|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования  **"МИРЭА — Российский технологический университет"**  **РТУ МИРЭА**   |  | | --- | | Институт информационных технологий (ИИТ) | | Кафедра промышленной информатики (ПИ) | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ** | | | | |
|  | | | | |
| по дисциплине «Разработка баз данных» | | | | |
| **Студент группы** | ИМБО-02-22 |  |  |
|  | *(учебная группа, фамилия, имя, отчество студента)* |  | *(подпись студента)* |
| **Преподаватель** | Ст. Преподаватель Баев И.Б. |  |  |
| *(должность, ученая степень, звание, фамилия, имя, отчество преподавателя) (подпись преподавателя)* | | | |

|  |  |
| --- | --- |
| Работа выполнена | «\_\_» \_\_\_\_ 2024 г. |
| «Зачтено» | «\_\_» \_\_\_\_ 2024 г. |
|  |  |

Москва 2024

Содержание

[Описание предметной области 3](#_Toc181868419)

[физическая модель базы д анных 4](#_Toc181868420)

[Результата выполнения практических работ в SQL Command Line 5](#_Toc181868421)

[Результат выполнения практической работы по использованию СУБД 33](#_Toc181868422)

[Выборка данных (DQL) 33](#_Toc181868423)

[Процедуры 38](#_Toc181868424)

[Функции 42](#_Toc181868425)

[Триггеры 47](#_Toc181868426)

[Оконные функции 50](#_Toc181868427)

[Заключение 56](#_Toc181868428)

[Список использованных источников 57](#_Toc181868429)

Описание предметной области

В данной работе была рассмотрена

физическая модель базы д анных

На Рисунке 1 представлена физическая модель рассматриваемой предметной области.

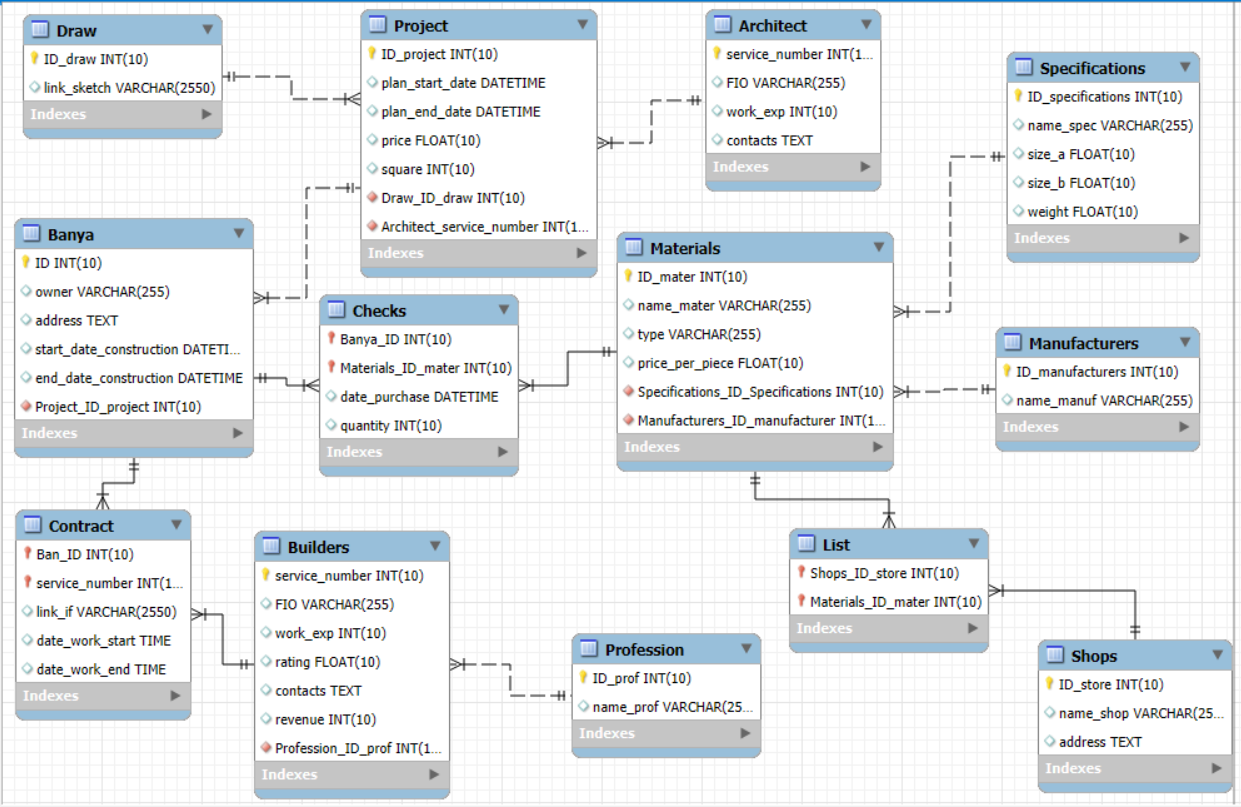


Рисунок 1 — Физическая модель предметной области

Результата выполнения практических работ в SQL Command Line

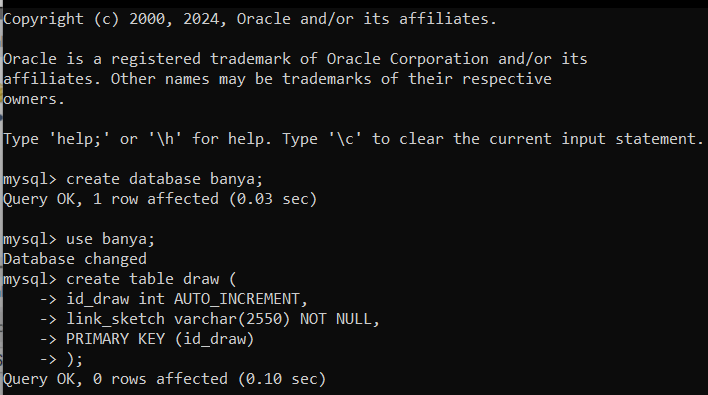
Для начала создадим БД banya. Результат представлен на Рисунке 2.

Рисунок 2 — Создание базы данных

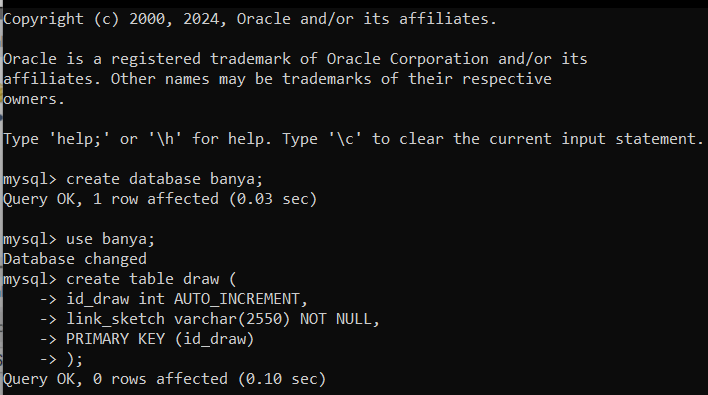
Для нашей работы выберем нашу БД banya. Результат представлен на Рисунке 3.

Рисунок 3 — Использование базы данных

Теперь необходимо создать и заполнить все необходимые таблицы. Каждый из столбцов имеет свой тип данных и ограничение. Начнем с центральной для данной задачи таблицы draw, которая содержит информацию о чертежах. Она включает в себя столбцы для уникального идентификатора каждого чертежа (id\_draw), ссылка на чертежи (link\_sketch).

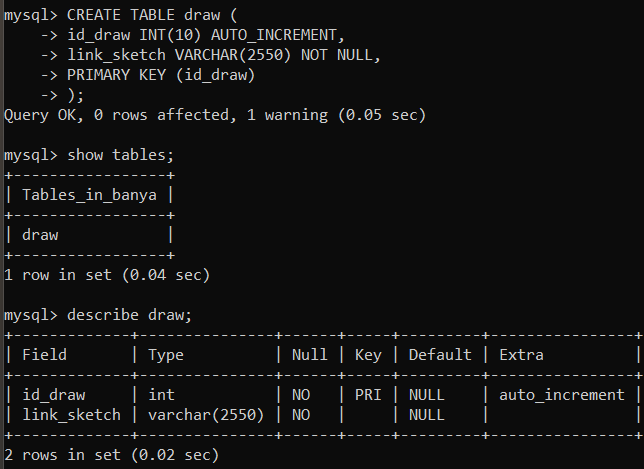


Рисунок 4 — Создание таблицы draw

В таблице architect содержит информацию об нанятых архитекторах, которые делают проекты. Она содержит столбцы служебный номер (service\_number), ФИО (fio), опыт работы в годах (work\_exp), контакты (contacts)

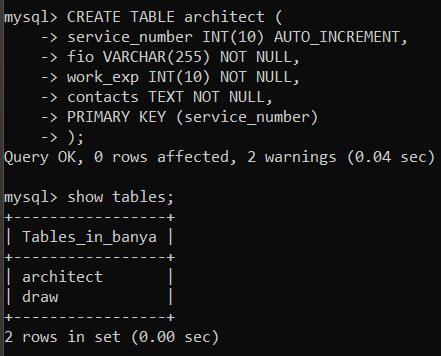


Рисунок 5 — Создание таблицы architect

Таблица profession хранятся название профессии. Она содержит столбцы для идентификатора професии (id\_prof), название профессии (name\_prof).

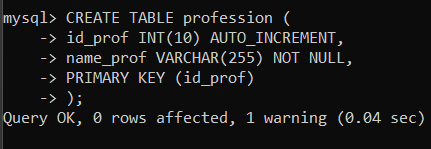


Рисунок 6 — Создание таблицы profession

Таблица shops предназначена для местонахождении магазин стройматериалов. Она включает в себя столбцы для уникального идентификатора магазина (id\_store), название магазина (name\_shop), и его адреса (address).

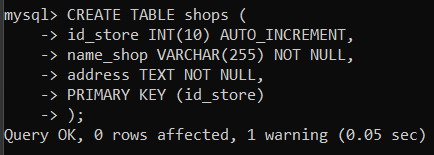


Рисунок 7 — Создание таблицы shops

Таблица manufacturers содержит название производителей купленных или некупленных материалов. Она включает столбцы для идентификаторов производителей (id\_manufacturers), названия производителя (name\_manuf).

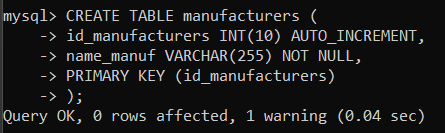


Рисунок 8 — Создание таблицы manufacturers

Таблица specifications хранит данные о характеристиках материалов. Она включает столбцы для идентификатора материала (id\_specifications), названия (name\_spec), size\_a\_metr (длина), size\_b\_metr (ширина), weight\_kilo (вес). Эта таблица обеспечивает подробную информацию о материалах.

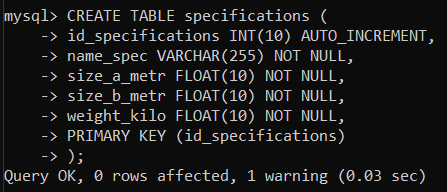


Рисунок 9 — Создание таблицы specifications

Таблица project содержит информацию об планах постройки. Она включает столбцы для идентификаторов проекта (id\_project), начало планирования постройки (plan\_start\_date), конец постройки по плану (plan\_end\_date), площадь (square), цена проекта (price), уникальные чертежи (draw\_id\_draw), сервисный номер архитектора (architect\_service\_number).

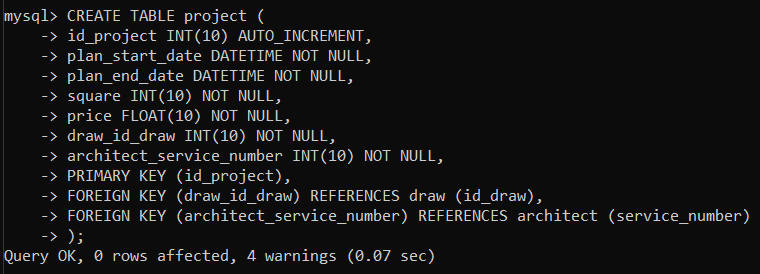


Рисунок 10 — Создание таблицы project

Таблица banya сведения о владельце бани. Она включает столбцы для уникального идентификатора владельца (id), имя владельца (owner), адрес, где находится баня (address), дата начало постройки (start\_date\_constuction), дата окончания постройки, идентификанция проекта (project\_id).

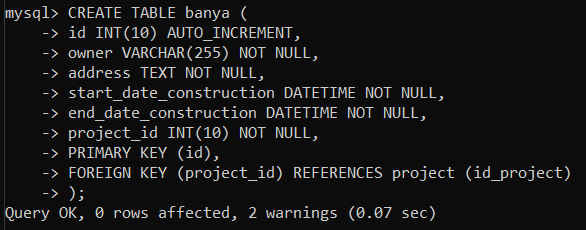


Рисунок 11 — Создание таблицы banya

Таблица materials содержит информацию о материалах,. Она включает столбцы для уникального идентификатора материала (id\_mater), названия материала (name\_mater), тип (type), цена за одну штуку (price\_per\_piece), идентификатор характеристики (specifications\_id), идентификатор производителя (manufacturers\_id).

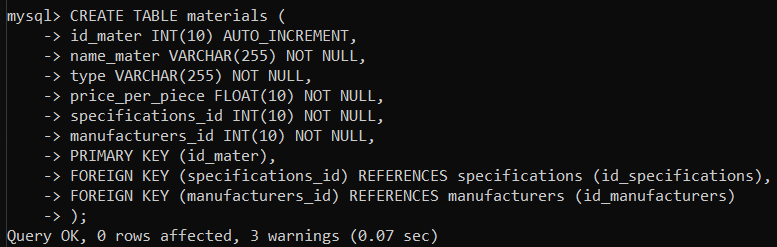


Рисунок 12 — Создание таблицы materials

Таблица list связывает материалы и магазины. Она включает столбцы для идентификаторов магазина и материала.

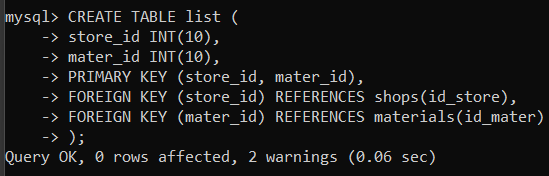


Рисунок 13 — Создание таблицы list

Таблица checks содержит данные о купленных материалах. Она включает столбцы для идентификаторов бани и материала, дата покупки (data\_purchase), в каком количестве купили (quantity).

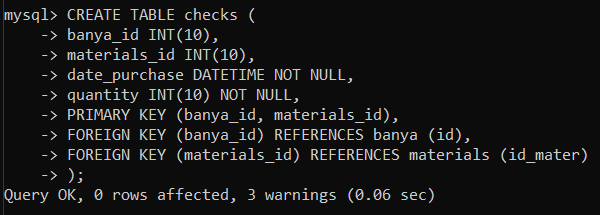


Рисунок 14 — Создание таблицы checks

Таблица builders хранит данные о всех строителях. Она включает столбцы для уникального номера строителя (service\_number), ФИО (fio), опыт работы (work\_exp), контакты (contacts), зарплата (revenue), идентификатор професии (profession\_id).

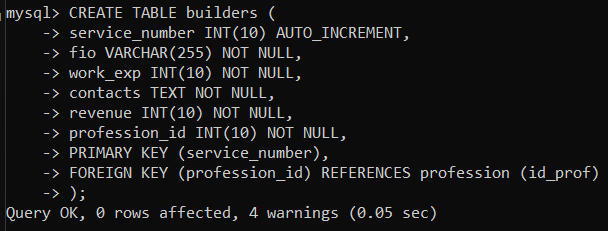


Рисунок 15 — Создание таблицы builders

Таблица contract содержит все договоры. Она включает столбцы для уникального идентификатора бани (ban\_id), идентификатора строителя (service\_number), ссылка на условия работы (link\_if), начало рабочей смены (date\_work\_start), конец рабочей смены (date\_work\_end).

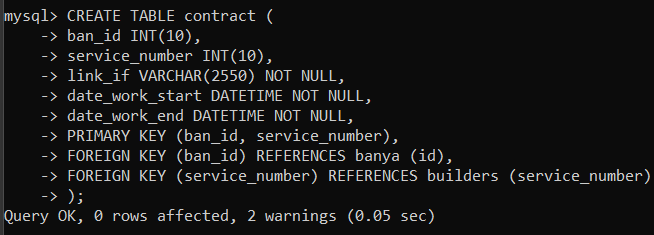


Рисунок 16 — Создание таблицы contract

Произведем просмотр созданных таблиц в базе данных.

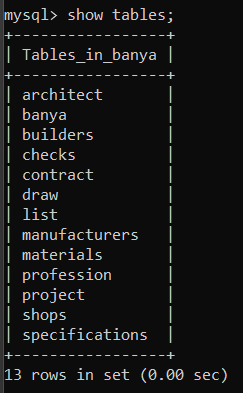


Рисунок 17 — Просмотр таблиц базы данных banya

Внесем данные в таблицы shops и architect. Для этого введем представленные ниже команду.

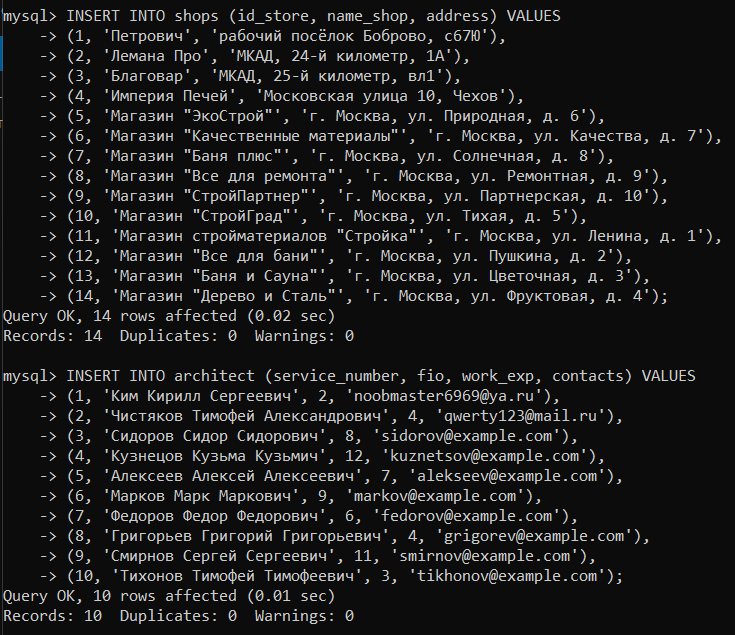


Рисунок 18 — Заполнение таблиц shops и architect

Внесем данные в таблицы manufacturers и profession. Для этого введем представленные ниже команду.

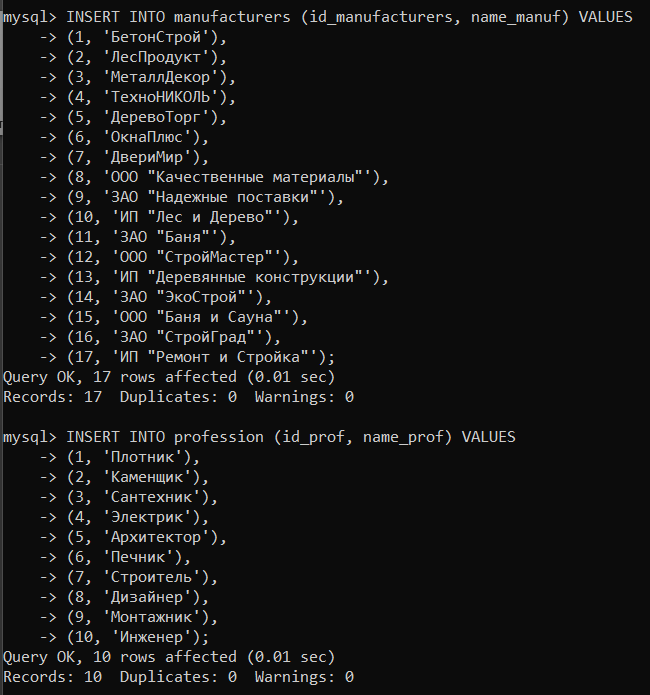


Рисунок 19 — Заполнение таблиц manufacrurers и profession

Внесем данные в таблицы draw и specifications. Для этого введем представленные ниже команду.

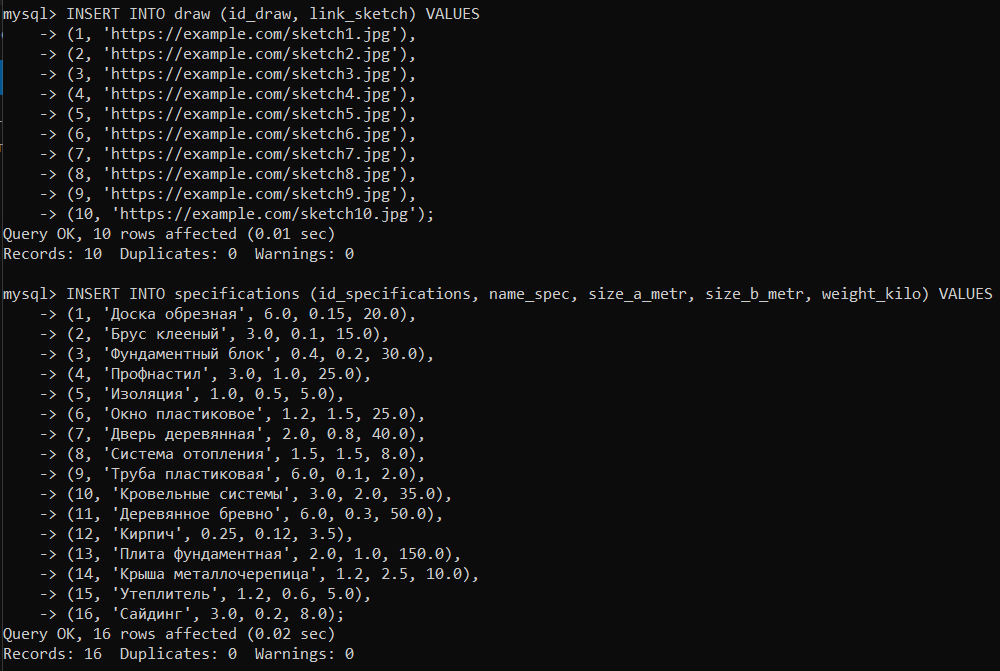


Рисунок 20 — Заполнение таблиц draw и specifications

Внесем данные в таблицы project и banya. Для этого введем представленные ниже команду.

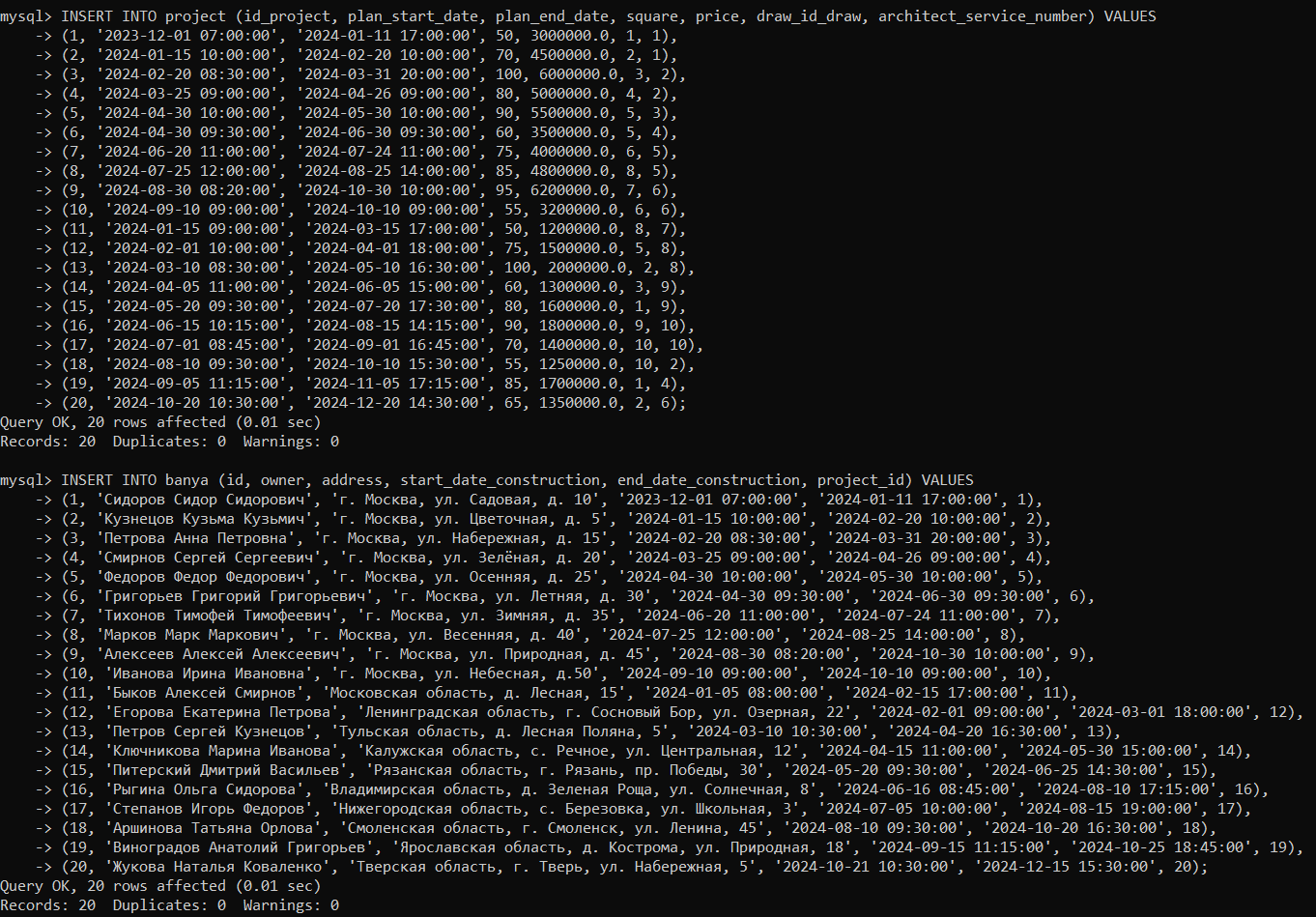


Рисунок 21 — Заполнение таблиц project и banya

Внесем данные в таблицу materials. Для этого введем представленные ниже команду.

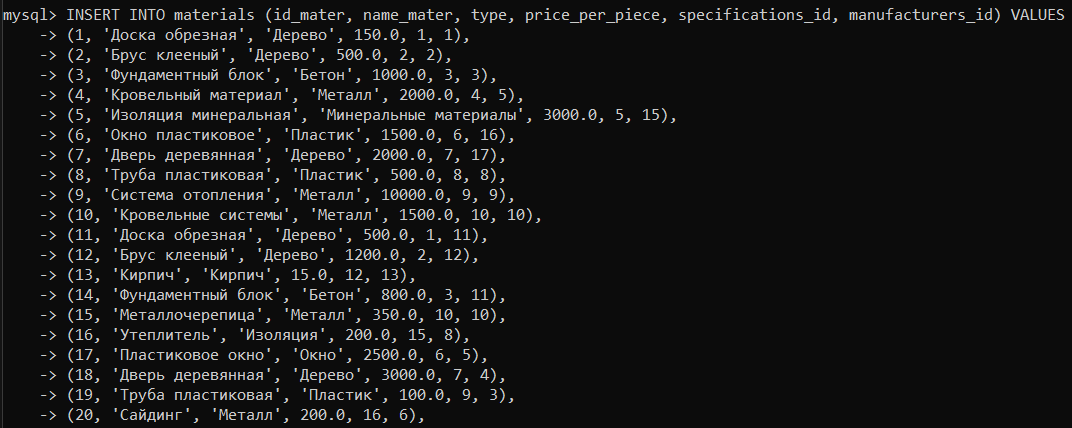


Рисунок 22 — Заполнение таблиц materials

Внесем данные в таблицы list. Для этого введем представленные ниже команду.

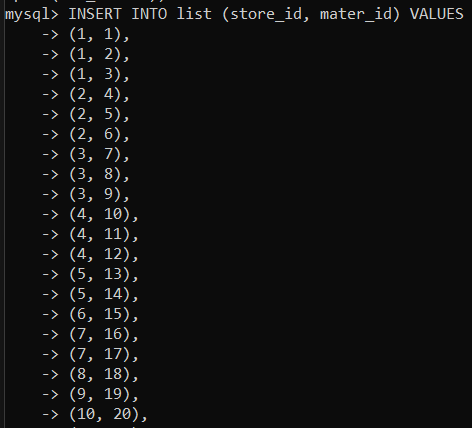


Рисунок 23 — Заполнение таблиц list

Внесем данные в таблицу checks. Для этого введем представленные ниже команду.

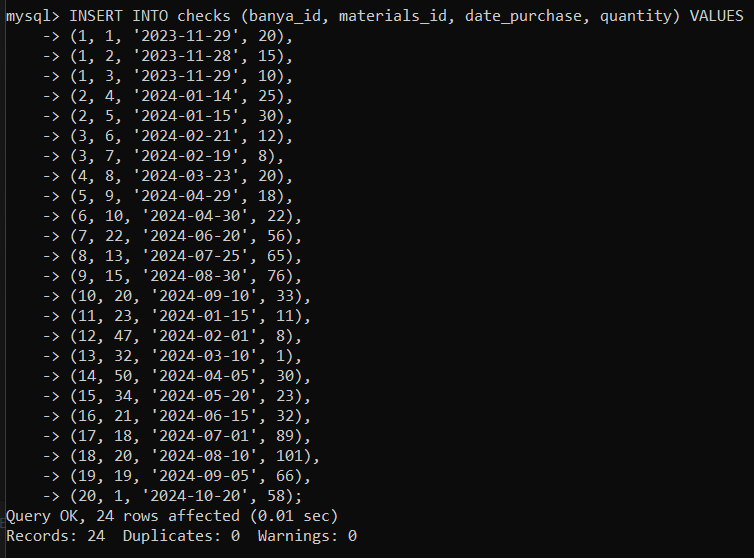


Рисунок 24 — Заполнение таблицы checks

Внесем данные в таблицы builders и contract. Для этого введем представленные ниже команду.

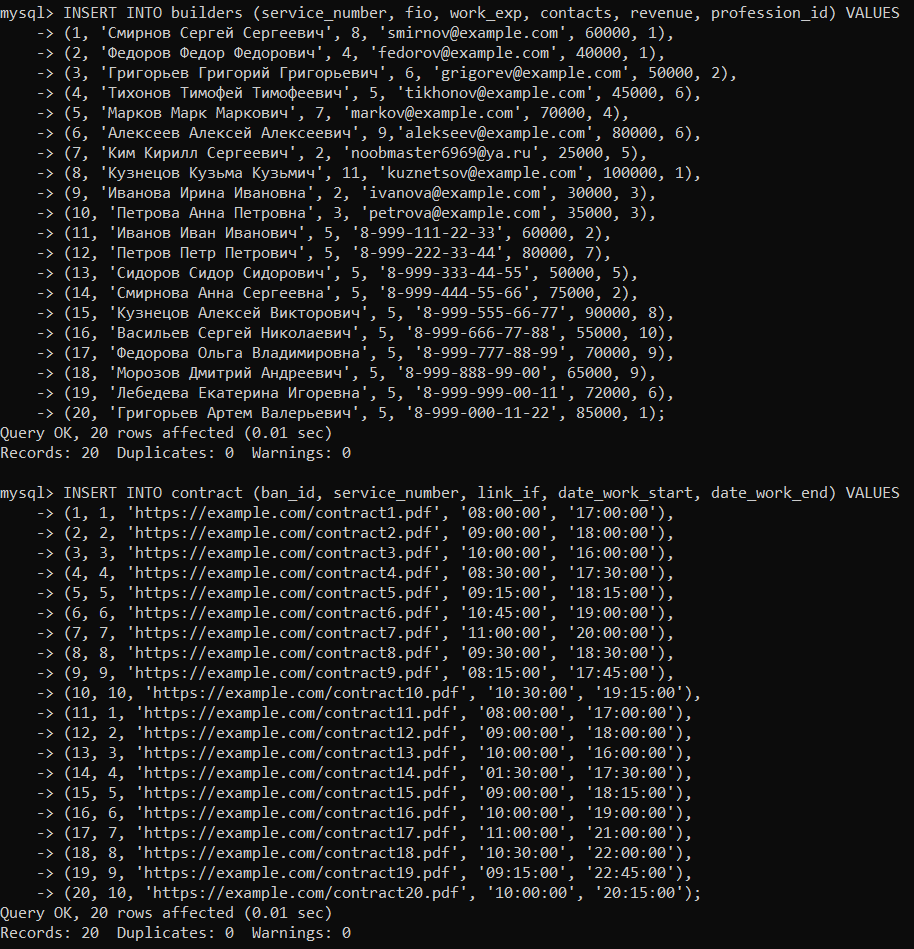


Рисунок 25 — Заполнение таблицы builders и contract

Просмотрим все таблицы после заполнения при помощи команды SELECT \* FROM «название таблицы»;

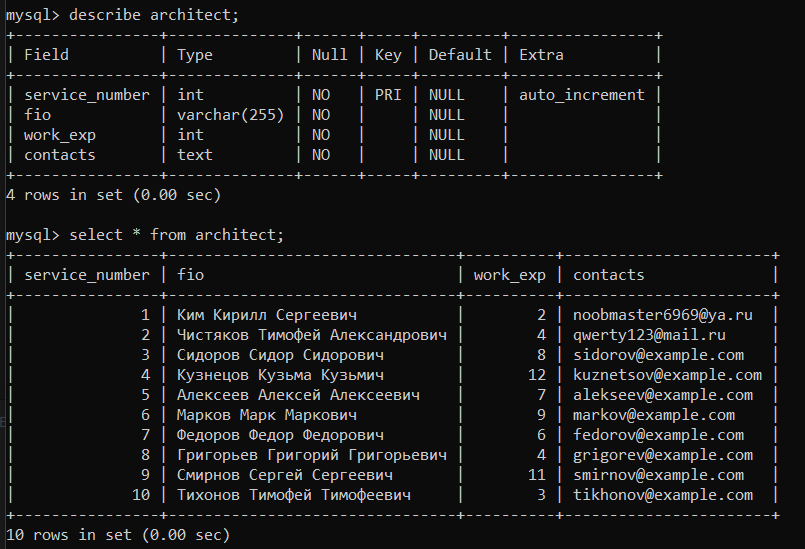


Рисунок 26 — Содержание таблицы architect

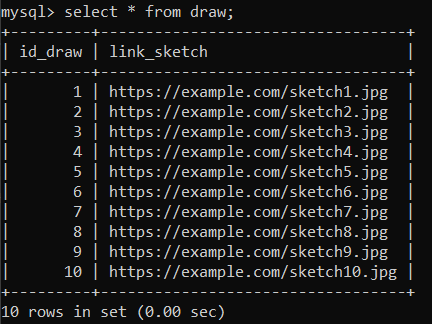


Рисунок 27 — Содержание таблицы draw

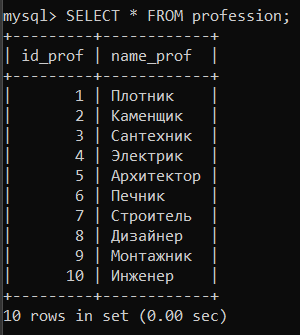


Рисунок 28 — Содержание таблицы profession

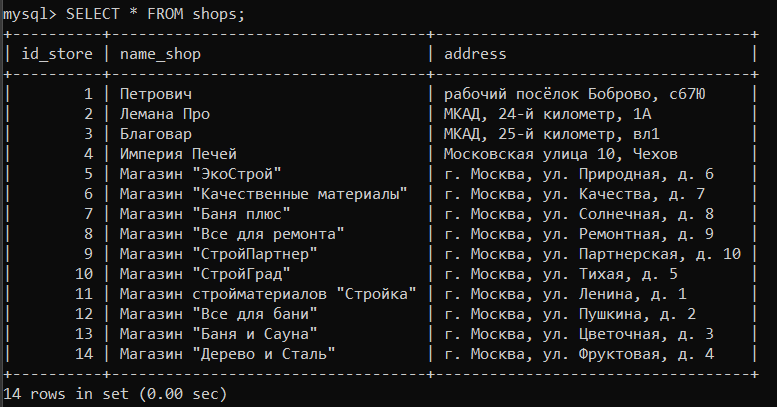


Рисунок 29 — Содержание таблицы shops

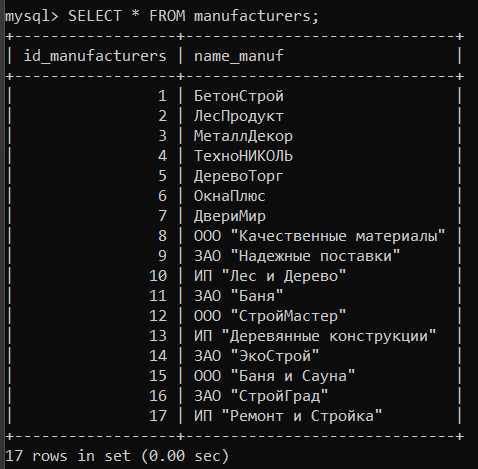


Рисунок 30 — Содержание таблицы manufacturers

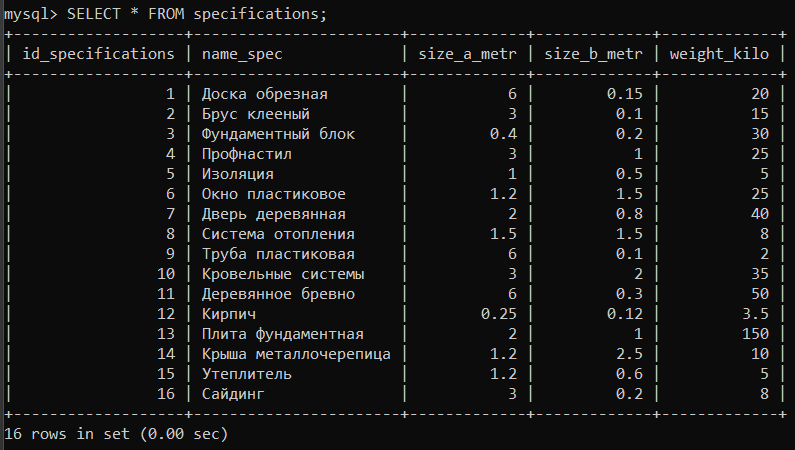


Рисунок 31 — Содержание таблицы specifications

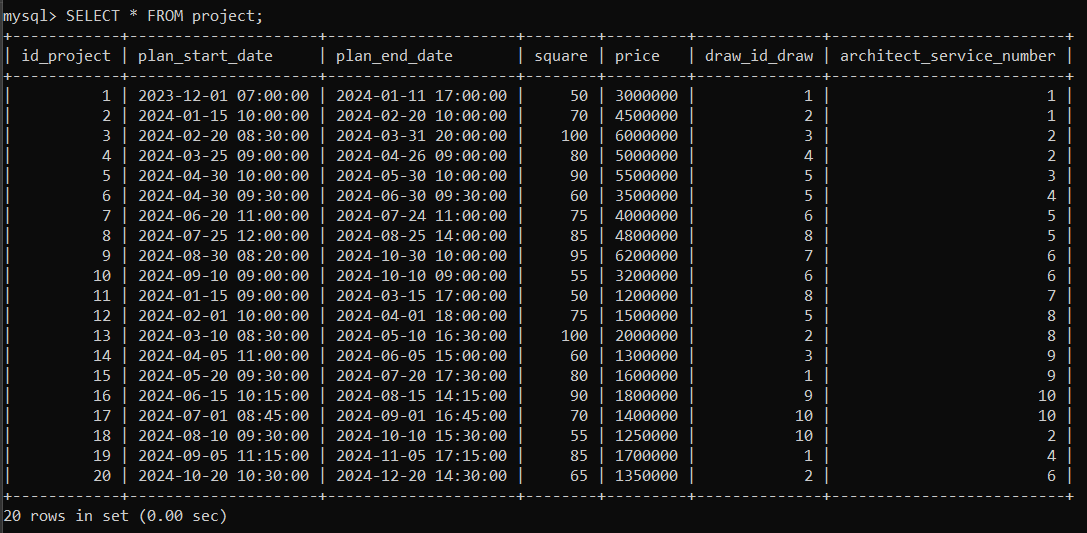


Рисунок 32 — Содержание таблицы project

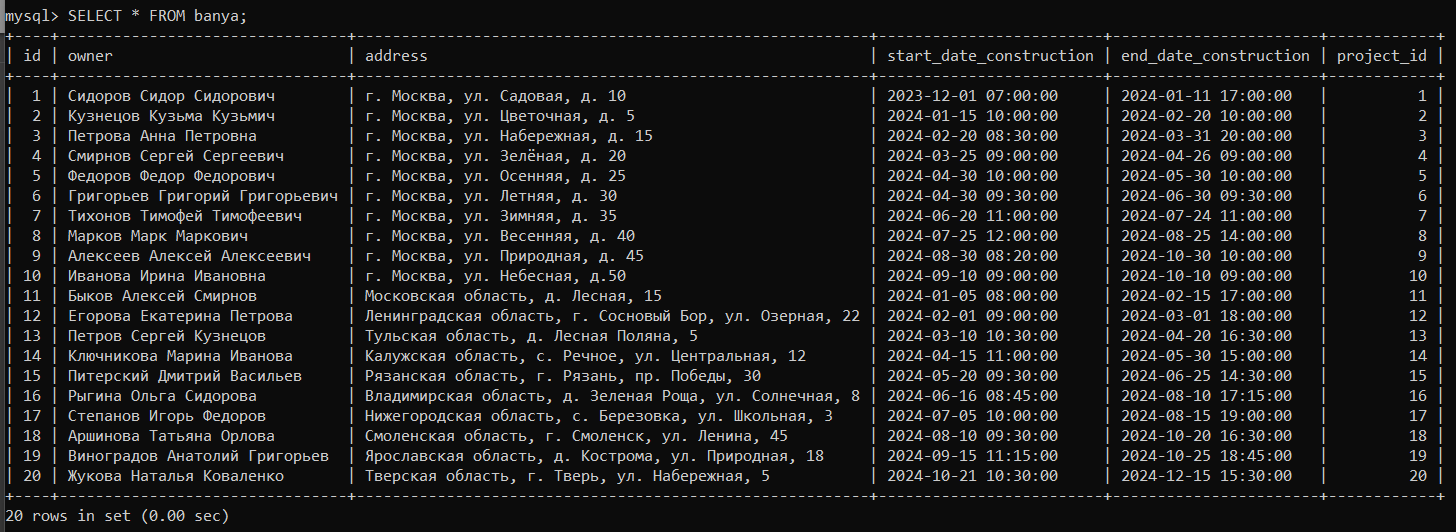


Рисунок 33 — Содержание таблицы banya

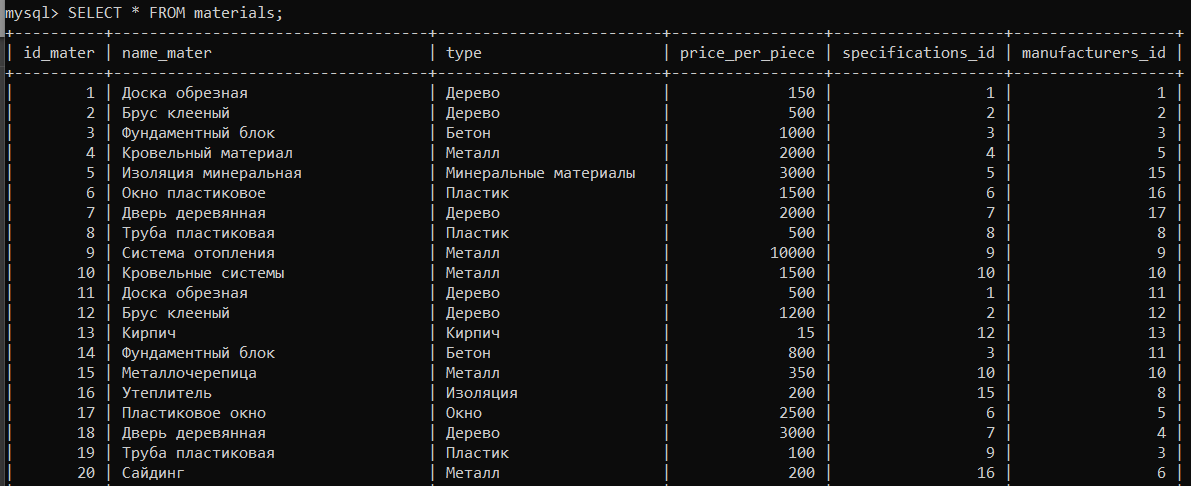


Рисунок 34 — Содержание таблицы materials

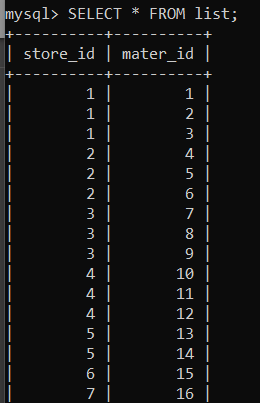


Рисунок 35 — Содержание таблицы list

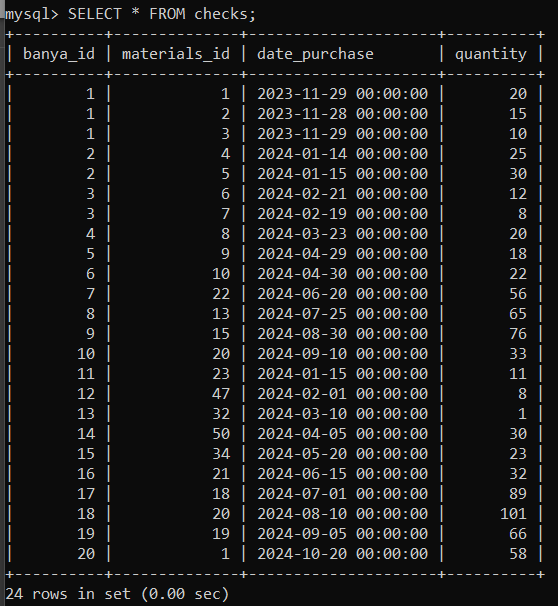


Рисунок 36 — Содержание таблицы checks

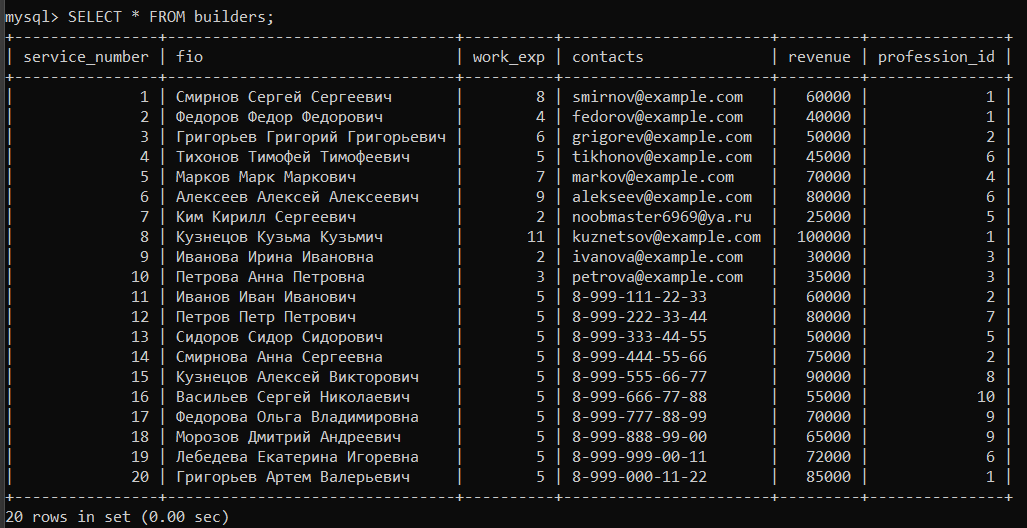


Рисунок 36 — Содержание таблицы builders

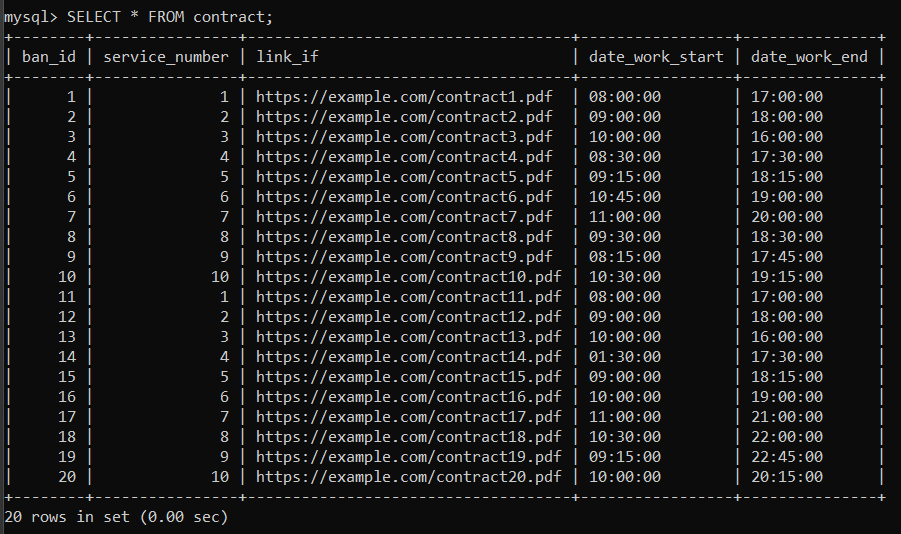


Рисунок 37 — Содержание таблицы contract

Просмотр столбца из таблицы при помощи команды SELECT «имя» FROM «имя»;

SELECT fio FROM architect;

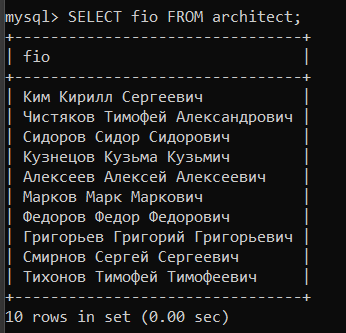


Рисунок 38 — Вывод столбца fio

Просмотр несколько столбцов из таблицы при помощи команды SELECT «имя», «имя» FROM «имя»;

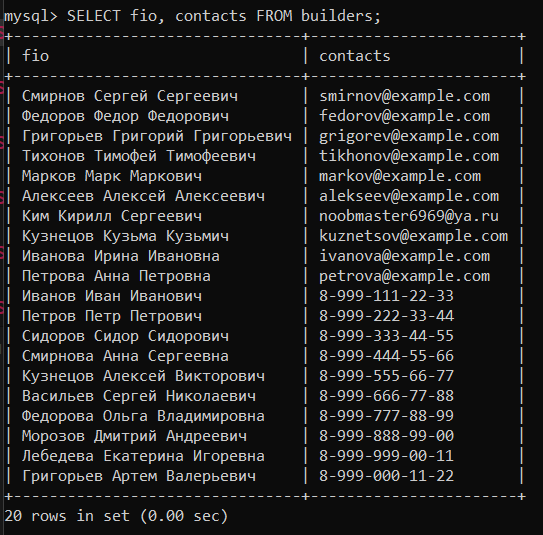


Рисунок 39 — Вывод столбцов fio и contacts строителей

Сортировка столбца по фамилиям по возрастанию при помощи команды SELECT \* FROM имя\_таблицы ORDER BY имя\_столбца\_сортировки;



Рисунок 40 — Сортировка материалов в порядке убывания

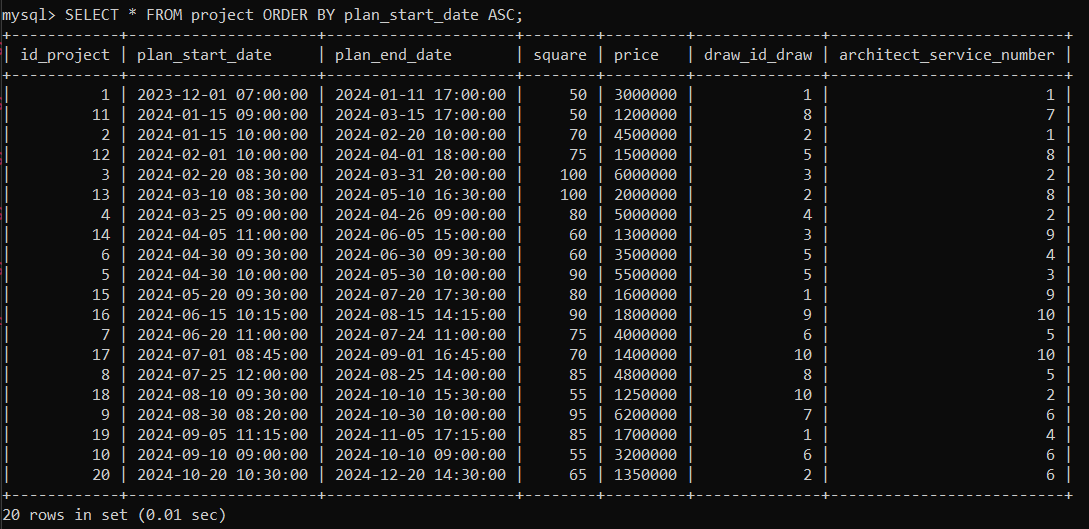


Рисунок 41 — Сортировка проектов по дате начало планирования в порядке возрастания

Очень часто бывает, что все информация из таблицы не нужна. Например, необходимо узнать, архитектор, который имеет опыт работы «4 года». Для этого в SQL есть ключевое слово WHERE. Отсортируем значения в столбцах по заданному условию.

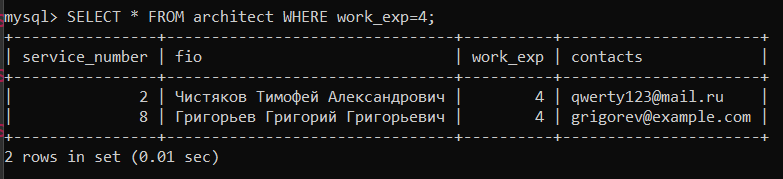


Рисунок 42 — Условие для таблицы architect

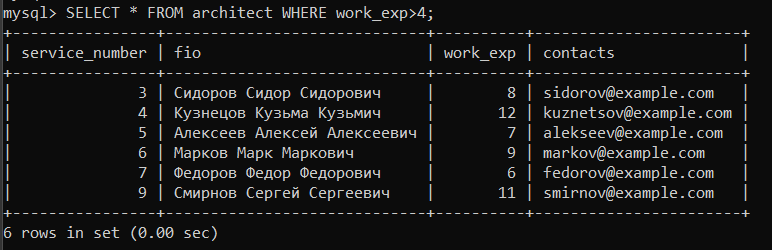


Рисунок 43 — Условие для таблицы architect

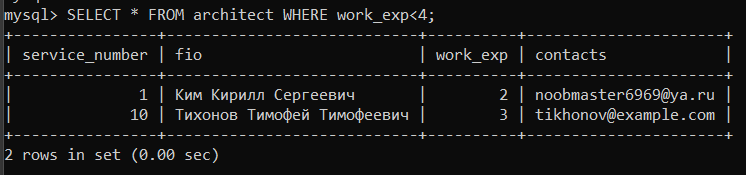


Рисунок 44 — Условие для таблицы architect

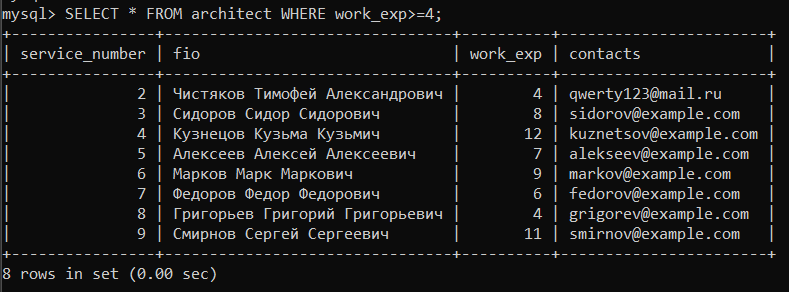


Рисунок 45 — Условие для таблицы architect

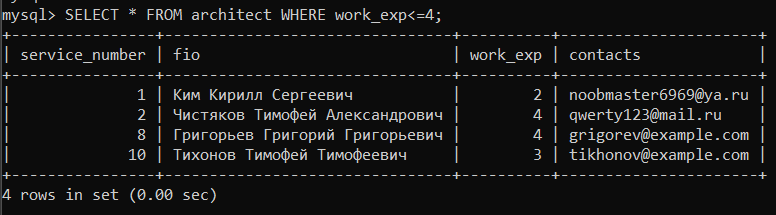


Рисунок 46 — Условие для таблицы architect

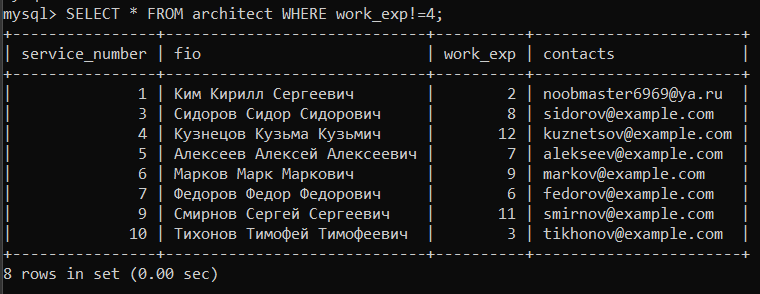


Рисунок 47 — Условие для таблицы architect

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 48 — Условие для таблицы architect

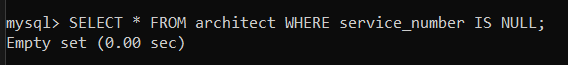


Рисунок 49 — Отбираются значения не имеющие значения в указанном поле

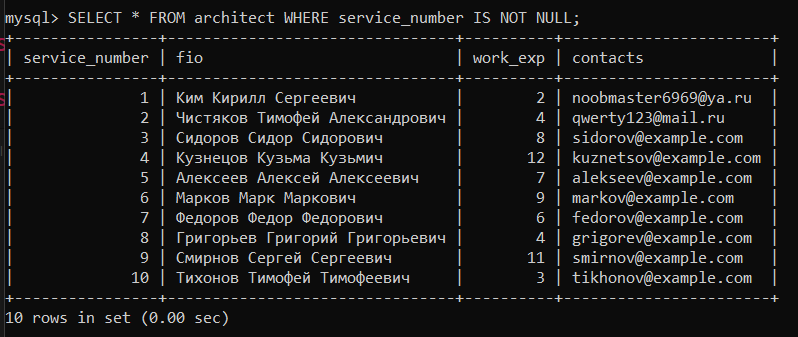


Рисунок 50 — Отбираются значения имеющие значения в указанном поле

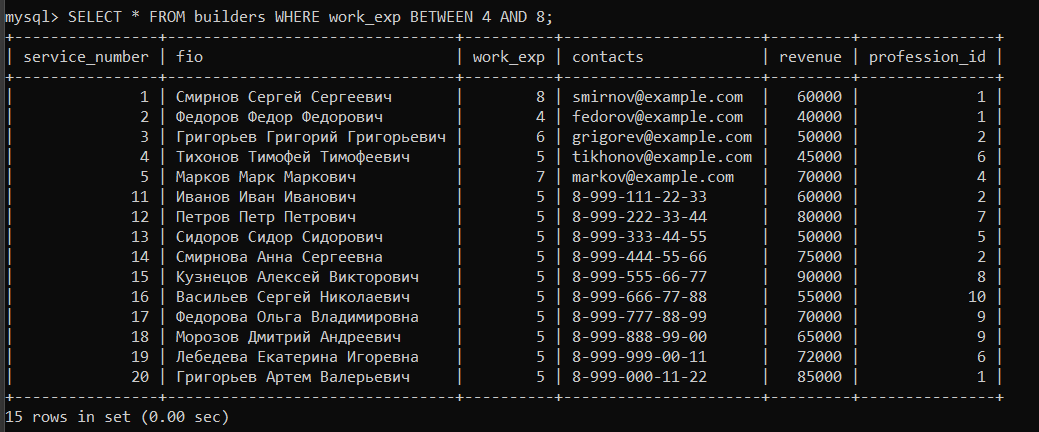


Рисунок 51 — Выборка строителей с опытом работы в определенном диапазоне

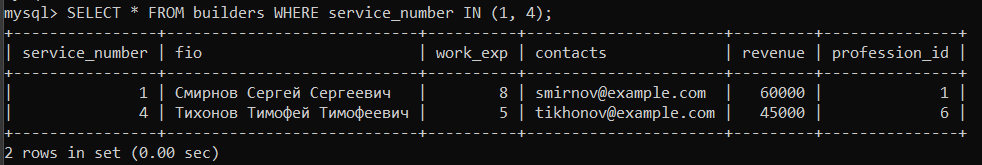


Рисунок 52 — Отбираются значения соответствующие указанным

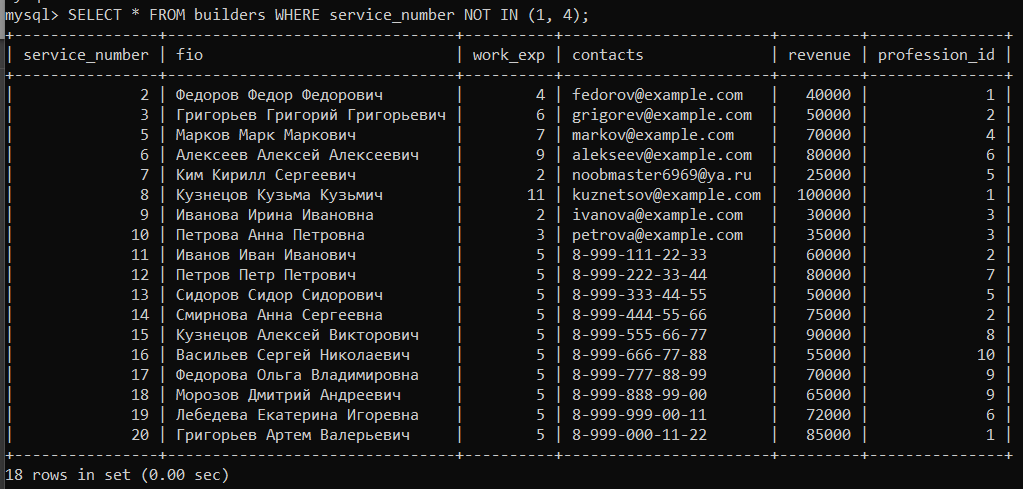


Рисунок 53 — Отбираются значения не соответствующие указанным

Выведем из таблицы shops строчки, где в списке продуктов встречается «МКАД».

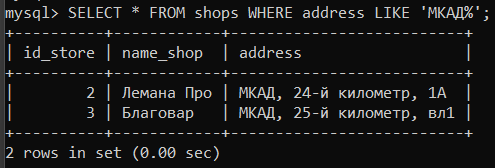


Рисунок 54 — Отбираются значения, соответствующие образцу с использованием LIKE

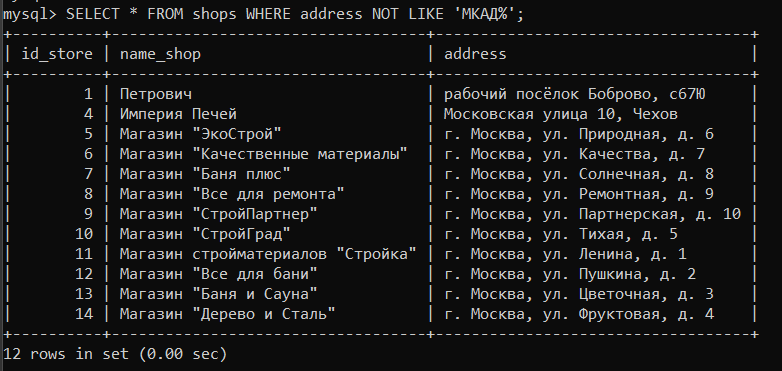


Рисунок 55 — Отбираются не соответствующие значения с использованием NOT LIKE

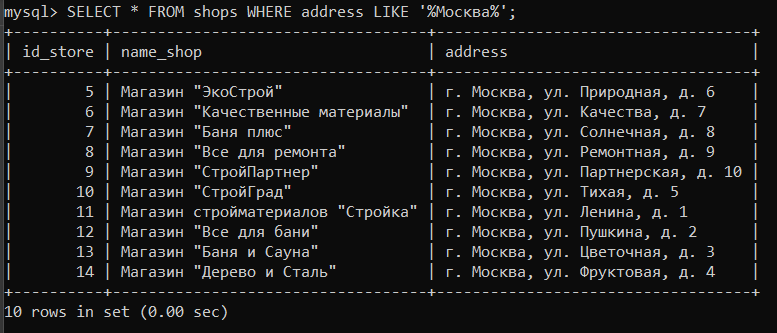


Рисунок 56 — Метасимвол

**Изменение данных в таблице.**

Забыли добавить в таблице builders – rep (рейтинг).

Для добавления столбцов в таблицу используется оператор ALTER TABLE — ADD COLUMN.

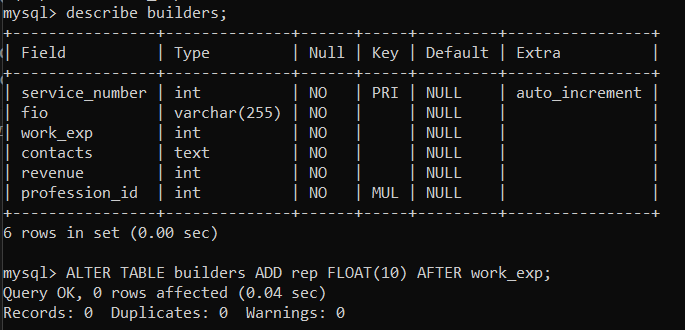


Рисунок 57 — Добавление нового столбца в таблицу

Запишем значения в новый столбец.

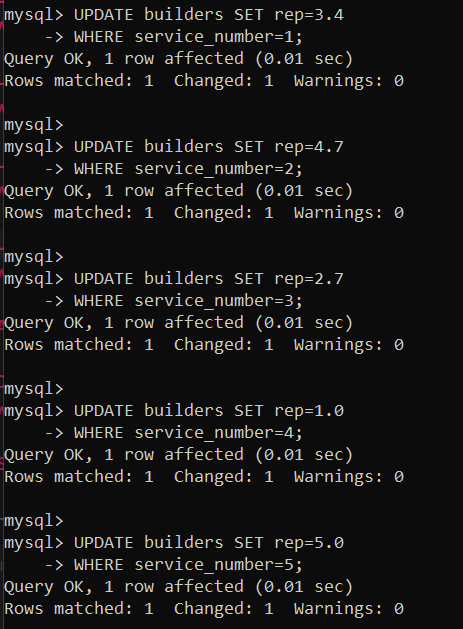


Рисунок 58 — Добавления значений

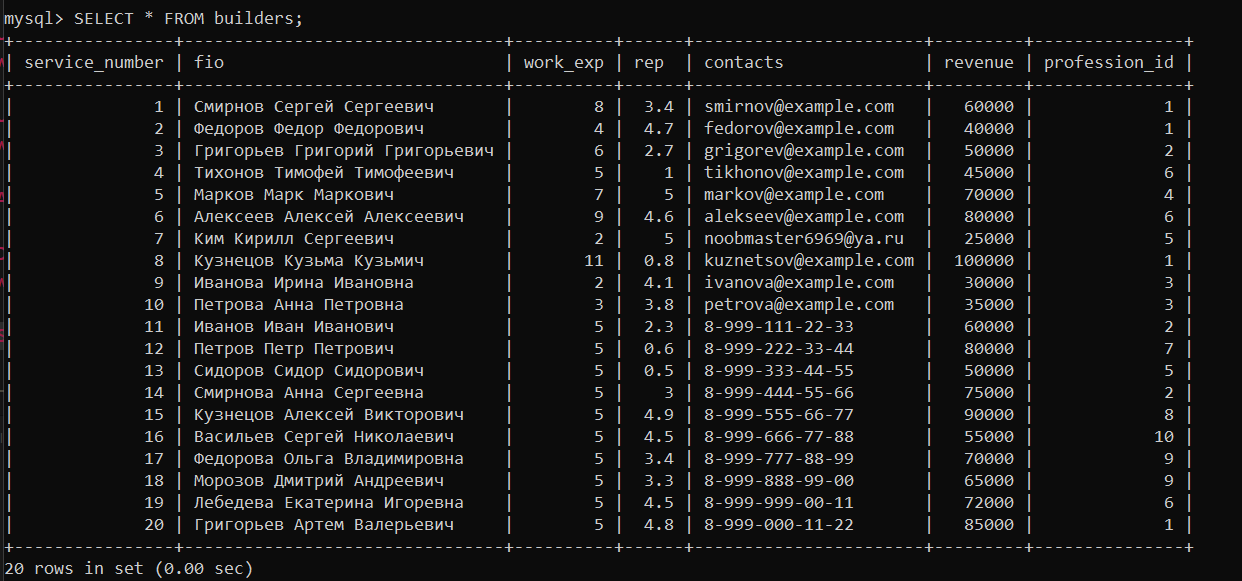


Рисунок 55 — Обновление информации

Переименовать столбец с изменением типа данных командой ALTER TABLE builders CHANGE COLUMN rep rating FLOAT(10);

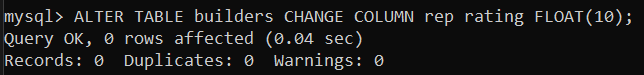


Рисунок 56 — Замена названия столбца

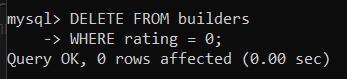


Рисунок 57 — Удаление записей с 0 рейтингом

Операция соединения. Вывести металл и название производителя:

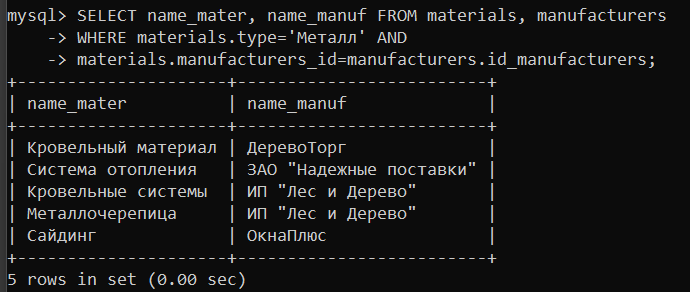


Рисунок 58 — Вывод информации по критериям

Операция объединения. Вывести название материалов и цену:

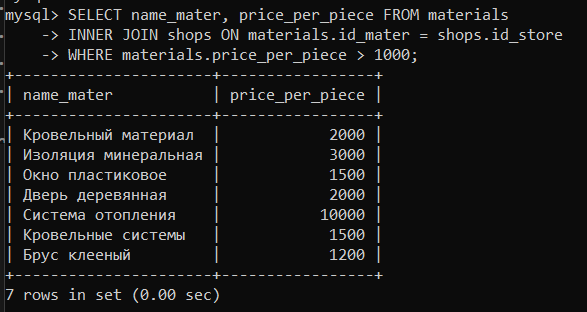


Рисунок 59 — Вывод информации по критериям

Операция пересечения. Вывести название магазина, адрес, что купить какой материл:

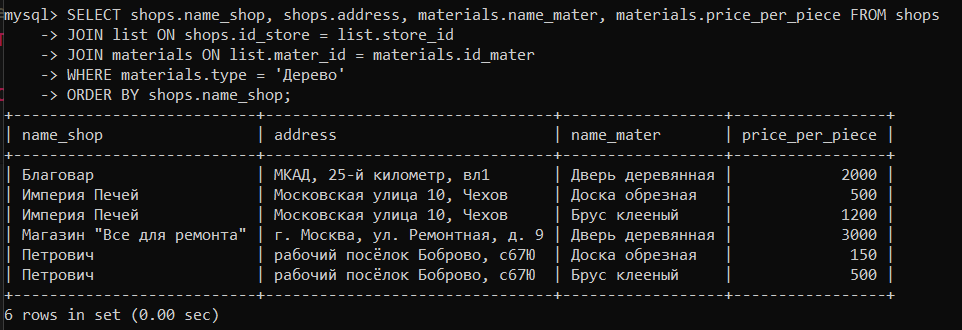


Рисунок 60 — Вывод информации по критериям

Операция разности. Вывести ФИО строителя и опыт работы:

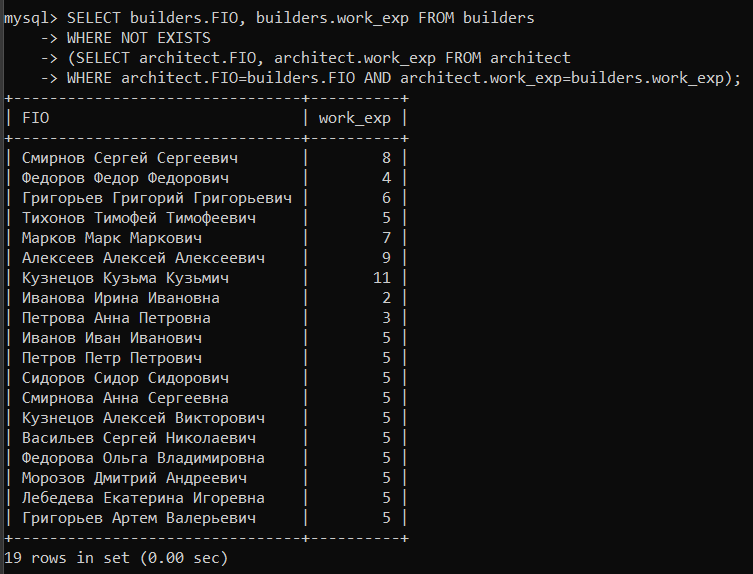


Рисунок 61 — Вывод информации по критериям

Операция группировки (GROUP BY).

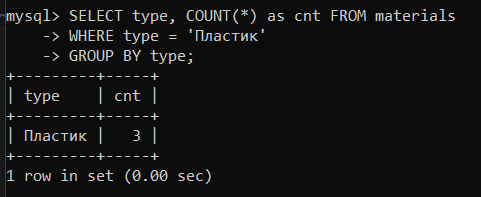


Рисунок 62 — Вывод количество пластиков

Операция сортировки. Запрос для получения списка материалов, отсортированных по цене за единицу.

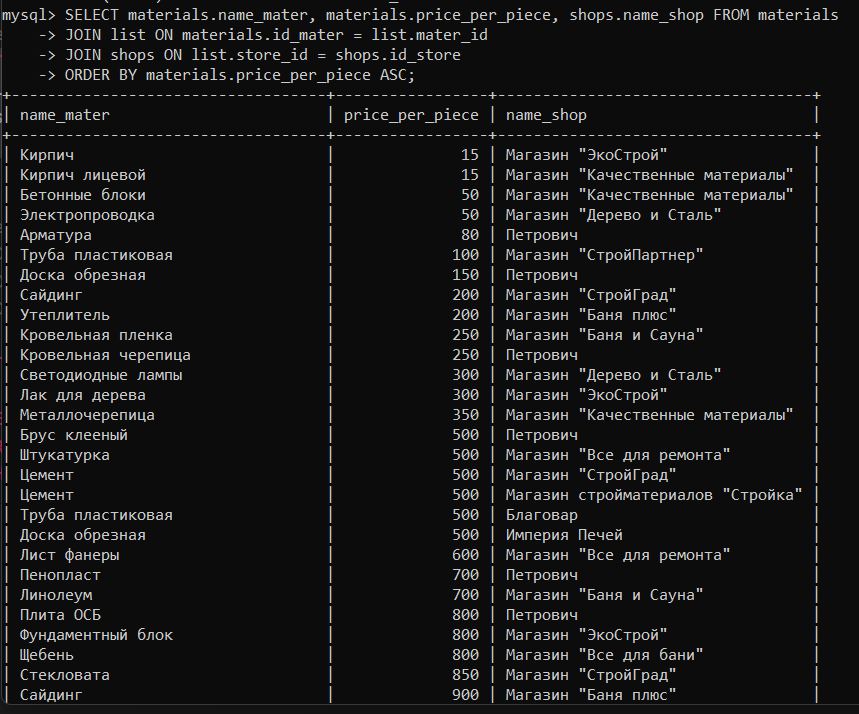


Рисунок 63 — Вывод материалов

Результат выполнения практической работы по использованию СУБД

Выборка данных (DQL)

Запрос 1. Операция проекции. Осуществляется выбор только части полей таблицы, т.е. производится вертикальная выборка данных.

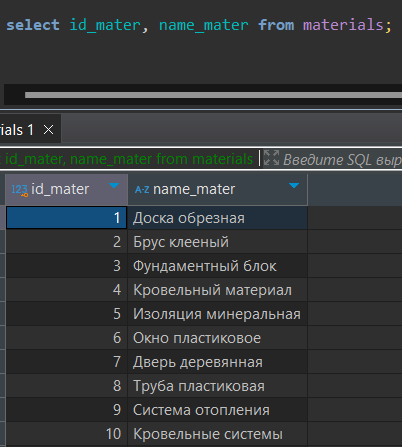


Рисунок 64 — Выборка только id и названия материалов

Запрос 2. Операция селекции. Осуществляется горизонтальная выборка – в результат попадают только записи, удовлетворяющие условию.

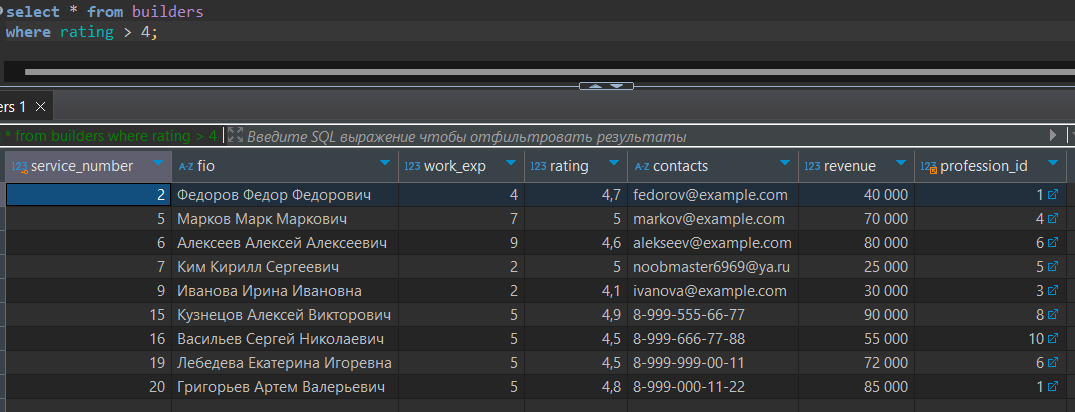


Рисунок 65 — Выборка строителей больше 4 рейтинга

Запрос 3. Операции соединения (JOIN). Здесь следует выделить декартово произведение и на его основе соединение по условию, а также естественное соединение (по одноименным полям или равенству полей с одинаковым смыслом).

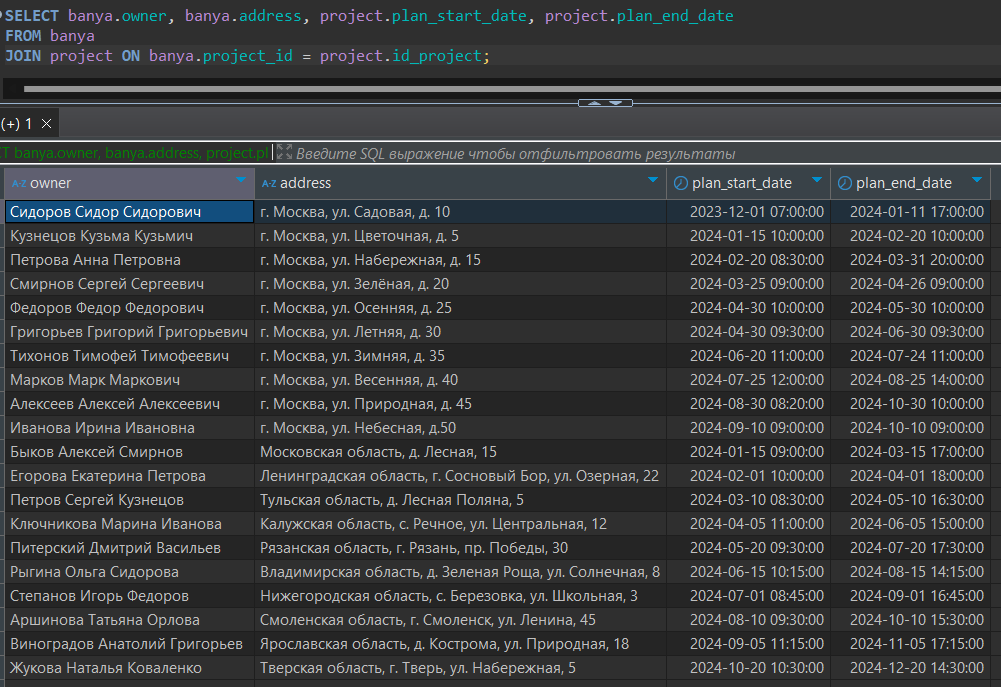


Рисунок 66 — Декартово произведение banya и project

Запрос 4. Операция объединения. Объединим список архитекторов и строителей, чтобы получить общий список людей, работающих в строительной сфере. Теоретико-множественные операции часто можно записать с помощью логических операций, примененных в конструкции WHERE запроса. Также можно задать логическим ИЛИ.

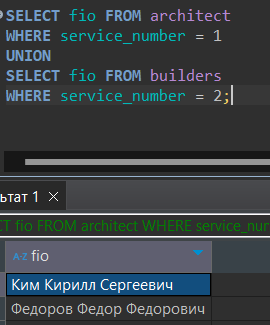


Рисунок 67 — Объединение architect и builders

Запрос 5. Операция пересечения. В простых случаях эту операцию можно описать с помощью логической операции AND. В более сложных случаях эта операция определяется чаще всего с помощью подзапроса и ключевого слова EXISTS, которое показывает наличие похожего элемента во множестве, которое задается подзапросом.

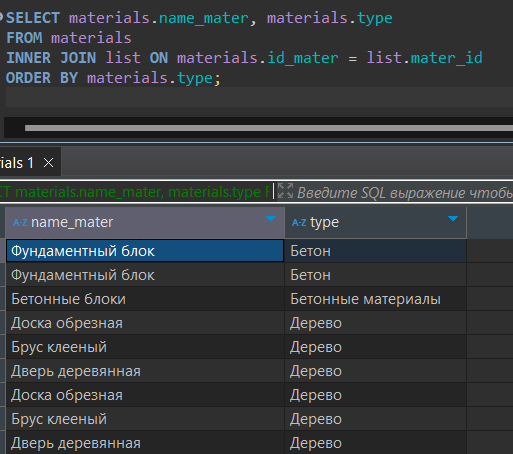


Рисунок 68 — Пересечение list и materials

Запрос 6. Операция разности. Эта операция также определяется часто с помощью подзапроса с ключевым словом NOT EXISTS, которое показывает отсутствие элемента во множестве, задаваемом подзапросом.

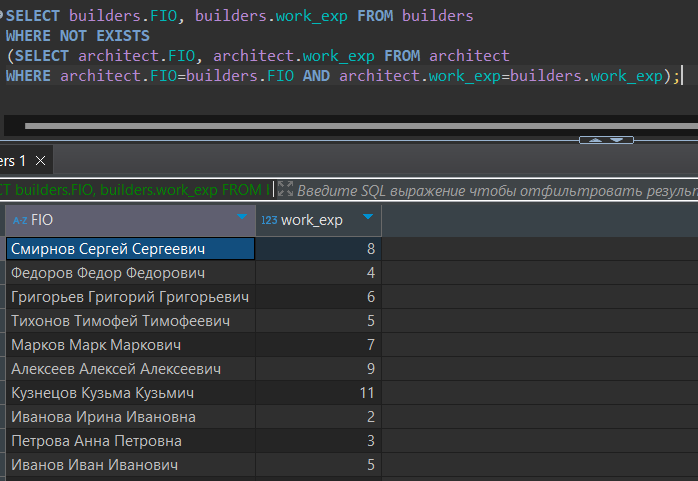


Рисунок 69 — Строители без архитекторов

Запрос 7. Операция группировки. Эта операция связана со своеобразной сверткой таблицы по полям группировки. Помимо полей группировки результат запроса может содержать итоговые агрегирующие функции по группам (COUNT, SUM, AVG, MAX, MIN).

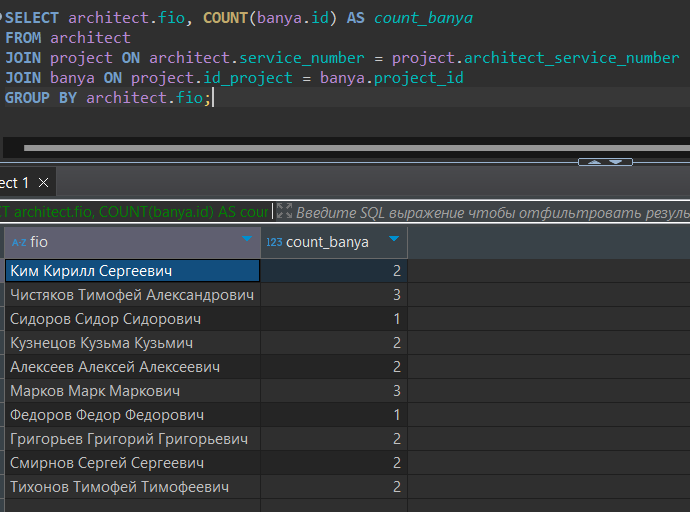


Рисунок 70 — Количество бань, построенным каждым архитекром

Запрос 8. Операция сортировки.

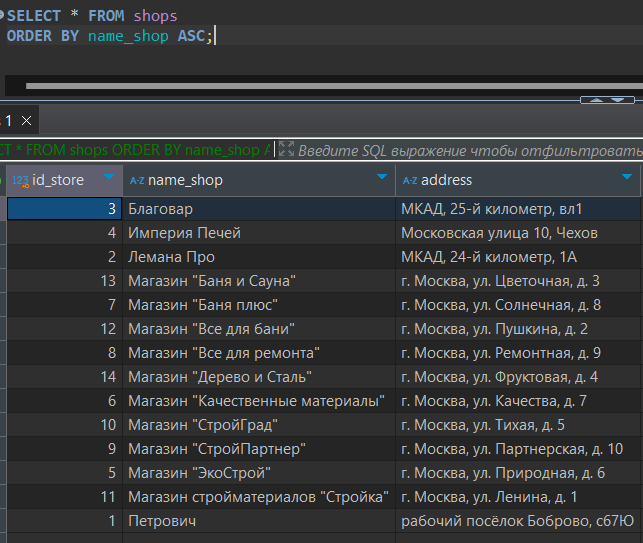


Рисунок 71 — Сортировка название магазинов в алфавитном порядке

Процедуры

1. Процедура для добавления архитектора. Код представлен в Листинге 1. Результат работы процедуры представлен на Рисунке 72.

*Листинг 1 — Процедура для добавления архитектора*

|  |
| --- |
| CREATE PROCEDURE AddArchitect (IN p\_fio VARCHAR(255), IN p\_work\_exp INT, IN p\_contacts TEXT)  BEGIN  INSERT INTO architect (fio, work\_exp, contacts) VALUES (p\_fio, p\_work\_exp, p\_contacts);  END; |

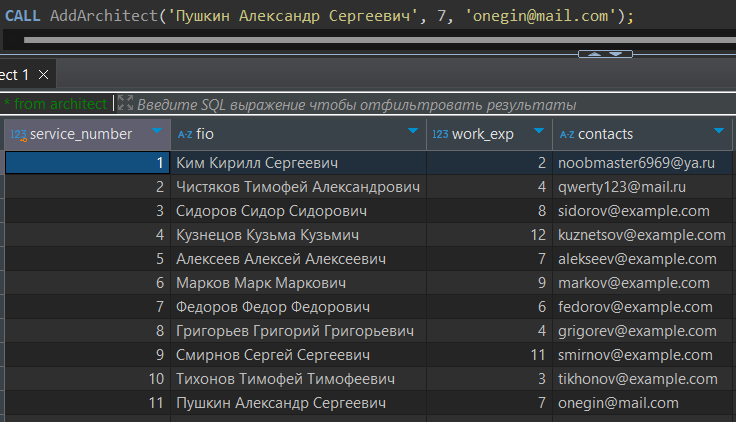


Рисунок 72 — Процедура для добавления архитектора

1. Процедура для получения списка материалов, испульзуемых в конкретной бане. Код представлен в Листинге 2. Результат работы процедуры представлен на Рисунке 73.

*Листинг 2 — Процедура для получения списка материалов, используемых в конкретной бане*

|  |
| --- |
| CREATE PROCEDURE get\_materials\_by\_banya(  IN p\_banya\_id INT  )  BEGIN  SELECT materials.name\_mater, checks.quantity  FROM banya  JOIN checks ON banya.id = checks.banya\_id  JOIN materials ON checks.materials\_id = materials.id\_mater  WHERE banya.id = p\_banya\_id;  END; |

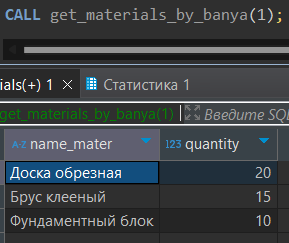


Рисунок 73 — Процедура для получения списка материалов, используемых в конкретной бане

1. Процедура для получения информации сколько часов работал строитель. Код представлен в Листинге 3. Результат работы процедуры представлен на Рисунке 74.

*Листинг 3 — Процедура для получения информации сколько часов работал строитель*

|  |
| --- |
| CREATE PROCEDURE get\_builders\_by\_banya(  IN p\_banya\_id INT  )  BEGIN  SELECT builders.fio, timediff(contract.date\_work\_end,contract.date\_work\_start)  FROM banya  JOIN contract ON banya.id = contract.ban\_id  JOIN builders ON contract.service\_number = builders.service\_number  WHERE banya.id = p\_banya\_id;  END; |

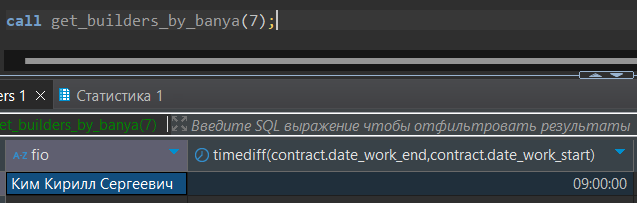


Рисунок 74 — Процедура для получения информации сколько часов строитель работал

1. Процедура обновления даты окончания строительства бани. Код представлен в Листинге 4. Результат работы процедуры представлен на Рисунке 75.

*Листинг 4 — Процедура для обновления даты окончания строительства бани*

|  |
| --- |
| CREATE PROCEDURE UpdateDate(IN id\_p INT, IN end\_date DATETIME)  BEGIN  UPDATE banya  SET end\_date\_construction = end\_date  WHERE id = id\_p;  END; |

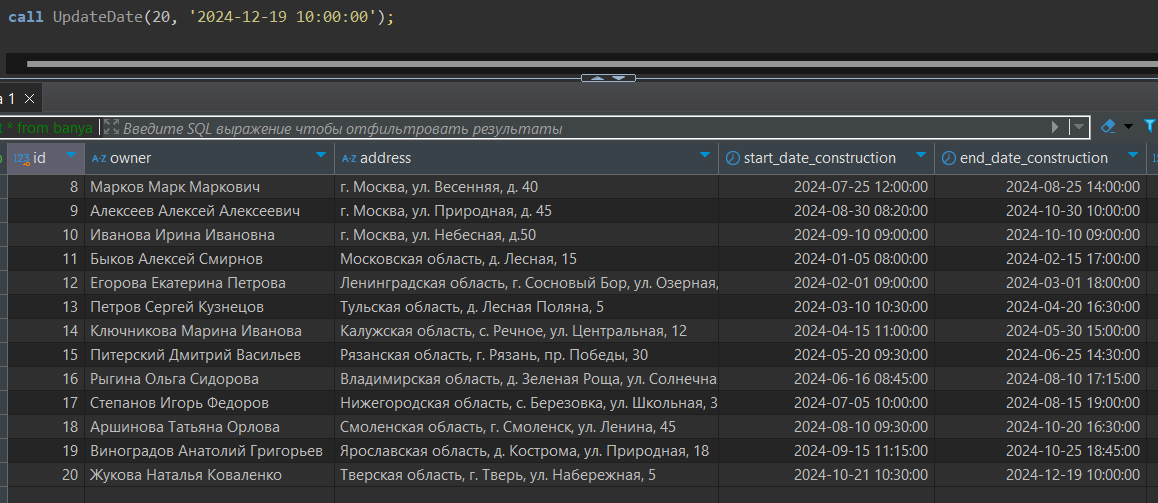


Рисунок 75 — Процедура для обновления даты окончания строительства бани

1. Процедура для получения размер материала (произведение). Код представлен в Листинге 5. Результат работы процедуры представлен на Рисунке 76.

*Листинг 5 — Процедура для получения размер материала*

|  |
| --- |
| CREATE PROCEDURE calculate\_material\_area(IN p\_type VARCHAR(255))  BEGIN  SELECT name\_spec, size\_a\_metr \* size\_b\_metr FROM specifications  JOIN materials ON materials.specifications\_id = specifications.id\_specifications  WHERE materials.type = p\_type;  END; |

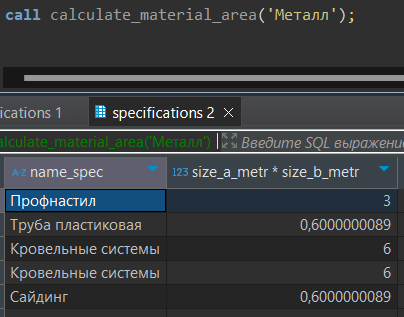


Рисунок 76 — Процедура для получения размер материала

Функции

1. Функция для получения всей стоимости строительства бани, включая покупки материалов. Код представлен в Листинге 6. Результат работы процедуры представлен на Рисунке 77.

*Листинг 6 — Функция для получения всей стоимости строительства бани, включая покупки материалов*

|  |
| --- |
| CREATE FUNCTION GetConstructionCost (p\_project\_id INT)  RETURNS FLOAT  DETERMINISTIC  BEGIN  DECLARE total\_cost FLOAT;  SELECT SUM(materials.price\_per\_piece \* checks.quantity) +  (SELECT price FROM project WHERE id\_project = p\_project\_id) INTO total\_cost  FROM materials  JOIN checks ON materials.id\_mater = checks.materials\_id  JOIN banya ON checks.banya\_id = banya.id  WHERE banya.project\_id = p\_project\_id;  RETURN total\_cost;  END; |

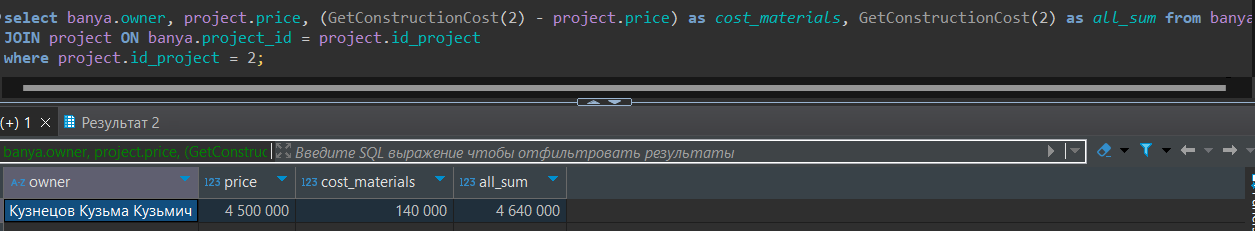


Рисунок 77 — Функция для получения всей стоимости строительства бани, включая покупки материалов

1. Функция для подсчета количества материалов по типу. Код представлен в Листинге 7. Результат работы процедуры представлен на Рисунке 78.

*Литстинг 7 —* *Функция для подсчета количества материалов по типу*

|  |
| --- |
| CREATE FUNCTION CountMaterialsByType (p\_type VARCHAR(255))  RETURNS INT  DETERMINISTIC  BEGIN  DECLARE material\_count INT;    SELECT COUNT(\*) INTO material\_count FROM materials WHERE type = p\_type;    RETURN material\_count;  END; |

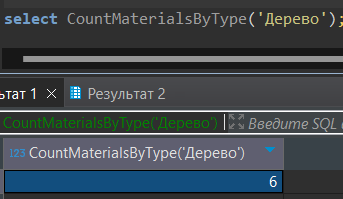


Рисунок 78 — Функция для подсчета количества материалов по типу

1. Функция на наличия материала в спике. Код представлен в Листинге 8. Результат работы процедуры представлен на Рисунке 79.

*Литстинг 8 —* *Функция на наличие материала в списке*

|  |
| --- |
| CREATE FUNCTION IsMaterialAvailable (p\_material\_id INT) RETURNS BOOLEAN  DETERMINISTIC  BEGIN  DECLARE available BOOLEAN;  SELECT COUNT(\*) > 0 INTO available FROM list WHERE mater\_id = p\_material\_id;    RETURN available;  END; |

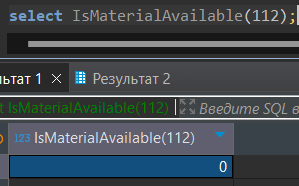


Рисунок 79 — Функция на наличие материала в списке

1. Функция для подсчета количество дней построенных бань летом. Код представлен в Листинге 9. Результат работы процедуры представлен на Рисунке 80.

*Литстинг 9 — Функция для подсчета количество дней построенных бань летом*

|  |
| --- |
| CREATE FUNCTION count\_summer\_construction\_days(m1 INT, m2 INT, m3 INT)  RETURNS INT  DETERMINISTIC  BEGIN  DECLARE cnt INT;  SELECT COUNT(\*) into cnt FROM banya WHERE MONTH(start\_date\_construction) IN (m1, m2, m3);  RETURN cnt;  END; |

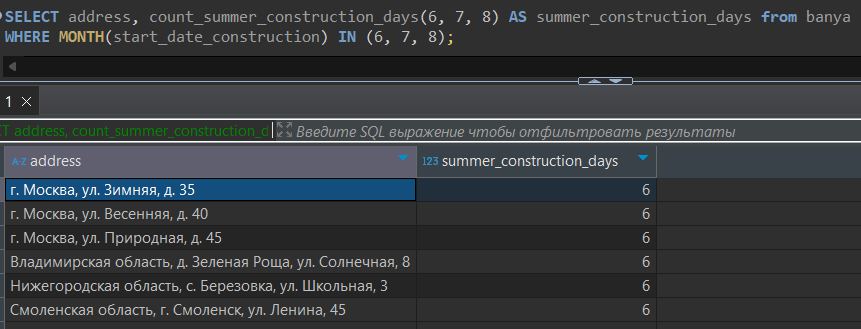


Рисунок 80 — Функция для подсчета количество дней построенных бань летом

1. Функция для подсчета архитектор делает проект. Код представлен в Листинге 10. Результат работы процедуры представлен на Рисунке 81.

*Литстинг 10 — Функция для подсчета количество дней построенных бань летом*

|  |
| --- |
| CREATE FUNCTION average\_construction\_time(service\_num INT)  RETURNS INT  DETERMINISTIC  begin  DECLARE avg\_time INT;  SELECT AVG(TIMESTAMPDIFF(DAY, plan\_start\_date, plan\_end\_date)) INTO avg\_time  FROM architect join project on architect.service\_number = project.architect\_service\_number  where architect.service\_number = service\_num;  RETURN avg\_time;  END; |

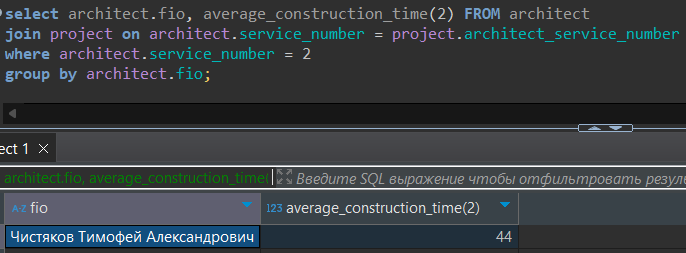


Рисунок 81 — Функция для подсчета сколько в среднем архитектор делает проект

Триггеры

1. Триггер для проверки уникальности владельца бани. Этот триггер предотвращает создание нескольких записей о бане с одинаковым владельцем. Код представлен в Листинге 11.

*Листинг 11 — Триггер проверки уникальности владельца бани*

|  |
| --- |
| CREATE TRIGGER check\_unique\_banya\_owner  BEFORE INSERT ON banya  FOR EACH ROW  BEGIN  DECLARE owner\_count INT;  SELECT COUNT(\*) INTO owner\_count FROM banya WHERE owner = NEW.owner;  IF owner\_count > 0 THEN  SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Owner already has a banya';  END IF;  END; |

1. Триггер для автоматического расчета стоимости проекта. Этот триггер автоматически обновляет стоимость проекта при добавлении новой записи о материале в таблицу checks. Код представлен в Листинге 12.

*Листинг 12 — Триггер автоматического расчета стоимоти проекта*

|  |
| --- |
| CREATE TRIGGER update\_project\_price  AFTER INSERT ON checks  FOR EACH ROW  BEGIN  DECLARE total\_price FLOAT DEFAULT 0;  SELECT SUM(m.price\_per\_piece \* c.quantity) INTO total\_price  FROM materials m  JOIN checks c ON m.id\_mater = c.materials\_id  WHERE c.banya\_id = NEW.banya\_id;  UPDATE project  SET price = total\_price  WHERE id\_project = (SELECT project\_id FROM banya WHERE id = NEW.banya\_id);  END; |

1. Триггер для проверки наличия необходимых материалов перед началом строительства. Этот триггер проверяет наличие всех необходимых материалов перед началом строительства бани. Код представлен в Листинге 13.

*Листинг 13 — Триггер проверки наличия необходимых материалов перед началом строительства*

|  |
| --- |
| CREATE TRIGGER check\_materials\_before\_start  BEFORE UPDATE ON banya  FOR EACH ROW  BEGIN  DECLARE missing\_materials INT;  SELECT COUNT(\*) INTO missing\_materials  FROM materials m  LEFT JOIN checks c ON m.id\_mater = c.materials\_id AND c.banya\_id = NEW.id  WHERE c.quantity IS NULL;  IF missing\_materials > 0 THEN  SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Необходимые материалы отсутствуют для начала строительства';  END IF;  END; |

1. Этот триггер для удаления материалов из списка магазинов. Он удаляет материалы из таблицы list, если материал был удален из таблицы materials. Код представлен в Листинге 14.

*Листинг 14 — Триггер удаления материалов из списка магазинов*

|  |
| --- |
| CREATE TRIGGER delete\_material\_from\_list  AFTER DELETE ON materials  FOR EACH ROW  BEGIN  DELETE FROM list WHERE mater\_id = OLD.id\_mater;  END; |

1. Триггер на добавление новой бани для уведомления архитектора. Код представлен в Листинге 15.

*Листинг 15 — обновления статуса испытания одобрении в нескольких странах*

|  |
| --- |
| CREATE TRIGGER NotifyArchitect AFTER INSERT ON banya  FOR EACH ROW  BEGIN  INSERT INTO notifications (architect\_service\_number, message)  VALUES (NEW.project\_id.architect\_service\_number, CONCAT('Новая баня построена для ', NEW.owner));  END; |

Оконные функции

Ниже приведены примеры оконных функций из списка:

1. **Агрегатные функции**

*Листинг 16 — Агрегатные функции COUNT*

|  |
| --- |
| SELECT architect.fio, COUNT(project.id\_project) as CountProject  FROM architect JOIN project ON project.architect\_service\_number = architect.service\_number  GROUP BY architect.fio; |

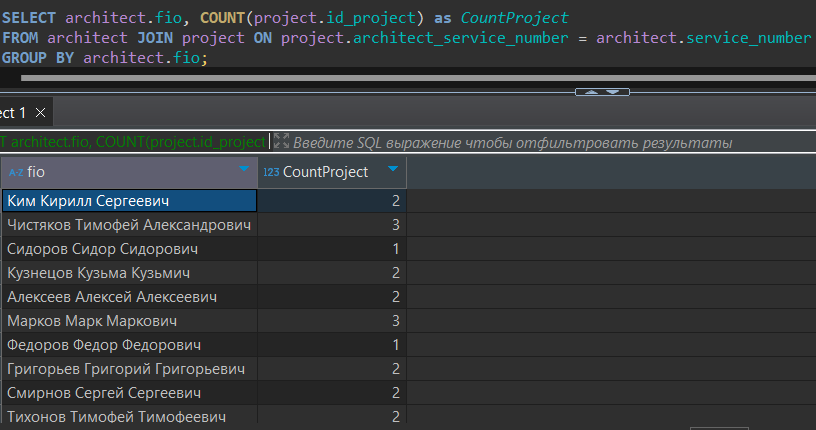


Рисунок 82 — Вызов агрегатных функций COUNT

*Листинг 17 — Агрегатные функции SUM*

|  |
| --- |
| SELECT banya.owner, banya.start\_date\_construction, SUM(project.price) OVER (PARTITION BY banya.start\_date\_construction) as p  FROM banya JOIN project ON project.id\_project = banya.project\_id; |

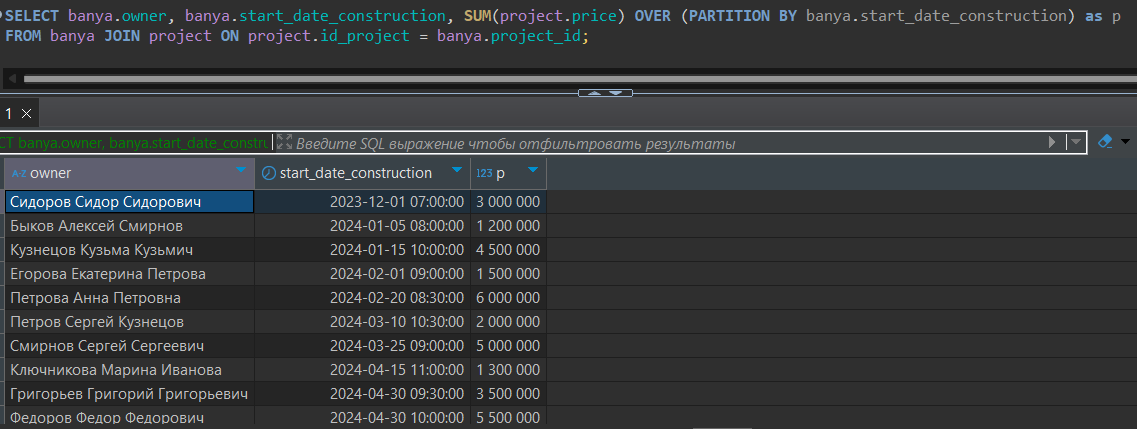


Рисунок 83 — Вызов агрегатных функций SUM

*Листинг 18 — Агрегатные функции AVG*

|  |
| --- |
| SELECT banya.owner, AVG(DATEDIFF(project.plan\_end\_date, project.plan\_start\_date)) OVER (PARTITION BY banya.owner) as avg\_const\_time  FROM banya JOIN project ON project.id\_project = banya.project\_id; |

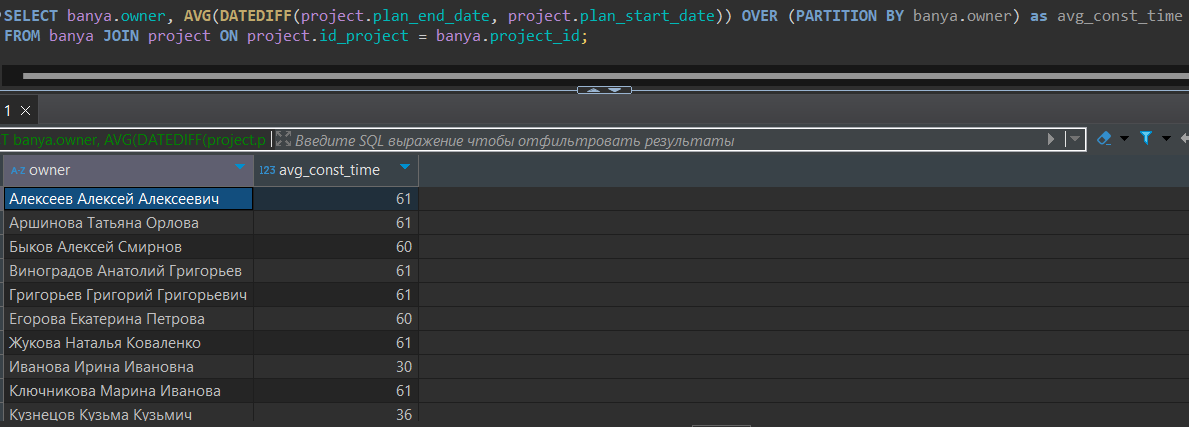


Рисунок 84 — Вызов агрегатных функций AVG

*Листинг 19 — Агрегатные функции MIN*

|  |
| --- |
| SELECT materials.name\_mater, specifications.name\_spec, specifications.weight\_kilo,  MIN(specifications.weight\_kilo) OVER (PARTITION BY materials.name\_mater) as min\_weight  FROM materials JOIN specifications ON materials.specifications\_id = specifications.id\_specifications; |

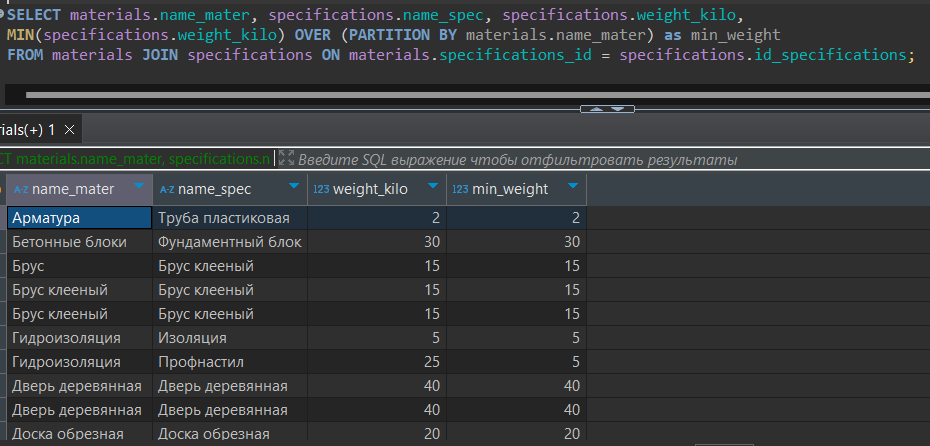


Рисунок 85 — Вызов агрегатных функций MIN

*Листинг 20 — Агрегатные функции MAX*

|  |
| --- |
| SELECT materials.name\_mater, specifications.name\_spec, materials.price\_per\_piece,  MAX(materials.price\_per\_piece) OVER (PARTITION BY materials.name\_mater) as max\_price  FROM materials JOIN specifications ON materials.specifications\_id = specifications.id\_specifications; |

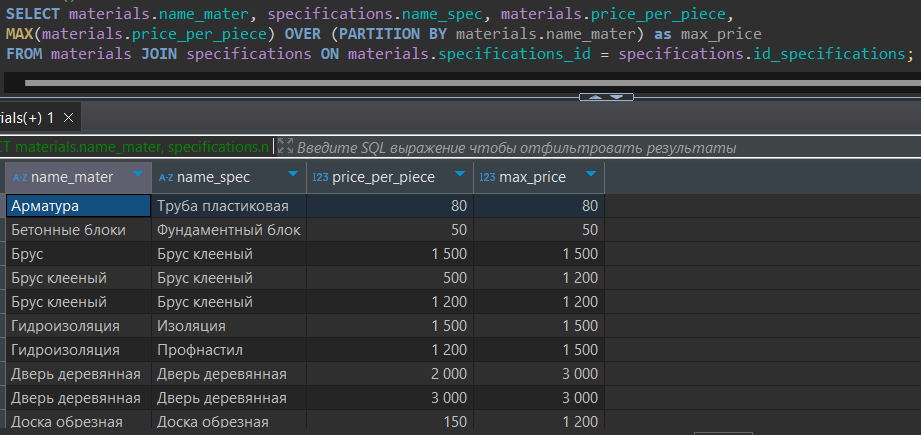


Рисунок 86 — Вызов агрегатных функций MAX

1. **Ранжирующие функции**

*Литстинг 21 — Ранжирующие функции*

|  |
| --- |
| SELECT banya.owner, banya.address, banya.start\_date\_construction, banya.end\_date\_construction,  ROW\_NUMBER() OVER (ORDER BY banya.end\_date\_construction) AS RowNumber,  RANK() OVER (ORDER BY banya.end\_date\_construction DESC) AS SalesRank,  DENSE\_RANK() OVER (ORDER BY banya.end\_date\_construction DESC) AS DenseRank,  NTILE(4) OVER (ORDER BY banya.end\_date\_construction DESC) AS PriceGroup  FROM banya; |

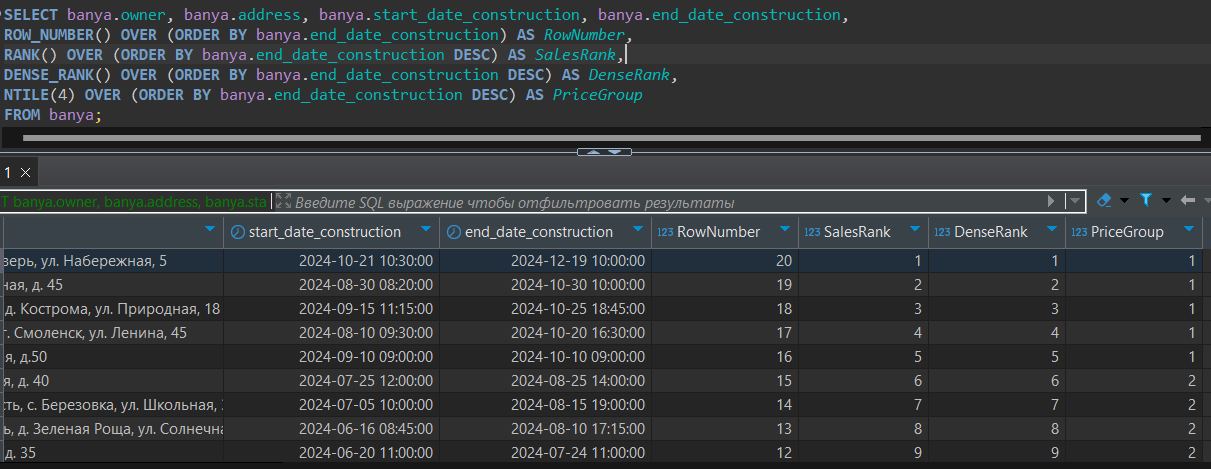


Рисунок 87 — Вызов ранжирующих функций

1. **Функции смещения**

*Литстинг 22 — Функции смещения*

|  |
| --- |
| SELECT banya.owner, project.plan\_start\_date, project.plan\_end\_date,  LAG(project.plan\_start\_date, 1) OVER (ORDER BY project.plan\_start\_date) AS prev\_value,  LEAD(project.plan\_start\_date, 1) OVER (ORDER BY project.plan\_start\_date) AS next\_value,  FIRST\_VALUE(project.plan\_start\_date) OVER (ORDER BY project.plan\_start\_date) AS first\_value,  LAST\_VALUE(project.plan\_start\_date) OVER (ORDER BY project.plan\_start\_date) AS last\_value  FROM banya JOIN project ON project.id\_project = banya.project\_id; |

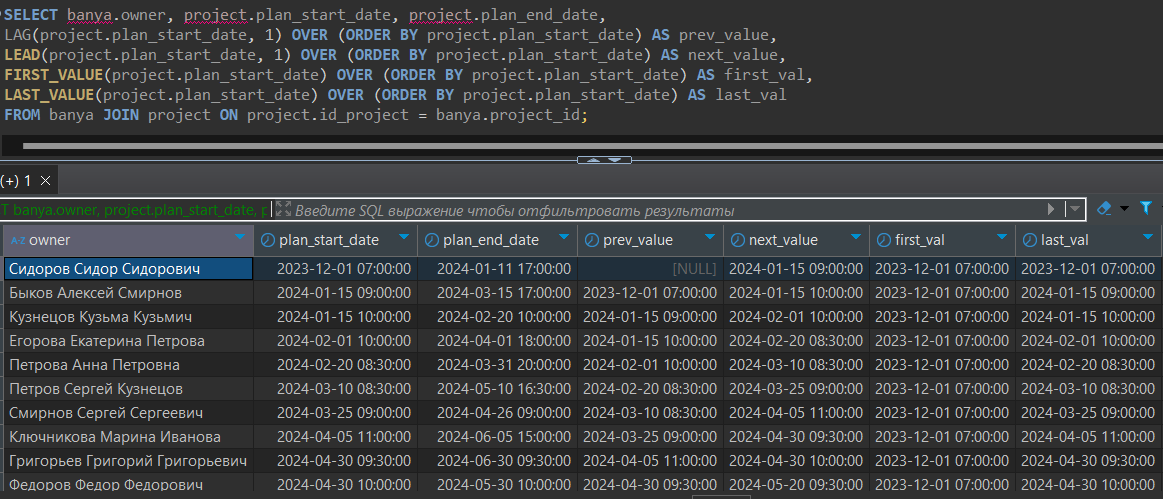


Рисунок 88 — Вызов функций смещения

Заключение

В течение семестра была создана база данных

Список использованных источников

1. Дейт, К. Дж. Введение в системы баз данных / учебное пособие / К. Дж. Дейт; — Москва: «Вильяме» 2005. — 1316 с.
2. Дьяков И.А. Базы данных. Язык SQL: Учебное пособие. — Тамбов: ТГТУ, 2004. — 80 с.
3. Шварц Б., Зайцев П., Ткаченко В., Заводны Дж., Ленц А., Бэллинг Д. MySQL. Оптимизация производительности, 2-е издание. — Пер. с англ. — Спб.: Символ-Плюс, 2010. — 832 с.: ил.
4. Э.В. Сысоев, Е.В. Бурцева. Базы данных: лекции к курсу. — Тамбов : Изд-во Тамб. гос. техн. ун-та, 2007. — 48 с.
5. Баженова И.Ю. Разработка распределенных приложений баз данных: Курс лекций. — М.: МГУ им. М.В. Ломоносова, 2006. — 203 с.
6. Зиборов В.В. MS Visual C++ 2010 в среде .NET. Библиотека программиста. — Спб.: Питер, 2012. - 320 с.: ил.