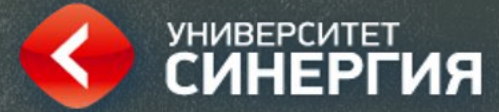
The background is a dark, textured surface with faint, light-colored chalk-like drawings. These include a globe in the upper left, a large letter 'V' in the top left, a microscope on the left side, a stack of books at the bottom left, a plus sign and a cross in the bottom left, an open book with handwritten text in the bottom center, and a large percentage sign and other symbols in the bottom right.

# Практическое занятие

Преподаватель - Борисова Татьяна Андреевна

# Сегодня на занятии



- Повторение предыдущего материала
- Выполнение запросов с SELECT (полная выборка, выборка и элементарные арифметические операции, DISTINCT, COUNT)
- Обсуждение домашнего задания

# Структура SQL запросов

Общая структура запроса выглядит следующим образом:

**SELECT** ('столбцы или \* для выбора всех столбцов; обязательно')

**FROM** ('таблица; обязательно')

**WHERE** ('условие/фильтрация, например, city = 'Moscow'; необязательно')

**GROUP BY** ('столбец, по которому хотим сгруппировать данные; необязательно')

**HAVING** ('условие/фильтрация на уровне сгруппированных данных; необязательно')

**ORDER BY** ('столбец, по которому хотим отсортировать вывод; необязательно')



# Пояснение

SELECT, FROM — обязательные элементы запроса, которые определяют выбранные столбцы, их порядок и источник данных.

WHERE — необязательный элемент запроса, который используется, когда нужно отфильтровать данные по нужному условию. Очень часто внутри элемента where используются IN / NOT IN для фильтрации столбца по нескольким значениям, AND / OR для фильтрации таблицы по нескольким столбцам.

GROUP BY — необязательный элемент запроса, с помощью которого можно задать агрегацию по нужному столбцу (например, если нужно узнать какое количество клиентов живет в каждом из городов).

При использовании GROUP BY обязательно:

1. перечень столбцов, по которым делается разрез, был одинаковым внутри SELECT и внутри GROUP BY,
2. агрегатные функции (SUM, AVG, COUNT, MAX, MIN) должны быть также указаны внутри SELECT с указанием столбца, к которому такая функция применяется.

# Пояснение

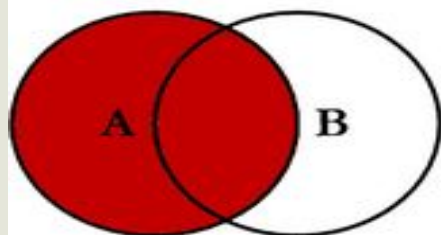
HAVING — необязательный элемент запроса, который отвечает за фильтрацию на уровне сгруппированных данных (по сути, WHERE, но только на уровень выше).

ORDER BY — необязательный элемент запроса, который отвечает за сортировку таблицы.

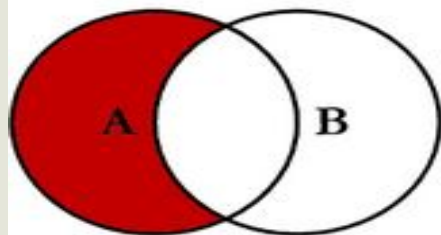
JOIN — необязательный элемент, используется для объединения таблиц по ключу, который присутствует в обеих таблицах. Перед ключом ставится оператор ON.

# Пояснение

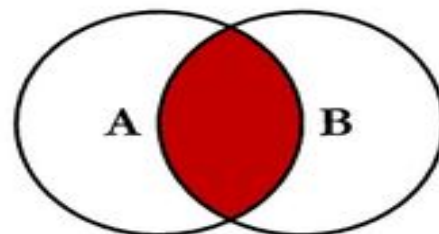
## SQL JOINS



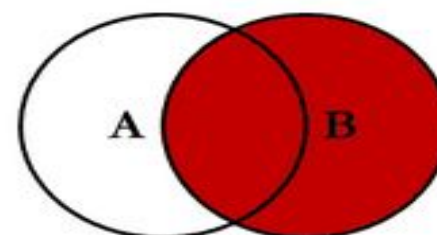
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key
```



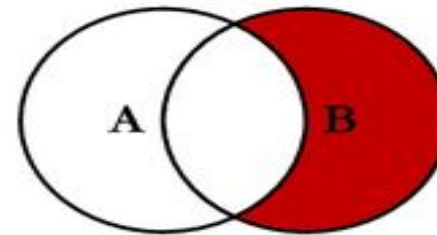
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key  
WHERE B.Key IS NULL
```



```
SELECT <select_list>  
FROM TableA A  
INNER JOIN TableB B  
ON A.Key = B.Key
```



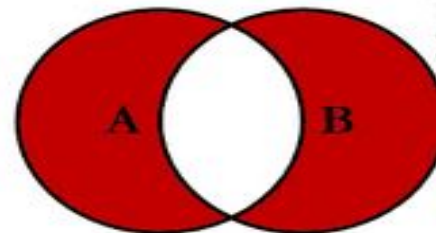
```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key
```



```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key  
WHERE A.Key IS NULL
```



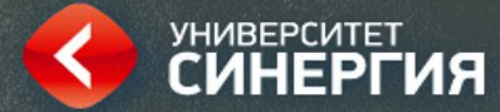
```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key
```



```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key  
WHERE A.Key IS NULL  
OR B.Key IS NULL
```



# Домашнее задание



1. Выбрать все данные из таблицы customers
2. Выбрать все записи из таблицы customers, но только колонки "имя контакта" и "город"
3. Выбрать все записи из таблицы orders, но взять две колонки: идентификатор заказа и колонку, значение в которой мы рассчитываем как разницу между датой отгрузки и датой формирования заказа.
4. Выбрать все уникальные города в которых "зарегистрированы" заказчики
5. Выбрать все уникальные сочетания городов и стран в которых "зарегистрированы" заказчики
6. Посчитать количество заказчиков
7. Посчитать количество уникальных стран в которых "зарегистрированы" заказчики

*СПАСИБО ЗА  
ВНИМАНИЕ!*

