

Санкт-Петербургский Политехнический Университет Петра Великого

Институт компьютерных наук и технологий

Кафедра компьютерных систем и программных технологий

ОТЧЕТ
по лабораторной работе

«Язык SQL-DDL»

Базы данных

Работу выполнил студент

группа 43501/3 Дьячков В.В.

Работу принял преподаватель

_____ Мясов А.В.

Санкт-Петербург

2018

Содержание

| | | |
|----------|--|----------|
| 1 | Цель работы | 3 |
| 2 | Программа работы | 3 |
| 3 | Теоретическая информация | 3 |
| 4 | Стандартные запросы | 3 |
| 4.1 | Выборка всех данных | 3 |
| 4.2 | Выборка с использованием логических операций | 3 |
| 4.3 | Запрос с вычисляемым полем | 3 |
| 4.4 | Выборка с использованием сортировки | 3 |
| 4.5 | Запрос с вычислением совокупных характеристик таблиц | 3 |
| 4.6 | Выборка из связанных таблиц | 4 |
| 4.7 | Запрос с использованием группировки | 4 |
| 4.8 | Вложенный запрос | 4 |
| 4.9 | Вставка записей | 4 |
| 4.10 | Изменение записей | 4 |
| 4.11 | Удаление записей по условию | 4 |
| 4.12 | Удаление с использованием вложенного запроса | 4 |
| 5 | Запросы в соответствии с заданием преподавателя | 4 |
| 5.1 | Рейтинг городов по кварталам | 4 |
| 5.2 | Клиенты, имеющие наибольший средний рост стоимости путевки | 5 |
| 6 | Выводы | 6 |

1. Цель работы

Познакомиться с языком создания запросов управления данными SQL-DML.

2. Программа работы

1. Изучение SQL-DML.
2. Выполнение всех запросов из списка стандартных запросов. Демонстрация результатов преподавателю.
3. Получение у преподавателя и реализация SQL-запросов в соответствии с индивидуальным заданием. Демонстрация результатов преподавателю.
4. Сохранение в БД выполненных запросов SELECT в виде представлений, запросов INSERT, UPDATE или DELETE – в виде ХП. Выкладывание скрипта в GitLab.

3. Теоретическая информация

4. Стандартные запросы

4.1. Выборка всех данных

Сделайте выборку всех данных из каждой таблицы

4.2. Выборка с использованием логических операций

Сделайте выборку данных из одной таблицы при нескольких условиях, с использованием логических операций, LIKE, BETWEEN, IN (не менее 3-х разных примеров)

4.3. Запрос с вычисляемым полем

Создайте в запросе вычисляемое поле

4.4. Выборка с использованием сортировки

Сделайте выборку всех данных с сортировкой по нескольким полям

4.5. Запрос с вычислением совокупных характеристик таблиц

Создайте запрос, вычисляющий несколько совокупных характеристик таблиц

4.6. Выборка из связанных таблиц

Сделайте выборку данных из связанных таблиц (не менее двух примеров)

4.7. Запрос с использованием группировки

Создайте запрос, рассчитывающий совокупную характеристику с использованием группировки, наложите ограничение на результат группировки

4.8. Вложенный запрос

Придумайте и реализуйте пример использования вложенного запроса

4.9. Вставка записей

С помощью оператора INSERT добавьте в каждую таблицу по одной записи

4.10. Изменение записей

С помощью оператора UPDATE измените значения нескольких полей у всех записей, отвечающих заданному условию

4.11. Удаление записей по условию

С помощью оператора DELETE удалите запись, имеющую максимальное (минимальное) значение некоторой совокупной характеристики

4.12. Удаление с использованием вложенного запроса

С помощью оператора DELETE удалите записи в главной таблице, на которые не ссылается подчиненная таблица (используя вложенный запрос)

5. Запросы в соответствии с заданием преподавателя

5.1. Рейтинг городов по кварталам

Задание: Вывести рейтинг городов по кварталам. В рейтинге 5 городов, которые больше всего ездят в каком-то квартале.

```
1 WITH summary AS (  
2     WITH quarter_summary AS (  
3         SELECT city.id AS city_id,  
4             EXTRACT(QUARTER FROM "from") AS quarter,  
5             COUNT(*) AS count  
6     FROM reservation
```

```

7      JOIN room ON room.id = reservation.room_id
8      JOIN room_type ON room_type.id = room.room_type_id
9      JOIN hotel ON hotel.id = room_type.hotel_id
10     JOIN city ON hotel.city_id = city.id
11     GROUP BY city.id, quarter
12 )
13 SELECT ROW_NUMBER() OVER (PARTITION BY quarter ORDER BY SUM(count) DESC) AS row_num,
14        city_id AS city_id,
15        quarter AS quarter,
16        SUM(count) AS reservations
17 FROM quarter_summary
18 GROUP BY quarter, city_id
19 )
20 SELECT summary.quarter AS quarter,
21        city.name AS city_name,
22        country.name AS country_name,
23        summary.reservations AS reservations
24 FROM summary
25     JOIN city ON summary.city_id = city.id
26     JOIN country ON city.country_id = country.id
27 WHERE summary.row_num < 2
28 ORDER BY summary.quarter, summary.reservations DESC;

```

Листинг 1: quarter.sql

5.2. Клиенты, имеющие наибольший средний рост стоимости путевки

Задание: Вывести 5 клиентов, которые имеют наибольший средний рост стоимости путевки.

```

1 WITH summary AS (
2     WITH increase_summary AS (
3         SELECT ROW_NUMBER() OVER (PARTITION BY user_id ORDER BY "from") AS row_num,
4                price AS price,
5                user_id AS user_id
6         FROM reservation
7         WHERE is_paid AND user_id IN (
8             SELECT user_id
9             FROM reservation
10            WHERE is_paid
11            GROUP BY user_id
12            HAVING COUNT(*) > 10
13        )
14    )
15    SELECT curr.user_id AS user_id,
16           AVG((curr.price - prev.price) :: numeric) :: money AS avg_diff
17    FROM increase_summary curr
18         JOIN increase_summary prev
19         ON curr.user_id = prev.user_id AND curr.row_num = prev.row_num + 1
20    GROUP BY curr.user_id
21    ORDER BY avg_diff DESC
22    LIMIT 5
23 )
24 SELECT usr.id, usr.name, usr.email, usr.phone_number, reservations, total, avg_diff
25 FROM summary
26     JOIN "user" AS usr ON user_id = usr.id
27     JOIN (

```

```

28         SELECT user_id    AS user_id,
29                COUNT(*)    AS reservations,
30                SUM(price)  AS total
31        FROM reservation
32        GROUP BY user_id
33    ) AS total ON total.user_id = usr.id;
34
35 SELECT *
36 FROM reservation
37 WHERE user_id = 578 AND is_paid
38 ORDER BY "from";

```

Листинг 2: travellers.sql

6. Выводы