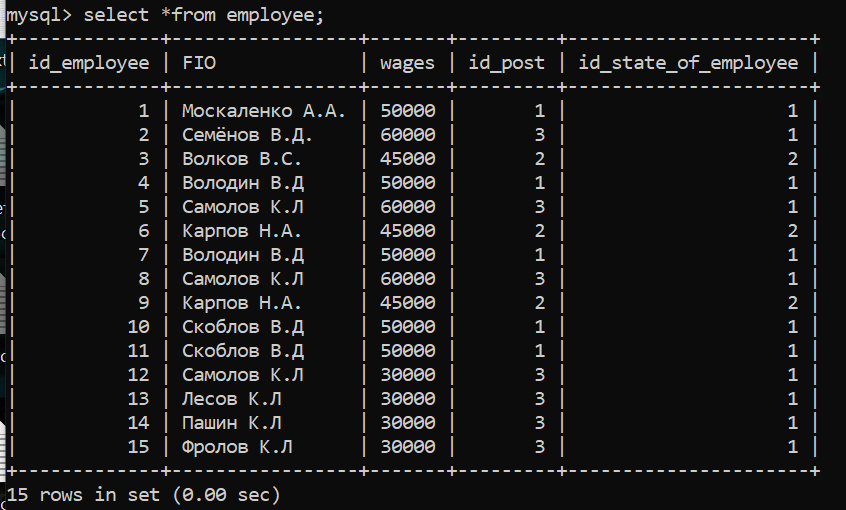


Рис. 9 – Запрос к таблице

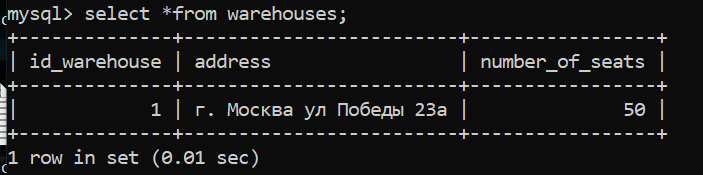


Рис. 10 – Запрос к таблице

Из запроса на рисунке 9 непонятно, какие именно данные содержит таблица, поэтому здесь нам потребуются более сложные запросы, включающие ключевое слово JOIN.

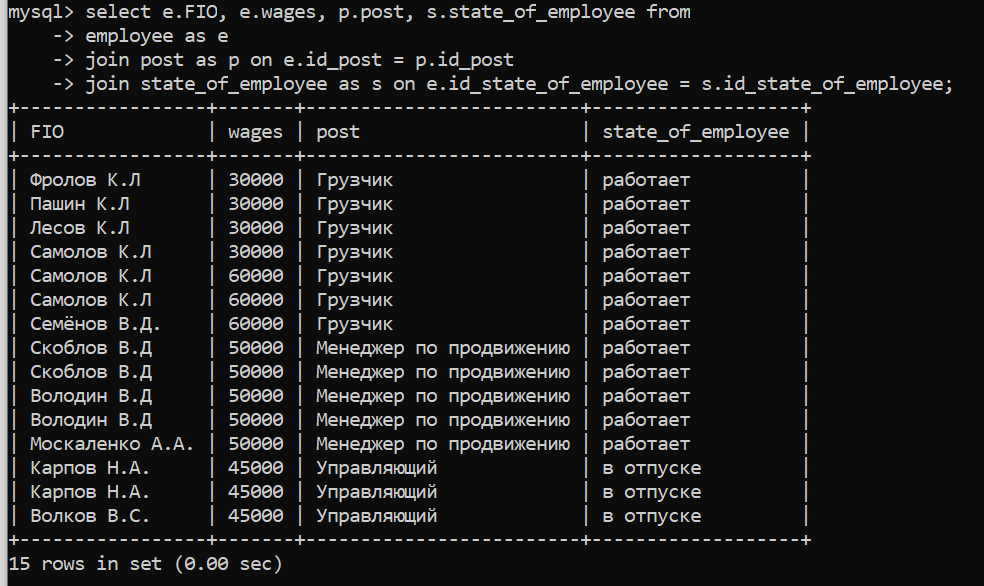
****

Рис. 11 - Запрос, использующий конструкцию *join*

Также с помощью конструкции *where* можно выводить только требуемые нам столбцы

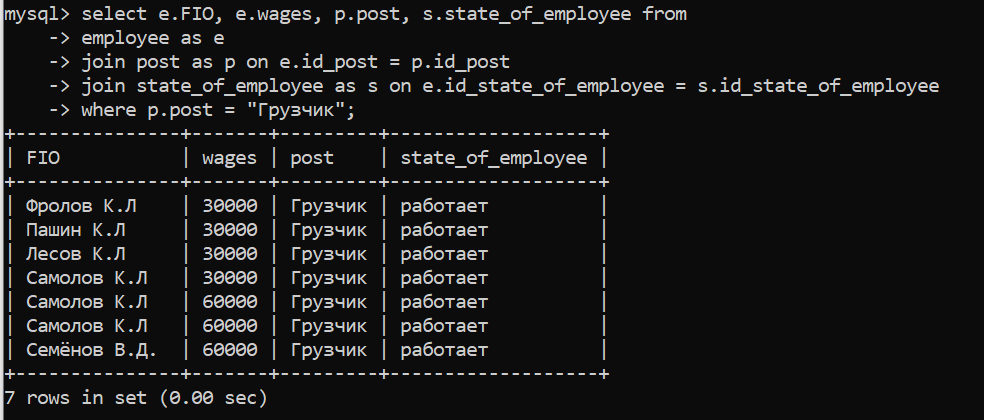
****

Рис. 12 - Запрос с ключевым словом *where*

Давайте также опробуем конструкцию вложенных запросов. Выведем ФИО сотрудников, которые занимают должность «Управляющий».

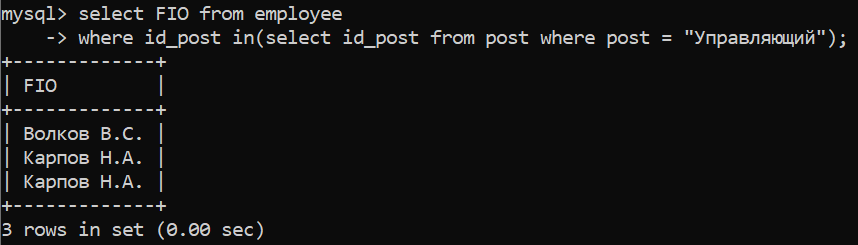


Рис. 13 - Конструкция вложенных запросов

Давайте попробуем изменить данные внутри таблицы используя команды SET и WHERE.

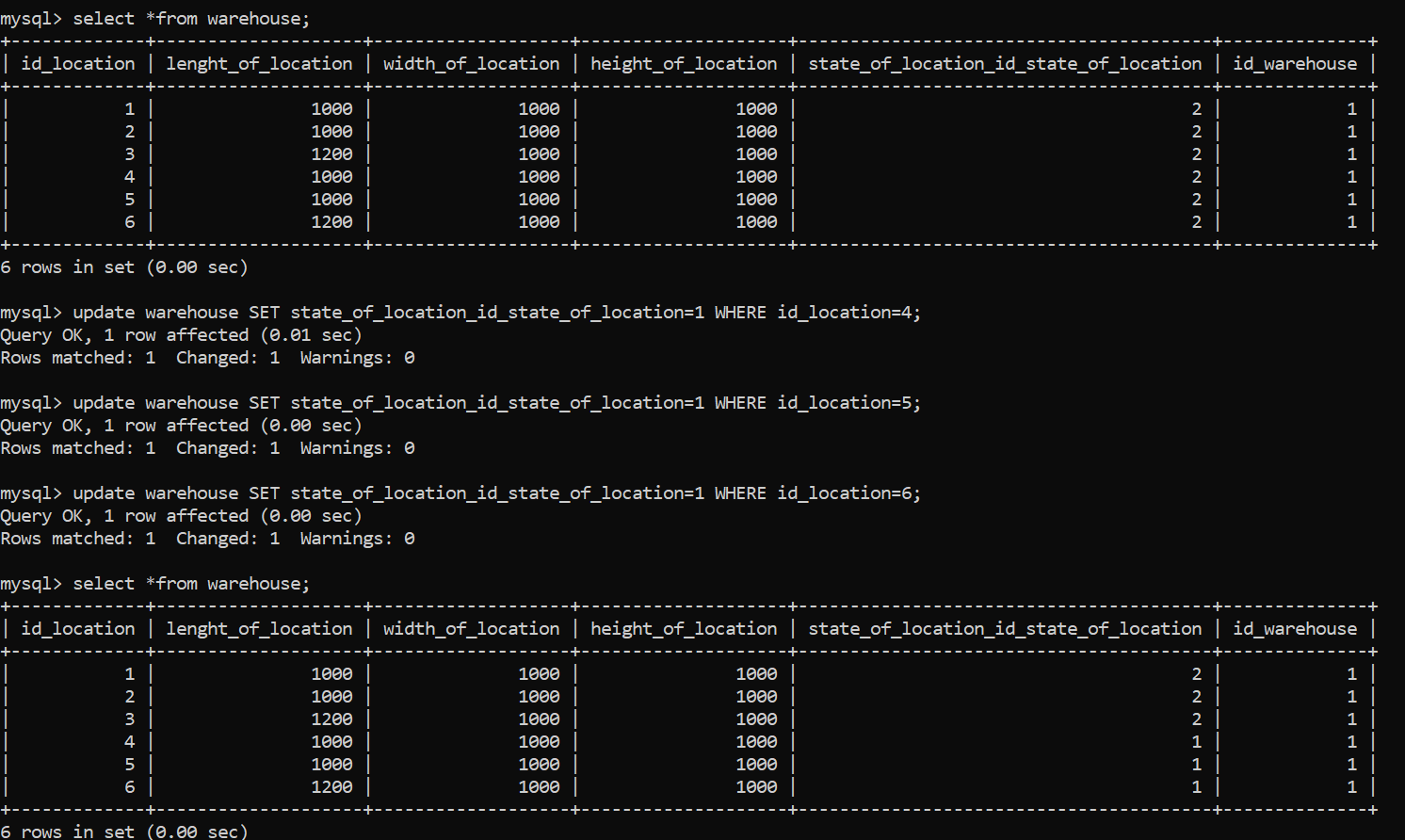


Рис. 14 – Изменение данных

Хранимые процедуры, функции и триггеры

Перейдём к созданию хранимых процедур. Создавать процедуры будем в среде Workbench, а вызывать непосредственно в консоли.

Процедуры нужны для упрощения работы с таблицей. Когда часто требуется писать трудные запросы, их легче поместить в процедуру и уже вызывать её. Давайте напишем сложных запрос в хранимую процедуру и посмотрим на её выполнение.

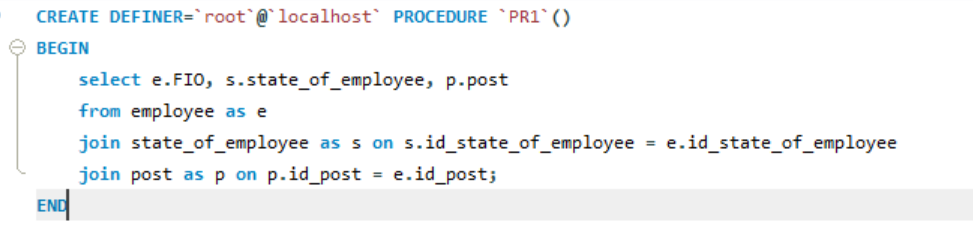


Рис. 15 – Код процедуры

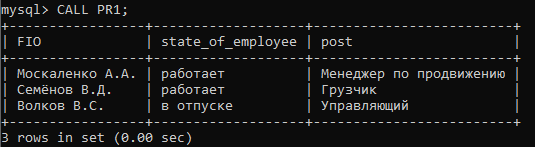


Рис. 16 - Вызов процедуры

В процедуры можно передавать аргументы, и получать значения из них.

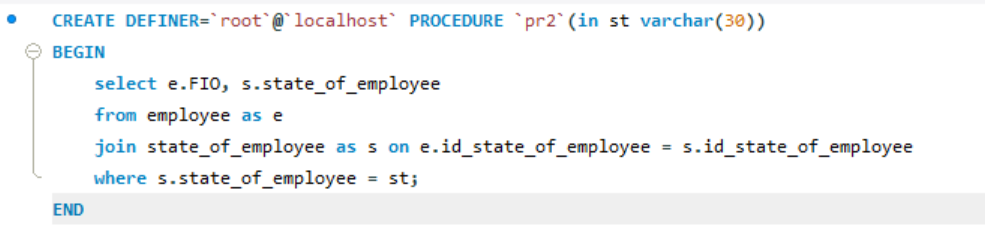


Рис. 17 – Код процедуры

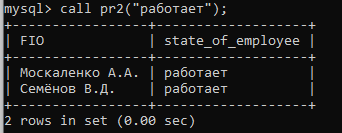


Рис. 18 – Вызов процедуры

Теперь напишем триггер. Предположим, что наши кинотеатры имеют ограниченное количество вакансий на одну должность (не более 2-х человек на должность), поэтому при попытке добавления нового сотрудника будет проверяться сколько сотрудников уже состоит на данной должности. Если количество сотрудников уже равняется трём, то вставка в таблицу будет запрещена.

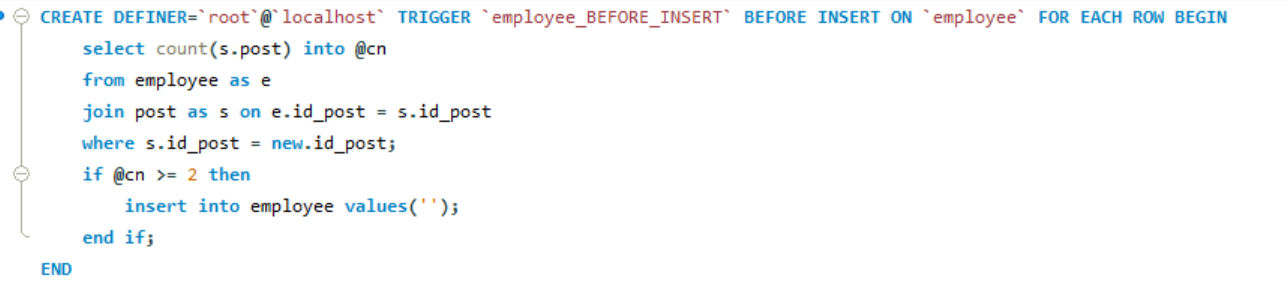


Рис. 19 – Код триггера

В теле триггера мы считаем какое кол-во сотрудников у нас есть на данный момент с конкретной должностью и если их количество больше 2, то выполняется заведомо неверная команда.

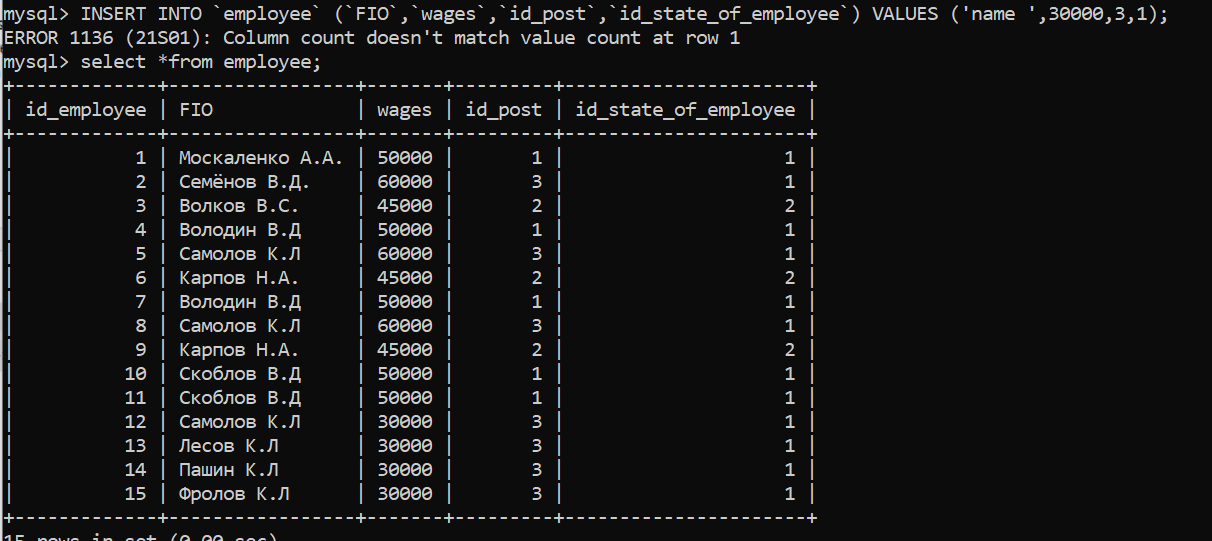


Рис. 20 – Выполнение триггера

Создадим функцию. Отличие функции от процедуры в том, что функция всегда должна возвращать какое-то значение (что подчёркивается путём её вызова), и в ней нельзя проводить выборку данных с последующим выводом.

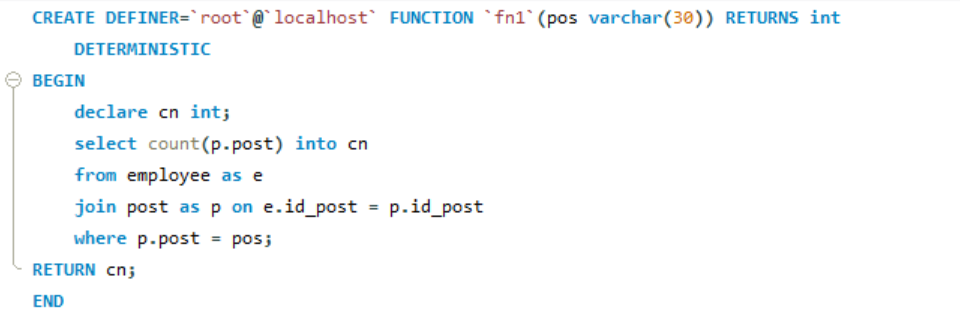


Рис. 21 – Код функции

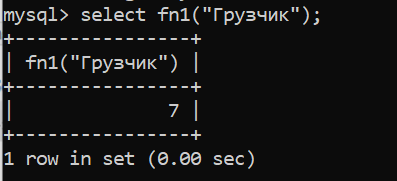


Рис. 22 – Результат выполнения функции

Вывод

В процессе выполнения данных практических работ были получены умения построения логической и физической базы данных. Изучен язык запросов SQL, а именно: создание таблиц, вставка и удаление данных, изменение данных, выборка данных из таблиц. Также было изучено как создавать свои хранимые процедуры, триггеры и функции.

Список используемой литературы

1. Лекции РТУ МИРЭА по курсу «Разработка баз данных», Богомольная Г.В.
2. Методические материалы по SQL