

# SQL



## Выборка данных

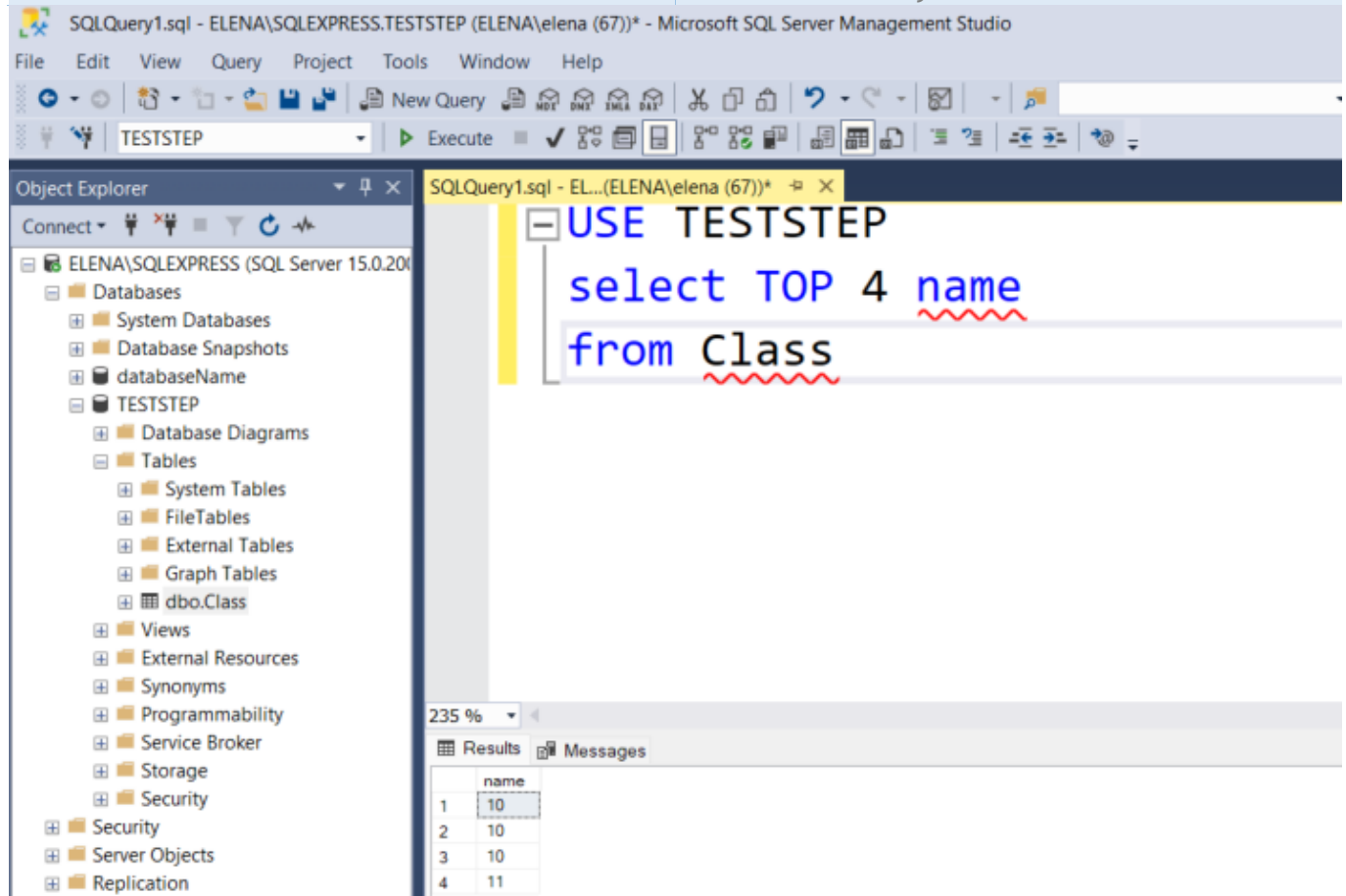
Создать БД	<code>CREATE DATABASE teststep;</code>
Изменить наименование БД	<code>ALTER DATABASE teststep MODIFY NAME = teststep2;</code>
Удалить БД	<code>DROP DATABASE teststep;</code>
Создать таблицу	<code>USE teststep CREATE TABLE students (-- PRIMARY KEY = UNIQUE + NOT NULL id INT PRIMARY KEY, fio NVARCHAR(50) NOT NULL, groupNumber NVARCHAR(10));</code>
Изменить наименование таблицы	<code>USE teststep EXEC sp_rename 'students', 'Students';</code>
Удалить таблицу	<code>USE teststep DROP TABLE students;</code>
Добавить поле в таблицу	<code>USE teststep ALTER TABLE students ADD letter NVARCHAR(5);</code>
Изменить тип поля в таблице	<code>USE teststep ALTER TABLE students ALTER COLUMN letter NVARCHAR(10);</code>
Удалить поле из таблицы	<code>USE teststep ALTER TABLE students DROP COLUMN letter;</code>
Вставить строку с информацией в таблицу	<code>USE teststep INSERT INTO students (id, fio, groupNumber) VALUES (1, N'Иванов Иван Иванович', N'EKO-11021'), (2, N'Петров Петр Петрович', N'EKO-11022'), (3, N'Сидоров Сидор Сидорович', N'EKO-11021');</code>
Выборка данных из таблицы	<code>USE teststep SELECT * FROM students;</code>
Выборка данных из таблицы с псевдонимами	<code>USE teststep SELECT id AS ИД, fio AS ФИО FROM students;</code>
Обновить (редактировать) данные поля в таблице по условию	<code>USE teststep UPDATE students SET groupNumber = N'EKO-11023' WHERE id = 3;</code>
Удалить строку в таблице по условию	<code>USE teststep DELETE FROM students WHERE id = 3;</code>
Очистить данные таблицы	<code>USE teststep TRUNCATE TABLE students;</code>

# SQL

В языке **T-SQL** совместно с предложением **SELECT** можно использовать оператор **TOP**, который позволяет получить первые записи.

Выборка ограниченного количества записей (строк) из таблицы

```
USE teststep
SELECT TOP 2 *
FROM students;
```



Если в таблице вашей базы данных определенное поле содержит большое количество повторяющихся значений, а ваша задача избавиться от них в результирующей таблице, то вы можете использовать для этого оператор **DISTINCT**, который следует написать перед именем требуемого столбца, и тем самым вы исключите из результата все повторения. Применение команды **SELECT DISTINCT** позволяет получить лишь те данные, которые не повторяются в таблице.

Выборка уникальных данных из таблицы

```
USE teststep
SELECT DISTINCT *
FROM students;
```

При помощи оператора **SELECT** вы прописываете что вам нужно, в операторе **FROM** вы указываете откуда необходимо взять требуемую информацию, а в операторе **WHERE** находится условие, которое конкретизирует конечный результат. В тех случаях, когда необходимо отфильтровать получаемые данные, задав определенные условия, в **SQL**-запросе предусмотрено использование оператора **WHERE**, после которого прописываются требуемые условия. Для того чтобы в **SQL**-запросе указать необходимые ограничения, в языке **T-SQL** используются следующие операторы сравнения: **=** — равно; **<>** — не равно; **>** — больше; **>=** — больше или равно; **!>** — не больше чем; **<** — меньше; **<=** — меньше или равно; **!<** — не меньше чем.

# SQL

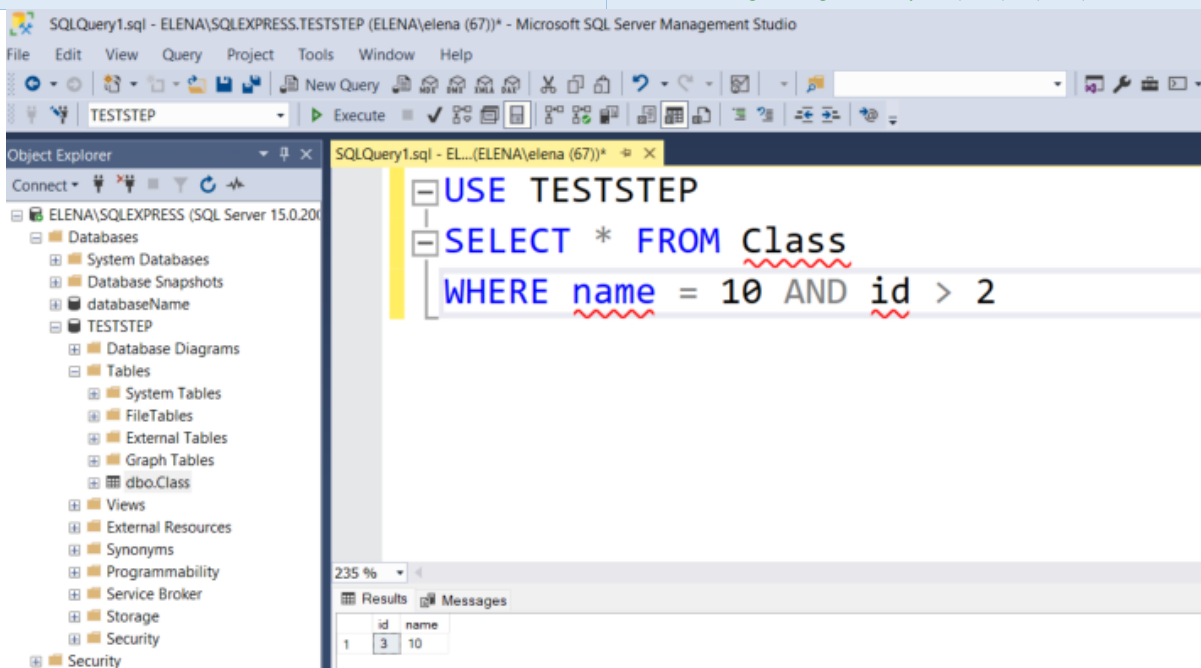
Выборка данных из таблицы по условию

```
USE teststep
SELECT *
FROM students
WHERE id = 1;
```

Для того чтобы в операторе **WHERE** задать несколько условий, необходимо использовать операторы объединения **AND** и **OR**. Оператор **AND** объединяет два условия и возвращает истину, если они оба верны. Оператор **OR** объединяет два условия и возвращает истину, если хотя бы одно из условий верно. Для того чтобы получить требуемый результат необходимо осуществить проверку соответствующих значений на кратность числу два, используя операцию деление по модулю (%). Существует еще один оператор — **NOT**, который используется, когда необходимо задать противоположное условие. В тех случаях, когда вам требуется проверить значения столбцов на неопределенность (значение **NULL**), использование операторов сравнения (**=**, **<>**, **>**, **<**, и т.д.) невозможно. Вместо них вам необходимо использовать оператор **IS NULL**.

Выборка данных из таблицы по составному условию

```
USE teststep
SELECT *
FROM students
WHERE id > 1 AND id < 3;
-- AND OR NOT = != > < <= >=
```

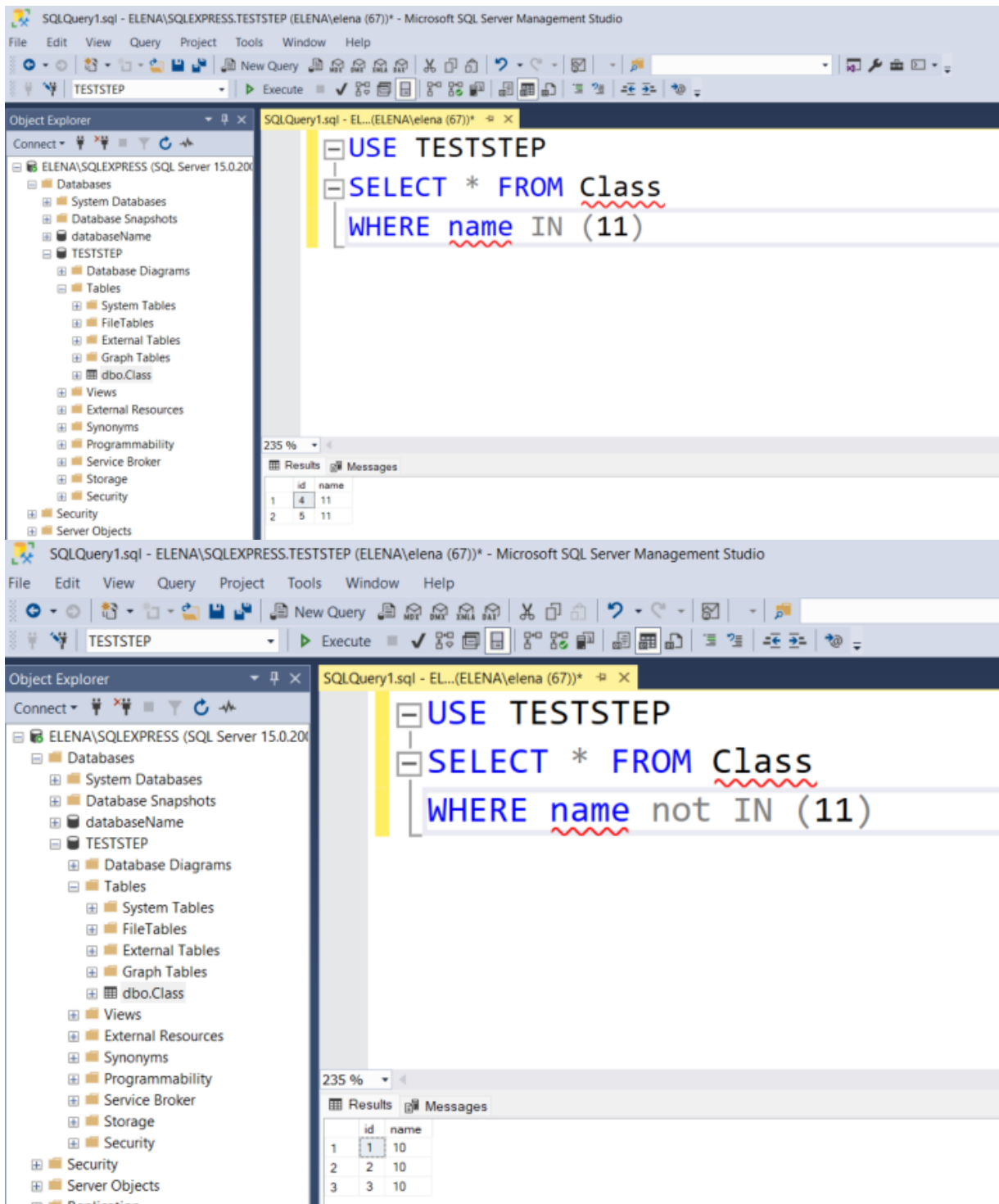


Для того чтобы улучшить читабельность подобных запросов можно использовать ключевое слово **IN**, которое обеспечивает сравнение значения указанного столбца с любым значением из указанного множества.

Выборка данных из таблицы по условию вхождения в множество

```
USE teststep
SELECT *
FROM students
WHERE id IN (1);
-- IN NOT IN
```

# SQL

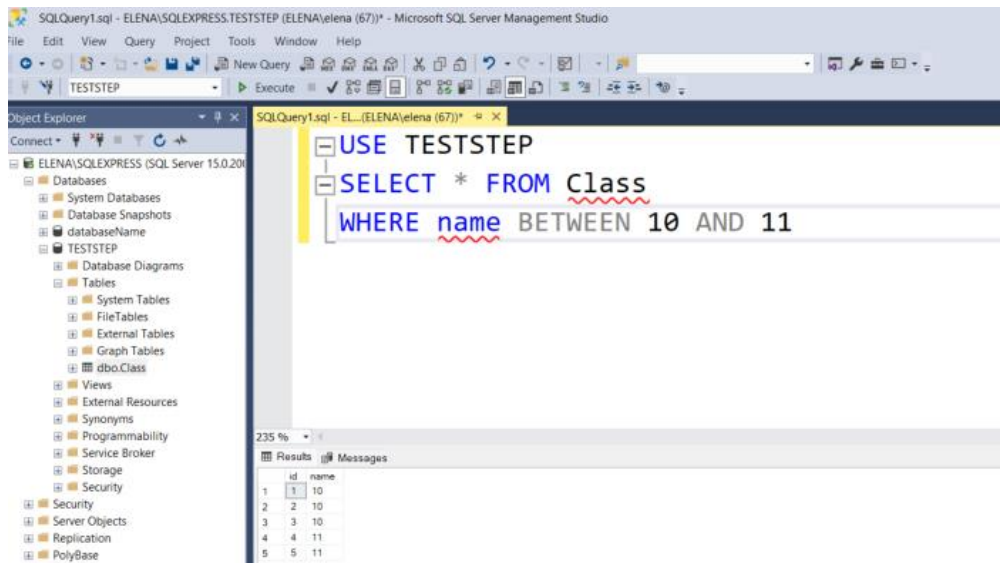


При написании SQL-запросов, в которых требуется проверка на попадание значения определенного столбца в некий диапазон можно использовать оператор **AND**, а можно, в качестве альтернативы, применить ключевое слово **BETWEEN**. Ключевое слово **BETWEEN** прописывается после имени столбца, а после него необходимо указать нижнюю и верхнюю границу диапазона, разделенных оператором **AND**, границы диапазона также учитываются. Также с ключевым словом **BETWEEN** можно использовать оператор **NOT**.

Выборка данных из таблицы по условию вхождения в диапазон

```
USE teststep
SELECT *
FROM students
WHERE id BETWEEN 1 AND 3;
```

# SQL



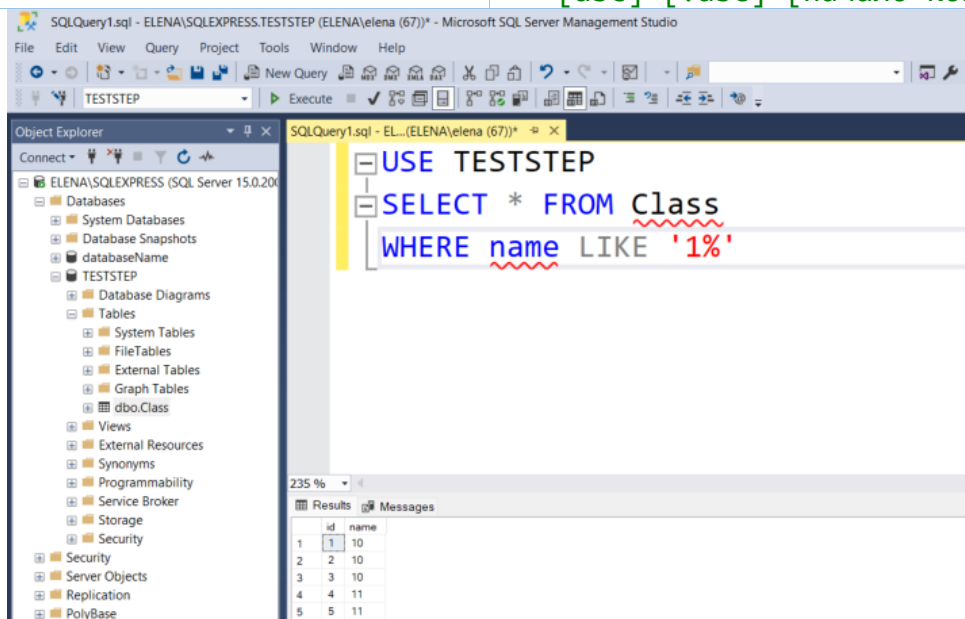
В тех случаях, когда необходимо осуществить поиск по текстовым полям таблиц, существует возможность использовать шаблон, который можно задать, применив ключевое слово **LIKE**. Для того чтобы сформировать блон необходимо указывать определенные служебные символы или комбинации этих символов: **%** — соответствует любой последовательности символов от 0 и более; **\_** — представляет любой одиночный символ; **[]** — задает последовательность или диапазон возможных символов; **[^]** — задает последовательность или диапазон символов, которые должны отсутствовать.

Выборка данных из таблицы по условию вхождения в фильтр

```
USE teststep
SELECT *
FROM students
WHERE fio LIKE N'%Иван%';
-- % _
```

Выборка данных из таблицы по условию вхождения в фильтр регулярного выражения

```
USE teststep
SELECT *
FROM students
WHERE id LIKE '[1-2]';
-- [abc] [!abc] [начало-конец]
```



# SQL

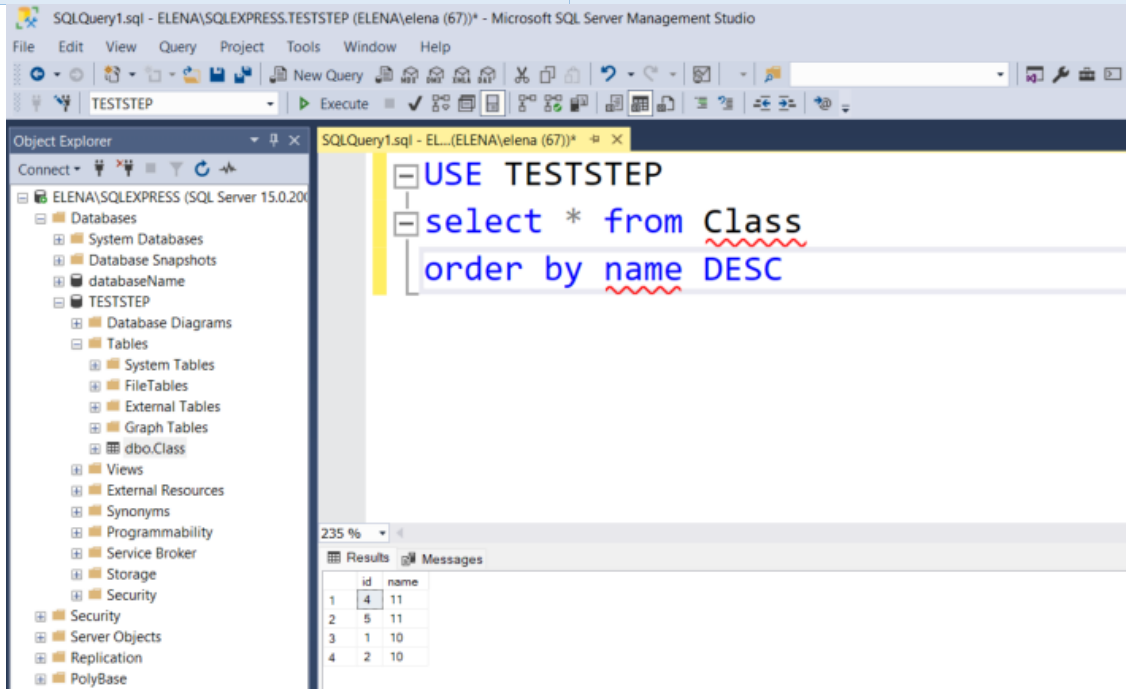
Внутри **WHERE** часто применяются:

- математические знаки сравнения (**=**, **<**, **>**, **<=**, **>=**, **<>**);
- для фильтрования таблицы одновременно по нескольким значениям столбцов – **AND** (выполнение всех условий) / **OR** (выполнение хотя бы одного условия);
- для фильтрования в таблице столбцов сразу по нескольким значениям – **IN** (включение) / **NOT IN** (исключение);
- команды проверки: **BETWEEN** – на расположение значения в определенном промежутке (в числовом или текстовом выражении)
- **LIKE** – по заданному шаблону (в операторе используются два оператора: «%» – ни одного, один либо несколько символов, «\_» – один символ). Позволяет сравнивать не только числовые показатели, но и текстовые.

При написании некоторых **SQL**-запросов существует необходимость упорядочить полученную информацию, тогда при написании запросов следует использовать предложение **ORDER BY**. При помощи оператора **ORDER BY** можно указать количество столбцов, по которым производится сортировка и направление этой сортировки. Чтобы полученные данные были отсортированы по убыванию необходимо указать ключевое слово **DESC** (от descending) после соответствующего поля. Сортировка по возрастанию является сортировкой по умолчанию и ее можно не указывать, но если вы хотите сделать это явным образом, то следует воспользоваться ключевым словом **ASC** (от ascending).

Выборка данных из таблицы с сортировкой

```
USE teststep
SELECT *
FROM students
ORDER BY id DESC;
-- ASC DESC
```







## Выборка данных

Выборка данных из таблицы	<pre>USE teststep SELECT * FROM students;</pre>
Выборка данных из таблицы с псевдонимами	<pre>USE teststep SELECT id AS ИД, fio AS ФИО FROM students;</pre>
Выборка уникальных данных из таблицы	<pre>USE teststep SELECT DISTINCT * FROM students;</pre>
Выборка ограниченного количества записей (строк) из таблицы	<pre>USE teststep SELECT TOP 2 * FROM students;</pre>
Выборка данных из таблицы по условию	<pre>USE teststep SELECT * FROM students WHERE id = 1;</pre>
Выборка данных из таблицы по составному условию	<pre>USE teststep SELECT * FROM students WHERE id &gt; 1 AND id &lt; 3; -- AND OR NOT = != &gt; &lt; &lt;= &gt;=</pre>
Выборка данных из таблицы по условию вхождения в множество	<pre>USE teststep SELECT * FROM students WHERE id IN (1); -- IN NOT IN</pre>
Выборка данных из таблицы по условию вхождения в диапазон	<pre>USE teststep SELECT * FROM students WHERE id BETWEEN 1 AND 3;</pre>
Выборка данных из таблицы по условию вхождения в фильтр	<pre>USE teststep SELECT * FROM students WHERE fio LIKE N'%Иван%'; -- % _</pre>
Выборка данных из таблицы по условию вхождения в фильтр регулярного выражения	<pre>USE teststep SELECT * FROM students WHERE id LIKE '[1-2]'; -- [abc] [!abc] [начало-конец]</pre>
Выборка данных из таблицы с сортировкой	<pre>USE teststep SELECT * FROM students ORDER BY id DESC; -- ASC DESC</pre>



## Практическая работа

**\* Запросы необходимо сохранить для проверки**

1. Вывести содержимое таблицы палат.
2. Вывести фамилии и телефоны всех врачей.
3. Вывести все этажи без повторений, на которых располагаются палаты.
4. Вывести названия заболеваний под именем "Name of Disease" и степень их тяжести под именем "Severity of Disease".
5. Использовать выражение FROM для любых трех таблиц базы данных, используя для них псевдонимы.
6. Вывести названия отделений, расположенных в корпусе 5 и имеющих фонд финансирования менее 30000.
7. Вывести названия отделений, расположенных в 3-м корпусе с фондом финансирования в диапазоне от 12000 до 15000.
8. Вывести названия палат, расположенных в корпусах 4 и 5 на 1-м этаже.
9. Вывести названия, корпуса и фонды финансирования отделений, расположенных в корпусах 3 или 6 и имеющих фонд финансирования меньше 11000 или больше 25000.
10. Вывести фамилии врачей, чья зарплата (сумма ставки и надбавки) превышает 1500.
11. Вывести фамилии врачей, у которых половина зарплаты превышает трехкратную надбавку.
12. Вывести названия обследований без повторений, проводимых в первые три дня недели с 12:00 до 15:00.
13. Вывести названия и номера корпусов отделений, расположенных в корпусах 1 или 3.
14. Вывести названия заболеваний всех степеней тяжести, кроме 1-й и 2-й.
15. Вывести названия отделений, которые не располагаются в 2-м или 4-м корпусе.
16. Вывести названия отделений, которые располагаются в 1-м или 5-м корпусе.
17. Вывести фамилии врачей, начинающиеся на букву "N".
18. Вывести 5 первых отделений из таблицы Отделения.
19. Вывести таблицу с информацией о докторах с сортировкой по убыванию.
20. Вывести из каждой таблицы 3 первых уникальных значения с псевдонимами, условием (исходя из данных таблицы), сортировкой по убыванию.





## Домашняя работа

### \* Прислать запросы для проверки

1. Вывести таблицу кафедр, но расположить ее поля в обратном порядке.
2. Вывести названия групп и их рейтинги с уточнением имен полей именем таблицы.
3. Вывести для преподавателей их фамилию, процент ставки по отношению к надбавке и процент ставки по отношению к зарплате (сумма ставки и надбавки).
4. Вывести таблицу факультетов в виде одного поля в следующем формате: "The dean of faculty [faculty] is [dean].".
5. Вывести фамилии преподавателей, которые являются профессорами и ставка которых превышает 1050.
6. Вывести названия кафедр, фонд финансирования которых меньше 11000 или больше 25000.
7. Вывести названия факультетов кроме факультета "Computer Science".
8. Вывести фамилии и должности преподавателей, которые не являются профессорами.
9. Вывести фамилии, должности, ставки и надбавки ассистентов, у которых надбавка в диапазоне от 160 до 550.
10. Вывести фамилии и ставки ассистентов.
11. Вывести фамилии и должности преподавателей, которые были приняты на работу до 01.01.2000.
12. Вывести названия кафедр, которые в алфавитном порядке располагаются до кафедры "Software Development". Выводимое поле должно иметь название "Name of Department".
13. Вывести фамилии ассистентов, имеющих зарплату (сумма ставки и надбавки) не более 1200.
14. Вывести названия групп 5-го курса, имеющих рейтинг в диапазоне от 2 до 4.
15. Вывести фамилии ассистентов со ставкой меньше 550 или надбавкой меньше 200.