

# Вебинар 6

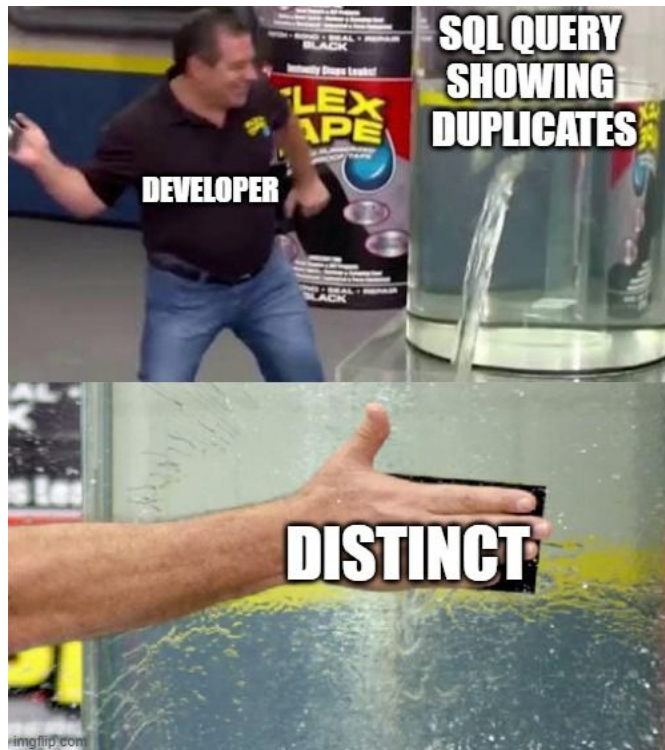
Группировка данных и объединение таблиц

# Выбор уникальных значений столбца

Из предыдущего занятия мы помним, что мы можем сделать выборку уникальных столбцов и строк с помощью ключевого слова **DISTINCT**:

```
SELECT DISTINCT имя_столбца FROM  
имя_таблицы
```

Но мы также можем использовать и оператор **GROUP BY** для этих (и других) целей.



# Оператор GROUP BY

Оператор GROUP BY – это команда, которая группирует при выборке данные, имеющие одинаковые значения в столбце.

Т.е. на выходе мы получаем таблицу, в которой есть только уникальные значения в столбце имя\_столбца.

```
SELECT имя_столбца FROM имя_таблицы GROUP BY  
имя_столбца;
```

Output:

char_id	name	weapon	hp
1	Al-Haitham	Sword	456
2	Jinx	Gun	785
3	Steve	Sword	400
4	Jett	Gun	672
5	Tracer	Gun	983
6	Lisa	Book	1234
7	Green Arrow	Bow	3333

name
Al-Haitham
Jinx
Steve
Jett
Tracer
Lisa
Green Arrow

# Групповые функции.

- Групповая функция **SUM()** суммирует все значения в столбце. При работе с оператором GROUP BY она суммирует все значения в столбце в рамках группы.
- Групповая функция **COUNT()** считает, сколько записей относится к группе.

```
SELECT столбец1, SUM(столбец2), COUNT(столбец3) FROM таблица  
GROUP BY столбец1;
```

name	sum(weapon)	count(hp)
Haitham	0	1
Jinx	0	1
Steve	0	1
Jett	0	1
Tracer	0	1
Lisa	0	1
Arrow	0	1

- Групповая функция **MIN()** вычисляет минимальное значение элементов столбца, относящихся к группе.
- Групповая функция **MAX()** вычисляет максимальное значение элементов столбца, относящихся к группе.
- Групповая функция **AVG()** вычисляет среднее значение элементов столбца, относящихся к группе.

# Объединение двух таблиц. Оператор UNION

Для начала - создадим еще одну таблицу в уже существующей БД.

Добавим еще несколько персонажей, для того, чтобы можно было провести операцию объединения таблиц.

**// Такой же алгоритм будет работать и с 3, и с 4 таблицами (и т.д.)**

```
1  create table mycharacters (char_id int primary key auto_increment,  
2  name char(20), weapon char(20), hp int);  
3  insert into mycharacters (name, weapon, hp)  
4  values ("Al-Haitham", "Sword", "456" ),  
5  ("Jinx", "Gun", "785"),  
6  ("Steve", "Sword", "400"),  
7  ("Jett", "Gun", "672"),  
8  ("Tracer", "Gun", "983"),  
9  ("Lisa", "Book", "1234"),  
10 ("Green Arrow", "Bow", "3333");  
11  
12 create table mycharacters2 (char_id int primary key auto_increment,  
13 name2 char(20), weapon2 char(20), hp2 int);  
14 insert into mycharacters2 (name2, weapon2, hp2)  
15 values ("Iter", "Sword", "444" ),  
16 ("Bi-Bi", "Gun", "900"),  
17 ("Lumen", "Sword", "400");
```

# Объединение двух таблиц. Оператор UNION

Объединение двух или более таблиц - это объединение данных из разных таблиц в одну.

**Важно.** При объединении количество столбцов во всех таблицах должно совпадать, иначе будет ошибка. Имена столбцов будут такие же, как в основной таблице, в которую добавляются данные из других таблиц.

Для объединения данных нужно воспользоваться командой **UNION**

// Так у нас выведется таблица, в которой все строки будут уникальными (т.е. ни одна строка не будет полностью совпадать с другой), при этом мы получим выборку из двух таблиц.

```
SELECT * FROM имя_таблицы1  
WHERE условие  
UNION SELECT * FROM имя_таблицы2  
WHERE условие
```

char_id	name	weapon	hp
2	Jinx	Gun	785
4	Jett	Gun	672
5	Tracer	Gun	983
6	Lisa	Book	1234
7	Green Arrow	Bow	3333
9	Bi-Bi	Gun	900

# Объединение двух таблиц. Оператор UNION

Если мы хотим получить все строки, в т.ч. и повторяющиеся, то нужно использовать служебное слово **ALL**

Для примера - продублируем строку с любым персонажем.

8	("Tracer", "Gun", "983"),
9	("Tracer", "Gun", "983"),
10	("Tracer", "Gun", "983"),

**Важно.** Имена колонок берутся из первой таблицы.

```
SELECT * FROM имя_таблицы1  
WHERE условие  
UNION ALL SELECT * FROM  
имя_таблицы2 WHERE условие
```

char_id	name	weapon	hp
2	Jinx	Gun	785
4	Jett	Gun	672
5	Tracer	Gun	983
6	Tracer	Gun	983
7	Tracer	Gun	983
8	Lisa	Book	1234
9	Green Arrow	Bow	3333
11	Bi-Bi	Gun	900

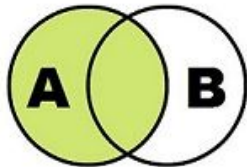
# Соединение таблиц. Оператор JOIN.

Соединение таблиц – это операция, когда таблицы сравниваются между собой построчно и появляется возможность вывода столбцов из всех таблиц, участвующих в соединении.

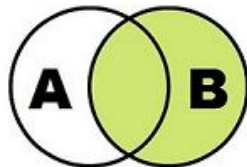
Есть несколько видов оператора JOIN.



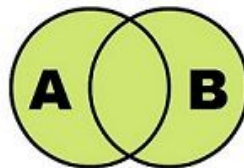
**INNER JOIN**



**LEFT JOIN**



**RIGHT JOIN**



**FULL JOIN**

// A - таблица 1, B - таблица 2.



# Соединение таблиц. Оператор INNER JOIN.

- **INNER JOIN** – внутреннее соединение. Выводит строки, только если условие соединения выполняется. Условие прописывается после ключевого слова ON. В запросах необязательно прописывать INNER – если написать только JOIN, то СУБД по умолчанию выполнить именно внутреннее соединение.

```
SELECT * FROM таблица1  
JOIN таблица2 ON условие;
```

Output:

char_id	name	weapon	hp	char_id	name2	weapon2	hp2
1	Al-Haitham	Sword	456	2	Bi-Bi	Gun	900
2	Jinx	Gun	785	2	Bi-Bi	Gun	900
3	Steve	Sword	400	2	Bi-Bi	Gun	900
4	Jett	Gun	672	2	Bi-Bi	Gun	900
5	Tracer	Gun	983	2	Bi-Bi	Gun	900
6	Lisa	Book	1234	2	Bi-Bi	Gun	900
7	Green Arrow	Bow	3333	2	Bi-Bi	Gun	900

// Убрали “дубли” строки с “Tracer”

// Написали: `select*from mycharacters join  
mycharacters2 on hp2 > 800;`

# Соединение таблиц.

## Оператор LEFT и RIGHT JOIN.

- **LEFT JOIN** и **RIGHT JOIN** называются левым и правым соединениями, или внешними. При таких соединениях из левой (для LEFT JOIN) или из правой таблицы (для RIGHT JOIN) попадает в результаты в любом случае.

**Левая таблица** - та, название которой указывается перед написанием ключевых слов [LEFT | RIGHT| INNER] JOIN, **правая таблица** – та, название которой указываем после них.

```
SELECT * FROM левая_таблица  
RIGHT JOIN правая_таблица  
ON условие;
```

Output:

char_id	name	weapon	hp	char_id	name2	weapon2	hp2
NULL	NULL	NULL	NULL	1	Iter	Sword	444
7	Green	Arrow	Bow	3333	2	Bi-Bi	Gun 900
6	Lisa	Book	1234	2	Bi-Bi	Gun	900
5	Tracer	Gun	983	2	Bi-Bi	Gun	900
4	Jett	Gun	672	2	Bi-Bi	Gun	900
3	Steve	Sword	400	2	Bi-Bi	Gun	900
2	Jinx	Gun	785	2	Bi-Bi	Gun	900
1	Al-Haitham	Sword	456	2	Bi-Bi	Gun	900
NULL	NULL	NULL	NULL	3	Lumen	Sword	400

Как вы думаете, почему при таком соединении появились “NULL”?

# Соединение таблиц. Оператор FULL JOIN.

- **FULL JOIN** - полное внешнее соединение. Возвращает все строки из всех таблиц, участвующих в соединении, соединив между собой те, которые подошли под условие ON.

```
SELECT * FROM таблица1  
FULL JOIN таблица2  
ON условие;
```

Output:

char_id	name	weapon	hp	char_id	name2	weapon2	hp2
1	Al-Haitham	Sword	456	2	Bi-Bi	Gun	900
2	Jinx	Gun	785	2	Bi-Bi	Gun	900
3	Steve	Sword	400	2	Bi-Bi	Gun	900
4	Jett	Gun	672	2	Bi-Bi	Gun	900
5	Tracer	Gun	983	2	Bi-Bi	Gun	900
6	Lisa	Book	1234	2	Bi-Bi	Gun	900
7	Green Arrow	Bow	3333	2	Bi-Bi	Gun	900