# Задание на практическое занятие №5

(часть 1)

# Операции с БД

Вопросы занятия:

1. Выборка данных. Команда SELECT
2. Фильтрация данных. Оператор WHERE
3. Обновление данных. Команда UPDATE
4. Удаление данных. Команда DELETE

Время – 4 часа

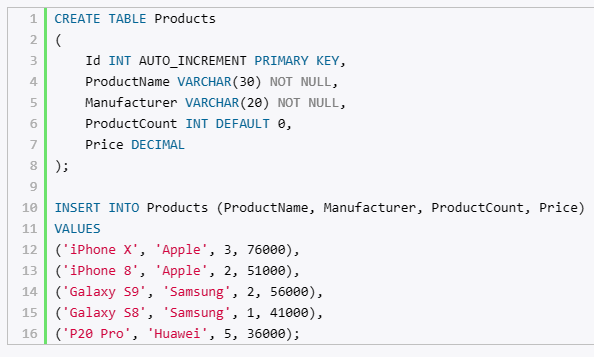
В задании приведены примеры выполнения операций с БД, которые вам необходимо отработать **с БД, которая создана вами в соответствии с индивидуальным заданием.**

# 1 Выборка данных. Команда SELECT

Для выборки данных из БД в MySQL применяется команда SELECT. В упрощенном виде она имеет следующий синтаксис:



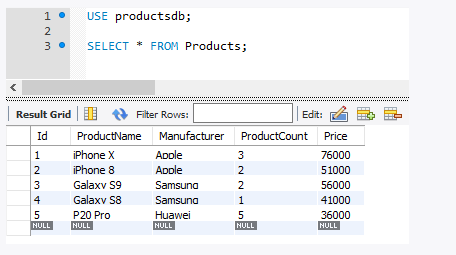
Например, пусть ранее была создана таблица Products, и в нее добавлены некоторые начальные данные:



Получим все объекты из этой таблицы:

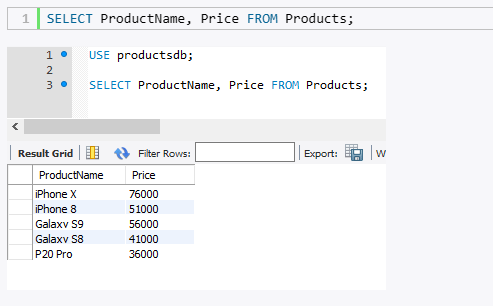


Символ звездочка \* указывает, что нам надо получить все столбцы.



Стоит отметить, что применение звездочки \* для получения данных считается не очень хорошей практикой, так как обычно необходимо получить данные по небольшому набору столбцов. Поэтому более оптимальный подход заключается в указании всех необходимых столбцов после слова SELECT. Исключение составляет тот случай, когда надо получить данные по абсолютно всем столбцам таблицы. Также использование символа \* может быть предпочтительно тогда, когда названия столбцов не известны.

Если необходимо получить данные не из всех, а из каких-то конкретных столбцов, тогда спецификации этих столбцов перечисляются через запятую после SELECT:



Спецификация столбца необязательно должна представлять его название. Это может быть любое выражение, например, результат арифметической операции. Так, выполним следующий запрос:

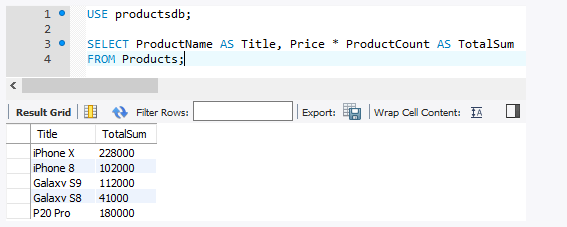


Здесь при выборке будут создаваться два столбца. Причем второй столбец представляет значение столбца Price, умноженное на значение столбца ProductCount, то есть совокупную стоимость товара.

С помощью оператора AS можно изменить название выходного столбца или определить его псевдоним:



Здесь для первого столбца определяется псевдоним Title, хотя в реальности он будет представлять столбец ProductName. Второй столбец TotalSum хранит произведение столбцов ProductCount и Price.



# 2 Фильтрация данных. Оператор WHERE

Зачастую необходимо извлекать не все данные из БД, а только те, которые соответствуют определенному условию. Для фильтрации данных в команде SELECT применяется оператор WHERE, после которого указывается условие:



Если условие истинно, то строка попадает в результирующую выборку. В качестве можно использовать операции сравнения, которые сравнивают два выражения:

=: сравнение на равенство

!=: сравнение на неравенство

<>: сравнение на неравенство

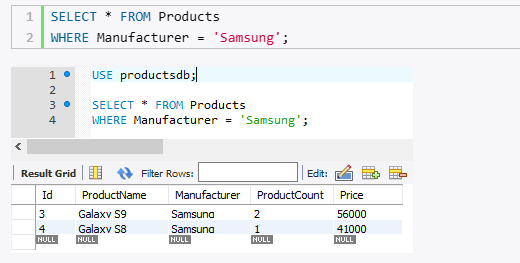
<: меньше чем

>: больше чем

<=: меньше чем или равно

>=: больше чем или равно

К примеру, выберем всех товары, производителем которых является компания Samsung:

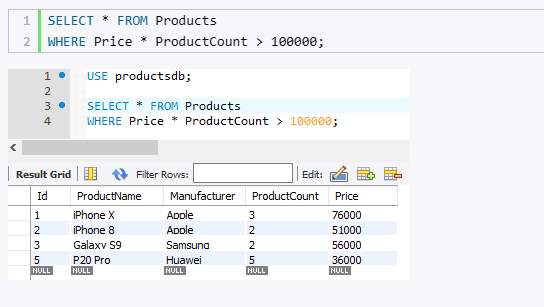


Стоит отметить, что для MySQL не важен регистр символов, и, к примеру, строка "Samsung" будет эквивалентна строке "SAMSUNG" или "sumSunG".

Другой пример - найдем все товары, количество которых меньше 3:



Критерий фильтрации может представлять и более сложное составное выражение. Например, найдем все товары, у которых совокупная стоимость больше 100 000:



Логические операторы

Логические операторы позволяют объединить несколько условий. В MySQL можно использовать следующие логические операторы:

- AND: операция логического И. Она объединяет два выражения:



Только если оба этих выражения одновременно истинны, то и общее условие оператора AND также будет истинно. То есть если и первое условие истинно, и второе.

- OR: операция логического ИЛИ. Она также объединяет два выражения:

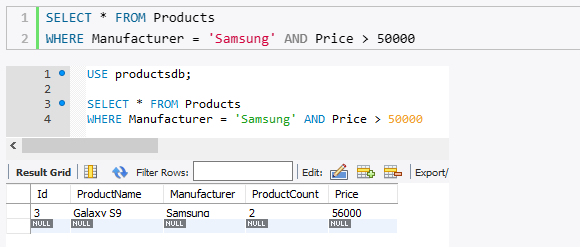


Если хотя бы одно из этих выражений истинно, то общее условие оператора OR также будет истинно. То есть если или первое условие истинно, или второе.

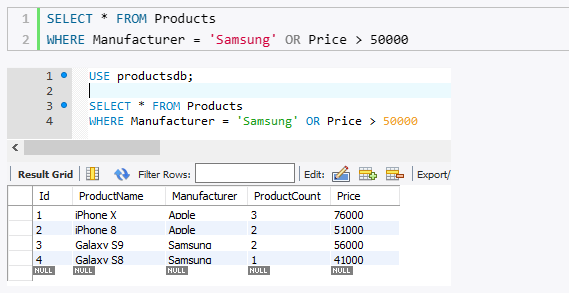
- NOT: операция логического отрицания. Если выражение в этой операции ложно, то общее условие истинно.



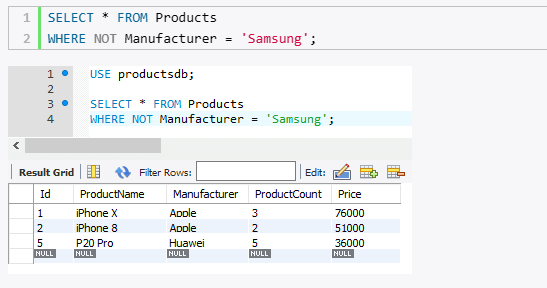
Например, выберем все товары, у которых производитель Samsung и одновременно цена больше 50000:



Теперь изменим оператор на OR. То есть выберем все товары, у которых либо производитель Samsung, либо цена больше 50000:



Применение оператора NOT - выберем все товары, у которых производитель не Samsung:



# Приоритет операций

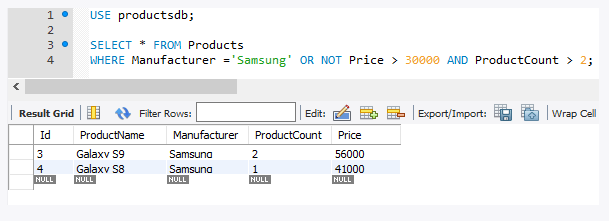
В одном условии при необходимости мы можем объединять несколько логических операций. Однако следует учитывать, что самой приоритетной операцией, которая выполняется в первую очередь, является NOT, менее приоритетная - AND и операция с наименьшим приоритетом - OR. Например:



В данном случае сначала вычисляется выражение NOT Price > 30000, то есть цена должна быть меньше или равна 30000.

Затем вычисляется выражение NOT Price > 30000 AND ProductCount > 2, то есть цена должна быть меньше или равна 30000 и одновременно количество товаров должно быть больше 2.

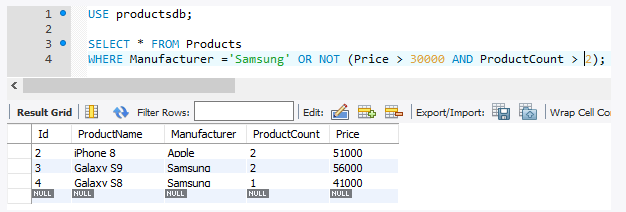
В конце вычисляется оператор OR - либо цена должна быть меньше или равна 30000 и одновременно количество товаров должно быть больше 2, либо производителем должен быть Samsung.



С помощью скобок можно переопределить приоритет операций:

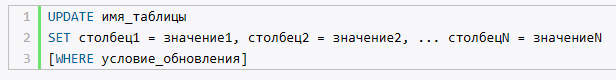


В данном случае находим товары, у которых либо производитель Samsung, либо одновременно цена товара меньше или равна 30000 и количество товаров меньше 3.



# 3 Обновление данных. Команда UPDATE

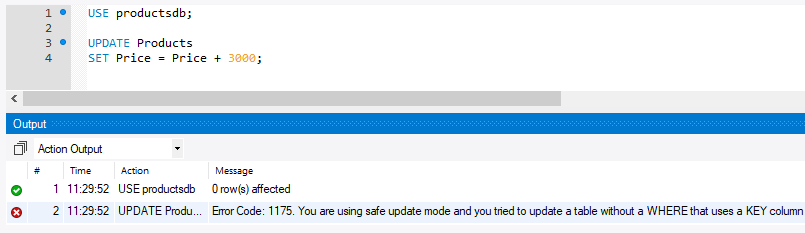
Команда UPDATE применяется для обновления уже имеющихся строк. Она имеет следующий формальный синтаксис:



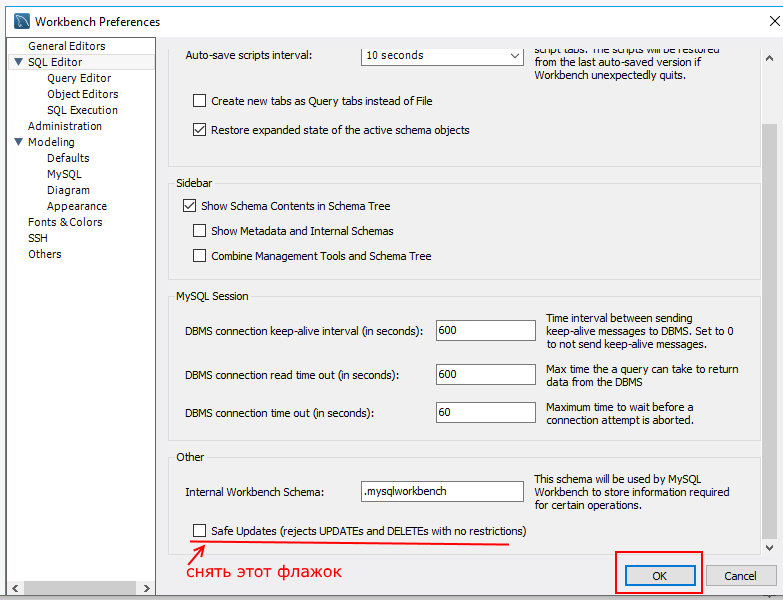
Например, увеличим у всех товаров цену на 3000:



Однако при выполнении данного запроса в MySQL Workbench мы можем столкнуться с ошибкой:



Ошибка говорит о том, что мы находимся в безопасном режиме. И чтобы его отключить, в MySQL Workbench надо перейти в меню Edit -> Preferences и в открывшемся окне перейти к пункту SQL Editor:

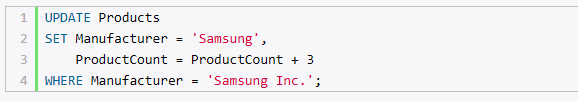


В открывшейся вкладке в самом низу надо снять флажок с поля "Safe Updates (reject UPDATEs and DELETEs with no restrictions)" и затем сохранить изменения, нажав на кнопку OK. После этого надо переподключиться к серверу.

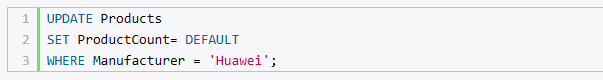
Используем выражение WHERE и изменим название производителя с "Samsung" на "Samsung Inc.":



Также можно обновлять сразу несколько столбцов:



При обновлении вместо конкретных значений и выражений мы можем использовать ключевые слова DEFAULT и NULL для установки соответственно значения по умолчанию или NULL:

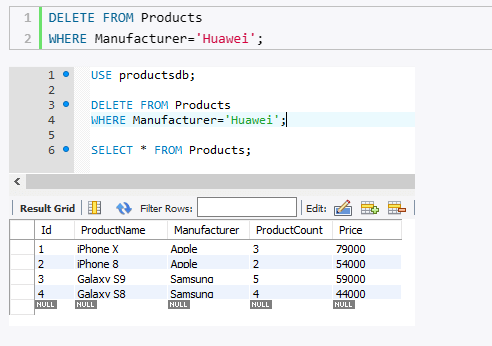


# 4 Удаление данных. Команда DELETE

Команда DELETE удаляет данные из БД. Она имеет следующий формальный синтаксис:



Например, удалим строки, у которых производитель - Huawei:



Или удалим все товары, производителем которых является Apple и которые имеют цену меньше 60000:



Если необходимо вовсе удалить все строки вне зависимости от условия, то условие можно не указывать:



# Создание базы данных

В программе MySQL Workbench существует 3 способа создания баз данных (БД).

1. Поэтапное создание БД и её компонентов с использованием SQL-команд с «мгновенным» применением изменений к БД.
2. Создание БД на основе модели MySQL Workbench.
3. Создание БД в результате выполнения скрипта, содержащего SQL-команды.

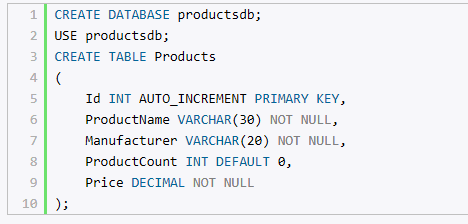
# Добавление данных. Команда INSERT

Для добавления данных в БД в MySQL используется команда INSERT, которая имеет следующий формальный синтаксис:



После выражения INSERT INTO в скобках можно указать список столбцов через запятую, в которые надо добавлять данные, и в конце после слова VALUES скобках перечисляют добавляемые для столбцов значения.

Например, пусть в базе данных productsdb есть следующая таблица Products:



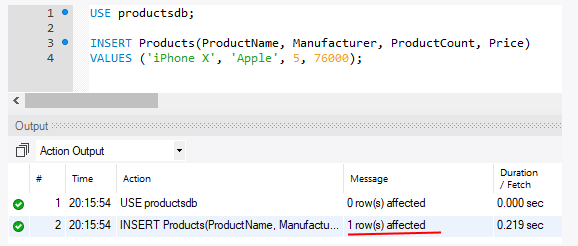
Добавим в эту таблицу одну строку с помощью следующего кода:



В данно случае значения будут передаваться столбцам по позиции. То есть стобцу ProductName передается строка "iPhone X", столбцу Manufacturer - строка "Apple" и так далее.

Важно, чтобы между значениями и типами данных столбцов было соответствие. Так, столбец ProductName представляет тип varchar, то есть строку. Соответственно этому столбцу мы можем передать строковое значение в одинарных кавычках. А стобец ProductCount представляет тип int, то есть целое число, поэтому данному столбцу нужно передать целые числа, но никак не строки.

После удачного выполнения в MySQL Workbench в поле вывода должны появиться зеленый маркер и сообщение "1 row(s) affected":



Необязательно при добавлении данных указывать значения абсолютно для всех столбцов таблицы. Например, в примере выше не указано значение для стобца Id. Но поскольку для данного столбца определен атрибут AUTO\_INCREMENT, то его значение будет автоматически генерироваться.

Также мы можем опускать при добавлении такие столбцы, которые поддерживают значение NULL или для которых указано значение по умолчанию, то есть для них определены атрибуты NULL или DEFAULT. Так, в таблице Products столбец ProductCount имеет значение по умолчанию - число 0. Поэтому мы можем при добавлении опустить этот столбец, и ему будет передаваться число 0:



С помощью ключевых слов DEFAULT и NULL можно указать, что в качестве значения будет использовать значение по умолчанию или NULL соответственно:

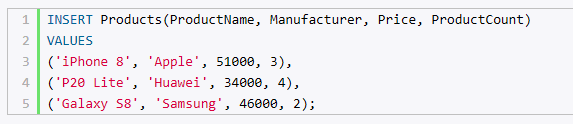


или



#### Множественное добавление

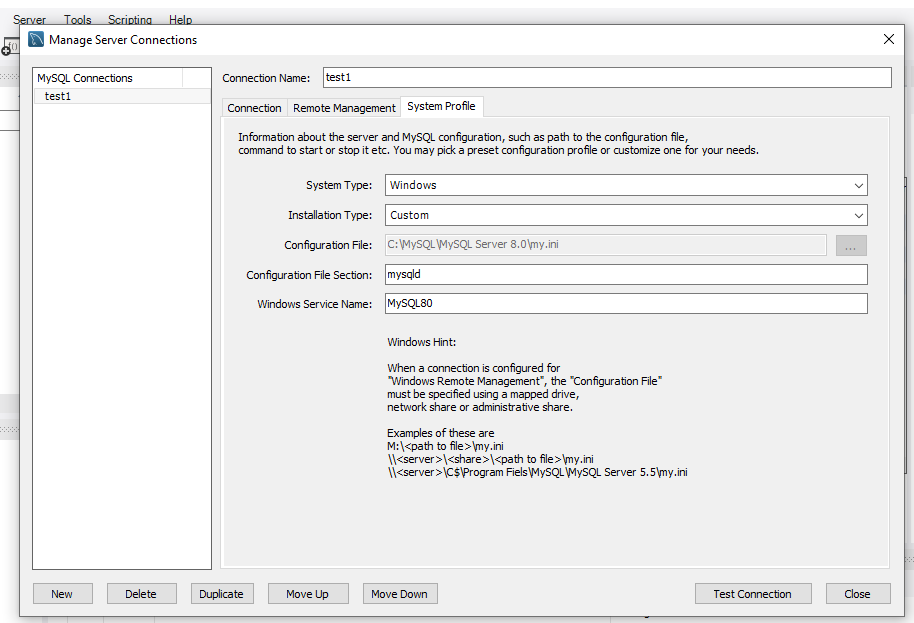
Также мы можем добавить сразу несколько строк:



В данном случае в таблицу будут добавлены три строки.

# Создание БД из ER-модели

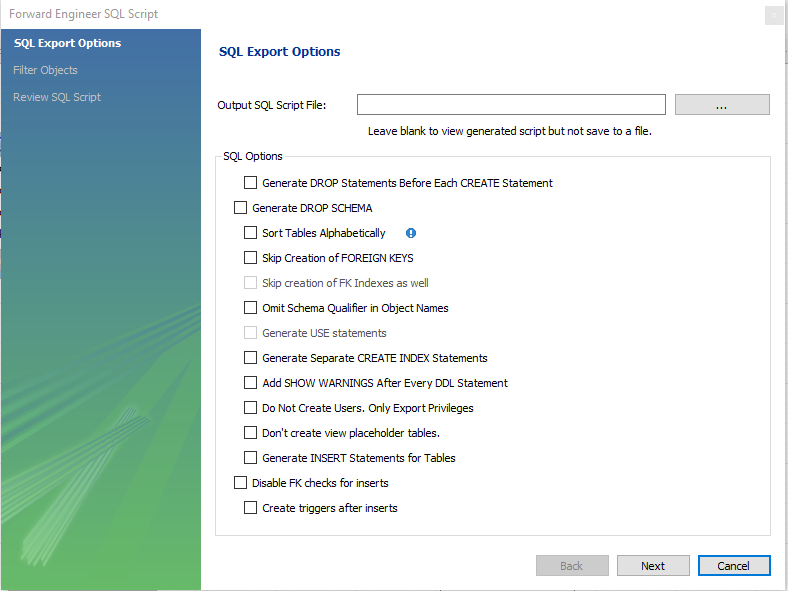
Создадим БД из созданной ER-модели на сервер mySQL. Для этого создадим новое подключение к базе данных (при необходимости), воспользовавшись разделом меню Server:



Откроем **EER Diagram** (в меню раздел File)**.**

**Далее:**

“File → Export→ Forward Engineer MySQL Create Script…”:



Отмечаем необходимые галочки c с настройками импорта. Сохраняем скрипт в  файл – пропишите директорию в поле сверху.

В следующем окне можно настроить – какие объекты мы будем экспортировать.

В результате мы получаем сгененрированный SQL-скрипт, позволяющих создать физическую базу данных, построенную в соответствии с вашей ER-моделью.

# Выполнение скрипта – создания базы данных и таблиц

“File → Open SQL Script…” → 

Устраните все ошибки (при их наличии).

Вы получили скрипт, позволяющий формировать вашу БД.

# Заполнение БД

# Отредактируйте созданный на предыдущем этапе скрипт, добавив в него комады Insert, позволяющие добавить в БД записи.

Количество записей в таблицах БД должны быть достаточным для выполнения сложных запросов.

Предусмотрите в скрипте команды удаления таблиц БД (DROP) для повторного выполнения скрипта.

Примеры готовых скриптов, позволяющих создавать и наполнять БД приведены в прилагаемых файлах.

Также примеры скриптов вы можете найти здесь: <https://dev.mysql.com/doc/index-other.html>

# Требования к оформлению отчета

1. Отчет должен содержать:
2. Титульный лист, содержащий: название работы, Тему, цель и задание к работе
3. Снимки экрана (скриншоты) процесса выполнения операций.
4. Выводы (что узнали, где можно применить полученные знания)