Создание запросов, содержащих оператор JOIN

В разделе **FROM** команды SELECT можно объединять 2 и более таблиц, используя оператор **JOIN**.

Для объедиения обычно используются PK и FK.
Нужно учитывать, что названия столбцов в запросе не могут повторяться, поэтому нужно писать:

- таблица.столбец

Синтаксис раздела FROM:
FROM table1 **JOIN** table2 **ON** table1.столбец1 = table2.столбец2
Здесь table1 – левая таблица, table2 – правая таблица. Вместо JOIN может быть другое соединение

Виды соединений:

- **внутреннее** (**INNER JOIN** или **JOIN**) — используется по умолчанию, возвращает записи, совпадающие в обеих таблицах по ключам
- **правое** (**RIGHT JOIN**) – возвращает все строки из правой таблицы (таблицы справа от JOIN) и соответствующие им строки из левой таблицы. Вместо отсутствующих совпадений в левой таблице пишется NULL
- **левое** (**LEFT JOIN**) – возвращает все строки из левой таблицы (таблицы слева от JOIN) и соответствующие им строки из правой таблицы. Вместо отсутствующих совпадений в правой таблице пишется NULL
- **полное внешнее** (**FULL JOIN**) – возвращает все строки из обеих таблицы. Вместо отсутсвующих совпадений в таблицах пишется NULL

Для удобства чтения и редактирования запроса следует писать JOIN отдельной строкой:

```
FROM table1
    JOIN table2 ON table2.столбец2 = table1.столбец1
```

Можно объединять более двух таблиц. Для этого нужно писать несколько JOIN:

```
FROM table1
    JOIN table2 ON table2.столбец2 = table1.столбец1
    JOIN table3 ON table3.столбец3 = …
    JOIN table4 ON table4.столбец4 = …
```

Таблицам в разделе FROM можно давать псевдонимы так же, как и псевдонимы столбцам

Примеры:

Получение данных о всех играх и датах их продаж	SELECT Games.*, Sales.date FROM Games JOIN Sales ON Sales.idGame = Games.idGame
Получение данных о всех играх и датах их продаж. Если игры не покупали, они все равно будут выведены	SELECT Games.*, Sales.date FROM Games LEFT JOIN Sales ON Sales.idGame = Games.idGame
Получение данных о всех играх, датах их продаж и пользователях	SELECT Games.*, User.*, Sales.date FROM Games JOIN Sales ON Sales.idGame = Games.idGame JOIN Users ON Users.idUser = Sales.idUser
Получение данных количестве продаж игр	SELECT Games.*, COUNT(Games.idGame) FROM Games JOIN Sales ON Sales.idGame = Games.idGame GROUP BY Games.idGame
Получение данных о всех играх и датах их продаж с указанием псевдонимов таблиц	SELECT Games.*, Sales.date FROM Games AS g JOIN Sales AS s ON s.idGame = g.idGame

Собственное объединение — таблица объединяется сама с собой (требует указания псевдонима):

FROM t1
    JOIN t1 **AS** t2 **ON** t1.столбец1 = t2.столбец2

Декартово произведение — каждой строке одной таблиы ставится в соответствие каждая строка другой таблицы (сравнение столбцов не нужно):

FROM t1 **JOIN** t2

Объединение результатов запросов

Синтаксис объединения результатов запросов:
SELECT … -- первый запрос на выборку
команда объединения
SELECT … -- второй запрос на выборку

Особенности:
- у запросов на выборку должны быть одинаковое количество и типы столбцов
- псевдонимы столбцов задаются в первом запросе
- сортировка выполняется в последнем запросе

Команды объединения:
- **UNION** – возвращает все строки из обоих запросов без дубликатов
- **UNION ALL** – возвращает все строки из обоих запросов с дубликатами
- **EXCEPT** – возвращает все строки первого запроса кроме совпавших со строками второго
- **INTERSECT** – возвращает все строки, совпавшие в первом и втором запросах

Примеры:

Выборка названий игр и имен пользователей с уточнением игра / пользователь во втором столбце	SELECT name, 'игра' FROM Games **UNION** SELECT name, 'пользователь' FROM Users
Выборка названий игр и имен пользователей с уточнением игра / пользователь во втором столбце с указанием псевдонимов и сортировкой	SELECT name, 'игра' AS Уточнение FROM Games **UNION** SELECT name, 'пользователь' FROM Users ORDER BY name
Выборка игр ценой до 1000 кроме игр категории RPG	SELECT * FROM Games WHERE price < 1000 **EXCEPT** SELECT * FROM Games WHERE category = 'RPG'

Создание подзапросов на выборку данных

Подзапрос — запрос в ( ), вложенный в SQL-запрос.

Правила написания подзапросов:

- в них нет сортировки
- из подзапроса можно обратиться к столбцам таблиц основного запроса
- в операциях сравнения подзапрос пишется справа от оператора сравнения
- для уменьшения количества сравнений в подзапросах, возвращающих таблицу или столбец, нужно писать DISTINCT

Виды подзапросов:

- подзапрос, возвращающий одно значение. Обычно в подзапросе используется агрегатная функция
- подзапрос, возвращающий список значений из одного столбца
- подзапрос, возвращающий набор строк или таблицу. Может использоваться как источник данных в разделе FROM, такому подзапросу нужно задавать псевдоним

Примеры:

Одно значение Вернуть игры с ценой выше средней. Подзапрос возвращает среднюю цену	SELECT * FROM Games WHERE price > **(SELECT AVG(price)** **FROM Games)**
Список значений Вернуть игры, которые не покупались. Подзапрос возвращает игры, которые покупали	SELECT * FROM Games WHERE idGame NOT IN **(SELECT DISTINCT idGame** **FROM Sales)**
Набор значений	SELECT ... FROM **(SELECT …** **FROM …) AS t**

Для упрощения разработки и тестирования следует:

1. проверить работу подзапроса как отдельного SQL-запроса,
2. проверить работу основного запроса (вместо подзапроса использовать константы)
3. включить подзапрос в основной запрос и проверить их совместную работу

Пример написания запроса, возвращающего игры с ценой выше средней:

1) написать запрос, возвращающий среднюю цену игр:

```
SELECT AVG(price)
FROM Games
```

2) написать запрос, возвращающий игр с ценой выше указанной:

```
SELECT *
FROM Games
WHERE price > 1000
```

3) объединить запрос и подзапрос:

```
SELECT *
FROM Games
WHERE price > (SELECT AVG(price)
                FROM Games)
```

Операторы, применяющиеся при работе с подзапросами

Операторы:

- **IN** – проверка вхождения в набор значений
- **ALL** – вернет истину, если условие верно для всех строк подзапроса
- **ANY** – вернет истину, если оно оказалось истинным хотя бы для одной строки подзапроса
- **EXISTS** – позволяет проверить, возвращает ли подзапрос хотя бы для одно значение

Примеры:

Вернуть игры, которые не покупались	SELECT * FROM Games WHERE idGame **NOT IN** (SELECT DISTINCT idGame FROM Sales)
Вернуть игры, цена которых больше цен игр категории RPG	SELECT * FROM Games WHERE price > **ALL** (SELECT DISTINCT price FROM Sales WHERE category = 'RPG')
Вернуть игры, цена которых меньше цен игр категории RPG	SELECT * FROM Games WHERE price < **ANY** (SELECT DISTINCT price FROM Sales WHERE category = 'RPG')
Вернуть игры, которые не покупались	SELECT * FROM Games WHERE **NOT EXISTS** (SELECT * FROM Sales WHERE Games.idGame = Sales.idGame)