

SQL

Зачем это всё?

Содержание

- CREATE TABLE, TRUNCATE, DROP
- INSERT INTO, UPDATE, DELETE
- SELECT
- GROUP BY, агрегации
- JOIN
- Оконные функции
- Транзакции

CREATE TABLE

```
CREATE TABLE Persons (  
    PersonID int,  
    LastName varchar(255),  
    FirstName varchar(255),  
    Address varchar(255),  
    City varchar(255)  
);
```

```
DROP TABLE Persons;
```

```
TRUNCATE TABLE Persons;
```

ALTER TABLE

- ALTER TABLE Customers ADD Email varchar(255);

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
89	White Clover Markets	Karl Jablonski	305 - 14th Ave. S. Suite 3B	Seattle	98128	USA
90	Wilman Kala	Matti Karttunen	Keskuskatu 45	Helsinki	21240	Finland
91	Wolski	Zbyszek	ul. Filtrowa 68	Walla	01-012	Poland
92	Cardinal	Tom B. Erichsen	Skagen 21	Stavanger	4006	Norway

INSERT INTO

- INSERT INTO
Customers
(CustomerName, ContactName, Address, City, PostalCode, Country)
VALUES
(
'Cardinal', 'Tom B. Erichsen', 'Skagen
21', 'Stavanger', '4006', 'Norway'),
(
'Cardinal', 'Tom B. Erichsen', 'Skagen
21', 'Stavanger', '4006', 'Norway');

UPDATE, DELETE

- UPDATE Customers
SET ContactName = 'Alfred Schmidt', City= 'Frankfurt'
WHERE CustomerID = 1;
- DELETE FROM Customers
WHERE CustomerID = 1;

SELECT

- `SELECT CustomerName, City FROM Customers;`
- `SELECT * FROM Customers;`
- `SELECT * FROM Customers ORDER BY CustomerID;`
- `SELECT * FROM Customers ORDER BY CustomerID DESC;`
- `SELECT * FROM Customers ORDER BY CustomerID LIMIT 10;`

SELECT + WHERE

- SELECT CustomerName, City FROM Customers
WHERE
Country = 'USA'
- SELECT CustomerName, City FROM Customers
WHERE
Country = 'USA' OR Country = 'Finland'
- SELECT CustomerName, City FROM Customers
WHERE
Country NOT IN ('USA', 'Finland')

SELECT + WHERE

- SELECT CustomerName, City FROM Customers
WHERE
Country LIKE 'U%'
- SELECT * FROM Students
WHERE
Age BETWEEN 3 AND 10

Вставляем результат одного запроса в другую таблицу

- INSERT INTO *table2*
SELECT * FROM *table1*
WHERE *condition*;
- INSERT INTO *table2* WITH TRUNCATE
SELECT * FROM *table1*
WHERE *condition*;

GROUP BY

- Считаем число заказов по городам:

```
SELECT city, COUNT(*) as orders  
FROM all_orders_table  
GROUP BY city
```

COUNT – агрегационная функция (комбинирует несколько значений в одно)

GROUP BY

- Считаем число заказов по городам:

```
SELECT city, COUNT(*) as orders  
FROM all_orders_table  
GROUP BY city
```

COUNT – агрегационная функция (комбинирует несколько значений в одно)

- Знаете еще такие функции?

GROUP BY

- Считаем число заказов по городам:

```
SELECT city, COUNT(*) as orders  
FROM all_orders_table  
GROUP BY city
```

COUNT – агрегационная функция (комбинирует несколько значений в одно)

- Знаете еще такие функции? MIN, MAX, AVG, SUM, *MEDIAN*

GROUP BY + HAVING

- Считаем число заказов по городам:

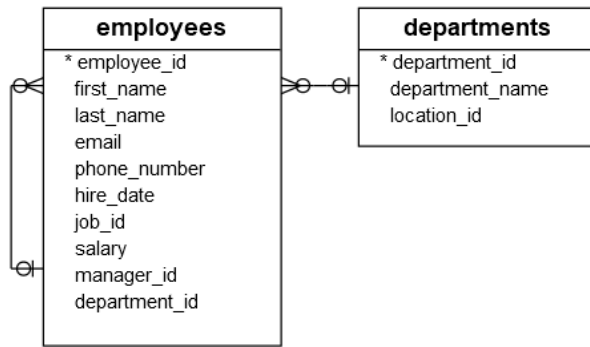
```
SELECT city, COUNT(*) as orders  
FROM all_orders_table  
GROUP BY city  
HAVING LENGTH(city) > 5
```


GROUP BY + HAVING

- Считаем число заказов по городам:

```
SELECT city, COUNT(DISTINCT user_id) as users  
FROM all_orders_table  
GROUP BY city  
HAVING LENGTH(city) > 5
```

Подзапросы



- Как посчитать среднюю зарплату по отделам (есть среднее средних)?

Подзапросы

SELECT

ROUND(AVG(average_salary), 0)

FROM

(SELECT AVG(salary) average_salary FROM employees
GROUP BY department_id) AS department_salary

JOIN

OrderID	CustomerID	OrderDate
10308	2	1996-09-18
10309	37	1996-09-19
10310	77	1996-09-20

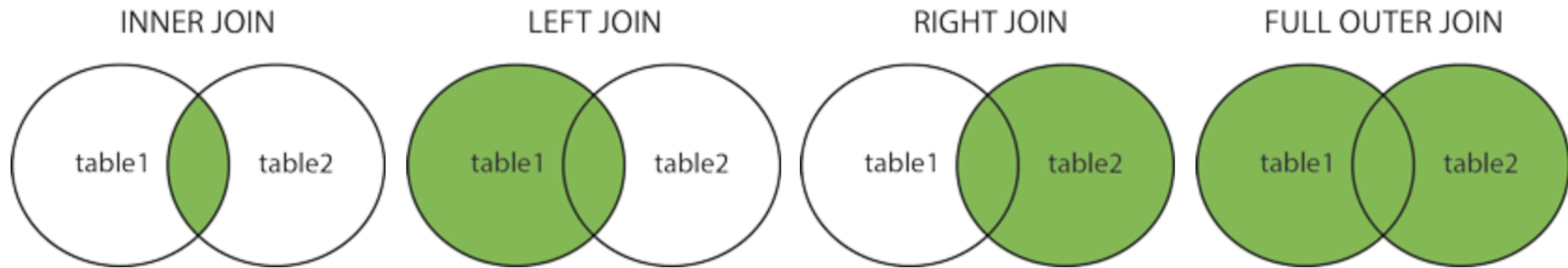
CustomerID	CustomerName	ContactName	Country
1	Alfreds Futterkiste	Maria Anders	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mexico

JOIN

- `SELECT Orders.OrderID, Customers.CustomerName, Orders.OrderDate
FROM Orders
INNER JOIN Customers
ON Orders.CustomerID = Customers.CustomerID;`

OrderID	CustomerName	OrderDate
10308	Ana Trujillo Emparedados y helados	9/18/1996
10365	Antonio Moreno Taquería	11/27/1996
10383	Around the Horn	12/16/1996
10355	Around the Horn	11/15/1996
10278	Berglunds snabbköp	8/12/1996

JOIN



Оконные функции

- Хотим посчитать среднее время между заказами для каждого пользователя
- Как это сделать?

Оконные функции

- Хотим посчитать среднее время между заказами для каждого пользователя
- Как это сделать?
- Нужно для каждой строки находить предыдущую

Оконные функции

```
select
  user_id,
  order_datetime,
  lag(order_datetime) as prev_order_datetime over (partition by
user_id order by order_datetime)
from orders
```

Что дальше?

- Практика на SQL – первое занятие в январе
- ORM – второе занятие в январе
- Контрольная – 18 января в 12:10
- Домашнее задание по SQL/ORM - ?

Ещё почитать

- <https://www.w3schools.com/sql/>
- <https://medium.com/better-programming/the-last-sql-guide-for-data-analysis-youll-ever-need-17ae10fa4a6f>
- <https://proglib.io/p/sql-for-20-minutes/>
- <https://www.postgresql.org/docs/9.1/tutorial-window.html>

И немного мотивации:

- <https://www.linkedin.com/pulse/why-you-should-learn-sql-brewster-knowlton/>
- <https://www.dataquest.io/blog/why-sql-is-the-most-important-language-to-learn/>