**Теория**:

1. <https://proselyte.net/tutorials/sql/>
2. <https://habr.com/post/305926/>
3. <https://habr.com/company/oleg-bunin/blog/348172/>

**Практика:**

**Задание 1**

**SCHEMA:**

Create table Person (PersonId int, FirstName varchar(255), LastName varchar(255))

Create table Address (AddressId int, PersonId int, City varchar(255), State varchar(255))

Truncate table Person

insert into Person (PersonId, LastName, FirstName) values ('1', 'Wang', 'Allen')

Truncate table Address

insert into Address (AddressId, PersonId, City, State) values ('1', '2', 'New York City', 'New York')

Write a SQL query for a report that provides the following information for each person in the Person table, regardless if there is an address for each of those people:

FirstName, LastName, City, State

SQL query requested:

SELECT FirstName, LastName, City, State FROM

Person P LEFT JOIN Address A ON P.PersonId = A.PersonId;

**Задание 2**

**SCHEMA:**

Create table If Not Exists Employee (Id int, Salary int)

Truncate table Employee

insert into Employee (Id, Salary) values ('1', '100')

insert into Employee (Id, Salary) values ('2', '200')

insert into Employee (Id, Salary) values ('3', '300')

Write a SQL query to get the second highest salary from the Employee table.

+----+--------+

| Id | Salary |

+----+--------+

| 1 | 100 |

| 2 | 200 |

| 3 | 300 |

+----+--------+

For example, given the above Employee table, the query should return 200 as the second highest salary. If there is no second highest salary, then the query should return null.

+---------------------+

| SecondHighestSalary |

+---------------------+

| 200 |

+---------------------+

SQL query requested:

SELECT IFNULL ((SELECT DISTINCT Salary FROM Employee ORDER BY Salary DESC LIMIT 1,1), NULL) AS "SecondHighestSalary";

**Задание 3**

**SCHEMA:**

Create table If Not Exists Employee (Id int, Name varchar(255), Salary int, ManagerId int)

Truncate table Employee

insert into Employee (Id, Name, Salary, ManagerId) values ('1', 'Joe', '70000', '3')

insert into Employee (Id, Name, Salary, ManagerId) values ('2', 'Henry', '80000', '4')

insert into Employee (Id, Name, Salary, ManagerId) values ('3', 'Sam', '60000', 'None')

insert into Employee (Id, Name, Salary, ManagerId) values ('4', 'Max', '90000', 'None')

The Employee table holds all employees including their managers. Every employee has an Id, and there is also a column for the manager Id.

+----+-------+--------+-----------+

| Id | Name | Salary | ManagerId |

+----+-------+--------+-----------+

| 1 | Joe | 70000 | 3 |

| 2 | Henry | 80000 | 4 |

| 3 | Sam | 60000 | NULL |

| 4 | Max | 90000 | NULL |

+----+-------+--------+-----------+

Given the Employee table, write a SQL query that finds out employees who earn more than their managers. For the above table, Joe is the only employee who earns more than his manager.

+----------+

| Employee |

+----------+

| Joe |

+----------+

SQL query requested:

SELECT a.Name AS Employee

FROM Employee a, Employee b

WHERE a.Salary>b.Salary AND b.Id = a.ManagerId;

**Задание 4**

**SCHEMA:**

Create table If Not Exists Person (Id int, Email varchar(255))

Truncate table Person

insert into Person (Id, Email) values ('1', 'a@b.com')

insert into Person (Id, Email) values ('2', 'c@d.com')

insert into Person (Id, Email) values ('3', 'a@b.com')

Write a SQL query to find all duplicate emails in a table named Person.

+----+---------+

| Id | Email |

+----+---------+

| 1 | a@b.com |

| 2 | c@d.com |

| 3 | a@b.com |

+----+---------+

For example, your query should return the following for the above table:

+---------+

| Email |

+---------+

| a@b.com |

+---------+

Note: All emails are in lowercase.

SQL query requested:

SELECT Email FROM Person

GROUP BY Email

HAVING COUNT(Email)>=2;

**Задание 5**

**SCHEMA:**

Create table If Not Exists Customers (Id int, Name varchar(255))

Create table If Not Exists Orders (Id int, CustomerId int)

Truncate table Customers

insert into Customers (Id, Name) values ('1', 'Joe')

insert into Customers (Id, Name) values ('2', 'Henry')

insert into Customers (Id, Name) values ('3', 'Sam')

insert into Customers (Id, Name) values ('4', 'Max')

Truncate table Orders

insert into Orders (Id, CustomerId) values ('1', '3')

insert into Orders (Id, CustomerId) values ('2', '1')

Suppose that a website contains two tables, the Customers table and the Orders table. Write a SQL query to find all customers who never order anything.

Table: Customers.

+----+-------+

| Id | Name |

+----+-------+

| 1 | Joe |

| 2 | Henry |

| 3 | Sam |

| 4 | Max |

+----+-------+

Table: Orders.

+----+------------+

| Id | CustomerId |

+----+------------+

| 1 | 3 |

| 2 | 1 |

+----+------------+

Using the above tables as example, return the following:

+-----------+

| Customers |

+-----------+

| Henry |

| Max |

+-----------+

SQL query requested:

SELECT DISTINCT Name FROM Customers

WHERE Id NOT IN (SELECT DISTINCT CustomerId FROM Orders);

**Задание 6**

**SCHEMA:**

Create table If Not Exists courses (student varchar(255), class varchar(255))

Truncate table courses

insert into courses (student, class) values ('A', 'Math')

insert into courses (student, class) values ('B', 'English')

insert into courses (student, class) values ('C', 'Math')

insert into courses (student, class) values ('D', 'Biology')

insert into courses (student, class) values ('E', 'Math')

insert into courses (student, class) values ('F', 'Computer')

insert into courses (student, class) values ('G', 'Math')

insert into courses (student, class) values ('H', 'Math')

insert into courses (student, class) values ('I', 'Math')

There is a table courses with columns: student and class

Please list out all classes which have more than or equal to 5 students.

For example, the table:

+---------+------------+

| student | class |

+---------+------------+

| A | Math |

| B | English |

| C | Math |

| D | Biology |

| E | Math |

| F | Computer |

| G | Math |

| H | Math |

| I | Math |

+---------+------------+

Should output:

+---------+

| class |

+---------+

| Math |

+---------+

SQL query requested:

SELECT class FROM courses

GROUP BY class

HAVING COUNT(class)>=5;