

## SQL

**Transact-SQL (T-SQL)** — вариант языка SQL, созданный Microsoft для Microsoft SQL Server.

**SQL** – регистронезависимый язык, но в T-SQL ключевые слова рекомендуется писать капслом.

Строковые значения и даты пишутся в одинарных кавычках: 'user' и '2022-01-29 08:35:00'

Числовые значения — как в программировании, разделитель — точка.

Строки можно складывать: 'фамилия' + ' ' + 'имя'

### Выполнение запросов (команд) в SQL Server Management Studio (SSMS):

- для выполнения всех команд скрипта нажать F5 или кнопку «Выполнить»
- для выполнения одной из команд выделить ее текст, нажать F5 или кнопку «Выполнить»

**Разделитель команд (;), его можно не писать, если в скрипте одна команда :**

SQL-запрос1;

SQL-запрос2;

<b>Однострочный комментарий:</b> -- текст комментария	<b>Многострочный комментарий:</b> /*текст комментария*/
--	--

## Синтаксис команды SELECT

**SELECT** – команда выборки (получения) данных

### Порядок разделов команды SELECT:

1. **SELECT** – перечисление столбцов/данных для вывода
2. **FROM** – источник данных (например, одна и более таблиц)
3. **WHERE** – условие отбора/фильтрации
4. **GROUP BY** – группировка данных
5. **HAVING** – условие отбора среди групп
6. **ORDER BY** – сортировка данных

### Порядок выполнения разделов команды SELECT:

- раздел SELECT выполняется предпоследним,
- раздел ORDER BY выполняется последним,
- остальные разделы — в указанном порядке: FROM, WHERE, GROUP BY, HAVING .

Раздел SELECT – единственный обязательный.

Раздел FROM обязателен, если данные должны браться из таблиц.

Каждый раздел следует писать с отдельной строки для повышения читаемости кода.

### Синтаксис раздела SELECT:

**SELECT** [DISTINCT ] [TOP n [PERCENT]] \* | списокВыборки

- [ ] – необязательные параметры
- вариант1 | вариант2 – вариант1 или вариант2
- DISTINCT – без дубликатов
- TOP – первые n записей (или первые n% записей, если есть PERCENT)
- \* – все столбцы
- списокВыборки – определенные столбцы, константы, выражения, функции

Названия столбцов могут быть даны в полной форме (требуется при объединении таблиц):

- **таблица.столбец:** Games.idGame
- **БД.схема.таблица.столбец:** pcs9115.dbo.Games.idGame

Столбцам можно задать псевдоним после AS, указывать AS необязательно:

• столбец AS Псевдоним	• столбец Псевдоним
------------------------	---------------------

Ключевые слова, пробелы и спецсимволы требуют указания названия в квадратных скобках: [название]

**Примеры:**

Выборка текущей даты	SELECT GETDATE()
Выборка все записей из таблицы Games	SELECT * FROM Games
Выборка первой записи из таблицы Games	SELECT TOP 1 * FROM Games
Выборка названия и цены из таблицы Games	SELECT name, price FROM Games
Выборка названия и цены из таблицы Games с указанием псевдонимов столбцов	SELECT name AS [Название игры], price AS Цена FROM Games
Выборка названия и цены из таблицы Games. Цена вычисляется в долларах	SELECT name Название, price/77.87 [Цена, \$] FROM Games

**Синтаксис раздела ORDER BY:**

**ORDER BY** столбец1 ASC | DESC, столбец2 ASC | DESC, ...

- ASC – сортировка по возрастанию, используется по умолчанию, можно не указывать
- DESC – сортировка по убыванию
- столбцы сортировки — любые из раздела SELECT

**Примеры:**

Сортировка по возрастанию названия	SELECT * FROM Games ORDER BY name
Сортировка по убыванию названия	SELECT * FROM Games ORDER BY name DESC
Сортировка по убыванию категории и возрастанию цены	SELECT * FROM Games ORDER BY category DESC, price

**Постраничный вывод**

Постраничный вывод (пагинация) – доступен только в отсортированных данных.

Для реализации используется:

- OFFSET – задает количество пропускаемых строк
- FETCH – задает количество отображаемых строк, является необязательным

Особенность: не может работать вместе с TOP

**Синтаксис:**

**ORDER BY** столбцы сортировки

**OFFSET** пропускаемоеКоличество **ROWS**

**FETCH** отображаемоеКоличество **NEXT ROWS ONLY**

**Примеры:**

Вывод всех строк кроме трех первых (здесь на странице — 3 строки)	SELECT * FROM Games ORDER BY name OFFSET 3 ROWS
Вывод трех строк с пятой страницы (здесь на странице — 3 строки, поэтому пропускается 12 строк)	SELECT * FROM Games ORDER BY name OFFSET 12 ROWS FETCH NEXT 3 ROWS ONLY

## Синтаксис раздела WHERE:

WHERE условие выборки данных

Операторы, используемые при составлении условий:

### 1) Логические:

И: условие1 <b>AND</b> условие2	ИЛИ: условие1 <b>OR</b> условие2	НЕ: <b>NOT</b> условие
---------------------------------	----------------------------------	------------------------

### 2) Операторы сравнения:

> и <, >= и <=, = и <>

Выборка игр категории RPG ценой меньше 1000	SELECT * FROM Games WHERE category = 'RPG' AND price < 1000
---	---

### 3) Оператор BETWEEN (значения, входящие в диапазон от минимального до максимального):

Условие для цены от 500 до 1000: WHERE price **BETWEEN** 500 **AND** 1000

Условие для цены не из диапазона от 500 до 1000: WHERE price **NOT BETWEEN** 500 **AND** 1000

### 4) Оператор IN (значения, входящие в указанный набор данных):

Условие для категории RPG или шутер: WHERE category **IN** ('RPG', 'Шутер')

Условие для категорий кроме RPG или шутер: WHERE category **NOT IN** ('RPG', 'Шутер')

### 5) Оператор IN (проверка на NULL-значения):

Условие для игр, у которых указано описание: WHERE description **IS NOT NULL**

Условие для игр, у которых не указано описание: WHERE description **IS NULL**

### 6) Оператор LIKE (поиск по подстроке):

% - замена любого количества любых символов (аналог \* в шаблонах на C#):

- 'текст%' — строки, начинающиеся с указанного текста
- '%текст' — строки, заканчивающиеся на указанный текст
- '%текст%' — строки, содержащие указанный текст в любой части строки

Условие для игр, где название начинается на W: WHERE name **LIKE** 'W%'

Условие для игр, где название не начинается на W: WHERE name **NOT LIKE** 'W%'

## Синтаксис раздела GROUP BY:

GROUP BY столбец1, столбец2, ...

### Особенности группировки:

- обычно применяется, если нужно использовать агрегатные (статистические) функции
- при использовании в SELECT можно указывать константы, агрегатные функции и выражения со столбцами, указанными в столбцах группировки

### Агрегатные функции:

<ul style="list-style-type: none"><li>• <b>MIN</b>(столбец) – минимум по столбцу</li><li>• <b>MAX</b>(столбец) – максимум по столбцу</li><li>• <b>SUM</b>(столбец) – сумма по столбцу</li><li>• <b>AVG</b>(столбец) – среднее по столбцу</li></ul>	<ul style="list-style-type: none"><li>• <b>COUNT</b>(*) – количество строк</li><li>• <b>COUNT</b>(столбец) – количество не NULL значений</li><li>• <b>COUNT</b>(DISTINCT столбец) – количество уникальных значений в столбце</li></ul>
--	--

### Примеры:

Вывод максимальной и минимальной цен игр	SELECT MIN(price) Минимум, MAX(price) Максимум FROM Games
Вывод категории и максимальной и минимальной цен для каждой категории игр	SELECT category, MIN(price), MAX(price) FROM Games GROUP BY category

### **Синтаксис раздела HAVING:**

**HAVING** условие выборки данных из групп

### **Особенности HAVING:**

- условия составляются как в WHERE
- в условиях можно использовать агрегатные функции
- применяется только вместе с группировкой

### **Пример:**

Получение списка категорий, в которых больше трех игр	SELECT category, COUNT(*) FROM Games GROUP BY category HAVING COUNT(*) > 3
---	---

### **Выражение CASE:**

CASE позволяет вернуть значение и может быть использован в SELECT

#### **1) CASE, проверяющий условие:**

<b>Общая форма:</b>  CASE WHEN условие1 THEN значение1 WHEN условие2 THEN значение2 ... ELSE значениеN END	<b>Пример:</b>  Добавление столбца с ценовой категорией игры: SELECT *, <u>CASE</u> <u>WHEN price &gt; 1000 THEN 'Дорогие'</u> <u>ELSE 'Дешевые'</u> <u>END AS 'Ценовая категория'</u> FROM Games
---	--

#### **2) CASE, проверяющий значение:**

<b>Общая форма:</b>  CASE проверяемоеЗначение WHEN сравниваемоеЗначение1 THEN значение1 WHEN сравниваемоеЗначение2 THEN значение2 ... ELSE значениеN END	<b>Пример:</b>  Изменение названия категории с RPG на РПГ: SELECT *, <u>CASE category</u> <u>WHEN 'RPG' THEN 'РПГ'</u> <u>ELSE category</u> <u>END</u> FROM Games
---	--