**ЗАДАНИЕ ДЛЯ САМОСТОЯТЕЛЬНОЙ ПРАКТИЧЕСКОЙ РАБОТЫ ПО ДИСЦИПЛИНЕ «СИСТЕМЫ ОБРАБОТКИ И ХРАНЕНИЯ ДАННЫХ»**

Изучить и создать выборку и сортировку данных. Изучить и применить операторы для изменения данных в таблицах.

Результат работы в виде отчета должен содержать:

* снимки экрана (скриншоты) выборки данных по различным параметрам (по каждому оператору);
* снимок экрана (скриншоты) сортировки данных;
* снимки экрана (скриншоты) применения операторов изменения данных в таблицах Вашей базы данных;
* добавить все практические работы по SQL в итоговых отчет.

**ТЕОРЕТИЧЕСКИЕ ОСНОВЫ.**

# Выборка и сортировка данных

Продолжаем изучать базовые знания SQL на пример БД forum есть три таблицы: users (пользователи), topics (темы) и posts (сообщения).

Если необходимо посмотреть, какие данные в них содержатся, то для этого в SQL существует оператор SELECT. Синтаксис его использования следующий:

1. SELECT что\_выбрать FROM откуда\_выбрать;

Вместо «что\_выбрать» нужно указать либо имя столбца, значения которого хотим увидеть, либо имена нескольких столбцов через запятую, либо символ звездочки (\*), означающий выбор всех столбцов таблицы. Вместо

«откуда\_выбрать» следует указать имя таблицы.

Давайте посмотрим все столбцы из таблицы architect:

SELECT \* FROM architect;

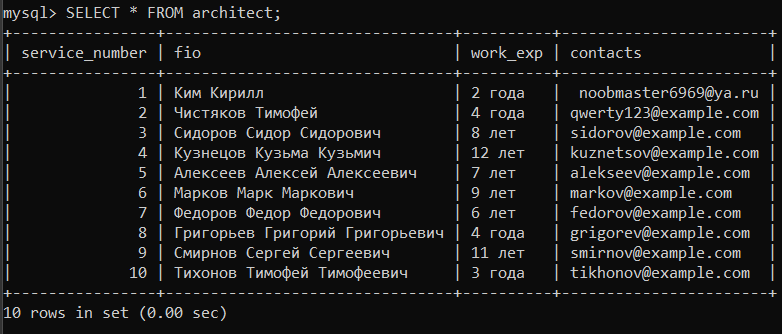


Рисунок 1 – Данные из таблицы architect

Вот и все данные, которые были внесены в эту таблицу. Но предположим, что нужно посмотреть только столбец fio. Для этого в запросе укажем имя этого столбца:

1. SELECT fio FROM architect;

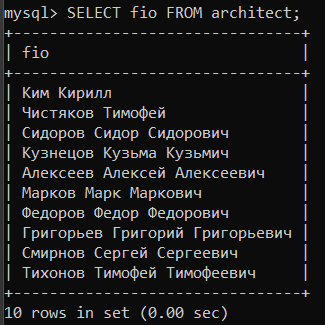


Рисунок 2 – Столбец fio из таблицы architect

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, синтаксис у такого запроса следующий:

1. SELECT имя\_столбца FROM имя\_таблицы WHERE условие;

Для нашего примера условием является work\_exp>4:

1. SELECT \* FROM architect WHERE work\_exp>4;
2. SELECT \* FROM architect WHERE work\_exp<4;

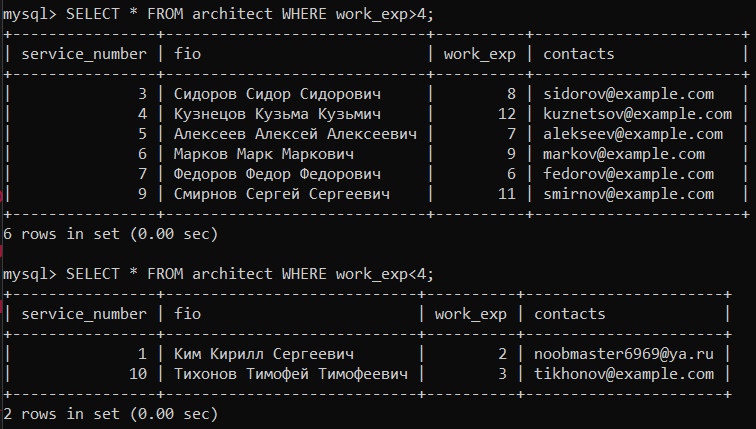


Рисунок 3 – Условие для таблицы architect

Для нашего примера условием является work\_exp>=4:

1. SELECT \* FROM architect WHERE work\_exp>=4;
2. SELECT \* FROM architect WHERE work\_exp<=4;

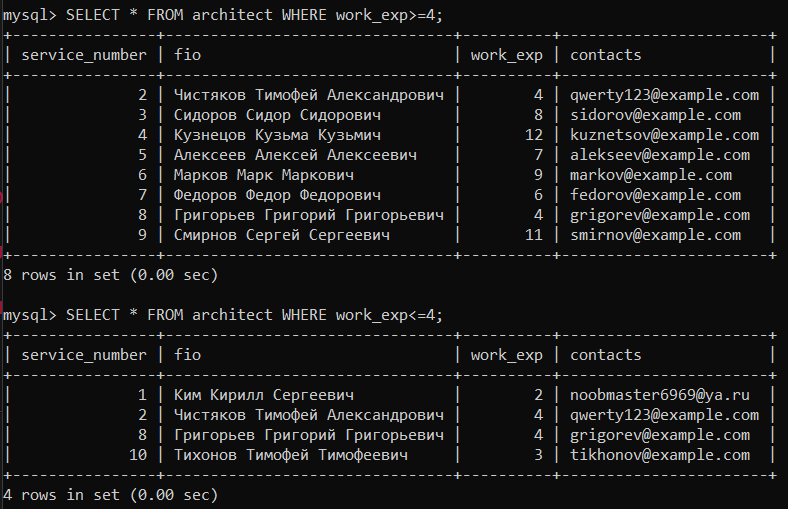


Рисунок 4 – Условие для таблицы architect

Для нашего примера условием является work\_exp!=4:

1. SELECT \* FROM architect WHERE work\_exp!=4;

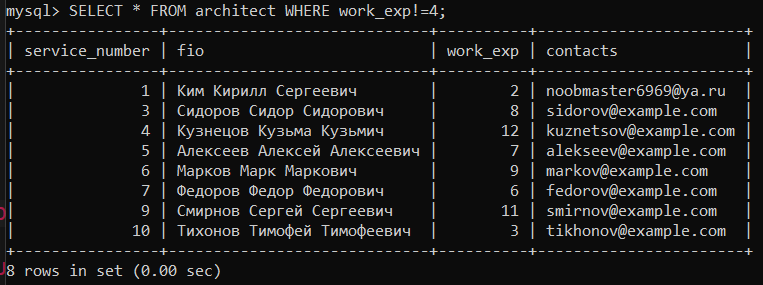


Рисунок 5 – Условие для таблицы architect

Выведем из таблицы architect строчки, где service\_number не пустой.

SELECT \* FROM architect WHERE service\_number IS NOT NULL;

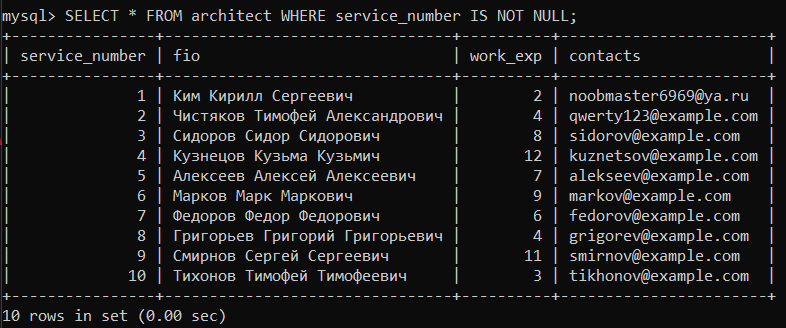


Рисунок 6 – Отбираются строки, имеющие значения в указанном поле

Выведем из таблицы architect строчки, где service\_number не пустой.

SELECT \* FROM architect WHERE service\_number IS NULL;



Рисунок 7 – Отбираются строки, не имеющие значения в указанном поле

Выведем из таблицы builders строчки, где work\_exp (опыт работы) от 4 до 8 лет.

SELECT \* FROM builders WHERE work\_exp BETWEEN 4 AND 8;

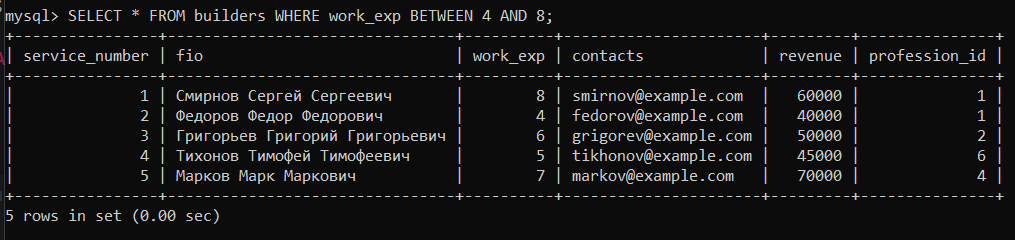


Рисунок 8 — Отбираются значения, находящиеся между указанными

Выведем из таблицы builders строчки, где service\_number равен 1 и 4.

SELECT \* FROM builders WHERE service\_number IN (1, 4);

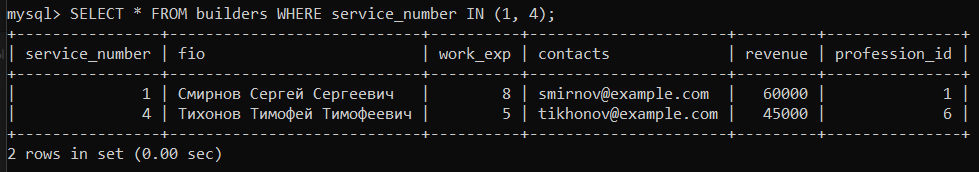


Рисунок 9 — Отбираются значения, соответствующие указанным

Выведем из таблицы builders строчки, где service\_number не равен 1 и 4.

SELECT \* FROM builders WHERE service\_number NOT IN (1, 4);

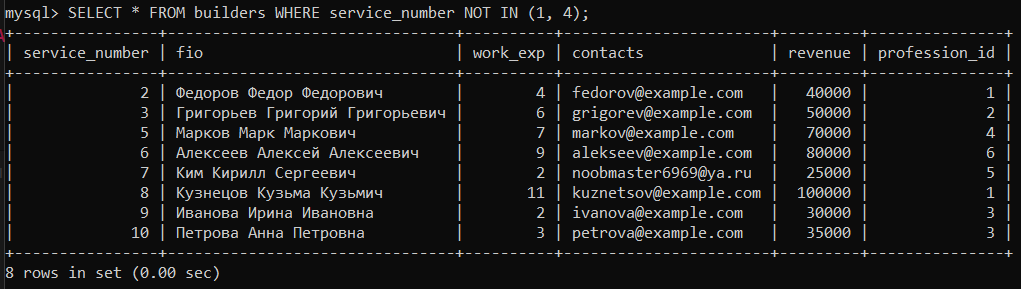


Рисунок 10 — Отбираются значения, кроме указанных

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

SELECT \* FROM shops WHERE address LIKE 'МКАД%';

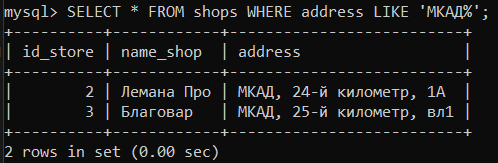


Рисунок 43 — Отбираются значения, соответствующие образцу

Выведем из таблицы shops строчки, где address не начинается с «МКАД».

SELECT \* FROM shops WHERE address NOT LIKE 'МКАД%';

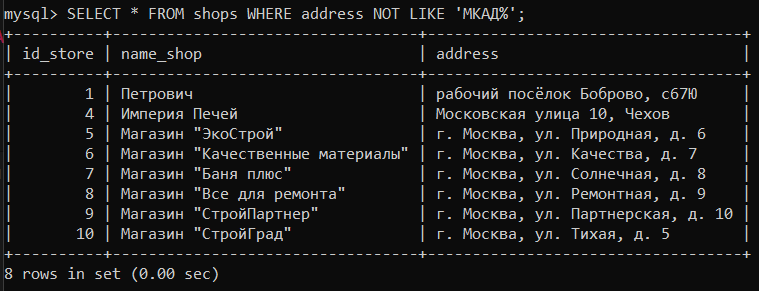


Рисунок 44 — Отбираются значения, не соответствующие образцу

Выведем из таблицы shops строчки, где address со второго символа начинается с «Москва».

SELECT \* FROM shops WHERE address LIKE '%Москва%';

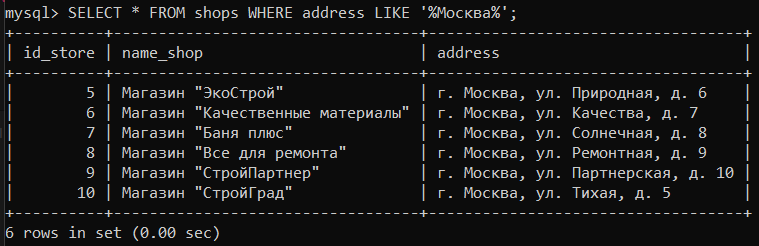
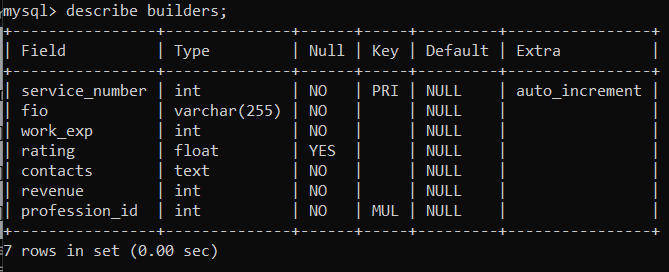


Рисунок 45 — Метасимвол \_

**Изменение данных в таблице.**

Забыли добавить в таблице builders – rating (рейтинг).



Для добавления столбцов в таблицу используется оператор ALTER TABLE — ADD COLUMN. Добавим столбец rating (рейтинг), будет после столбца work\_exp в таблице builders.

ALTER TABLE builders ADD COLUMN rating *FLOAT*(10) AFTER work\_exp;

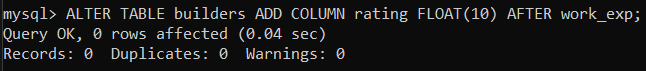
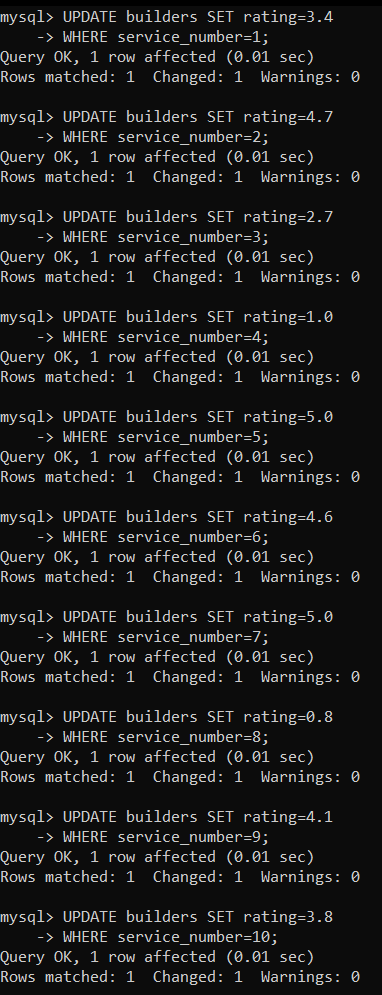


Рисунок 46 — Добавление нового столбца в таблицу

Для обновления уже существующих данных служит оператор UPDATE.

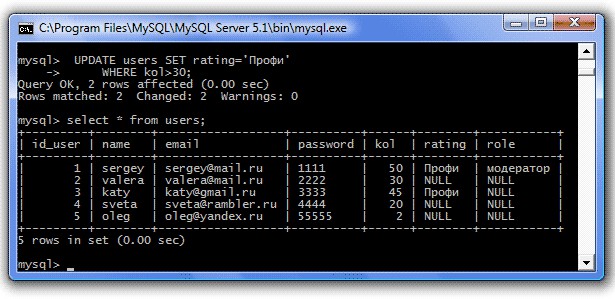
Давайте добавим строителям рейтинг.



**Рисунок 23 – Добавление**



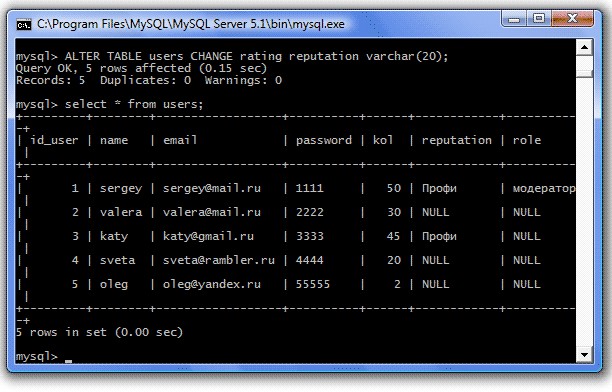
**Рисунок 13 – Замена информации**



Данные изменились в двух строках, согласно заданному условию. Понятно, что если в запросе опустить условие, то данные будут обновлены во всех строках таблицы.

Предположим, что не нравится название Рейтинг у нашего столбца, надо переименовать столбец в Репутация — reputation. Для изменения имени существующего столбца используется оператор CHANGE.

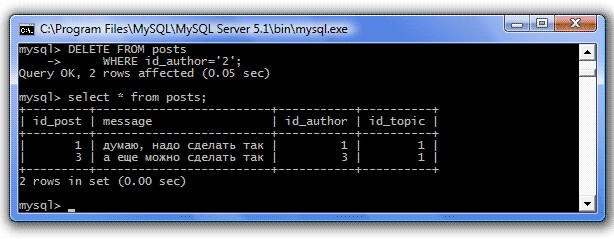
Давайте поменяем rating на reputation:



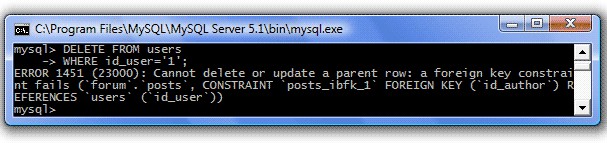
***Обратите внимание, что тип столбца надо указывать даже, если он не меняется.*** Кстати, если нам понадобится изменить только тип столбца, то будем использовать оператор MODIFY.

Рассмотрим — оператор DELETE, который позволяет удалять строки из таблицы.

Давайте из таблицы сообщений удалим те записи, которые оставлял пользователь valera (id=2):



***Если опустить условие, то из таблицы будут удалены все данные.***

Следует помнить, что данные СУБД даст удалить только в том случае, если они не являются внешними ключами для данных из других таблиц (поддержка целостности БД). Например, если удалить из таблицы users пользователя, который оставлял сообщения, то нам это не удастся.

Сначала надо удалить его сообщения, а уж потом и его самого.