

Лекция по дисциплине: «Основы проектирования баз данных»

для специальности: 09.02.07
«Информационные системы
и программирование»

Тема: Язык SQL

Подготовила преподаватель:
Мишанкина Лариса Геннадьевна

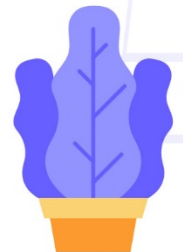


Цель:

Изучение структуры языка SQL

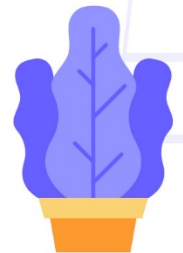
Задачи:

- ❖ познакомиться с основными понятиями и определениями языка;
- ❖ познакомиться с операторами языка SQL;
- ❖ познакомиться с достоинствами и недостатками языка SQL



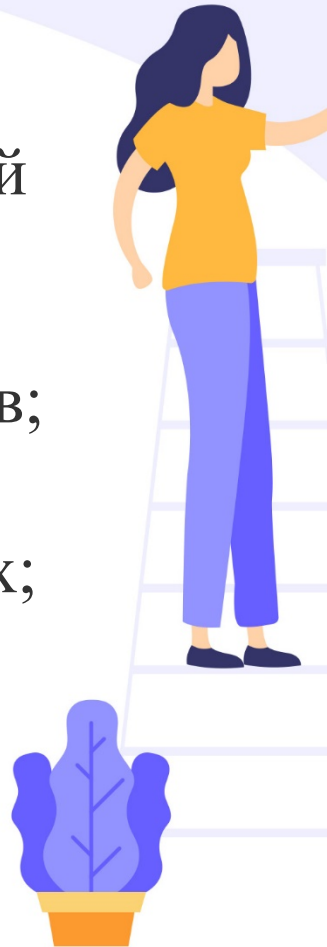
План:

- Основные достоинства языка SQL
- Общая структура SQL
- Операторы языка SQL
- Понятие агрегации данных
- Выводы
- Контрольные вопросы

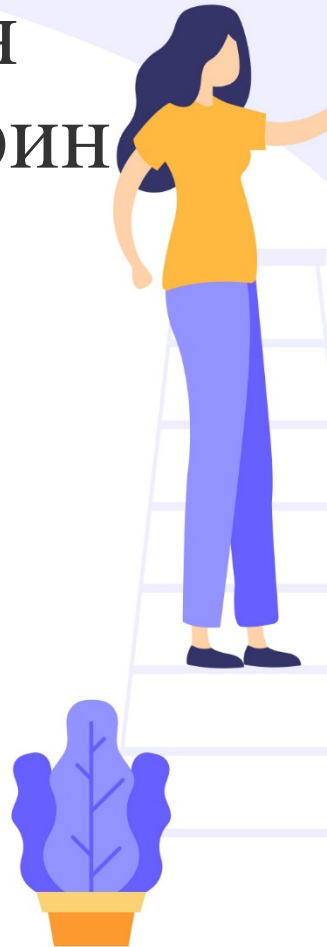


Основные достоинства языка SQL заключаются в следующем:

- стандартность;
- независимость от конкретных СУБД;
- возможность переноса с одной вычислительной системы на другую;
- реляционная основа языка;
- возможность создания интерактивных запросов;
- возможность программного доступа к БД;
- обеспечение различного представления данных;
- возможность динамического изменения и расширения структуры БД;
- поддержка архитектуры клиент-сервер.

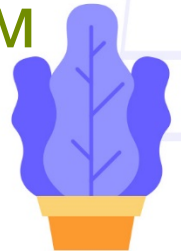
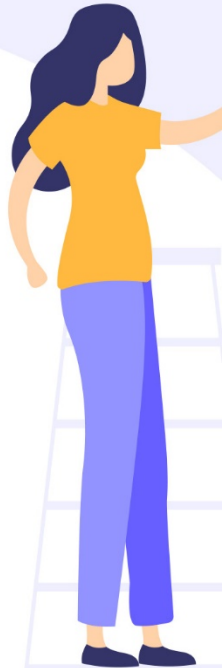


- Наиболее часто SQL используется для формирования выгрузок, витрин (с последующим построением отчетов на основе этих витрин) и администрирования баз данных.



Общая структура SQL

- **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')

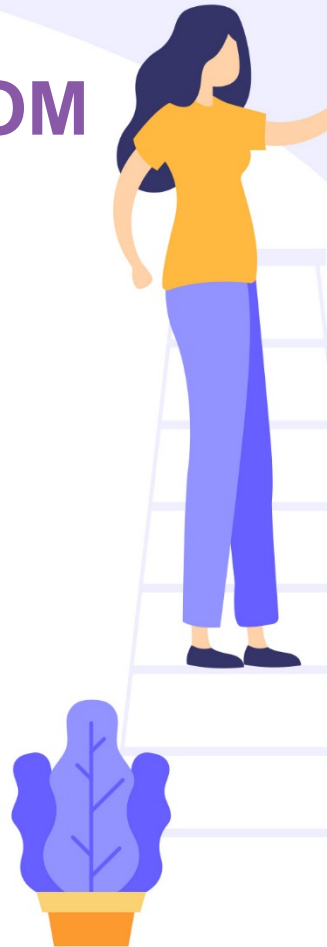


- **SELECT, FROM** — обязательные элементы запроса, которые определяют выбранные столбцы, их порядок и источник данных.
- Выбрать все (обозначается как *) из таблицы Customers:
- **SELECT * FROM** Customers



Выбрать столбцы CustomerID, CustomerName из таблицы Customers:

- **SELECT** CustomerID, CustomerName **FROM** Customers



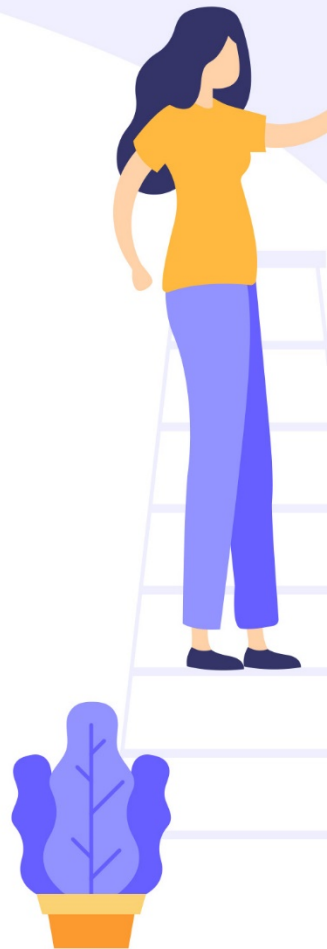
ОПЕРАТОР WHERE

- 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.



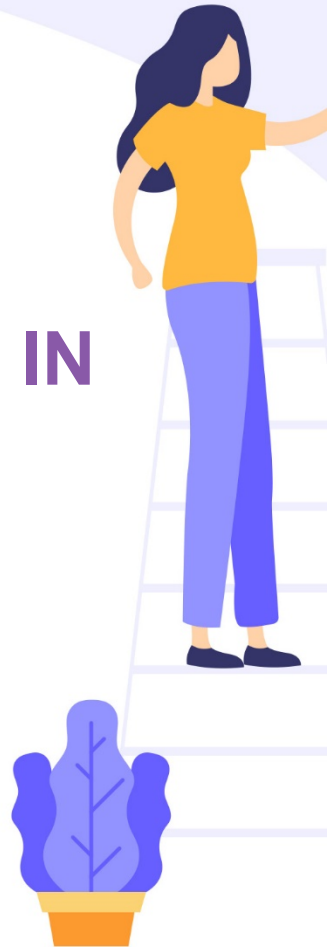
Фильтрация по одному условию и одному значению:

- **select** * **from** Customers **WHERE** City = 'London'



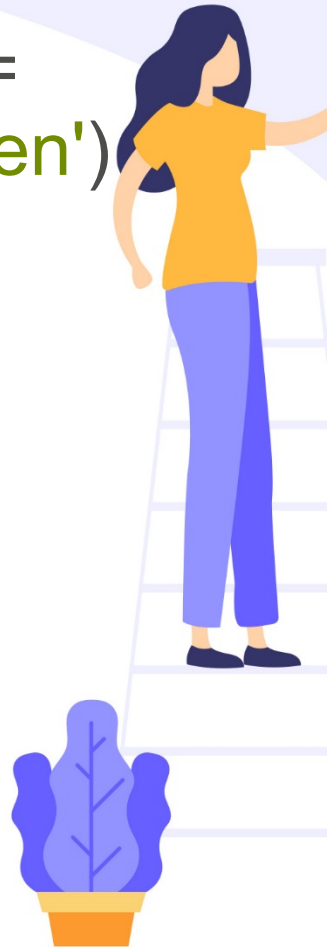
Фильтрация по одному условию и нескольким значениям с применением IN (включение) или NOT IN (исключение):

- **select * from Customers where City IN ('London', 'Berlin')**
- **select * from Customers where City NOT IN ('Madrid', 'Berlin', 'Bern')**



Фильтрация по нескольким условиям с применением AND (выполняются все условия) или OR (выполняется хотя бы одно условие) и несколькими значениями:

- **select * from Customers where Country = 'Germany' AND City not in ('Berlin', 'Aachen') AND CustomerID > 15**
- **select * from Customers where City in ('London', 'Berlin') OR CustomerID > 4**



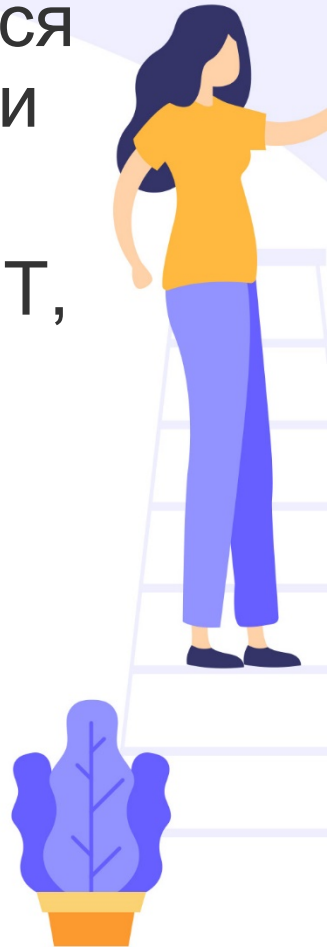
ОПЕРАТОР GROUP BY

- 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).



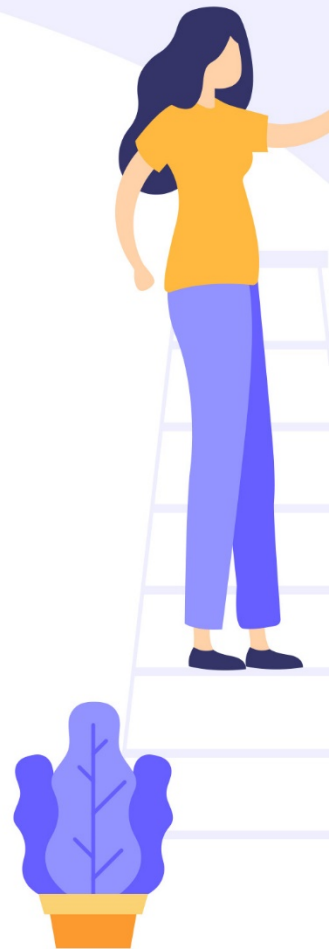
При использовании GROUP BY обязательно:

1. перечень столбцов, по которым делается разрез, должен быть 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.



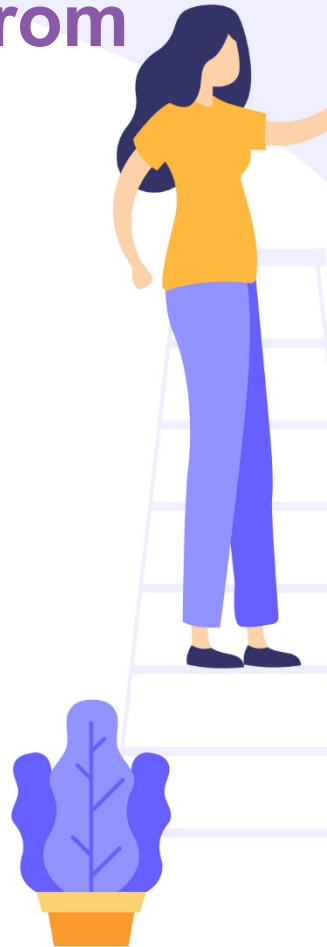
Группировка количества клиентов по городу:

- **select** City, **count**(CustomerID) **from** Customers **GROUP BY** City



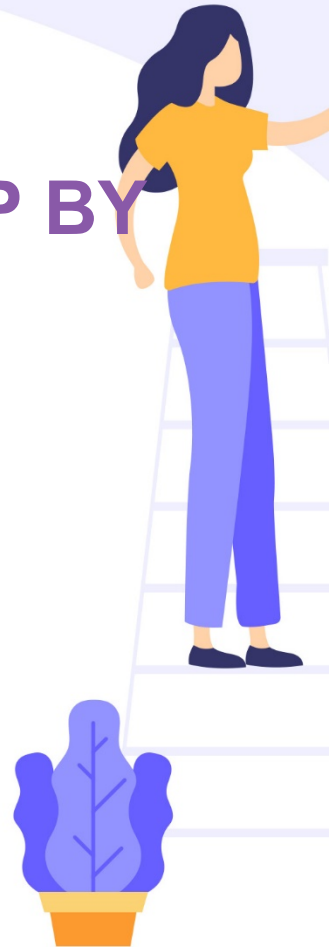
Группировка количества клиентов по стране и городу:

- **select** Country, City, **count**(CustomerID) **from** Customers **GROUP BY** Country, City



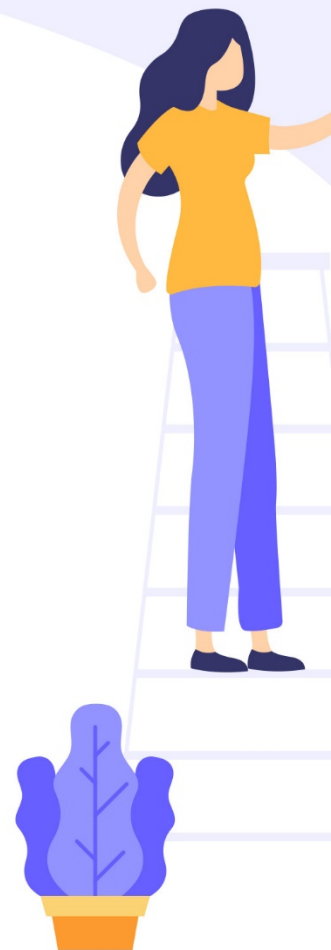
Группировка продаж по ID товара с разными агрегатными функциями:
количество заказов с данным товаром и
количество проданных штук товара:

- **select** ProductID, **COUNT**(OrderID),
SUM(Quantity) **from** OrderDetails **GROUP BY**
ProductID



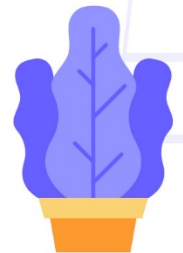
Группировка продаж с фильтрацией исходной таблицы. В данном случае на выходе будет таблица с количеством клиентов по городам Германии:

- **select** City, **count**(CustomerID) **from** Customers **WHERE** Country = 'Germany' **GROUP BY** City



Агрегация данных –

- **это** процесс сбора, обработки и объединения **данных** из различных источников в одну общую **базу данных**. Он позволяет упростить понимание информации и принимать взвешенные решения на основе полной картины **данных**.



ОПЕРАТОР HAVING

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

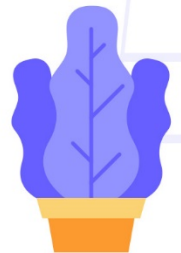
Фильтрация агрегированной таблицы с количеством клиентов по городам, в данном случае оставляем в выгрузке только те города, в которых не менее 5 клиентов:

- **select** City, **count**(CustomerID) **from** Customers **group by** City **HAVING** **count**(CustomerID) **>= 5**



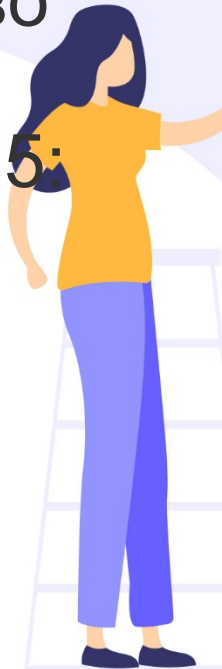
- В случае с переименованным столбцом внутри **HAVING** можно указать как и саму агрегирующую конструкцию `count(CustomerID)`, так и новое название столбца `number_of_clients`:

- **select** City, `count`(CustomerID) **as** number_of_clients **from** Customers **group by** City **HAVING** number_of_clients \geq 5



- Пример запроса, содержащего WHERE и HAVING. В данном запросе сначала фильтруется исходная таблица по пользователям, рассчитывается количество клиентов по городам и остаются только те города, где количество клиентов не менее 5:

- **select** City, **count**(CustomerID) **as** number_of_clients **from** Customers **WHERE** CustomerName **not in** ('Around the Horn','Drachenblut Delikatessend') **group by** City **HAVING** number_of_clients **>= 5**



ОПЕРАТОР ORDER BY

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

Простой пример сортировки по одному столбцу. В данном запросе осуществляется сортировка по городу, который указал клиент:

- **select** * **from** Customers **ORDER BY** City

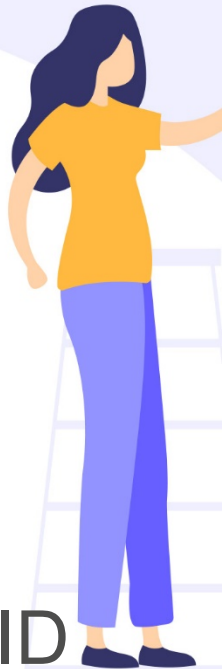


- Осуществлять сортировку можно и по нескольким столбцам, в этом случае сортировка происходит по порядку указанных столбцов:
- **select** * **from** Customers **ORDER BY** Country, City



- По умолчанию сортировка происходит по возрастанию для чисел и в алфавитном порядке для текстовых значений. Если нужна обратная сортировка, то в конструкции ORDER BY после названия столбца надо добавить DESC:

- **select** * **from** Customers **order by** CustomerID **DESC**



- Обратная сортировка по одному столбцу и сортировка по умолчанию по второму:
- **select * from Customers order by Country DESC, City**



• ОПЕРАТОР JOIN

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

Запрос, в котором соединяем таблицы Order и Customer по ключу CustomerID, при этом перед названием столбца ключа добавляется название таблицы через точку:

- **select * from** Orders **JOIN** Customers **ON** Orders.CustomerID = Customers.CustomerID



Выводы:

- Язык SQL используется для построения запросов и витрин в базах данных
- Общая структура языка проста и понятна пользователю
- Агрегация данных - это процесс сбора, обработки и объединения **данных** из различных источников в одну общую **базу данных**.



Контрольные вопросы:

1. Дайте определение кортежа
2. Дайте определение атрибута
3. Ограничение отношения это?
4. Проекция отношения это?
5. Что представляет собой пересечение отношений?
6. Что представляет собой разность отношений?
7. Что представляет собой произведение отношений?
8. Что представляет собой деление отношений?
9. При естественном соединении отношений что происходит?
10. При соединении отношений по условию что происходит?



ИСТОЧНИКИ:

- <https://habr.com/ru/articles/480838/>
- <https://Wikipedia.ru>

