



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

Ким К.С.

(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
ФИЗИЧЕСКАЯ МОДЕЛЬ БАЗЫ ДАННЫХ.....	4
РЕЗУЛЬТАТА ВЫПОЛНЕНИЯ ПРАКТИЧЕСКИХ РАБОТ В SQL COMMAND LINE .....	5
РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ ПРАКТИЧЕСКОЙ РАБОТЫ ПО ИСПОЛЬЗОВАНИЮ СУБД.....	33
Выборка данных (DQL) .....	33
Процедуры.....	38
Функции.....	42
Триггеры .....	47
Оконные функции .....	50
ЗАКЛЮЧЕНИЕ.....	56
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	57

## ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

В данной работе была рассмотрена база данных деятельности строительства бани. База данных содержит информацию о чертежах, проектах, клиентах (кто заказывает баню), купленных материалах, строителях и так далее.

Процесс начинается с заказа бани, клиент оставляет свои данные, а именно: ФИО, потом архитектор разрабатывает чертеж проекта. Клиент согласует или корректирует с архитектором проект бани. И компания вычисляет сколько уйдет проект: по времени, по цене. После компания покупает материалы, в определенных магазинах, какие будут нужны для строительства бани. К назначенному времени работники должны построить баню.

В работе также рассматривается профессия строителей. Потому что каждый строитель делает поэтапно: делают фундамент, каркас, установки печки, проведение электричества, отделка.

Также у строителей есть график работы и они должны строго соблюдать, по контракту назначенному строительной компанией.

# ФИЗИЧЕСКАЯ МОДЕЛЬ БАЗЫ ДАННЫХ

На Рисунке 1 представлена физическая модель рассматриваемой предметной области.

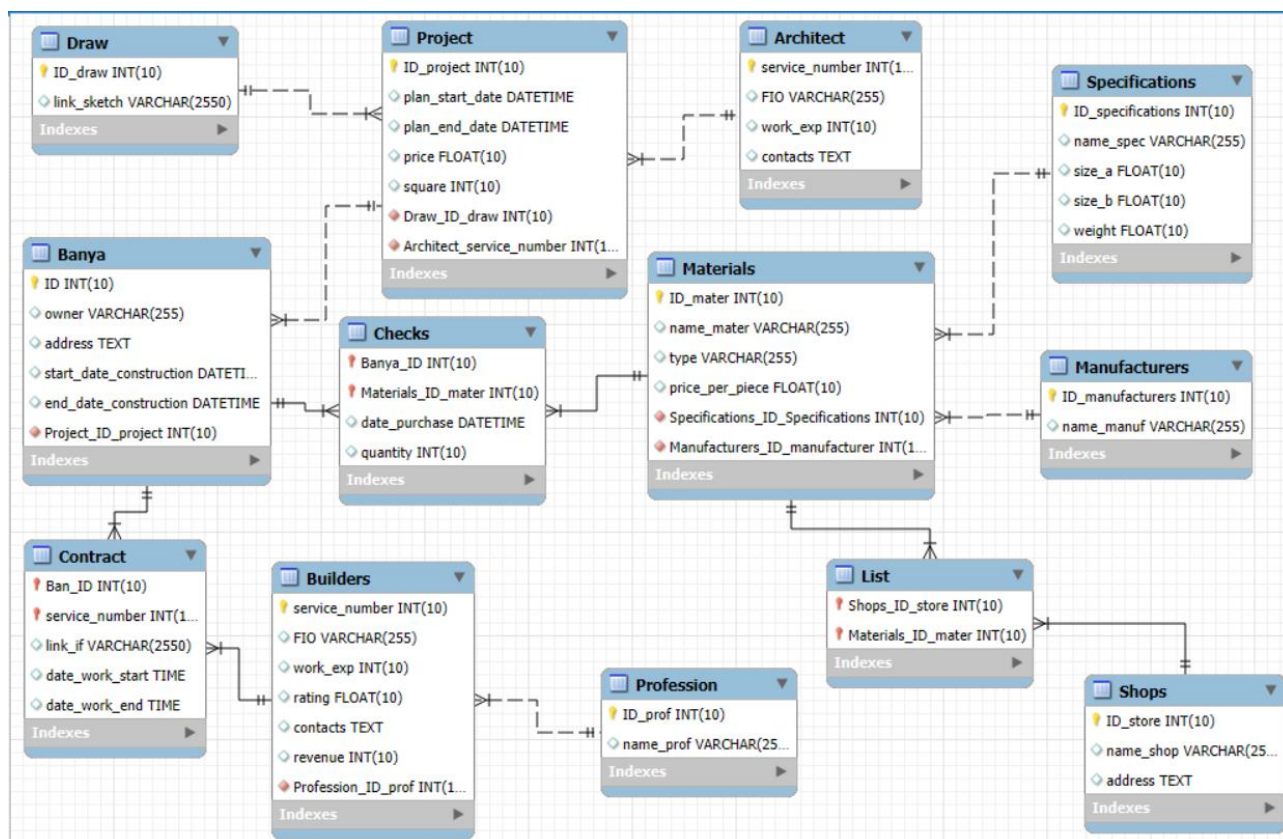
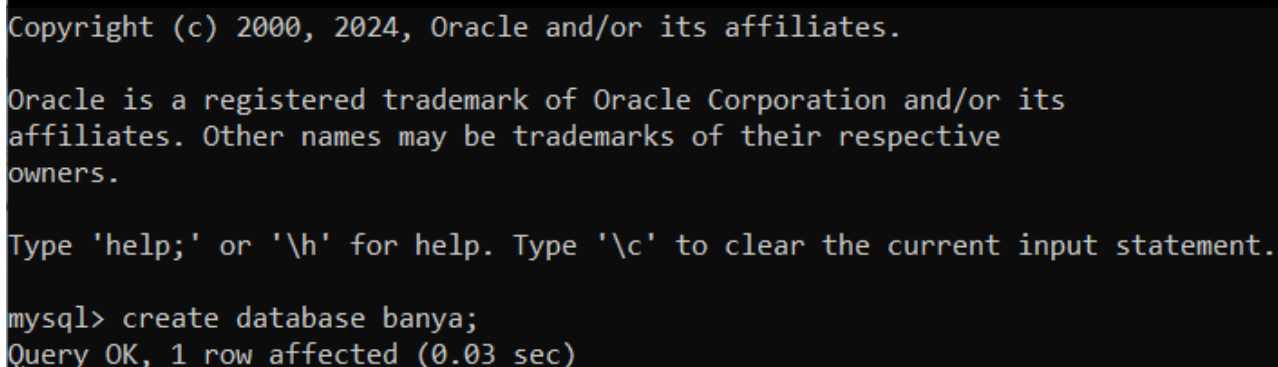


Рисунок 1 — Физическая модель предметной области

# РЕЗУЛЬТАТА ВЫПОЛНЕНИЯ ПРАКТИЧЕСКИХ РАБОТ В SQL COMMAND LINE

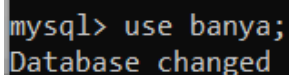
Для начала создадим БД banya. Результат представлен на Рисунке 2.



```
Copyright (c) 2000, 2024, Oracle and/or its affiliates.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
mysql> create database banya;  
Query OK, 1 row affected (0.03 sec)
```

Рисунок 2 — Создание базы данных

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



```
mysql> use banya;  
Database changed
```

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).

```
mysql> CREATE TABLE draw (
  -> id_draw INT(10) AUTO_INCREMENT,
  -> link_sketch VARCHAR(2550) NOT NULL,
  -> PRIMARY KEY (id_draw)
  -> );
Query OK, 0 rows affected, 1 warning (0.05 sec)

mysql> show tables;
+-----+
| Tables_in_banya |
+-----+
| draw             |
+-----+
1 row in set (0.04 sec)

mysql> describe draw;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id_draw    | int           | NO   | PRI | NULL    | auto_increment |
| link_sketch | varchar(2550) | NO   |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.02 sec)
```

**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)

```
mysql> CREATE TABLE architect (
  -> service_number INT(10) AUTO_INCREMENT,
  -> fio VARCHAR(255) NOT NULL,
  -> work_exp INT(10) NOT NULL,
  -> contacts TEXT NOT NULL,
  -> PRIMARY KEY (service_number)
  -> );
Query OK, 0 rows affected, 2 warnings (0.04 sec)

mysql> show tables;
+-----+
| Tables_in_banya |
+-----+
| architect        |
| draw             |
+-----+
2 rows in set (0.00 sec)
```

**Рисунок 5 — Создание таблицы architect**

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

```
mysql> CREATE TABLE profession (  
-> id_prof INT(10) AUTO_INCREMENT,  
-> name_prof VARCHAR(255) NOT NULL,  
-> PRIMARY KEY (id_prof)  
-> );  
Query OK, 0 rows affected, 1 warning (0.04 sec)
```

**Рисунок 6 — Создание таблицы profession**

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

```
mysql> CREATE TABLE shops (  
-> id_store INT(10) AUTO_INCREMENT,  
-> name_shop VARCHAR(255) NOT NULL,  
-> address TEXT NOT NULL,  
-> PRIMARY KEY (id_store)  
-> );  
Query OK, 0 rows affected, 1 warning (0.05 sec)
```

**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).

```
mysql> CREATE TABLE manufacturers (  
-> id_manufacturers INT(10) AUTO_INCREMENT,  
-> name_manuf VARCHAR(255) NOT NULL,  
-> PRIMARY KEY (id_manufacturers)  
-> );  
Query OK, 0 rows affected, 1 warning (0.04 sec)
```

**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.

```
mysql> CREATE TABLE specifications (
-> id_specifications INT(10) AUTO_INCREMENT,
-> name_spec VARCHAR(255) NOT NULL,
-> size_a_metr FLOAT(10) NOT NULL,
-> size_b_metr FLOAT(10) NOT NULL,
-> weight_kilo FLOAT(10) NOT NULL,
-> PRIMARY KEY (id_specifications)
-> );
Query OK, 0 rows affected, 1 warning (0.03 sec)
```

**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).

```
mysql> CREATE TABLE project (
-> id_project INT(10) AUTO_INCREMENT,
-> plan_start_date DATETIME NOT NULL,
-> plan_end_date DATETIME NOT NULL,
-> square INT(10) NOT NULL,
-> price FLOAT(10) NOT NULL,
-> draw_id_draw INT(10) NOT NULL,
-> architect_service_number INT(10) NOT NULL,
-> PRIMARY KEY (id_project),
-> FOREIGN KEY (draw_id_draw) REFERENCES draw (id_draw),
-> FOREIGN KEY (architect_service_number) REFERENCES architect (service_number)
-> );
Query OK, 0 rows affected, 4 warnings (0.07 sec)
```

**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, идентификация проекта (project\_id).

```
mysql> CREATE TABLE banya (
-> id INT(10) AUTO_INCREMENT,
-> owner VARCHAR(255) NOT NULL,
-> address TEXT NOT NULL,
-> start_date_construction DATETIME NOT NULL,
-> end_date_construction DATETIME NOT NULL,
-> project_id INT(10) NOT NULL,
-> PRIMARY KEY (id),
-> FOREIGN KEY (project_id) REFERENCES project (id_project)
-> );
Query OK, 0 rows affected, 2 warnings (0.07 sec)
```

**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`).

```
mysql> CREATE TABLE materials (  
-> id_mater INT(10) AUTO_INCREMENT,  
-> name_mater VARCHAR(255) NOT NULL,  
-> type VARCHAR(255) NOT NULL,  
-> price_per_piece FLOAT(10) NOT NULL,  
-> specifications_id INT(10) NOT NULL,  
-> manufacturers_id INT(10) NOT NULL,  
-> PRIMARY KEY (id_mater),  
-> FOREIGN KEY (specifications_id) REFERENCES specifications (id_specifications),  
-> FOREIGN KEY (manufacturers_id) REFERENCES manufacturers (id_manufacturers)  
-> );  
Query OK, 0 rows affected, 3 warnings (0.07 sec)
```

**Рисунок 12 — Создание таблицы `materials`**

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

```
mysql> CREATE TABLE list (  
-> store_id INT(10),  
-> mater_id INT(10),  
-> PRIMARY KEY (store_id, mater_id),  
-> FOREIGN KEY (store_id) REFERENCES shops(id_store),  
-> FOREIGN KEY (mater_id) REFERENCES materials(id_mater)  
-> );  
Query OK, 0 rows affected, 2 warnings (0.06 sec)
```

**Рисунок 13 — Создание таблицы `list`**

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

```
mysql> CREATE TABLE checks (  
-> banya_id INT(10),  
-> materials_id INT(10),  
-> date_purchase DATETIME NOT NULL,  
-> quantity INT(10) NOT NULL,  
-> PRIMARY KEY (banya_id, materials_id),  
-> FOREIGN KEY (banya_id) REFERENCES banya (id),  
-> FOREIGN KEY (materials_id) REFERENCES materials (id_mater)  
-> );  
Query OK, 0 rows affected, 3 warnings (0.06 sec)
```

**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 профессии (`profession_id`).

```
mysql> CREATE TABLE builders (  
-> service_number INT(10) AUTO_INCREMENT,  
-> fio VARCHAR(255) NOT NULL,  
-> work_exp INT(10) NOT NULL,  
-> contacts TEXT NOT NULL,  
-> revenue INT(10) NOT NULL,  
-> profession_id INT(10) NOT NULL,  
-> PRIMARY KEY (service_number),  
-> FOREIGN KEY (profession_id) REFERENCES profession (id_prof)  
-> );  
Query OK, 0 rows affected, 4 warnings (0.05 sec)
```

**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`).

```
mysql> CREATE TABLE contract (  
-> ban_id INT(10),  
-> service_number INT(10),  
-> link_if VARCHAR(2550) NOT NULL,  
-> date_work_start DATETIME NOT NULL,  
-> date_work_end DATETIME NOT NULL,  
-> PRIMARY KEY (ban_id, service_number),  
-> FOREIGN KEY (ban_id) REFERENCES banya (id),  
-> FOREIGN KEY (service_number) REFERENCES builders (service_number)  
-> );  
Query OK, 0 rows affected, 2 warnings (0.05 sec)
```

**Рисунок 16 — Создание таблицы `contract`**

Произведем просмотр созданных таблиц в базе данных.

```
mysql> show tables;
+-----+
| Tables_in_banya |
+-----+
| architect         |
| banya             |
| builders          |
| checks            |
| contract          |
| draw              |
| list              |
| manufacturers     |
| materials         |
| profession        |
| project           |
| shops             |
| specifications    |
+-----+
13 rows in set (0.00 sec)
```

Рисунок 17 — Просмотр таблиц базы данных banya

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

```
mysql> INSERT INTO shops (id_store, name_shop, address) VALUES
-> (1, 'Петрович', 'рабочий посёлок Боброво, с67Ю'),
-> (2, 'Лемана Про', 'МКАД, 24-й километр, 1А'),
-> (3, 'Благовар', 'МКАД, 25-й километр, вл1'),
-> (4, 'Империя Печей', 'Московская улица 10, Чехов'),
-> (5, 'Магазин "ЭкоСтрой"', 'г. Москва, ул. Природная, д. 6'),
-> (6, 'Магазин "Качественные материалы"', 'г. Москва, ул. Качества, д. 7'),
-> (7, 'Магазин "Баня плюс"', 'г. Москва, ул. Солнечная, д. 8'),
-> (8, 'Магазин "Все для ремонта"', 'г. Москва, ул. Ремонтная, д. 9'),
-> (9, 'Магазин "СтройПартнер"', 'г. Москва, ул. Партнерская, д. 10'),
-> (10, 'Магазин "СтройГрад"', 'г. Москва, ул. Тихая, д. 5'),
-> (11, 'Магазин стройматериалов "Стройка"', 'г. Москва, ул. Ленина, д. 1'),
-> (12, 'Магазин "Все для бани"', 'г. Москва, ул. Пушкина, д. 2'),
-> (13, 'Магазин "Баня и Сауна"', 'г. Москва, ул. Цветочная, д. 3'),
-> (14, 'Магазин "Дерево и Сталь"', 'г. Москва, ул. Фруктовая, д. 4');
Query OK, 14 rows affected (0.02 sec)
Records: 14 Duplicates: 0 Warnings: 0

mysql> INSERT INTO architect (service_number, fio, work_exp, contacts) VALUES
-> (1, 'Ким Кирилл Сергеевич', 2, 'noobmaster6969@ya.ru'),
-> (2, 'Чистяков Тимофей Александрович', 4, 'qwerty123@mail.ru'),
-> (3, 'Сидоров Сидор Сидорович', 8, 'sidorov@example.com'),
-> (4, 'Кузнецов Кузьма Кузьмич', 12, 'kuznetsov@example.com'),
-> (5, 'Алексеев Алексей Алексеевич', 7, 'alekseev@example.com'),
-> (6, 'Марков Марк Маркович', 9, 'markov@example.com'),
-> (7, 'Федоров Федор Федорович', 6, 'fedorov@example.com'),
-> (8, 'Григорьев Григорий Григорьевич', 4, 'grigorev@example.com'),
-> (9, 'Смирнов Сергей Сергеевич', 11, 'smirnov@example.com'),
-> (10, 'Тихонов Тимофей Тимофеевич', 3, 'tikhonov@example.com');
Query OK, 10 rows affected (0.01 sec)
Records: 10 Duplicates: 0 Warnings: 0
```

Рисунок 18 — Заполнение таблиц shops и architect

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

```
mysql> INSERT INTO manufacturers (id_manufacturers, name_manuf) VALUES
-> (1, 'БетонСтрой'),
-> (2, 'ЛесПродукт'),
-> (3, 'МеталлДекор'),
-> (4, 'ТехноНИКОЛЬ'),
-> (5, 'ДеревоТорг'),
-> (6, 'ОкнаПлюс'),
-> (7, 'ДвериМир'),
-> (8, 'ООО "Качественные материалы"'),
-> (9, 'ЗАО "Надежные поставки"'),
-> (10, 'ИП "Лес и Дерево"'),
-> (11, 'ЗАО "Баня"'),
-> (12, 'ООО "СтройМастер"'),
-> (13, 'ИП "Деревянные конструкции"'),
-> (14, 'ЗАО "ЭкоСтрой"'),
-> (15, 'ООО "Баня и Сауна"'),
-> (16, 'ЗАО "СтройГрад"'),
-> (17, 'ИП "Ремонт и Стройка");
Query OK, 17 rows affected (0.01 sec)
Records: 17  Duplicates: 0  Warnings: 0

mysql> INSERT INTO profession (id_prof, name_prof) VALUES
-> (1, 'Плотник'),
-> (2, 'Каменщик'),
-> (3, 'Сантехник'),
-> (4, 'Электрик'),
-> (5, 'Архитектор'),
-> (6, 'Печник'),
-> (7, 'Строитель'),
-> (8, 'Дизайнер'),
-> (9, 'Монтажник'),
-> (10, 'Инженер');
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

**Рисунок 19 — Заполнение таблиц manufacturers и profession**

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

```
mysql> INSERT INTO draw (id_draw, link_sketch) VALUES
-> (1, 'https://example.com/sketch1.jpg'),
-> (2, 'https://example.com/sketch2.jpg'),
-> (3, 'https://example.com/sketch3.jpg'),
-> (4, 'https://example.com/sketch4.jpg'),
-> (5, 'https://example.com/sketch5.jpg'),
-> (6, 'https://example.com/sketch6.jpg'),
-> (7, 'https://example.com/sketch7.jpg'),
-> (8, 'https://example.com/sketch8.jpg'),
-> (9, 'https://example.com/sketch9.jpg'),
-> (10, 'https://example.com/sketch10.jpg');
Query OK, 10 rows affected (0.01 sec)
Records: 10 Duplicates: 0 Warnings: 0

mysql> INSERT INTO specifications (id_specifications, name_spec, size_a_metr, size_b_metr, weight_kilo) VALUES
-> (1, 'Доска обрезная', 6.0, 0.15, 20.0),
-> (2, 'Брус клееный', 3.0, 0.1, 15.0),
-> (3, 'Фундаментный блок', 0.4, 0.2, 30.0),
-> (4, 'Профнастил', 3.0, 1.0, 25.0),
-> (5, 'Изоляция', 1.0, 0.5, 5.0),
-> (6, 'Окно пластиковое', 1.2, 1.5, 25.0),
-> (7, 'Дверь деревянная', 2.0, 0.8, 40.0),
-> (8, 'Система отопления', 1.5, 1.5, 8.0),
-> (9, 'Труба пластиковая', 6.0, 0.1, 2.0),
-> (10, 'Кровельные системы', 3.0, 2.0, 35.0),
-> (11, 'Деревянное бревно', 6.0, 0.3, 50.0),
-> (12, 'Кирпич', 0.25, 0.12, 3.5),
-> (13, 'Плита фундаментная', 2.0, 1.0, 150.0),
-> (14, 'Крыша металлочерепица', 1.2, 2.5, 10.0),
-> (15, 'Утеплитель', 1.2, 0.6, 5.0),
-> (16, 'Сайдинг', 3.0, 0.2, 8.0);
Query OK, 16 rows affected (0.02 sec)
Records: 16 Duplicates: 0 Warnings: 0
```

**Рисунок 20 — Заполнение таблиц draw и specifications**

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

```
mysql> INSERT INTO project (id_project, plan_start_date, plan_end_date, square, price, draw_id_draw, architect_service_number) VALUES
-> (1, '2023-12-01 07:00:00', '2024-01-11 17:00:00', 50, 3000000.0, 1, 1),
-> (2, '2024-01-15 10:00:00', '2024-02-20 10:00:00', 70, 4500000.0, 2, 1),
-> (3, '2024-02-20 08:30:00', '2024-03-31 20:00:00', 100, 6000000.0, 3, 2),
-> (4, '2024-03-25 09:00:00', '2024-04-26 09:00:00', 80, 5000000.0, 4, 2),
-> (5, '2024-04-30 10:00:00', '2024-05-30 10:00:00', 90, 5500000.0, 5, 3),
-> (6, '2024-04-30 09:30:00', '2024-06-30 09:30:00', 60, 3500000.0, 5, 4),
-> (7, '2024-06-20 11:00:00', '2024-07-24 11:00:00', 75, 4000000.0, 6, 5),
-> (8, '2024-07-25 12:00:00', '2024-08-25 14:00:00', 85, 4800000.0, 8, 5),
-> (9, '2024-08-30 08:20:00', '2024-10-30 10:00:00', 95, 6200000.0, 7, 6),
-> (10, '2024-09-10 09:00:00', '2024-10-10 09:00:00', 55, 3200000.0, 6, 6),
-> (11, '2024-01-15 09:00:00', '2024-03-15 17:00:00', 50, 1200000.0, 8, 7),
-> (12, '2024-02-01 10:00:00', '2024-04-01 18:00:00', 75, 1500000.0, 5, 8),
-> (13, '2024-03-10 08:30:00', '2024-05-10 16:30:00', 100, 2000000.0, 2, 8),
-> (14, '2024-04-05 11:00:00', '2024-06-05 15:00:00', 60, 1300000.0, 3, 9),
-> (15, '2024-05-20 09:30:00', '2024-07-20 17:30:00', 80, 1600000.0, 1, 9),
-> (16, '2024-06-15 10:15:00', '2024-08-15 14:15:00', 90, 1800000.0, 9, 10),
-> (17, '2024-07-01 08:45:00', '2024-09-01 16:45:00', 70, 1400000.0, 10, 10),
-> (18, '2024-08-10 09:30:00', '2024-10-10 15:30:00', 55, 1250000.0, 10, 2),
-> (19, '2024-09-05 11:15:00', '2024-11-05 17:15:00', 85, 1700000.0, 1, 4),
-> (20, '2024-10-20 10:30:00', '2024-12-20 14:30:00', 65, 1350000.0, 2, 6);
Query OK, 20 rows affected (0.01 sec)
Records: 20 Duplicates: 0 Warnings: 0

mysql> INSERT INTO banya (id, owner, address, start_date_construction, end_date_construction, project_id) VALUES
-> (1, 'Сидоров Сидор Сидорович', 'г. Москва, ул. Садовая, д. 10', '2023-12-01 07:00:00', '2024-01-11 17:00:00', 1),
-> (2, 'Кузнецов Кузьма Кузьмич', 'г. Москва, ул. Цветочная, д. 5', '2024-01-15 10:00:00', '2024-02-20 10:00:00', 2),
-> (3, 'Петрова Анна Петровна', 'г. Москва, ул. Набережная, д. 15', '2024-02-20 08:30:00', '2024-03-31 20:00:00', 3),
-> (4, 'Смирнов Сергей Сергеевич', 'г. Москва, ул. Зелёная, д. 20', '2024-03-25 09:00:00', '2024-04-26 09:00:00', 4),
-> (5, 'Федоров Федор Федорович', 'г. Москва, ул. Осенняя, д. 25', '2024-04-30 10:00:00', '2024-05-30 10:00:00', 5),
-> (6, 'Григорьев Григорий Григорьевич', 'г. Москва, ул. Летняя, д. 30', '2024-04-30 09:30:00', '2024-06-30 09:30:00', 6),
-> (7, 'Тихонов Тимофей Тимофеевич', 'г. Москва, ул. Зимняя, д. 35', '2024-06-20 11:00:00', '2024-07-24 11:00:00', 7),
-> (8, 'Марков Марк Маркович', 'г. Москва, ул. Весенняя, д. 40', '2024-07-25 12:00:00', '2024-08-25 14:00:00', 8),
-> (9, 'Алексеев Алексей Алексеевич', 'г. Москва, ул. Природная, д. 45', '2024-08-30 08:20:00', '2024-10-30 10:00:00', 9),
-> (10, 'Иванова Ирина Ивановна', 'г. Москва, ул. Небесная, д. 50', '2024-09-10 09:00:00', '2024-10-10 09:00:00', 10),
-> (11, 'Быков Алексей Смирнов', 'Московская область, д. Лесная, 15', '2024-01-05 08:00:00', '2024-02-15 17:00:00', 11),
-> (12, 'Егорова Екатерина Петрова', 'Ленинградская область, г. Сосновый Бор, ул. Озерная, 22', '2024-02-01 09:00:00', '2024-03-01 18:00:00', 12),
-> (13, 'Петров Сергей Кузнецов', 'Тульская область, д. Лесная Поляна, 5', '2024-03-10 10:30:00', '2024-04-20 16:30:00', 13),
-> (14, 'Ключикова Марина Ивановна', 'Калужская область, с. Речное, ул. Центральная, 12', '2024-04-15 11:00:00', '2024-05-30 15:00:00', 14),
-> (15, 'Питерский Дмитрий Васильев', 'Рязанская область, г. Рязань, пр. Победы, 30', '2024-05-20 09:30:00', '2024-06-25 14:30:00', 15),
-> (16, 'Рыгина Ольга Сидорова', 'Владимирская область, д. Зеленая Роща, ул. Солнечная, 8', '2024-06-16 08:45:00', '2024-08-10 17:15:00', 16),
-> (17, 'Степанов Игорь Федоров', 'Нижегородская область, с. Березовка, ул. Школьная, 3', '2024-07-05 10:00:00', '2024-08-15 19:00:00', 17),
-> (18, 'Аршинова Татьяна Орлова', 'Смоленская область, г. Смоленск, ул. Ленина, 45', '2024-08-10 09:30:00', '2024-10-20 16:30:00', 18),
-> (19, 'Виноградов Анатолий Григорьев', 'Ярославская область, д. Кострома, ул. Природная, 18', '2024-09-15 11:15:00', '2024-10-25 18:45:00', 19),
-> (20, 'Жукова Наталья Коваленко', 'Тверская область, г. Тверь, ул. Набережная, 5', '2024-10-21 10:30:00', '2024-12-15 15:30:00', 20);
Query OK, 20 rows affected (0.01 sec)
Records: 20 Duplicates: 0 Warnings: 0
```

**Рисунок 21 — Заполнение таблиц project и banya**

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

```
mysql> INSERT INTO materials (id_mater, name_mater, type, price_per_piece, specifications_id, manufacturers_id) VALUES
-> (1, 'Доска обрезная', 'Дерево', 150.0, 1, 1),
-> (2, 'Брус клееный', 'Дерево', 500.0, 2, 2),
-> (3, 'Фундаментный блок', 'Бетон', 1000.0, 3, 3),
-> (4, 'Кровельный материал', 'Металл', 2000.0, 4, 5),
-> (5, 'Изоляция минеральная', 'Минеральные материалы', 3000.0, 5, 15),
-> (6, 'Окно пластиковое', 'Пластик', 1500.0, 6, 16),
-> (7, 'Дверь деревянная', 'Дерево', 2000.0, 7, 17),
-> (8, 'Труба пластиковая', 'Пластик', 500.0, 8, 8),
-> (9, 'Система отопления', 'Металл', 10000.0, 9, 9),
-> (10, 'Кровельные системы', 'Металл', 1500.0, 10, 10),
-> (11, 'Доска обрезная', 'Дерево', 500.0, 1, 11),
-> (12, 'Брус клееный', 'Дерево', 1200.0, 2, 12),
-> (13, 'Кирпич', 'Кирпич', 15.0, 12, 13),
-> (14, 'Фундаментный блок', 'Бетон', 800.0, 3, 11),
-> (15, 'Металлочерепица', 'Металл', 350.0, 10, 10),
-> (16, 'Утеплитель', 'Изоляция', 200.0, 15, 8),
-> (17, 'Пластиковое окно', 'Окно', 2500.0, 6, 5),
-> (18, 'Дверь деревянная', 'Дерево', 3000.0, 7, 4),
-> (19, 'Труба пластиковая', 'Пластик', 100.0, 9, 3),
-> (20, 'Сайдинг', 'Металл', 200.0, 16, 6),
```

Рисунок 22 — Заполнение таблиц materials

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

```
mysql> INSERT INTO list (store_id, mater_id) VALUES
-> (1, 1),
-> (1, 2),
-> (1, 3),
-> (2, 4),
-> (2, 5),
-> (2, 6),
-> (3, 7),
-> (3, 8),
-> (3, 9),
-> (4, 10),
-> (4, 11),
-> (4, 12),
-> (5, 13),
-> (5, 14),
-> (6, 15),
-> (7, 16),
-> (7, 17),
-> (8, 18),
-> (9, 19),
-> (10, 20),
```

Рисунок 23 — Заполнение таблиц list

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

```
mysql> INSERT INTO checks (banya_id, materials_id, date_purchase, quantity) VALUES
-> (1, 1, '2023-11-29', 20),
-> (1, 2, '2023-11-28', 15),
-> (1, 3, '2023-11-29', 10),
-> (2, 4, '2024-01-14', 25),
-> (2, 5, '2024-01-15', 30),
-> (3, 6, '2024-02-21', 12),
-> (3, 7, '2024-02-19', 8),
-> (4, 8, '2024-03-23', 20),
-> (5, 9, '2024-04-29', 18),
-> (6, 10, '2024-04-30', 22),
-> (7, 22, '2024-06-20', 56),
-> (8, 13, '2024-07-25', 65),
-> (9, 15, '2024-08-30', 76),
-> (10, 20, '2024-09-10', 33),
-> (11, 23, '2024-01-15', 11),
-> (12, 47, '2024-02-01', 8),
-> (13, 32, '2024-03-10', 1),
-> (14, 50, '2024-04-05', 30),
-> (15, 34, '2024-05-20', 23),
-> (16, 21, '2024-06-15', 32),
-> (17, 18, '2024-07-01', 89),
-> (18, 20, '2024-08-10', 101),
-> (19, 19, '2024-09-05', 66),
-> (20, 1, '2024-10-20', 58);
Query OK, 24 rows affected (0.01 sec)
Records: 24  Duplicates: 0  Warnings: 0
```

Рисунок 24 — Заполнение таблицы checks

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



```
mysql> INSERT INTO builders (service_number, fio, work_exp, contacts, revenue, profession_id) VALUES
-> (1, 'Смирнов Сергей Сергеевич', 8, 'smirnov@example.com', 60000, 1),
-> (2, 'Федоров Федор Федорович', 4, 'fedorov@example.com', 40000, 1),
-> (3, 'Григорьев Григорий Григорьевич', 6, 'grigorev@example.com', 50000, 2),
-> (4, 'Тихонов Тимофей Тимофеевич', 5, 'tikhonov@example.com', 45000, 6),
-> (5, 'Марков Марк Маркович', 7, 'markov@example.com', 70000, 4),
-> (6, 'Алексеев Алексей Алексеевич', 9, 'alekseev@example.com', 80000, 6),
-> (7, 'Ким Кирилл Сергеевич', 2, 'noobmaster6969@ya.ru', 25000, 5),
-> (8, 'Кузнецов Кузьма Кузьмич', 11, 'kuznetsov@example.com', 100000, 1),
-> (9, 'Иванова Ирина Ивановна', 2, 'ivanova@example.com', 30000, 3),
-> (10, 'Петрова Анна Петровна', 3, 'petrova@example.com', 35000, 3),
-> (11, 'Иванов Иван Иванович', 5, '8-999-111-22-33', 60000, 2),
-> (12, 'Петров Петр Петрович', 5, '8-999-222-33-44', 80000, 7),
-> (13, 'Сидоров Сидор Сидорович', 5, '8-999-333-44-55', 50000, 5),
-> (14, 'Смирнова Анна Сергеевна', 5, '8-999-444-55-66', 75000, 2),
-> (15, 'Кузнецов Алексей Викторович', 5, '8-999-555-66-77', 90000, 8),
-> (16, 'Васильев Сергей Николаевич', 5, '8-999-666-77-88', 55000, 10),
-> (17, 'Федорова Ольга Владимировна', 5, '8-999-777-88-99', 70000, 9),
-> (18, 'Морозов Дмитрий Андреевич', 5, '8-999-888-99-00', 65000, 9),
-> (19, 'Лебедева Екатерина Игоревна', 5, '8-999-999-00-11', 72000, 6),
-> (20, 'Григорьев Артем Валерьевич', 5, '8-999-000-11-22', 85000, 1);
Query OK, 20 rows affected (0.01 sec)
Records: 20 Duplicates: 0 Warnings: 0

mysql> INSERT INTO contract (ban_id, service_number, link_if, date_work_start, date_work_end) VALUES
-> (1, 1, 'https://example.com/contract1.pdf', '08:00:00', '17:00:00'),
-> (2, 2, 'https://example.com/contract2.pdf', '09:00:00', '18:00:00'),
-> (3, 3, 'https://example.com/contract3.pdf', '10:00:00', '16:00:00'),
-> (4, 4, 'https://example.com/contract4.pdf', '08:30:00', '17:30:00'),
-> (5, 5, 'https://example.com/contract5.pdf', '09:15:00', '18:15:00'),
-> (6, 6, 'https://example.com/contract6.pdf', '10:45:00', '19:00:00'),
-> (7, 7, 'https://example.com/contract7.pdf', '11:00:00', '20:00:00'),
-> (8, 8, 'https://example.com/contract8.pdf', '09:30:00', '18:30:00'),
-> (9, 9, 'https://example.com/contract9.pdf', '08:15:00', '17:45:00'),
-> (10, 10, 'https://example.com/contract10.pdf', '10:30:00', '19:15:00'),
-> (11, 1, 'https://example.com/contract11.pdf', '08:00:00', '17:00:00'),
-> (12, 2, 'https://example.com/contract12.pdf', '09:00:00', '18:00:00'),
-> (13, 3, 'https://example.com/contract13.pdf', '10:00:00', '16:00:00'),
-> (14, 4, 'https://example.com/contract14.pdf', '01:30:00', '17:30:00'),
-> (15, 5, 'https://example.com/contract15.pdf', '09:00:00', '18:15:00'),
-> (16, 6, 'https://example.com/contract16.pdf', '10:00:00', '19:00:00'),
-> (17, 7, 'https://example.com/contract17.pdf', '11:00:00', '21:00:00'),
-> (18, 8, 'https://example.com/contract18.pdf', '10:30:00', '22:00:00'),
-> (19, 9, 'https://example.com/contract19.pdf', '09:15:00', '22:45:00'),
-> (20, 10, 'https://example.com/contract20.pdf', '10:00:00', '20:15:00');
Query OK, 20 rows affected (0.01 sec)
Records: 20 Duplicates: 0 Warnings: 0
```

**Рисунок 25 — Заполнение таблицы builders и contract**

Посмотрим все таблицы после заполнения при помощи команды SELECT

\* FROM «название таблицы»;



```
mysql> describe architect;
```

Field	Type	Null	Key	Default	Extra
service_number	int	NO	PRI	NULL	auto_increment
fio	varchar(255)	NO		NULL	
work_exp	int	NO		NULL	
contacts	text	NO		NULL	

```
4 rows in set (0.00 sec)
```

```
mysql> select * from architect;
```

service_number	fio	work_exp	contacts
1	Ким Кирилл Сергеевич	2	noobmaster6969@ya.ru
2	Чистяков Тимофей Александрович	4	qwerty123@mail.ru
3	Сидоров Сидор Сидорович	8	sidorov@example.com
4	Кузнецов Кузьма Кузьмич	12	kuznetsov@example.com
5	Алексеев Алексей Алексеевич	7	alekseev@example.com
6	Марков Марк Маркович	9	markov@example.com
7	Федоров Федор Федорович	6	fedorov@example.com
8	Григорьев Григорий Григорьевич	4	grigorev@example.com
9	Смирнов Сергей Сергеевич	11	smirnov@example.com
10	Тихонов Тимофей Тимофеевич	3	tikhonov@example.com

```
10 rows in set (0.00 sec)
```

Рисунок 26 — Содержание таблицы architect

```
mysql> select * from draw;
```

id_draw	link_sketch
1	https://example.com/sketch1.jpg
2	https://example.com/sketch2.jpg
3	https://example.com/sketch3.jpg
4	https://example.com/sketch4.jpg
5	https://example.com/sketch5.jpg
6	https://example.com/sketch6.jpg
7	https://example.com/sketch7.jpg
8	https://example.com/sketch8.jpg
9	https://example.com/sketch9.jpg
10	https://example.com/sketch10.jpg

```
10 rows in set (0.00 sec)
```

Рисунок 27 — Содержание таблицы draw

```
mysql> SELECT * FROM profession;
```

id_prof	name_prof
1	Плотник
2	Каменщик
3	Сантехник
4	Электрик
5	Архитектор
6	Печник
7	Строитель
8	Дизайнер
9	Монтажник
10	Инженер

```
10 rows in set (0.00 sec)
```

Рисунок 28 — Содержание таблицы profession

```
mysql> SELECT * FROM shops;
```

id_store	name_shop	address
1	Петрович	рабочий посёлок Боброво, с67Ю
2	Лемана Про	МКАД, 24-й километр, 1А
3	Благовар	МКАД, 25-й километр, вл1
4	Империя Печей	Московская улица 10, Чехов
5	Магазин "ЭкоСтрой"	г. Москва, ул. Природная, д. 6
6	Магазин "Качественные материалы"	г. Москва, ул. Качества, д. 7
7	Магазин "Баня плюс"	г. Москва, ул. Солнечная, д. 8
8	Магазин "Все для ремонта"	г. Москва, ул. Ремонтная, д. 9
9	Магазин "СтройПартнер"	г. Москва, ул. Партнерская, д. 10
10	Магазин "СтройГрад"	г. Москва, ул. Тихая, д. 5
11	Магазин стройматериалов "Стройка"	г. Москва, ул. Ленина, д. 1
12	Магазин "Все для бани"	г. Москва, ул. Пушкина, д. 2
13	Магазин "Баня и Сауна"	г. Москва, ул. Цветочная, д. 3
14	Магазин "Дерево и Сталь"	г. Москва, ул. Фруктовая, д. 4

```
14 rows in set (0.00 sec)
```

Рисунок 29 — Содержание таблицы shops

```
mysql> SELECT * FROM manufacturers;
```

id_manufacturers	name_manuf
1	БетонСтрой
2	ЛесПродукт
3	МеталлДекор
4	ТехноНИКОЛЬ
5	ДеревоТорг
6	ОкнаПлюс
7	ДвериМир
8	ООО "Качественные материалы"
9	ЗАО "Надежные поставки"
10	ИП "Лес и Дерево"
11	ЗАО "Баня"
12	ООО "СтройМастер"
13	ИП "Деревянные конструкции"
14	ЗАО "ЭкоСтрой"
15	ООО "Баня и Сауна"
16	ЗАО "СтройГрад"
17	ИП "Ремонт и Стройка"

```
17 rows in set (0.00 sec)
```

Рисунок 30 — Содержание таблицы manufacturers

```
mysql> SELECT * FROM specifications;
```

id_specifications	name_spec	size_a_metr	size_b_metr	weight_kilo
1	Доска обрезная	6	0.15	20
2	Брус клееный	3	0.1	15
3	Фундаментный блок	0.4	0.2	30
4	Профнастил	3	1	25
5	Изоляция	1	0.5	5
6	Окно пластиковое	1.2	1.5	25
7	Дверь деревянная	2	0.8	40
8	Система отопления	1.5	1.5	8
9	Труба пластиковая	6	0.1	2
10	Кровельные системы	3	2	35
11	Деревянное бревно	6	0.3	50
12	Кирпич	0.25	0.12	3.5
13	Плита фундаментная	2	1	150
14	Крыша металлочерепица	1.2	2.5	10
15	Утеплитель	1.2	0.6	5
16	Сайдинг	3	0.2	8

```
16 rows in set (0.00 sec)
```

Рисунок 31 — Содержание таблицы specifications

```
mysql> SELECT * FROM project;
```

id_project	plan_start_date	plan_end_date	square	price	draw_id_draw	architect_service_number
1	2023-12-01 07:00:00	2024-01-11 17:00:00	50	3000000	1	1
2	2024-01-15 10:00:00	2024-02-20 10:00:00	70	4500000	2	1
3	2024-02-20 08:30:00	2024-03-31 20:00:00	100	6000000	3	2
4	2024-03-25 09:00:00	2024-04-26 09:00:00	80	5000000	4	2
5	2024-04-30 10:00:00	2024-05-30 10:00:00	90	5500000	5	3
6	2024-04-30 09:30:00	2024-06-30 09:30:00	60	3500000	5	4
7	2024-06-20 11:00:00	2024-07-24 11:00:00	75	4000000	6	5
8	2024-07-25 12:00:00	2024-08-25 14:00:00	85	4800000	8	5
9	2024-08-30 08:20:00	2024-10-30 10:00:00	95	6200000	7	6
10	2024-09-10 09:00:00	2024-10-10 09:00:00	55	3200000	6	6
11	2024-01-15 09:00:00	2024-03-15 17:00:00	50	1200000	8	7
12	2024-02-01 10:00:00	2024-04-01 18:00:00	75	1500000	5	8
13	2024-03-10 08:30:00	2024-05-10 16:30:00	100	2000000	2	8
14	2024-04-05 11:00:00	2024-06-05 15:00:00	60	1300000	3	9
15	2024-05-20 09:30:00	2024-07-20 17:30:00	80	1600000	1	9
16	2024-06-15 10:15:00	2024-08-15 14:15:00	90	1800000	9	10
17	2024-07-01 08:45:00	2024-09-01 16:45:00	70	1400000	10	10
18	2024-08-10 09:30:00	2024-10-10 15:30:00	55	1250000	10	2
19	2024-09-05 11:15:00	2024-11-05 17:15:00	85	1700000	1	4
20	2024-10-20 10:30:00	2024-12-20 14:30:00	65	1350000	2	6

```
20 rows in set (0.00 sec)
```

Рисунок 32 — Содержание таблицы project

```
mysql> SELECT * FROM banya;
```

id	owner	address	start_date_construction	end_date_construction	project_id
1	Сидоров Сидор Сидорович	г. Москва, ул. Садовая, д. 10	2023-12-01 07:00:00	2024-01-11 17:00:00	1
2	Кузнецов Кузьма Кузьмич	г. Москва, ул. Цветочная, д. 5	2024-01-15 10:00:00	2024-02-20 10:00:00	2
3	Петрова Анна Петровна	г. Москва, ул. Набережная, д. 15	2024-02-20 08:30:00	2024-03-31 20:00:00	3
4	Смирнов Сергей Сергеевич	г. Москва, ул. Зеленая, д. 20	2024-03-25 09:00:00	2024-04-26 09:00:00	4
5	Федоров Федор Федорович	г. Москва, ул. Осенняя, д. 25	2024-04-30 10:00:00	2024-05-30 10:00:00	5
6	Григорьев Григорий Григорьевич	г. Москва, ул. Летняя, д. 30	2024-04-30 09:30:00	2024-06-30 09:30:00	6
7	Тихонов Тимофей Тимофеевич	г. Москва, ул. Зимняя, д. 35	2024-06-20 11:00:00	2024-07-24 11:00:00	7
8	Марков Марк Маркович	г. Москва, ул. Весенняя, д. 40	2024-07-25 12:00:00	2024-08-25 14:00:00	8
9	Алексеев Алексей Алексеевич	г. Москва, ул. Природная, д. 45	2024-08-30 08:20:00	2024-10-30 10:00:00	9
10	Иванова Ирина Ивановна	г. Москва, ул. Небесная, д. 50	2024-09-10 09:00:00	2024-10-10 09:00:00	10
11	Быков Алексей Смирнов	Московская область, д. Лесная, 15	2024-01-05 08:00:00	2024-02-15 17:00:00	11
12	Егорова Екатерина Петрова	Ленинградская область, г. Сосновый Бор, ул. Озерная, 22	2024-02-01 09:00:00	2024-03-01 18:00:00	12
13	Петров Сергей Кузнецов	Тульская область, д. Лесная Поляна, 5	2024-03-10 10:30:00	2024-04-20 16:30:00	13
14	Ключникова Марина Иванова	Калужская область, с. Речное, ул. Центральная, 12	2024-04-15 11:00:00	2024-05-30 15:00:00	14
15	Питерский Дмитрий Васильев	Рязанская область, г. Рязань, пр. Победы, 30	2024-05-20 09:30:00	2024-06-25 14:30:00	15
16	Рыгина Ольга Сидорова	Владимирская область, д. Зеленая Роща, ул. Солнечная, 8	2024-06-16 08:45:00	2024-08-10 17:15:00	16
17	Степанов Игорь Федоров	Нижегородская область, с. Березовка, ул. Школьная, 3	2024-07-05 10:00:00	2024-08-15 19:00:00	17
18	Аришнова Татьяна Орлова	Смоленская область, г. Смоленск, ул. Ленина, 45	2024-08-10 09:30:00	2024-10-20 16:30:00	18
19	Виноградов Анатолий Григорьев	Ярославская область, д. Кострома, ул. Природная, 18	2024-09-15 11:15:00	2024-10-25 18:45:00	19
20	Жукова Наталья Коваленко	Тверская область, г. Тверь, ул. Набережная, 5	2024-10-21 10:30:00	2024-12-15 15:30:00	20

20 rows in set (0.00 sec)

Рисунок 33 — Содержание таблицы banya

```
mysql> SELECT * FROM materials;
```

id_mater	name_mater	type	price_per_piece	specifications_id	manufacturers_id
1	Доска обрезная	Дерево	150	1	1
2	Брус клееный	Дерево	500	2	2
3	Фундаментный блок	Бетон	1000	3	3
4	Кровельный материал	Металл	2000	4	5
5	Изоляция минеральная	Минеральные материалы	3000	5	15
6	Окно пластиковое	Пластик	1500	6	16
7	Дверь деревянная	Дерево	2000	7	17
8	Труба пластиковая	Пластик	500	8	8
9	Система отопления	Металл	10000	9	9
10	Кровельные системы	Металл	1500	10	10
11	Доска обрезная	Дерево	500	1	11
12	Брус клееный	Дерево	1200	2	12
13	Кирпич	Кирпич	15	12	13
14	Фундаментный блок	Бетон	800	3	11
15	Металлочерепица	Металл	350	10	10
16	Утеплитель	Изоляция	200	15	8
17	Пластиковое окно	Окно	2500	6	5
18	Дверь деревянная	Дерево	3000	7	4
19	Труба пластиковая	Пластик	100	9	3
20	Сайдинг	Металл	200	16	6

Рисунок 34 — Содержание таблицы materials

```
mysql> SELECT * FROM list;
```

store_id	mater_id
1	1
1	2
1	3
2	4
2	5
2	6
3	7
3	8
3	9
4	10
4	11
4	12
5	13
5	14
6	15
7	16

Рисунок 35 — Содержание таблицы list

```
mysql> SELECT * FROM checks;
```

banya_id	materials_id	date_purchase	quantity
1	1	2023-11-29 00:00:00	20
1	2	2023-11-28 00:00:00	15
1	3	2023-11-29 00:00:00	10
2	4	2024-01-14 00:00:00	25
2	5	2024-01-15 00:00:00	30
3	6	2024-02-21 00:00:00	12
3	7	2024-02-19 00:00:00	8
4	8	2024-03-23 00:00:00	20
5	9	2024-04-29 00:00:00	18
6	10	2024-04-30 00:00:00	22
7	22	2024-06-20 00:00:00	56
8	13	2024-07-25 00:00:00	65
9	15	2024-08-30 00:00:00	76
10	20	2024-09-10 00:00:00	33
11	23	2024-01-15 00:00:00	11
12	47	2024-02-01 00:00:00	8
13	32	2024-03-10 00:00:00	1
14	50	2024-04-05 00:00:00	30
15	34	2024-05-20 00:00:00	23
16	21	2024-06-15 00:00:00	32
17	18	2024-07-01 00:00:00	89
18	20	2024-08-10 00:00:00	101
19	19	2024-09-05 00:00:00	66
20	1	2024-10-20 00:00:00	58

24 rows in set (0.00 sec)

Рисунок 36 — Содержание таблицы checks

```
mysql> SELECT * FROM builders;
```

service_number	fio	work_exp	contacts	revenue	profession_id
1	Смирнов Сергей Сергеевич	8	smirnov@example.com	60000	1
2	Федоров Федор Федорович	4	fedorov@example.com	40000	1
3	Григорьев Григорий Григорьевич	6	grigorev@example.com	50000	2
4	Тихонов Тимофей Тимофеевич	5	tikhonov@example.com	45000	6
5	Марков Марк Маркович	7	markov@example.com	70000	4
6	Алексеев Алексей Алексеевич	9	alekseev@example.com	80000	6
7	Ким Кирилл Сергеевич	2	noobmaster6969@ya.ru	25000	5
8	Кузнецов Кузьма Кузьмич	11	kuznetsov@example.com	100000	1
9	Иванова Ирина Ивановна	2	ivanova@example.com	30000	3
10	Петрова Анна Петровна	3	petrova@example.com	35000	3
11	Иванов Иван Иванович	5	8-999-111-22-33	60000	2
12	Петров Петр Петрович	5	8-999-222-33-44	80000	7
13	Сидоров Сидор Сидорович	5	8-999-333-44-55	50000	5
14	Смирнова Анна Сергеевна	5	8-999-444-55-66	75000	2
15	Кузнецов Алексей Викторович	5	8-999-555-66-77	90000	8
16	Васильев Сергей Николаевич	5	8-999-666-77-88	55000	10
17	Федорова Ольга Владимировна	5	8-999-777-88-99	70000	9
18	Морозов Дмитрий Андреевич	5	8-999-888-99-00	65000	9
19	Лебедева Екатерина Игоревна	5	8-999-999-00-11	72000	6
20	Григорьев Артем Валерьевич	5	8-999-000-11-22	85000	1

20 rows in set (0.00 sec)

Рисунок 37 — Содержание таблицы builders

```
mysql> SELECT * FROM contract;
```

ban_id	service_number	link_if	date_work_start	date_work_end
1	1	https://example.com/contract1.pdf	08:00:00	17:00:00
2	2	https://example.com/contract2.pdf	09:00:00	18:00:00
3	3	https://example.com/contract3.pdf	10:00:00	16:00:00
4	4	https://example.com/contract4.pdf	08:30:00	17:30:00
5	5	https://example.com/contract5.pdf	09:15:00	18:15:00
6	6	https://example.com/contract6.pdf	10:45:00	19:00:00
7	7	https://example.com/contract7.pdf	11:00:00	20:00:00
8	8	https://example.com/contract8.pdf	09:30:00	18:30:00
9	9	https://example.com/contract9.pdf	08:15:00	17:45:00
10	10	https://example.com/contract10.pdf	10:30:00	19:15:00
11	1	https://example.com/contract11.pdf	08:00:00	17:00:00
12	2	https://example.com/contract12.pdf	09:00:00	18:00:00
13	3	https://example.com/contract13.pdf	10:00:00	16:00:00
14	4	https://example.com/contract14.pdf	01:30:00	17:30:00
15	5	https://example.com/contract15.pdf	09:00:00	18:15:00
16	6	https://example.com/contract16.pdf	10:00:00	19:00:00
17	7	https://example.com/contract17.pdf	11:00:00	21:00:00
18	8	https://example.com/contract18.pdf	10:30:00	22:00:00
19	9	https://example.com/contract19.pdf	09:15:00	22:45:00
20	10	https://example.com/contract20.pdf	10:00:00	20:15:00

```
20 rows in set (0.00 sec)
```

Рисунок 38 — Содержание таблицы contract

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

```
SELECT fio FROM architect;
```

```
mysql> SELECT fio FROM architect;
```

fio
Ким Кирилл Сергеевич
Чистяков Тимофей Александрович
Сидоров Сидор Сидорович
Кузнецов Кузьма Кузьмич
Алексеев Алексей Алексеевич
Марков Марк Маркович
Федоров Федор Федорович
Григорьев Григорий Григорьевич
Смирнов Сергей Сергеевич
Тихонов Тимофей Тимофеевич

```
10 rows in set (0.00 sec)
```

Рисунок 39 — Вывод столбца fio

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



```
mysql> SELECT fio, contacts FROM builders;
```

fio	contacts
Смирнов Сергей Сергеевич	smirnov@example.com
Федоров Федор Федорович	fedorov@example.com
Григорьев Григорий Григорьевич	grigorev@example.com
Тихонов Тимофей Тимофеевич	tikhonov@example.com
Марков Марк Маркович	markov@example.com
Алексеев Алексей Алексеевич	alekseev@example.com
Ким Кирилл Сергеевич	noobmaster6969@ya.ru
Кузнецов Кузьма Кузьмич	kuznetsov@example.com
Иванова Ирина Ивановна	ivanova@example.com
Петрова Анна Петровна	petrova@example.com
Иванов Иван Иванович	8-999-111-22-33
Петров Петр Петрович	8-999-222-33-44
Сидоров Сидор Сидорович	8-999-333-44-55
Смирнова Анна Сергеевна	8-999-444-55-66
Кузнецов Алексей Викторович	8-999-555-66-77
Васильев Сергей Николаевич	8-999-666-77-88
Федорова Ольга Владимировна	8-999-777-88-99
Морозов Дмитрий Андреевич	8-999-888-99-00
Лебедева Екатерина Игоревна	8-999-999-00-11
Григорьев Артем Валерьевич	8-999-000-11-22

```
20 rows in set (0.00 sec)
```

**Рисунок 40 — Вывод столбцов fio и contacts строителей**

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

```
mysql> SELECT * FROM materials ORDER BY price_per_piece DESC;
```

id_mater	name_mater	type	price_per_piece	specifications_id	manufacturers_id
40	Системы отопления (котел)	Отопление	25000	12	4
9	Система отопления	Металл	10000	9	9
5	Изоляция минеральная	Минеральные материалы	3000	5	15
18	Дверь деревянная	Дерево	3000	7	4
17	Пластиковое окно	Окно	2500	6	5
7	Дверь деревянная	Дерево	2000	7	17
4	Кровельный материал	Металл	2000	4	5
50	Электропроводка	Электрика	2000	3	17
37	Кровельные материалы (профнастил)	Кровля	1800	4	8
6	Окно пластиковое	Пластик	1500	6	16
10	Кровельные системы	Металл	1500	10	10
32	Брус	Деревянные материалы	1500	2	10
24	Гидроизоляция	Изоляция	1500	5	6
39	Теплоизоляционные плиты	Изоляция	1300	13	9
12	Брус клееный	Дерево	1200	2	12
31	Доска обрезная	Деревянные материалы	1200	1	11

**Рисунок 41 — Сортировка материалов в порядке убывания**

```
mysql> SELECT * FROM project ORDER BY plan_start_date ASC;
```

id_project	plan_start_date	plan_end_date	square	price	draw_id_draw	architect_service_number
1	2023-12-01 07:00:00	2024-01-11 17:00:00	50	3000000	1	1
11	2024-01-15 09:00:00	2024-03-15 17:00:00	50	1200000	8	7
2	2024-01-15 10:00:00	2024-02-20 10:00:00	70	4500000	2	1
12	2024-02-01 10:00:00	2024-04-01 18:00:00	75	1500000	5	8
3	2024-02-20 08:30:00	2024-03-31 20:00:00	100	6000000	3	2
13	2024-03-10 08:30:00	2024-05-10 16:30:00	100	2000000	2	8
4	2024-03-25 09:00:00	2024-04-26 09:00:00	80	5000000	4	2
14	2024-04-05 11:00:00	2024-06-05 15:00:00	60	1300000	3	9
6	2024-04-30 09:30:00	2024-06-30 09:30:00	60	3500000	5	4
5	2024-04-30 10:00:00	2024-05-30 10:00:00	90	5500000	5	3
15	2024-05-20 09:30:00	2024-07-20 17:30:00	80	1600000	1	9
16	2024-06-15 10:15:00	2024-08-15 14:15:00	90	1800000	9	10
7	2024-06-20 11:00:00	2024-07-24 11:00:00	75	4000000	6	5
17	2024-07-01 08:45:00	2024-09-01 16:45:00	70	1400000	10	10
8	2024-07-25 12:00:00	2024-08-25 14:00:00	85	4800000	8	5
18	2024-08-10 09:30:00	2024-10-10 15:30:00	55	1250000	10	2
9	2024-08-30 08:20:00	2024-10-30 10:00:00	95	6200000	7	6
19	2024-09-05 11:15:00	2024-11-05 17:15:00	85	1700000	1	4
10	2024-09-10 09:00:00	2024-10-10 09:00:00	55	3200000	6	6
20	2024-10-20 10:30:00	2024-12-20 14:30:00	65	1350000	2	6

```
20 rows in set (0.01 sec)
```

**Рисунок 42 — 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.

```
mysql> SELECT * FROM architect WHERE work_exp=4;
```

service_number	fio	work_exp	contacts
2	Чистяков Тимофей Александрович	4	qwerty123@mail.ru
8	Григорьев Григорий Григорьевич	4	grigorev@example.com

```
2 rows in set (0.01 sec)
```

**Рисунок 43 — Условие для таблицы architect**

```
mysql> SELECT * FROM architect WHERE work_exp>4;
```

service_number	fio	work_exp	contacts
3	Сидоров Сидор Сидорович	8	sidorov@example.com
4	Кузнецов Кузьма Кузьмич	12	kuznetsov@example.com
5	Алексеев Алексей Алексеевич	7	alekseev@example.com
6	Марков Марк Маркович	9	markov@example.com
7	Федоров Федор Федорович	6	fedorov@example.com
9	Смирнов Сергей Сергеевич	11	smirnov@example.com

```
6 rows in set (0.00 sec)
```

**Рисунок 44 — Условие для таблицы architect**



```
mysql> SELECT * FROM architect WHERE work_exp<4;
```

service_number	fio	work_exp	contacts
1	Ким Кирилл Сергеевич	2	noobmaster6969@ya.ru
10	Тихонов Тимофей Тимофеевич	3	tikhonov@example.com

```
2 rows in set (0.00 sec)
```

Рисунок 45 — Условие для таблицы architect

```
mysql> SELECT * FROM architect WHERE work_exp>=4;
```

service_number	fio	work_exp	contacts
2	Чистяков Тимофей Александрович	4	qwerty123@mail.ru
3	Сидоров Сидор Сидорович	8	sidorov@example.com
4	Кузнецов Кузьма Кузьмич	12	kuznetsov@example.com
5	Алексеев Алексей Алексеевич	7	alekseev@example.com
6	Марков Марк Маркович	9	markov@example.com
7	Федоров Федор Федорович	6	fedorov@example.com
8	Григорьев Григорий Григорьевич	4	grigorev@example.com
9	Смирнов Сергей Сергеевич	11	smirnov@example.com

```
8 rows in set (0.00 sec)
```

Рисунок 46 — Условие для таблицы architect

```
mysql> SELECT * FROM architect WHERE work_exp<=4;
```

service_number	fio	work_exp	contacts
1	Ким Кирилл Сергеевич	2	noobmaster6969@ya.ru
2	Чистяков Тимофей Александрович	4	qwerty123@mail.ru
8	Григорьев Григорий Григорьевич	4	grigorev@example.com
10	Тихонов Тимофей Тимофеевич	3	tikhonov@example.com

```
4 rows in set (0.00 sec)
```

Рисунок 47 — Условие для таблицы architect

```
mysql> SELECT * FROM architect WHERE work_exp!=4;
```

service_number	fio	work_exp	contacts
1	Ким Кирилл Сергеевич	2	noobmaster6969@ya.ru
3	Сидоров Сидор Сидорович	8	sidorov@example.com
4	Кузнецов Кузьма Кузьмич	12	kuznetsov@example.com
5	Алексеев Алексей Алексеевич	7	alekseev@example.com
6	Марков Марк Маркович	9	markov@example.com
7	Федоров Федор Федорович	6	fedorov@example.com
9	Смирнов Сергей Сергеевич	11	smirnov@example.com
10	Тихонов Тимофей Тимофеевич	3	tikhonov@example.com

```
8 rows in set (0.00 sec)
```

Рисунок 48 — Условие для таблицы architect

```
mysql> SELECT * FROM studies WHERE id_medication != 1;
```

id_medication	id_study	start_date	end_date	status	created_at	updated_at
2	2	2023-08-15	2023-09-15	completed	2024-10-17 14:05:18	2024-10-17 14:05:18
3	3	2023-07-01	2023-07-31	completed	2024-10-17 14:05:18	2024-10-17 14:05:18
4	4	2023-08-05	NULL	in_process	2024-10-17 14:05:18	2024-10-17 14:05:18
5	5	2023-06-20	2023-07-20	canceled	2024-10-17 14:05:18	2024-10-17 14:05:18
6	6	2023-09-10	2023-10-10	in_process	2024-10-17 14:05:18	2024-10-17 14:05:18
7	7	2023-08-20	2023-09-20	completed	2024-10-17 14:05:18	2024-10-17 14:05:18
8	8	2023-07-10	NULL	in_process	2024-10-17 14:05:18	2024-10-17 14:05:18
9	9	2023-06-01	2023-06-30	completed	2024-10-17 14:05:18	2024-10-17 14:05:18
10	10	2023-09-25	NULL	in_process	2024-10-17 14:05:18	2024-10-17 14:05:18

```
9 rows in set (0.00 sec)
```

Рисунок 49 — Условие для таблицы architect

```
mysql> SELECT * FROM architect WHERE service_number IS NULL;
Empty set (0.00 sec)
```

Рисунок 50 — Отбираются значения не имеющие значения в указанном поле

```
mysql> SELECT * FROM architect WHERE service_number IS NOT NULL;
```

service_number	fio	work_exp	contacts
1	Ким Кирилл Сергеевич	2	noobmaster6969@ya.ru
2	Чистяков Тимофей Александрович	4	qwerty123@mail.ru
3	Сидоров Сидор Сидорович	8	sidorov@example.com
4	Кузнецов Кузьма Кузьмич	12	kuznetsov@example.com
5	Алексеев Алексей Алексеевич	7	alekseev@example.com
6	Марков Марк Маркович	9	markov@example.com
7	Федоров Федор Федорович	6	fedorov@example.com
8	Григорьев Григорий Григорьевич	4	grigorev@example.com
9	Смирнов Сергей Сергеевич	11	smirnov@example.com
10	Тихонов Тимофей Тимофеевич	3	tikhonov@example.com

```
10 rows in set (0.00 sec)
```

Рисунок 51 — Отбираются значения имеющие значения в указанном поле

```
mysql> SELECT * FROM builders WHERE work_exp BETWEEN 4 AND 8;
```

service_number	fio	work_exp	contacts	revenue	profession_id
1	Смирнов Сергей Сергеевич	8	smirnov@example.com	60000	1
2	Федоров Федор Федорович	4	fedorov@example.com	40000	1
3	Григорьев Григорий Григорьевич	6	grigorev@example.com	50000	2
4	Тихонов Тимофей Тимофеевич	5	tikhonov@example.com	45000	6
5	Марков Марк Маркович	7	markov@example.com	70000	4
11	Иванов Иван Иванович	5	8-999-111-22-33	60000	2
12	Петров Петр Петрович	5	8-999-222-33-44	80000	7
13	Сидоров Сидор Сидорович	5	8-999-333-44-55	50000	5
14	Смирнова Анна Сергеевна	5	8-999-444-55-66	75000	2
15	Кузнецов Алексей Викторович	5	8-999-555-66-77	90000	8
16	Васильев Сергей Николаевич	5	8-999-666-77-88	55000	10
17	Федорова Ольга Владимировна	5	8-999-777-88-99	70000	9
18	Морозов Дмитрий Андреевич	5	8-999-888-99-00	65000	9
19	Лебедева Екатерина Игоревна	5	8-999-999-00-11	72000	6
20	Григорьев Артем Валерьевич	5	8-999-000-11-22	85000	1

```
15 rows in set (0.00 sec)
```

**Рисунок 52 — Выборка строителей с опытом работы в определенном диапазоне**

```
mysql> SELECT * FROM builders WHERE service_number IN (1, 4);
```

service_number	fio	work_exp	contacts	revenue	profession_id
1	Смирнов Сергей Сергеевич	8	smirnov@example.com	60000	1
4	Тихонов Тимофей Тимофеевич	5	tikhonov@example.com	45000	6

```
2 rows in set (0.00 sec)
```

**Рисунок 53 — Отбираются значения соответствующие указанным**

```
mysql> SELECT * FROM builders WHERE service_number NOT IN (1, 4);
```

service_number	fio	work_exp	contacts	revenue	profession_id
2	Федоров Федор Федорович	4	fedorov@example.com	40000	1
3	Григорьев Григорий Григорьевич	6	grigorev@example.com	50000	2
5	Марков Марк Маркович	7	markov@example.com	70000	4
6	Алексеев Алексей Алексеевич	9	alekseev@example.com	80000	6
7	Ким Кирилл Сергеевич	2	noobmaster6969@ya.ru	25000	5
8	Кузнецов Кузьма Кузьмич	11	kuznetsov@example.com	100000	1
9	Иванова Ирина Ивановна	2	ivanova@example.com	30000	3
10	Петрова Анна Петровна	3	petrova@example.com	35000	3
11	Иванов Иван Иванович	5	8-999-111-22-33	60000	2
12	Петров Петр Петрович	5	8-999-222-33-44	80000	7
13	Сидоров Сидор Сидорович	5	8-999-333-44-55	50000	5
14	Смирнова Анна Сергеевна	5	8-999-444-55-66	75000	2
15	Кузнецов Алексей Викторович	5	8-999-555-66-77	90000	8
16	Васильев Сергей Николаевич	5	8-999-666-77-88	55000	10
17	Федорова Ольга Владимировна	5	8-999-777-88-99	70000	9
18	Морозов Дмитрий Андреевич	5	8-999-888-99-00	65000	9
19	Лебедева Екатерина Игоревна	5	8-999-999-00-11	72000	6
20	Григорьев Артем Валерьевич	5	8-999-000-11-22	85000	1

```
18 rows in set (0.00 sec)
```

**Рисунок 54 — Отбираются значения не соответствующие указанным**

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

```
mysql> SELECT * FROM shops WHERE address LIKE 'МКАД%';
```

id_store	name_shop	address
2	Лемана Про	МКАД, 24-й километр, 1А
3	Благовар	МКАД, 25-й километр, вл1

```
2 rows in set (0.00 sec)
```

Рисунок 55 — Отбираются значения, соответствующие образцу с использованием LIKE

```
mysql> SELECT * FROM shops WHERE address NOT LIKE 'МКАД%';
```

id_store	name_shop	address
1	Петрович	рабочий посёлок Боброво, с67Ю
4	Империя Печей	Московская улица 10, Чехов
5	Магазин "ЭкоСтрой"	г. Москва, ул. Природная, д. 6
6	Магазин "Качественные материалы"	г. Москва, ул. Качества, д. 7
7	Магазин "Баня плюс"	г. Москва, ул. Солнечная, д. 8
8	Магазин "Все для ремонта"	г. Москва, ул. Ремонтная, д. 9
9	Магазин "СтройПартнер"	г. Москва, ул. Партнерская, д. 10
10	Магазин "СтройГрад"	г. Москва, ул. Тихая, д. 5
11	Магазин стройматериалов "Стройка"	г. Москва, ул. Ленина, д. 1
12	Магазин "Все для бани"	г. Москва, ул. Пушкина, д. 2
13	Магазин "Баня и Сауна"	г. Москва, ул. Цветочная, д. 3
14	Магазин "Дерево и Сталь"	г. Москва, ул. Фруктовая, д. 4

```
12 rows in set (0.00 sec)
```

Рисунок 56 — Отбираются не соответствующие значения с использованием NOT LIKE

```
mysql> SELECT * FROM shops WHERE address LIKE '%Москва%';
```

id_store	name_shop	address
5	Магазин "ЭкоСтрой"	г. Москва, ул. Природная, д. 6
6	Магазин "Качественные материалы"	г. Москва, ул. Качества, д. 7
7	Магазин "Баня плюс"	г. Москва, ул. Солнечная, д. 8
8	Магазин "Все для ремонта"	г. Москва, ул. Ремонтная, д. 9
9	Магазин "СтройПартнер"	г. Москва, ул. Партнерская, д. 10
10	Магазин "СтройГрад"	г. Москва, ул. Тихая, д. 5
11	Магазин стройматериалов "Стройка"	г. Москва, ул. Ленина, д. 1
12	Магазин "Все для бани"	г. Москва, ул. Пушкина, д. 2
13	Магазин "Баня и Сауна"	г. Москва, ул. Цветочная, д. 3
14	Магазин "Дерево и Сталь"	г. Москва, ул. Фруктовая, д. 4

```
10 rows in set (0.00 sec)
```

Рисунок 57 — Метасимвол

## Изменение данных в таблице.

Забыли добавить в таблице builders – rep (рейтинг).

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

```
mysql> describe builders;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| service_number | int           | NO   | PRI | NULL    | auto_increment |
| fio            | varchar(255)  | NO   |     | NULL    |                |
| work_exp       | int           | NO   |     | NULL    |                |
| contacts       | text          | NO   |     | NULL    |                |
| revenue        | int           | NO   |     | NULL    |                |
| profession_id  | int           | NO   | MUL | NULL    |                |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> ALTER TABLE builders ADD rep FLOAT(10) AFTER work_exp;
Query OK, 0 rows affected (0.04 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

Рисунок 58 — Добавление нового столбца в таблицу

Запишем значения в новый столбец.

```
mysql> UPDATE builders SET rep=3.4
-> WHERE service_number=1;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql>
mysql> UPDATE builders SET rep=4.7
-> WHERE service_number=2;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql>
mysql> UPDATE builders SET rep=2.7
-> WHERE service_number=3;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql>
mysql> UPDATE builders SET rep=1.0
-> WHERE service_number=4;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql>
mysql> UPDATE builders SET rep=5.0
-> WHERE service_number=5;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

Рисунок 59 — Добавления значений

```
mysql> SELECT * FROM builders;
```

service_number	fio	work_exp	rep	contacts	revenue	profession_id
1	Смирнов Сергей Сергеевич	8	3.4	smirnov@example.com	60000	1
2	Федоров Федор Федорович	4	4.7	fedorov@example.com	40000	1
3	Григорьев Григорий Григорьевич	6	2.7	grigorev@example.com	50000	2
4	Тихонов Тимофей Тимофеевич	5	1	tikhonov@example.com	45000	6
5	Марков Марк Маркович	7	5	markov@example.com	70000	4
6	Алексеев Алексей Алексеевич	9	4.6	alekseev@example.com	80000	6
7	Ким Кирилл Сергеевич	2	5	noobmaster6969@ya.ru	25000	5
8	Кузнецов Кузьма Кузьмич	11	0.8	kuznetsov@example.com	100000	1
9	Иванова Ирина Ивановна	2	4.1	ivanova@example.com	30000	3
10	Петрова Анна Петровна	3	3.8	petrova@example.com	35000	3
11	Иванов Иван Иванович	5	2.3	8-999-111-22-33	60000	2
12	Петров Петр Петрович	5	0.6	8-999-222-33-44	80000	7
13	Сидоров Сидор Сидорович	5	0.5	8-999-333-44-55	50000	5
14	Смирнова Анна Сергеевна	5	3	8-999-444-55-66	75000	2
15	Кузнецов Алексей Викторович	5	4.9	8-999-555-66-77	90000	8
16	Васильев Сергей Николаевич	5	4.5	8-999-666-77-88	55000	10
17	Федорова Ольга Владимировна	5	3.4	8-999-777-88-99	70000	9
18	Морозов Дмитрий Андреевич	5	3.3	8-999-888-99-00	65000	9
19	Лебедева Екатерина Игоревна	5	4.5	8-999-999-00-11	72000	6
20	Григорьев Артем Валерьевич	5	4.8	8-999-000-11-22	85000	1

```
20 rows in set (0.00 sec)
```

**Рисунок 60 — Обновление информации**

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

```
mysql> ALTER TABLE builders CHANGE COLUMN rep rating FLOAT(10);
Query OK, 0 rows affected (0.04 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

**Рисунок 61 — Замена названия столбца**

```
mysql> DELETE FROM builders
-> WHERE rating = 0;
Query OK, 0 rows affected (0.00 sec)
```

**Рисунок 62 — Удаление записей с 0 рейтингом**

Операция соединения. Вывести металл и название производителя:

```
mysql> SELECT name_mater, name_manuf FROM materials, manufacturers
-> WHERE materials.type='Металл' AND
-> materials.manufacturers_id=manufacturers.id_manufacturers;
```

name_mater	name_manuf
Кровельный материал	ДеревоТорг
Система отопления	ЗАО "Надежные поставки"
Кровельные системы	ИП "Лес и Дерево"
Металлочерепица	ИП "Лес и Дерево"
Сайдинг	ОкнаПлюс

```
5 rows in set (0.00 sec)
```

**Рисунок 63 — Вывод информации по критериям**



Операция объединения. Вывести название материалов и цену:

```
mysql> SELECT name_mater, price_per_piece FROM materials
-> INNER JOIN shops ON materials.id_mater = shops.id_store
-> WHERE materials.price_per_piece > 1000;
```

name_mater	price_per_piece
Кровельный материал	2000
Изоляция минеральная	3000
Окно пластиковое	1500
Дверь деревянная	2000
Система отопления	10000
Кровельные системы	1500
Брус клееный	1200

7 rows in set (0.00 sec)

Рисунок 64 — Вывод информации по критериям

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

```
mysql> SELECT shops.name_shop, shops.address, materials.name_mater, materials.price_per_piece FROM shops
-> JOIN list ON shops.id_store = list.store_id
-> JOIN materials ON list.mater_id = materials.id_mater
-> WHERE materials.type = 'Дерево'
-> ORDER BY shops.name_shop;
```

name_shop	address	name_mater	price_per_piece
Благовар	МКАД, 25-й километр, вл1	Дверь деревянная	2000
Империя Печей	Московская улица 10, Чехов	Доска обрезная	500
Империя Печей	Московская улица 10, Чехов	Брус клееный	1200
Магазин "Все для ремонта"	г. Москва, ул. Ремонтная, д. 9	Дверь деревянная	3000
Петрович	рабочий посёлок Боброво, с67Ю	Доска обрезная	150
Петрович	рабочий посёлок Боброво, с67Ю	Брус клееный	500

6 rows in set (0.00 sec)

Рисунок 65 — Вывод информации по критериям

Операция разности. Вывести ФИО строителя и опыт работы:

```
mysql> SELECT builders.FIO, builders.work_exp FROM builders
-> WHERE NOT EXISTS
-> (SELECT architect.FIO, architect.work_exp FROM architect
-> WHERE architect.FIO=builders.FIO AND architect.work_exp=builders.work_exp);
```

FIO	work_exp
Смирнов Сергей Сергеевич	8
Федоров Федор Федорович	4
Григорьев Григорий Григорьевич	6
Тихонов Тимофей Тимофеевич	5
Марков Марк Маркович	7
Алексеев Алексей Алексеевич	9
Кузнецов Кузьма Кузьмич	11
Иванова Ирина Ивановна	2
Петрова Анна Петровна	3
Иванов Иван Иванович	5
Петров Петр Петрович	5
Сидоров Сидор Сидорович	5
Смирнова Анна Сергеевна	5
Кузнецов Алексей Викторович	5
Васильев Сергей Николаевич	5
Федорова Ольга Владимировна	5
Морозов Дмитрий Андреевич	5
Лебедева Екатерина Игоревна	5
Григорьев Артем Валерьевич	5

```
19 rows in set (0.00 sec)
```

**Рисунок 66 — Вывод информации по критериям**

Операция группировки (GROUP BY).

```
mysql> SELECT type, COUNT(*) as cnt FROM materials
-> WHERE type = 'Пластик'
-> GROUP BY type;
```

type	cnt
Пластик	3

```
1 row in set (0.00 sec)
```

**Рисунок 67 — Вывод количество пластиков**

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



```
mysql> SELECT materials.name_mater, materials.price_per_piece, shops.name_shop FROM materials
-> JOIN list ON materials.id_mater = list.mater_id
-> JOIN shops ON list.store_id = shops.id_store
-> ORDER BY materials.price_per_piece ASC;
```

name_mater	price_per_piece	name_shop
Кирпич	15	Магазин "ЭкоСтрой"
Кирпич лицевой	15	Магазин "Качественные материалы"
Бетонные блоки	50	Магазин "Качественные материалы"
Электропроводка	50	Магазин "Дерево и Сталь"
Арматура	80	Петрович
Труба пластиковая	100	Магазин "СтройПартнер"
Доска обрезная	150	Петрович
Сайдинг	200	Магазин "СтройГрад"
Утеплитель	200	Магазин "Баня плюс"
Кровельная пленка	250	Магазин "Баня и Сауна"
Кровельная черепица	250	Петрович
Светодиодные лампы	300	Магазин "Дерево и Сталь"
Лак для дерева	300	Магазин "ЭкоСтрой"
Металлочерепица	350	Магазин "Качественные материалы"
Брус клееный	500	Петрович
Штукатурка	500	Магазин "Все для ремонта"
Цемент	500	Магазин "СтройГрад"
Цемент	500	Магазин стройматериалов "Стройка"
Труба пластиковая	500	Благовар
Доска обрезная	500	Империя Печей
Лист фанеры	600	Магазин "Все для ремонта"
Пенопласт	700	Петрович
Линолеум	700	Магазин "Баня и Сауна"
Плита ОСБ	800	Петрович
Фундаментный блок	800	Магазин "ЭкоСтрой"
Щебень	800	Магазин "Все для бани"
Стекловата	850	Магазин "СтройГрад"
Сайдинг	900	Магазин "Баня плюс"

Рисунок 68 — 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.

```
select id_mater, name_mater from materials;
```

id_mater	name_mater
1	Доска обрезная
2	Брус клееный
3	Фундаментный блок
4	Кровельный материал
5	Изоляция минеральная
6	Окно пластиковое
7	Дверь деревянная
8	Труба пластиковая
9	Система отопления
10	Кровельные системы

Рисунок 69 — Выборка только id и названия материалов

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

```
select * from builders
where rating > 4;
```

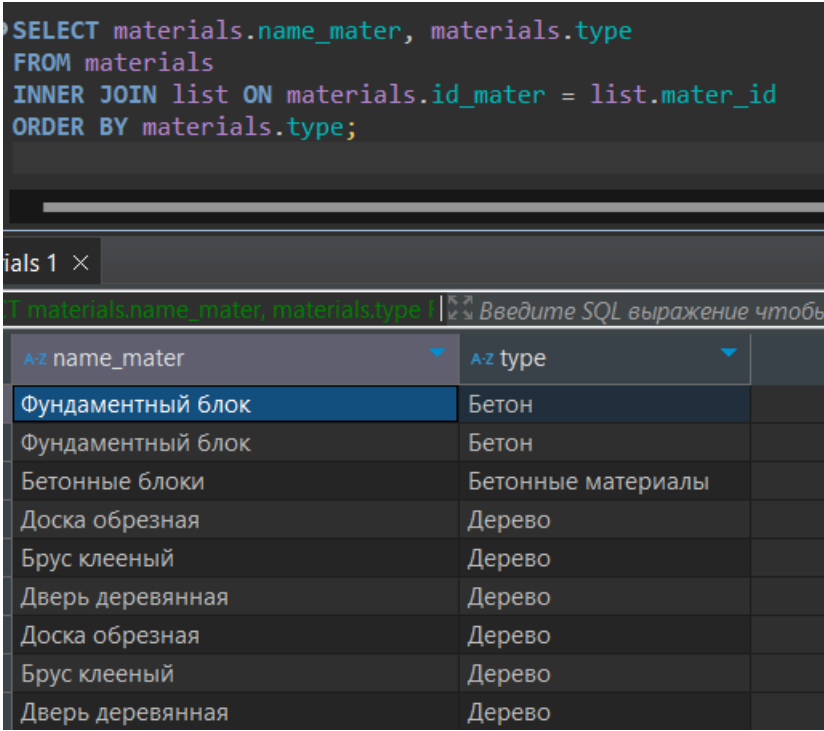
service_number	fio	work_exp	rating	contacts	revenue	profession_id
2	Федоров Федор Федорович	4	4,7	fedorov@example.com	40 000	1
5	Марков Марк Маркович	7	5	markov@example.com	70 000	4
6	Алексеев Алексей Алексеевич	9	4,6	alekseev@example.com	80 000	6
7	Ким Кирилл Сергеевич	2	5	noobmaster6969@ya.ru	25 000	5
9	Иванова Ирина Ивановна	2	4,1	ivanova@example.com	30 000	3
15	Кузнецов Алексей Викторович	5	4,9	8-999-555-66-77	90 000	8
16	Васильев Сергей Николаевич	5	4,5	8-999-666-77-88	55 000	10
19	Лебедева Екатерина Игоревна	5	4,5	8-999-999-00-11	72 000	6
20	Григорьев Артем Валерьевич	5	4,8	8-999-000-11-22	85 000	1

Рисунок 70 — 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).



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.



The screenshot shows a SQL query in a dark-themed editor:

```
SELECT materials.name_mater, materials.type
FROM materials
INNER JOIN list ON materials.id_mater = list.mater_id
ORDER BY materials.type;
```

Below the query, the results are displayed in a table with two columns: 'name\_mater' and 'type'. The first row is highlighted in blue.

A-z name_mater	A-z type
Фундаментный блок	Бетон
Фундаментный блок	Бетон
Бетонные блоки	Бетонные материалы
Доска обрезная	Дерево
Брус клееный	Дерево
Дверь деревянная	Дерево
Доска обрезная	Дерево
Брус клееный	Дерево
Дверь деревянная	Дерево

Рисунок 73 — 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.

```
SELECT builders.FIO, builders.work_exp FROM builders
WHERE NOT EXISTS
(SELECT architect.FIO, architect.work_exp FROM architect
WHERE architect.FIO=builders.FIO AND architect.work_exp=builders.work_exp);
```

ers 1 ×

1 builders.FIO, builders.work\_exp FROM 1 Введите SQL выражение чтобы отфильтровать результ

A-z FIO	123 work_exp	
Смирнов Сергей Сергеевич	8	
Федоров Федор Федорович	4	
Григорьев Григорий Григорьевич	6	
Тихонов Тимофей Тимофеевич	5	
Марков Марк Маркович	7	
Алексеев Алексей Алексеевич	9	
Кузнецов Кузьма Кузьмич	11	
Иванова Ирина Ивановна	2	
Петрова Анна Петровна	3	
Иванов Иван Иванович	5	

**Рисунок 74 — 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).

```
SELECT architect.fio, COUNT(banya.id) AS count_banya
FROM architect
JOIN project ON architect.service_number = project.architect_service_number
JOIN banya ON project.id_project = banya.project_id
GROUP BY architect.fio;
```

ect 1 ×

1 architect.fio, COUNT(banya.id) AS count\_banya Введите SQL выражение чтобы отфильтровать результ

A-z fio	123 count_banya	
Ким Кирилл Сергеевич	2	
Чистяков Тимофей Александрович	3	
Сидоров Сидор Сидорович	1	
Кузнецов Кузьма Кузьмич	2	
Алексеев Алексей Алексеевич	2	
Марков Марк Маркович	3	
Федоров Федор Федорович	1	
Григорьев Григорий Григорьевич	2	
Смирнов Сергей Сергеевич	2	
Тихонов Тимофей Тимофеевич	2	

**Рисунок 75 — Количество бань, построенным каждым архитектором**

## Запрос 8. Операция сортировки.

The screenshot shows a SQL query editor with the following query:

```
SELECT * FROM shops  
ORDER BY name_shop ASC;
```

Below the query editor, a table with 3 columns is displayed:

id_store	name_shop	address
3	Благовар	МКАД, 25-й километр, вл1
4	Империя Печей	Московская улица 10, Чехов
2	Лемана Про	МКАД, 24-й километр, 1А
13	Магазин "Баня и Сауна"	г. Москва, ул. Цветочная, д. 3
7	Магазин "Баня плюс"	г. Москва, ул. Солнечная, д. 8
12	Магазин "Все для бани"	г. Москва, ул. Пушкина, д. 2
8	Магазин "Все для ремонта"	г. Москва, ул. Ремонтная, д. 9
14	Магазин "Дерево и Сталь"	г. Москва, ул. Фруктовая, д. 4
6	Магазин "Качественные материалы"	г. Москва, ул. Качества, д. 7
10	Магазин "СтройГрад"	г. Москва, ул. Тихая, д. 5
9	Магазин "СтройПартнер"	г. Москва, ул. Партнерская, д. 10
5	Магазин "ЭкоСтрой"	г. Москва, ул. Природная, д. 6
11	Магазин стройматериалов "Стройка"	г. Москва, ул. Ленина, д. 1
1	Петрович	рабочий посёлок Боброво, с67Ю

Рисунок 76 — 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;
```

CALL AddArchitect('Пушкин Александр Сергеевич', 7, 'onegin@mail.com');

ect 1 ×

\* from architect | Введите SQL выражение чтобы отфильтровать результаты

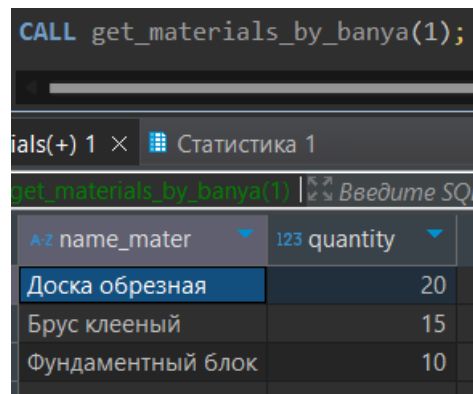
123 service_number	A-Z fio	123 work_exp	A-Z contacts
1	Ким Кирилл Сергеевич	2	noobmaster6969@ya.ru
2	Чистяков Тимофей Александрович	4	qwerty123@mail.ru
3	Сидоров Сидор Сидорович	8	sidorov@example.com
4	Кузнецов Кузьма Кузьмич	12	kuznetsov@example.com
5	Алексеев Алексей Алексеевич	7	alekseev@example.com
6	Марков Марк Маркович	9	markov@example.com
7	Федоров Федор Федорович	6	fedorov@example.com
8	Григорьев Григорий Григорьевич	4	grigorev@example.com
9	Смирнов Сергей Сергеевич	11	smirnov@example.com
10	Тихонов Тимофей Тимофеевич	3	tikhonov@example.com
11	Пушкин Александр Сергеевич	7	onegin@mail.com

**Рисунок 77 — Процедура для добавления архитектора**

2. Процедура для получения списка материалов, используемых 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;
```



A-Z name_mater	123 quantity
Доска обрезная	20
Брус клееный	15
Фундаментный блок	10

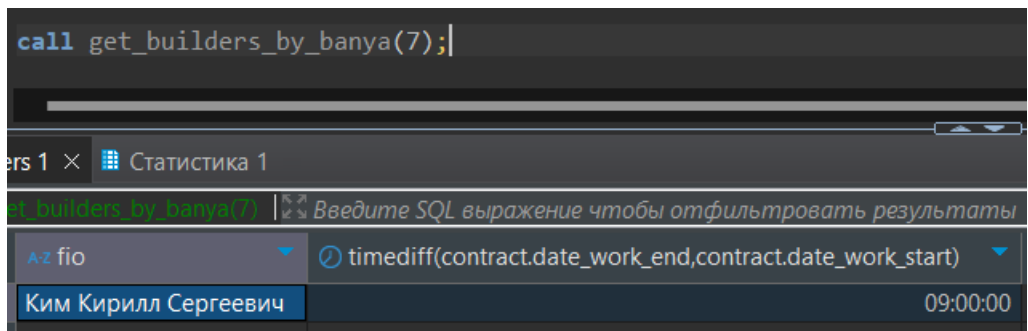
**Рисунок 78 — Процедура для получения списка материалов, используемых в конкретной бане**

3. 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 79 — Процедура для получения информации сколько часов строитель работал**

4. 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;
```

id	owner	address	start_date_construction	end_date_construction
8	Марков Марк Маркович	г. Москва, ул. Весенняя, д. 40	2024-07-25 12:00:00	2024-08-25 14:00:00
9	Алексеев Алексей Алексеевич	г. Москва, ул. Природная, д. 45	2024-08-30 08:20:00	2024-10-30 10:00:00
10	Иванова Ирина Ивановна	г. Москва, ул. Небесная, д.50	2024-09-10 09:00:00	2024-10-10 09:00:00
11	Быков Алексей Смирнов	Московская область, д. Лесная, 15	2024-01-05 08:00:00	2024-02-15 17:00:00
12	Егорова Екатерина Петрова	Ленинградская область, г. Сосновый Бор, ул. Озерная,	2024-02-01 09:00:00	2024-03-01 18:00:00
13	Петров Сергей Кузнецов	Тульская область, д. Лесная Поляна, 5	2024-03-10 10:30:00	2024-04-20 16:30:00
14	Ключникова Марина Иванова	Калужская область, с. Речное, ул. Центральная, 12	2024-04-15 11:00:00	2024-05-30 15:00:00
15	Питерский Дмитрий Васильев	Рязанская область, г. Рязань, пр. Победы, 30	2024-05-20 09:30:00	2024-06-25 14:30:00
16	Рыгина Ольга Сидорова	Владимирская область, д. Зеленая Роща, ул. Солнечна	2024-06-16 08:45:00	2024-08-10 17:15:00
17	Степанов Игорь Федоров	Нижегородская область, с. Березовка, ул. Школьная, 3	2024-07-05 10:00:00	2024-08-15 19:00:00
18	Аршинова Татьяна Орлова	Смоленская область, г. Смоленск, ул. Ленина, 45	2024-08-10 09:30:00	2024-10-20 16:30:00
19	Виноградов Анатолий Григорьев	Ярославская область, д. Кострома, ул. Природная, 18	2024-09-15 11:15:00	2024-10-25 18:45:00
20	Жукова Наталья Коваленко	Тверская область, г. Тверь, ул. Набережная, 5	2024-10-21 10:30:00	2024-12-19 10:00:00

**Рисунок 80 — Процедура для обновления даты окончания строительства бани**

5. 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;
```

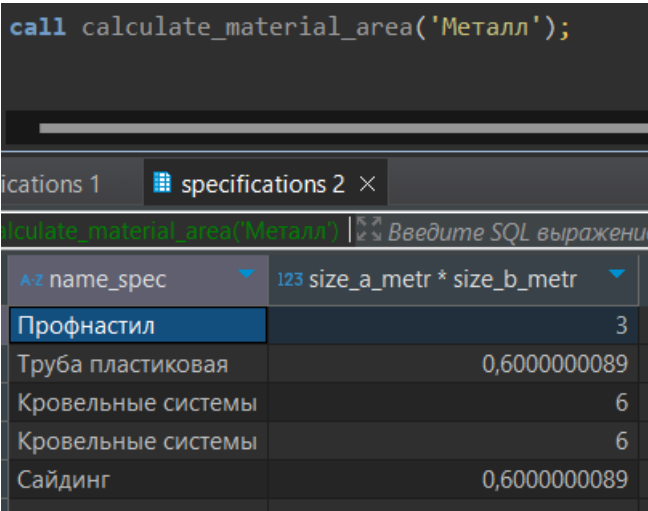


Рисунок 81 — Процедура для получения размер материала

Функции

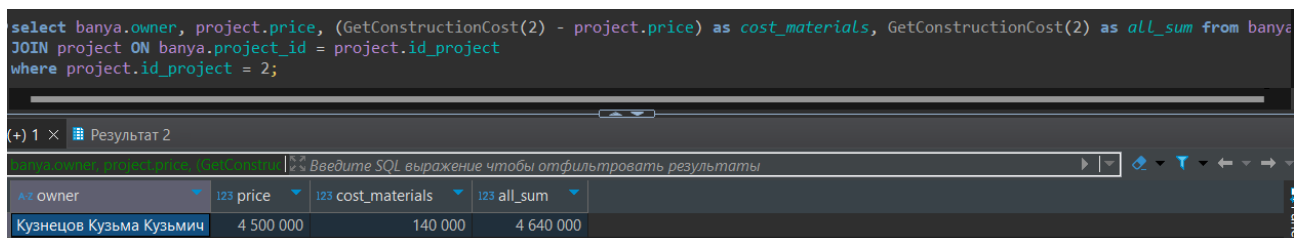
1. Функция для получения всей стоимости строительства бани, включая покупки материалов. Код представлен в Листинге 6. Результат работы процедуры представлен на Рисунке 77.

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;
```



The screenshot shows a SQL query window with the following query:

```
select banya.owner, project.price, (GetConstructionCost(2) - project.price) as cost_materials, GetConstructionCost(2) as all_sum from banya
JOIN project ON banya.project_id = project.id_project
where project.id_project = 2;
```

The result set is displayed in a table with the following data:

owner	price	cost_materials	all_sum
Кузнецов Кузьма Кузьмич	4 500 000	140 000	4 640 000

Рисунок 82 — Функция для получения всей стоимости строительства бани, включая покупки материалов

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

Листинг 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;
```

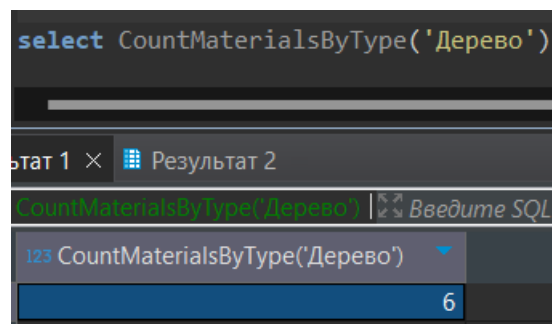


Рисунок 83 — Функция для подсчета количества материалов по типу

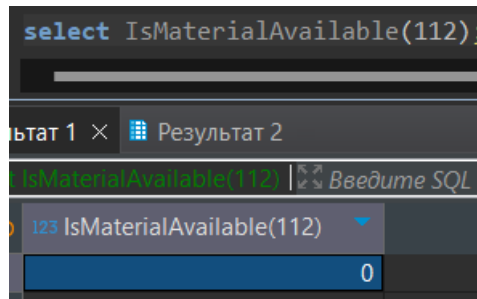
3. Функция на наличия материала в списке. Код представлен в Листинге 8. Результат работы процедуры представлен на Рисунке 79.

Листинг 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 84 — Функция на наличие материала в списке**

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

*Листинг 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;
```

```
SELECT address, count_summer_construction_days(6, 7, 8) AS summer_construction_days from banya
WHERE MONTH(start_date_construction) IN (6, 7, 8);
```

A-Z address	123 summer_construction_days
г. Москва, ул. Зимняя, д. 35	6
г. Москва, ул. Весенняя, д. 40	6
г. Москва, ул. Природная, д. 45	6
Владимирская область, д. Зеленая Роща, ул. Солнечная, 8	6
Нижегородская область, с. Березовка, ул. Школьная, 3	6
Смоленская область, г. Смоленск, ул. Ленина, 45	6

**Рисунок 85 — Функция для подсчета количество дней построенных бань летом**

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

*Листинг 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;
```

```

select architect.fio, average_construction_time(2) FROM architect
join project on architect.service_number = project.architect_service_number
where architect.service_number = 2
group by architect.fio;

```

architect.fio, average_construction_time(2)	
Чистяков Тимофей Александрович	44

**Рисунок 86** — 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;

```



2. 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 стоимости 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;
```

3. 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;

```

4. 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;

```

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

```
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;
```

Имя	CountProject
Ким Кирилл Сергеевич	2
Чистяков Тимофей Александрович	3
Сидоров Сидор Сидорович	1
Кузнецов Кузьма Кузьмич	2
Алексеев Алексей Алексеевич	2
Марков Марк Маркович	3
Федоров Федор Федорович	1
Григорьев Григорий Григорьевич	2
Смирнов Сергей Сергеевич	2
Тихонов Тимофей Тимофеевич	2

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;
```

1 x

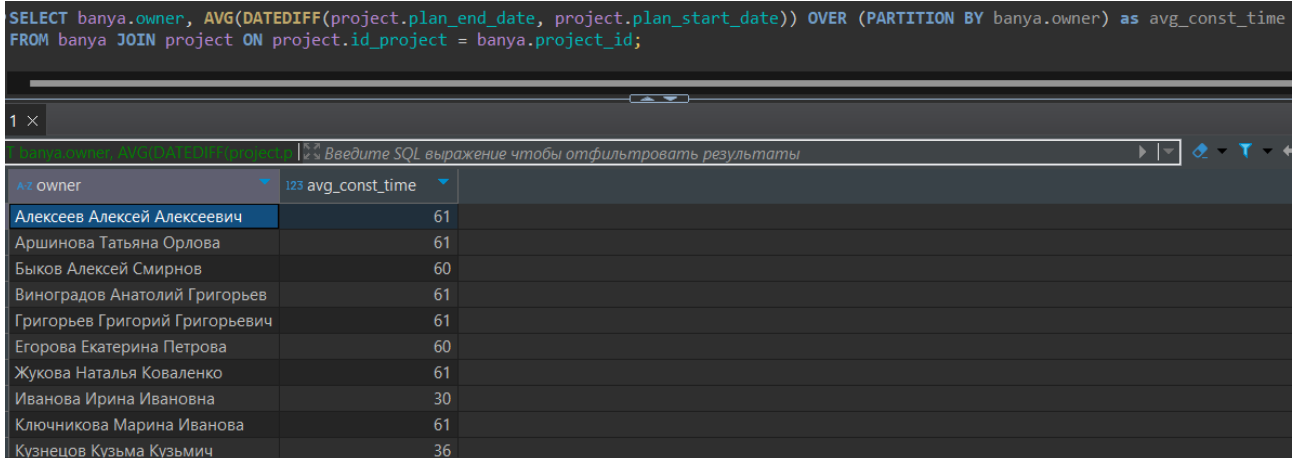
1 banya.owner, banya.start\_date\_construction | Введите SQL выражение чтобы отфильтровать результаты

owner	start_date_construction	sum p
Сидоров Сидор Сидорович	2023-12-01 07:00:00	3 000 000
Быков Алексей Смирнов	2024-01-05 08:00:00	1 200 000
Кузнецов Кузьма Кузьмич	2024-01-15 10:00:00	4 500 000
Егорова Екатерина Петрова	2024-02-01 09:00:00	1 500 000
Петрова Анна Петровна	2024-02-20 08:30:00	6 000 000
Петров Сергей Кузнецов	2024-03-10 10:30:00	2 000 000
Смирнов Сергей Сергеевич	2024-03-25 09:00:00	5 000 000
Ключникова Марина Иванова	2024-04-15 11:00:00	1 300 000
Григорьев Григорий Григорьевич	2024-04-30 09:30:00	3 500 000
Федоров Федор Федорович	2024-04-30 10:00:00	5 500 000

Листинг 18 — Агрегатные функции AVG

```
project.plan_start_date)) OVER (PARTITION BY banya.owner)
as avg_const_time

FROM banya JOIN project ON project.id_project =
banya.project_id;
```



The screenshot shows a SQL query in a database client. The query is: `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;` The results are displayed in a table with two columns: `owner` and `avg_const_time`. The table contains 12 rows of data.

owner	avg_const_time
Алексеев Алексей Алексеевич	61
Аршинова Татьяна Орлова	61
Быков Алексей Смирнов	60
Виноградов Анатолий Григорьев	61
Григорьев Григорий Григорьевич	61
Егорова Екатерина Петрова	60
Жукова Наталья Коваленко	61
Иванова Ирина Ивановна	30
Ключникова Марина Иванова	61
Кузнецов Кузьма Кузьмич	36

**Рисунок 89 — 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;
```

```
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;
```

A-Z name_mater	A-Z name_spec	123 weight_kilo	123 min_weight
Арматура	Труба пластиковая	2	2
Бетонные блоки	Фундаментный блок	30	30
Брус	Брус клееный	15	15
Брус клееный	Брус клееный	15	15
Брус клееный	Брус клееный	15	15
Гидроизоляция	Изоляция	5	5
Гидроизоляция	Профнастил	25	5
Дверь деревянная	Дверь деревянная	40	40
Дверь деревянная	Дверь деревянная	40	40
Доска обрезная	Доска обрезная	20	20

Рисунок 90 — 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;
```

```
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;
```

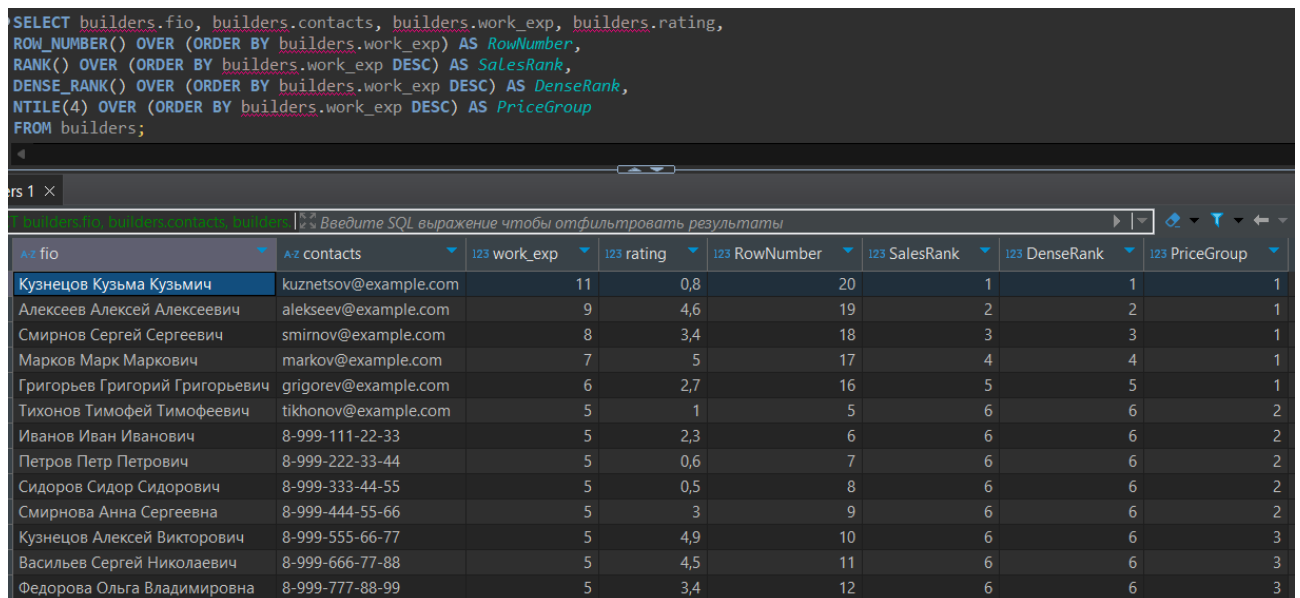
A-Z name_mater	A-Z name_spec	123 price_per_piece	123 max_price
Арматура	Труба пластиковая	80	80
Бетонные блоки	Фундаментный блок	50	50
Брус	Брус клееный	1 500	1 500
Брус клееный	Брус клееный	500	1 200
Брус клееный	Брус клееный	1 200	1 200
Гидроизоляция	Изоляция	1 500	1 500
Гидроизоляция	Профнастил	1 200	1 500
Дверь деревянная	Дверь деревянная	2 000	3 000
Дверь деревянная	Дверь деревянная	3 000	3 000
Доска обрезная	Доска обрезная	150	1 200

Рисунок 91 — Вызов агрегатных функций MAX

## 2) Ранжирующие функции

Литстинг 21 — Ранжирующие функции

```
SELECT      builders.fio,      builders.contacts,
builders.work_exp, builders.rating,
      ROW_NUMBER() OVER (ORDER BY builders.work_exp) AS
RowNumber,
      RANK() OVER (ORDER BY builders.work_exp DESC) AS
SalesRank,
      DENSE_RANK() OVER (ORDER BY builders.work_exp DESC)
AS DenseRank,
      NTILE(4) OVER (ORDER BY builders.work_exp DESC) AS
PriceGroup
FROM builders;
```



```
SELECT builders.fio, builders.contacts, builders.work_exp, builders.rating,
ROW_NUMBER() OVER (ORDER BY builders.work_exp) AS RowNumber,
RANK() OVER (ORDER BY builders.work_exp DESC) AS SalesRank,
DENSE_RANK() OVER (ORDER BY builders.work_exp DESC) AS DenseRank,
NTILE(4) OVER (ORDER BY builders.work_exp DESC) AS PriceGroup
FROM builders;
```

fio	contacts	work_exp	rating	RowNumber	SalesRank	DenseRank	PriceGroup
Кузнецов Кузьма Кузьмич	kuznetsov@example.com	11	0,8	20	1	1	1
Алексеев Алексей Алексеевич	alekseev@example.com	9	4,6	19	2	2	1
Смирнов Сергей Сергеевич	smirnov@example.com	8	3,4	18	3	3	1
Марков Марк Маркович	markov@example.com	7	5	17	4	4	1
Григорьев Григорий Григорьевич	grigorev@example.com	6	2,7	16	5	5	1
Тихонов Тимофей Тимофеевич	tikhonov@example.com	5	1	5	6	6	2
Иванов Иван Иванович	8-999-111-22-33	5	2,3	6	6	6	2
Петров Петр Петрович	8-999-222-33-44	5	0,6	7	6	6	2
Сидоров Сидор Сидорович	8-999-333-44-55	5	0,5	8	6	6	2
Смирнова Анна Сергеевна	8-999-444-55-66	5	3	9	6	6	2
Кузнецов Алексей Викторович	8-999-555-66-77	5	4,9	10	6	6	3
Васильев Сергей Николаевич	8-999-666-77-88	5	4,5	11	6	6	3
Федорова Ольга Владимировна	8-999-777-88-99	5	3,4	12	6	6	3

Рисунок 92 — Вызов ранжирующих функций

## 3) Функции смещения

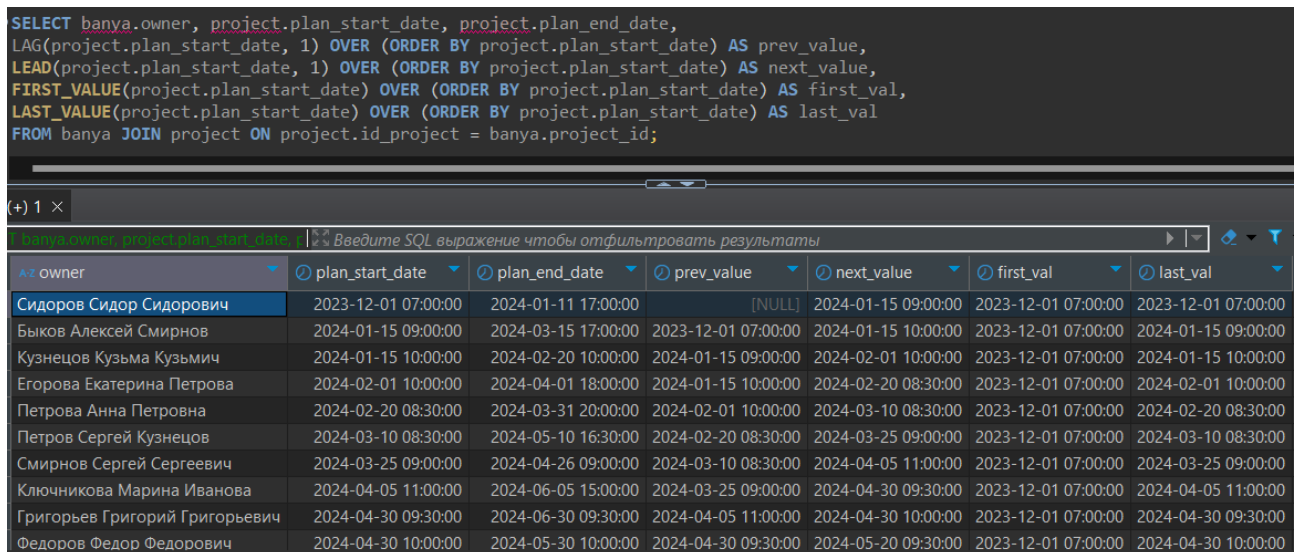
Литстинг 22 — Функции смещения



```

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;

```



The screenshot shows a SQL query execution in a database client. The query is the same as the one in the previous block. The results are displayed in a table with 7 columns: owner, plan\_start\_date, plan\_end\_date, prev\_value, next\_value, first\_val, and last\_val. The data is sorted by plan\_start\_date.

owner	plan_start_date	plan_end_date	prev_value	next_value	first_val	last_val
Сидоров Сидор Сидорович	2023-12-01 07:00:00	2024-01-11 17:00:00	[NULL]	2024-01-15 09:00:00	2023-12-01 07:00:00	2023-12-01 07:00:00
Быков Алексей Смирнов	2024-01-15 09:00:00	2024-03-15 17:00:00	2023-12-01 07:00:00	2024-01-15 10:00:00	2023-12-01 07:00:00	2024-01-15 09:00:00
Кузнецов Кузьма Кузьмич	2024-01-15 10:00:00	2024-02-20 10:00:00	2024-01-15 09:00:00	2024-02-01 10:00:00	2023-12-01 07:00:00	2024-01-15 10:00:00
Егорова Екатерина Петрова	2024-02-01 10:00:00	2024-04-01 18:00:00	2024-01-15 10:00:00	2024-02-20 08:30:00	2023-12-01 07:00:00	2024-02-01 10:00:00
Петрова Анна Петровна	2024-02-20 08:30:00	2024-03-31 20:00:00	2024-02-01 10:00:00	2024-03-10 08:30:00	2023-12-01 07:00:00	2024-02-20 08:30:00
Петров Сергей Кузнецов	2024-03-10 08:30:00	2024-05-10 16:30:00	2024-02-20 08:30:00	2024-03-25 09:00:00	2023-12-01 07:00:00	2024-03-10 08:30:00
Смирнов Сергей Сергеевич	2024-03-25 09:00:00	2024-04-26 09:00:00	2024-03-10 08:30:00	2024-04-05 11:00:00	2023-12-01 07:00:00	2024-03-25 09:00:00
Ключникова Марина Иванова	2024-04-05 11:00:00	2024-06-05 15:00:00	2024-03-25 09:00:00	2024-04-30 09:30:00	2023-12-01 07:00:00	2024-04-05 11:00:00
Григорьев Григорий Григорьевич	2024-04-30 09:30:00	2024-06-30 09:30:00	2024-04-05 11:00:00	2024-04-30 10:00:00	2023-12-01 07:00:00	2024-04-30 09:30:00
Федоров Федор Федорович	2024-04-30 10:00:00	2024-05-30 10:00:00	2024-04-30 09:30:00	2024-05-20 09:30:00	2023-12-01 07:00:00	2024-04-30 10:00:00

**Рисунок 93 — Вызов функций смещения**

## ЗАКЛЮЧЕНИЕ

В течение семестра была создана база данных для постройки бани.

На основе спроектированы физическая модель базы данных.

Была создана база данных для хранения информации о постройки бани, включая данные о проектах, материалах, строителях и контрактах.

Выполнены практические задания, изучены команды создания базы данных, создания таблиц, заполнения таблиц.

Изучен оператор SELECT. Написано множество операций проекции, селекции, соединения, объединения. Изучены встроенные в sql функции, такие как sum(), count(), max(), min(). Current\_date(), version() и так далее.

Написано множество процедур, функций и триггеров, а также сложных запросов.

Были получены навыки по использованию языка SQL для анализа и интерпретации данных.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Дейт, К. Дж. Введение в системы баз данных / учебное пособие / К. Дж. Дейт; — Москва: «Вильямс» 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.