

# GenTech #7 / Aug 15, 2023

## Базовые термины

- рефакторинг (кода) - перевыпуск (переработка) кода
- проект - кодовая база (как пример)
- алгоритм - идея (способ) решения задачи
- программа - реализация алгоритма
- стек (программный) - набор технологий (инструментов) для создания проекта

## Примеры запросов

```
SELECT * FROM Customers WHERE Country IN ('Germany', 'USA') --  
Country='Germany' -- OR -- Country='USA'
```

```
-- Вывести название и стоимость в USD одного самого дорогого проданного  
товара SELECT Products.ProductName, Products.Price * 1.1 AS Price_usd  
FROM OrderDetails JOIN Products ON  
OrderDetails.ProductID=Products.ProductID ORDER BY Products.Price DESC  
LIMIT 1
```

SQL ▾

 Copy

```
-- Вывести список стран, которые поставляют морепродукты SELECT DISTINCT  
Suppliers.Country FROM Products JOIN Suppliers ON  
Products.SupplierID=Suppliers.SupplierID JOIN Categories ON  
Products.CategoryID=Categories.CategoryID WHERE  
Categories.CategoryName='Seafood'
```

```
-- Вывести сумму проданного товара в `USA` SELECT SUM(Products.Price *  
OrderDetails.Quantity) AS total_sold_to_usa FROM OrderDetails JOIN Orders  
ON OrderDetails.OrderID=Orders.OrderID JOIN Customers ON  
Orders.CustomerID=Customers.CustomerID JOIN Products ON  
OrderDetails.ProductID=Products.ProductID WHERE Customers.Country='USA'
```

```
-- Вывести фамилии менеджеров, у которых отсутствуют заказы SELECT  
Employees.LastName FROM Employees LEFT JOIN Orders ON  
Employees.EmployeeID=Orders.EmployeeID WHERE Orders.EmployeeID IS NULL
```

```
-- Вывести стоимость заказа `10248` SELECT SUM(Products.Price *  
OrderDetails.Quantity) AS order_cost FROM OrderDetails JOIN Products ON  
OrderDetails.ProductID=Products.ProductID WHERE  
OrderDetails.OrderID=10248
```

```
-- Вывести среднюю стоимость напитка из `USA` SELECT AVG(Products.Price)  
AS avg_price FROM Products JOIN Suppliers ON  
Products.SupplierID=Suppliers.SupplierID JOIN Categories ON  
Products.CategoryID=Categories.CategoryID WHERE  
Categories.CategoryName='Beverages' AND Suppliers.Country='USA'
```

```
-- Вывести ко-во заказов, которое оформил клиент `1` SELECT COUNT(*) AS  
total_orders FROM Orders WHERE CustomerID=1
```

```
-- Вывести, сколько раз был продан товар 72 SELECT COUNT(*) AS sold_count  
FROM OrderDetails WHERE ProductID=72
```

```
-- Вывести ко-во поставщиков не из USA и не из Spain SELECT COUNT(*) AS  
total_not_usa_spain_suppliers FROM Suppliers WHERE NOT Country IN ('USA',  
'Spain')
```

```
-- Выводит два самых дорогих напитка из `USA` SELECT * FROM Products JOIN  
Categories ON Products.CategoryID=Categories.CategoryID JOIN Suppliers ON  
Products.SupplierID=Suppliers.SupplierID WHERE Categories.CategoryName  
LIKE 'beverages' AND Suppliers.Country LIKE 'usa' ORDER BY Products.Price  
DESC LIMIT 2
```

## Группировка в SQL (механизм агрегации) / оператор **GROUP BY**

### Категории запросов

1. **CRUD** - простые запросы
2. **Aggregation** - получение вычисленных данных

Группировка - объединение записей на основе общего признака с целью получения обобщенных данных

### Пример. Вывести страны покупателей

```
SELECT DISTINCT Country FROM Customers -- вывести количественное  
распределение клиентов по странам SELECT Country, COUNT(*) AS  
total_customers FROM Customers GROUP BY Country ORDER BY total_customers  
DESC
```

### Пример. Найти кол/распределение товаров по категориям

```
SELECT CategoryID, COUNT(*) AS total_products FROM Products GROUP BY  
CategoryID ORDER BY total_products DESC
```

### Задача. Вывести кол/распределение товаров по поставщикам

```
SELECT SupplierID, COUNT(*) AS total_products FROM Products GROUP BY
SupplierID ORDER BY total_products DESC -- SELECT Suppliers.SupplierName,
COUNT(*) AS total_products FROM Products JOIN Suppliers ON
Products.SupplierID=Suppliers.SupplierID GROUP BY Products.SupplierID
ORDER BY total_products DESC
```

## Проекция (полей) в БД

- инструмент оптимизации запросов, направленный на:
  - уменьшение времени выполнения запроса
  - снижение нагрузки на выч/ресурсы

### Задача. Найти сред/стоимость товара для каждого поставщика

(проекция: `название_поставщика`, `сред_стоимость`)

```
SELECT Suppliers.SupplierName, AVG(Products.Price) AS avg_price FROM
Products JOIN Suppliers ON Products.SupplierID=Suppliers.SupplierID GROUP
BY Suppliers.SupplierID
```

### Задача. Вывести страну, где больше всего клиентов

(проекция: `страна`, `ко_во_клиентов`)

```
SELECT Country, COUNT(*) AS total_customers FROM Customers GROUP BY
Country ORDER BY total_customers DESC LIMIT 1 -- LIMIT 1 OFFSET 1 --
вывести, кто на втором месте
```

### Задача. Вывести компанию, которая находится на третьем месте по количеству доставленных заказов

(проекция: `название_перевозчика`, `ко_во_заказов`)

```
SELECT Shippers.ShipperName, COUNT(*) AS total_orders FROM Orders JOIN
Shippers ON Orders.ShipperID=Shippers.ShipperID GROUP BY
Shippers.ShipperID ORDER BY total_orders DESC LIMIT 1 OFFSET 2
```

Задача. Вывести стоимость каждого заказа (проекция: **номер\_заказа** ,  
**стоимость\_заказа** )

```
SELECT OrderDetails.OrderID, SUM(OrderDetails.Quantity * Products.Price)
AS order_cost FROM OrderDetails JOIN Products ON
OrderDetails.ProductID=Products.ProductID GROUP BY OrderDetails.OrderID
ORDER BY order_cost DESC
```

Задача. Расчитать З/П менеджеров (ставка - **5%** от суммы проданных заказов)  
(проекция: **фамилия\_менеджера** , **з\_п** )

```
SELECT Employees.LastName, SUM(OrderDetails.Quantity * Products.Price) *
.05 AS salary FROM OrderDetails JOIN Products ON
OrderDetails.ProductID=Products.ProductID JOIN Orders ON
OrderDetails.OrderID=Orders.OrderID JOIN Employees ON
Orders.EmployeeID=Employees.EmployeeID GROUP BY Orders.EmployeeID ORDER
BY salary DESC
```

## Фильтрация агрегированных данных / оператор **HAVING**

- аналог **WHERE** , который позволяет отфильтровать агрегированные данные

Пример. Вывести менеджеров, у которых З/П от **2000** EUR

```
SELECT Employees.LastName, SUM(OrderDetails.Quantity * Products.Price) *  
.05 AS salary FROM OrderDetails JOIN Products ON  
OrderDetails.ProductID=Products.ProductID JOIN Orders ON  
OrderDetails.OrderID=Orders.OrderID JOIN Employees ON  
Orders.EmployeeID=Employees.EmployeeID GROUP BY Orders.EmployeeID HAVING  
salary >= 2000 ORDER BY salary DESC
```

**Задача. Вывести названия категорий, в которых средняя стоимость товаров выше 40 EUR**

```
SELECT Categories.CategoryName, AVG(Products.Price) AS avg_price FROM  
Products JOIN Categories ON Categories.CategoryID=Products.CategoryID  
GROUP BY Categories.CategoryID HAVING avg_price >= 40 ORDER BY avg_price  
DESC
```

## Работа с Git/GitHub

**Общий порядок (схема) работа в режиме с доступом на запись к репо**

1. Открыть имеющийся репо (или создать новый)
2. Добавить членов команды
3. Клонировать репо (скачать на компьютер)