

Основное задание

- SQL

1. Вывести отсортированный по количеству перелетов (по убыванию) и имени (по возрастанию) список пассажиров, совершивших хотя бы 1 полет.

```
SELECT name,  
  
        COUNT(*) AS count  
  
FROM Passenger, Pass_in_trip  
  
WHERE Pass_in_trip.passenger=Passenger.id  
  
GROUP BY Passenger.id  
  
ORDER BY COUNT (*) DESC, name ASC
```

2. Сколько времени обучающийся будет находиться в школе, учась со 2-го по 4-ый уч. предмет ?

По id учащегося можно определить его класс, далее с помощью id класса можно получить информацию из расписания, а дальше по расписанию рассчитывать время при помощи start_pair и end_pair.

Или же попробовать воспользоваться функцией timediff(), куда в качестве аргументов передать пары.

3. Выведите список комнат, которые были зарезервированы в течение 12 недели 2020 года.

```
SELECT *  
  
from Rooms  
  
        Inner Join (Select room_id, start_date,  
end_date from Reservations) as res  
  
                on res.room_id = Rooms.id and  
WEEK(res.start_date) >= 11 and WEEK(res.end_date) = 12
```

4. Какой(ие) кабинет(ы) пользуются самым большим спросом?

```
SELECT classroom
```

```
from (SELECT classroom, count(classroom) as cnt FROM
Schedule GROUP BY classroom) as classrooms

WHERE cnt =

      (SELECT MAX(cnt) FROM (SELECT classroom,
count(classroom) as cnt FROM Schedule GROUP BY classroom)
as classrooms)
```

5. -

6. -

Дополнительное задание

Основы Big Data

Код:

```
def mp(doc):
    for word in doc:
        yield word, 1

def shuffle(mapped_values):
    dict = {}
    for word, count in mapped_values:
        if (word in dict):
            dict[word].append(count)
        else:
            dict[word] = [count]
    return dict

def reduce(word, values):
    yield word, sum(values)
```

В репозитории приложен файл [map-reduce.py](#)

Основы баз данных

1. Перечисли способы получить количество записей в таблице

- Count()
- Через какой-нибудь графический интерфейс. Например pgAdmin
- Через orm с помощью свойства size (например)

2. Выполнится ли этот запрос:

```
SELECT
    order_id,
    order_code,
    SUM(order_value)
FROM
    orders
GROUP BY
    order_id
```

Во-первых, все будет зависеть от самой базы данных. Во вторых, от типов данных в таблице.

Например, используя SQLite и таблицу с данными, при которых данный запрос должен не сработать, он, как ни странно, сработает и выведет определенный результат.

Другая же ситуация обстоит с использованием, допустим, PostgreSQL. При наличии несопоставимых с запросом данных его нельзя будет осуществить.

Приведу пример. У нас есть таблица:

```
CREATE TABLE orders(
    order_id INT,
    order_code INT,
    order_value DATE
)
```

Осуществляя наш запрос, мы не сможем получить ответ, так как у даты нет функции суммы.

В-третьих, сам запрос неверный и требует наличия поля `order_code` в функции GROUP-BY.

Основы программирования

Решение находится в файле [array_task.py](#)