

Домашнее задание:

Изучите запрос, который выводит средний возраст студентов для каждого курса с типом "DESIGN":

```
SELECT c.name AS course_name, ROUND(AVG(s.age)) AS average_age
FROM courses c
JOIN subscriptions sub ON c.id = sub.course_id
JOIN students s ON sub.student_id = s.id
WHERE c.type = 'DESIGN'
GROUP BY c.name;
```

Задание: используя возможности языка программирования **Java**, выполните его без участия **SQL** функций, подобно тому примеру, что был на занятии

Запрос должен быть таким:

```
String query = "SELECT c.name AS course_name, c.type AS type, s.age " +
               "FROM courses c " +
               "JOIN subscriptions sub ON c.id = sub.course_id " +
               "JOIN students s ON sub.student_id = s.id";
```

но результат должен соответствовать. Успехов 😊

Скриншот работы с занятия:

```

package org.example;

import java.sql.*;
import java.util.HashMap;
import java.util.Map;

public class SubscriptionsStatistics {
    public static void main(String[] args) {
        // Параметры подключения к базе данных
        String url = "jdbc:mysql://localhost:3306/top"; // URL базы данных
        String user = "root"; // Логин пользователя
        String pass = "sHterBen44"; // Пароль пользователя

        try (Connection connection = DriverManager.getConnection(url, user, pass)) {
            // SQL-запрос для выбора данных о подписках
            String sql = "SELECT course_name, subscription_date FROM purchaselist";
            // Подготовленный SQL-запрос
            PreparedStatement preparedStatement = connection.prepareStatement(sql);
            // Выполнение запроса и получение результата
            ResultSet resultSet = preparedStatement.executeQuery();

            // Создание карты для хранения количества покупок на каждый курс
            Map<String, Integer> courseSubscriptionsCount = new HashMap<>();
            // Инициализация карты для хранения максимального месяца покупки для каждого
            // курса
            Map<String, Integer> courseSubscriptionMonths = new HashMap<>();

            // Счетчик курсов
            int courseCounter = 1;

            // Цикл для обхода результатов запроса
            while (resultSet.next()) {
                // Получение имени курса и даты подписки из результата запроса
                String courseName = resultSet.getString("course_name");
                String subscriptionDate = resultSet.getString("subscription_date");

                // Извлечение года и месяца из даты подписки
                int year = Integer.parseInt(subscriptionDate.substring(0, 4));
                int month = Integer.parseInt(subscriptionDate.substring(5, 7));

                // Проверка, что подписка была в 2018 году
                if (year == 2018) {
                    // Увеличение счетчика покупок для данного курса
                    courseSubscriptionsCount.put(courseName,
                        courseSubscriptionsCount.getOrDefault(courseName, 0) + 1);

                    // Запись максимального месяца покупки для данного курса
                    courseSubscriptionMonths.put(courseName,
                        Math.max(courseSubscriptionMonths.getOrDefault(courseName,
0), month));
                }
            }

            // Цикл для вывода результатов
            for (String courseName : courseSubscriptionsCount.keySet()) {
                int totalSubscriptions = courseSubscriptionsCount.get(courseName); //
Общее количество покупок
                int subscriptionMonths = courseSubscriptionMonths.get(courseName); //
Количество месяцев, за которые совершались покупки
                double averageSubscriptions = (double) totalSubscriptions /
subscriptionMonths; // Вычисление среднего количества покупок на курс за месяц

                // Округляем до 2 знаков после запятой
                String formattedAverage = String.format("%.2f", averageSubscriptions);

                // Вывод информации о курсе и округленном среднем количестве покупок
                System.out.println("Course " + courseCounter + ": " + courseName + ",

```

```
Average Subscriptions: " + formattedAverage);  
        courseCounter++;  
    }  
} catch (SQLException e) {  
    e.printStackTrace();  
}  
}  
}
```