

SQL, агрегация данных.

Существует 5 таблиц:

users (*user_id*, *user_email*, *subject_id*)
teacher (*teacher_id*, *email*, *group_id*)
student (*user_id*, *level*, *education_form*, *subject_id*)
subject (*subject_id*, *subject_title*)
group_student (*user_id*, *group_id*)

Задачи:

1. Найдите общее количество индивидуально обучающихся студентов для каждого уровня.

Решение:

```
SELECT level, education_form, count (*) FROM student  
WHERE education_form = 'personal' GROUP BY level, education_form;
```

2. Найдите общее количество учащихся школы с уровнем Advanced для каждой формы обучения

Решение:

```
SELECT education_form, level, count (*) FROM student  
WHERE level = 'Advanced' GROUP BY education_form, level;
```

3. Найдите общее количество учащихся школы в разрезе уровня и формы обучения, отсортированных по уровню в порядке возрастания и по форме обучения в порядке убывания.

Решение:

```
SELECT level, education_form, count (*) FROM student  
GROUP BY level, education_form ORDER BY level, education_form desc;
```

4. Найдите максимальное и минимальное id группы (одним запросом).

Решение:

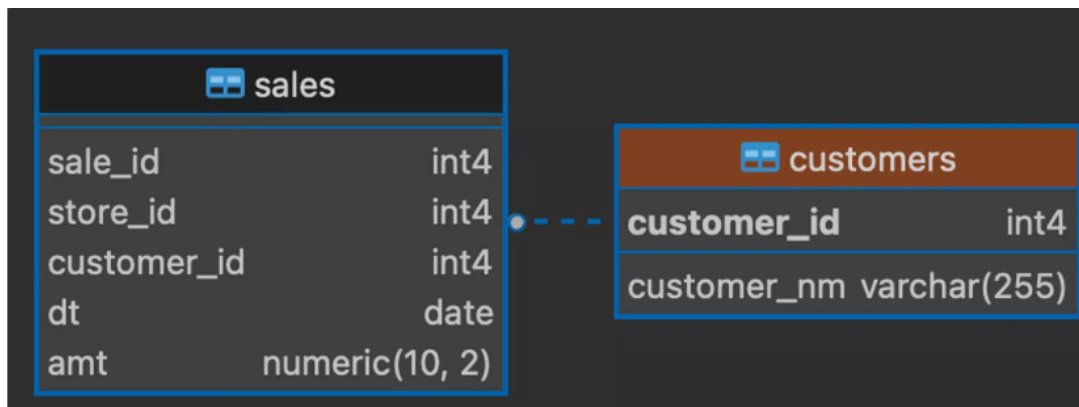
```
SELECT MIN (group_id), MAX (group_id) FROM group_student;
```

5. Найдите количество учеников и id группы, в которой наибольшее количество учеников.

Решение:

```
SELECT group_id, count (*) FROM group_student  
GROUP BY group_id ORDER BY count (*) desc LIMIT 1;
```

6. Дана схема БД:



Создайте и заполните таблицы.

Решение:

Создадим 2 таблицы sales и customers.

Sales (покупки)

sale_id	store_id	customer_id	dt	amt
11	3	5	2023.03.05	3587,87
12	9	2	2023.03.11	5712,93
13	3	4	2023.03.19	1684,14
14	7	2	2023.03.22	2944,51
15	2	1	2023.03.25	4351,68
16	3	3	2023.03.27	6774,38

Customers (Покупатели)

customer_id	customer_nm
1	Елена
2	Егор
3	Елена
4	Ксения
5	Николай

```

CREATE TABLE sales (
  sale_id      INTEGER,
  store_id     INTEGER,
  customer_id  INTEGER,
  dt           DATE,
  amt          NUMERIC (10, 2));
  
```

```
INSERT INTO sales (sale_id, store_id, customer_id, dt, amt)
VALUES
(11, 3, 5, '2023.03.05', 3587.87),
(12, 9, 2, '2023.03.11', 5712.93),
(13, 3, 4, '2023.03.19', 1684.14),
(14, 7, 2, '2023.03.22', 2944.51),
(15, 2, 1, '2023.03.25', 4351.68),
(16, 3, 3, '2023.03.27', 6774.38);
```

```
CREATE TABLE customers (
    customer_id    INTEGER,
    customer_nm    VARCHAR (255));
INSERT INTO customers (customer_id, customer_nm)
VALUES
(1, 'Елена'),
(2, 'Егор'),
(3, 'Елена'),
(4, 'Ксения'),
(5, 'Николай');
```

7. Выведите максимальную сумму покупки.

Решение:

```
SELECT MAX (amt) FROM sales;
```

8. Выведите минимальную дату покупки.

Решение:

```
SELECT MIN (dt) FROM sales;
```

9. Выведите среднюю сумму покупок для customer_id = 1.

Решение:

```
SELECT AVG (amt) FROM sales WHERE customer_id = 1;
```

10. Выведите минимальную и максимальную сумму покупки, совершенную в магазине с store_id = 3 или покупателем с customer_id = 2.

Решение:

```
SELECT MIN (amt), MAX (amt) FROM sales
WHERE store_id = 3 OR customer_id = 2;
```

11. Выведите количество уникальных имен покупателей.

Решение:

```
SELECT count (DISTINCT customer_nm) FROM customers;
```