Лабораторная работа №4

Запросы на выборку и модификацию данных. Представления. Работа с индексами

1. Цель работы: овладеть практическими навыками создания представлений и запросов на выборку данных к базе данных PostgreSQL, использования подзапросов при модификации данных и индексов.

2. Програмное обеспечение: СУБД PostgreSQL, pgAdmin 4.

3. Практическое задание:

1. Создать запросы и представления на выборку данных к базе данных PostgreSQL (согласно индивидуальному заданию, часть 2 и 3).
2. Составить 3 запроса на модификацию данных (INSERT, UPDATE, DELETE) **с использованием подзапросов**.
3. Изучить графическое представление запросов и просмотреть историю запросов.
4. Создать простой и составной индексы для двух произвольных запросов и сравнить время выполнения запросов без индексов и с индексами. Для получения плана запроса использовать команду EXPLAIN.

Выполнение заданий

Перед выполнением заданий, база данных была очищена и обновлена в соответствии с запросом:

Очистка:

```
TRUNCATE TABLE public.cleaning RESTART IDENTITY CASCADE;
TRUNCATE TABLE public.client RESTART IDENTITY CASCADE;
TRUNCATE TABLE public.cost_room_class RESTART IDENTITY CASCADE;
TRUNCATE TABLE public.hotel RESTART IDENTITY CASCADE;
TRUNCATE TABLE public.promotion RESTART IDENTITY CASCADE;
TRUNCATE TABLE public.registration RESTART IDENTITY CASCADE;
TRUNCATE TABLE public.room RESTART IDENTITY CASCADE;
TRUNCATE TABLE public.room_class RESTART IDENTITY CASCADE;
TRUNCATE TABLE public.workers RESTART IDENTITY CASCADE;
```

Заполнение:

```
    -- Заполнение таблицы hotel
INSERT INTO hotel (name, address, rating)
SELECT
  'Hotel ' || s,
  'Address ' || s,
  (s % 5) + 1
FROM generate_series(1, 100) s;

-- Заполнение таблицы room_class
INSERT INTO room_class (person_amount, class, room_left,
   isable_refund)
SELECT
  (s % 4) + 1,
  'Class ' || s,
  (s % 10) + 1,
  s % 2 = 0
FROM generate_series(1, 10) s;

-- Заполнение таблицы room
INSERT INTO room (ID_hotel, ID_class, room_num, is_free, is_clean)
SELECT
  (s % 100) + 1,
  (s % 10) + 1,
  s,
  s % 2 = 0,
  s % 2 = 0
FROM generate_series(1, 1000) s;

-- Заполнение таблицы workers
INSERT INTO workers (ID_contract, start_contract, end_contract,
   extras, name, job)
```

```
SELECT
  s,
  '2024-01-01',
  '2024-12-31',
  'Extra ' || s,
  'Worker ' || s,
  'Job ' || s
FROM generate_series(1, 200) s;

-- Заполнение таблицы cleaning
INSERT INTO cleaning (ID_room, scheduled_date, is_done, ID_executor)
SELECT
  (s % 1000) + 1,
  '2024-01-01'::date + (s % 30),
  s % 2 = 0,
  (s % 200) + 1
FROM generate_series(1, 3000) s;

-- Заполнение таблицы client
INSERT INTO client (First_Name, Last_Name, Passport_id, phone, email
    , residence_address)
SELECT
  'FirstName ' || s,
  'LastName ' || s,
  s,
  1000000000 + s,
  'client' || s || '@example.com',
  'Address ' || s
FROM generate_series(1, 500) s;

-- Заполнение таблицы registration
INSERT INTO registration (ID_Client, ID_room, ID_executor,
    is_archived, residence_address, booking_date, arrival_date,
    departure_date, payment_method, reg_status, payment_status)
SELECT
  (s % 500) + 1,
  (s % 1000) + 1,
  (s % 200) + 1,
  s % 2 = 0,
  'Address ' || s,
  '2024-01-01'::date + (s % 30),
  '2024-01-02'::date + (s % 30),
  '2024-01-03'::date + (s % 30),
  CASE WHEN s % 2 = 0 THEN 'card' ELSE 'cash' END,
  'забронирован',
  s % 2 = 0
```

```
FROM generate_series(1, 1000) s;

-- Заполнение таблицы cost_room_class
INSERT INTO cost_room_class (date_start, date_end, cost_per_day,
    ID_class)
SELECT
  '2024-01-01',
  '2024-12-31',
  (s % 100) + 1,
  (s % 10) + 1
FROM generate_series(1, 20) s;

-- Заполнение таблицы promotion
INSERT INTO promotion (ID_class, condition, date_start, date_end,
    prom_cost_per_day, amount)
SELECT
  (s % 10) + 1,
  'Condition ' || s,
  '2024-01-01',
  '2024-12-31',
  (s % 100) + 1,
  (s % 100) + 1
FROM generate_series(1, 50) s;
```

1. Создание запросов и представлений на выборку данных к БД

Запросы:
1)список всех 2-местных номеров отелей, с ценой менее 200 т.р., данные упорядочены в порядке уменьшения стоимости

Запрос:

```
SELECT r.room_num, h.name, crc.cost_per_day
FROM room r
JOIN hotel h ON r.ID_hotel = h.ID_hotel
JOIN cost_room_class crc ON r.ID_class = crc.ID_class
WHERE crc.cost_per_day < 200 AND
      (SELECT rc.person_amount FROM room_class rc WHERE rc.ID_class
    = r.ID_class) = 2
ORDER BY crc.cost_per_day DESC;
```

Ответ:

594	144	Hotel 45	5.00
595	564	Hotel 65	5.00
596	34	Hotel 35	5.00
597	504	Hotel 5	5.00
598	44	Hotel 45	5.00
599	224	Hotel 25	5.00
600	454	Hotel 55	5.00

2)Все записи регистрации постояльцев, которые выехали из отелей в течение двух последних недель.

Запрос:

```
SELECT *
FROM registration
WHERE departure_date > CURRENT_DATE - INTERVAL '2 weeks';
```

Ответ:

653	982	483	983	183	true	Address 982
654	983	484	984	184	false	Address 983
655	984	485	985	185	true	Address 984
656	985	486	986	186	false	Address 985
657	986	487	987	187	true	Address 986
658	987	488	988	188	false	Address 987
659	988	489	989	189	true	Address 988

3)Общий суточный доход каждого отеля за последний месяц:

```
SELECT h.ID_hotel, h.name, SUM(crc.cost_per_day) AS total_income
FROM registration reg
JOIN room r ON reg.ID_room = r.ID_room
JOIN hotel h ON r.ID_hotel = h.ID_hotel
JOIN cost_room_class crc ON r.ID_class = crc.ID_class
WHERE reg.arrival_date > CURRENT_DATE - INTERVAL '1 month'
GROUP BY h.ID_hotel;
```

Ответ:

	id_hotel [PK] integer	name character varying (255)	total_income numeric
91	91	Hotel 91	920.00
92	92	Hotel 92	140.00
93	93	Hotel 93	160.00
94	94	Hotel 94	180.00
95	95	Hotel 95	200.00
96	96	Hotel 96	220.00
97	97	Hotel 97	240.00
98	98	Hotel 98	260.00
99	99	Hotel 99	280.00
100	100	Hotel 100	300.00

4)Список свободных номеров одного из отелей на текущий день:

```sql
-- Например, для отеля с ID = 1
SELECT r.room_num
FROM room r
WHERE r.is_free = true AND r.ID_hotel = 1;
```

Ответ:

	room_num integer
1	100
2	200
3	300
4	400
5	500
6	600
7	700
8	800
9	900
10	1000

5)Общие потери от незанятых номеров за текущий день по всей сети:

```sql
SELECT SUM(crc.cost_per_day) AS total_loss
FROM room r
JOIN cost_room_class crc ON r.ID_class = crc.ID_class
WHERE r.is_free = true;
```

Ответ:

	total_loss numeric
1	18000.00

6)Отель с наибольшим количеством незанятых номеров на текущие сутки:

```sql
WITH RoomCounts AS (
  SELECT h.ID_hotel, h.name, COUNT(*) AS unoccupied_rooms
  FROM room r
  JOIN hotel h ON r.ID_hotel = h.ID_hotel
  WHERE r.is_free = true
  GROUP BY h.ID_hotel
),
MaxRooms AS (
  SELECT MAX(unoccupied_rooms) AS max_unoccupied
  FROM RoomCounts
)
SELECT rc.name, rc.unoccupied_rooms
FROM RoomCounts rc
JOIN MaxRooms mr ON rc.unoccupied_rooms = mr.max_unoccupied
ORDER BY rc.unoccupied_rooms DESC;
```

Ответ:

	name character varying (255)	unoccupied_rooms bigint
1	Hotel 87	10

7)Самый популярный тип номеров за последний год:

```sql
WITH BookingCounts AS (
  SELECT rc.ID_class, rc.class, COUNT(*) AS total_bookings
  FROM registration reg
  JOIN room r ON reg.ID_room = r.ID_room
  JOIN room_class rc ON r.ID_class = rc.ID_class
  WHERE reg.arrival_date > CURRENT_DATE - INTERVAL '1 year'
  GROUP BY rc.ID_class
),
MaxBookings AS (
```

```
  SELECT MAX(total_bookings) AS max_bookings
  FROM BookingCounts
)
SELECT bc.class, bc.total_bookings
FROM BookingCounts bc
JOIN MaxBookings mb ON bc.total_bookings = mb.max_bookings
ORDER BY rc.ID_class;
```

Ответ:

	class character varying (18)	total_bookings bigint
1	Class 8	100

Представления:
1)Представление для Турагентов (Поиск Свободных Номеров в Отелях)

```
CREATE VIEW available_rooms AS
SELECT h.name AS hotel_name, r.room_num, rc.class
FROM room r
JOIN hotel h ON r.ID_hotel = h.ID_hotel
JOIN room_class rc ON r.ID_class = rc.ID_class
WHERE r.is_free = true;
```

2)Представление для Владельца Компании (Информация о Доходах Каждого Отеля в Сети за Прошедший Месяц)

```
CREATE VIEW monthly_hotel_revenues AS
SELECT h.name AS hotel_name, SUM(crc.cost_per_day) AS total_revenue
FROM registration reg
JOIN room r ON reg.ID_room = r.ID_room
JOIN hotel h ON r.ID_hotel = h.ID_hotel
JOIN cost_room_class crc ON r.ID_class = crc.ID_class
WHERE reg.departure_date > CURRENT_DATE - INTERVAL '1 month'
GROUP BY h.name;
```

2. Составление 3-х запросов на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов

1)Уборка для номеров, из которых клиенты выехали в последний месяц.

```
INSERT INTO cleaning (ID_room, scheduled_date, is_done, ID_executor)
SELECT r.ID_room, CURRENT_DATE, false, (SELECT MIN(ID_executor) FROM
    workers)
FROM registration reg
JOIN room r ON reg.ID_room = r.ID_room
WHERE reg.departure_date > CURRENT_DATE - INTERVAL '1 month'
AND NOT EXISTS (
```

```
    SELECT 1 FROM cleaning c WHERE c.ID_room = reg.ID_room AND c.
    scheduled_date = CURRENT_DATE
)
GROUP BY r.ID_room;
```

2)Увеличение стоимости за день для всех номеров класса 'Class 1', которые сейчас заняты:

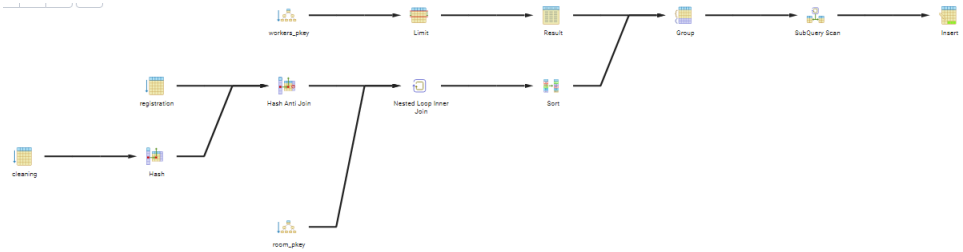
```
UPDATE cost_room_class
SET cost_per_day = cost_per_day + 10
WHERE ID_class = (
    SELECT ID_class
    FROM room_class
    WHERE class = 'Class 1'
);
```

3)Удаление записей об уборках, которые зарегистрированы неделю назад:

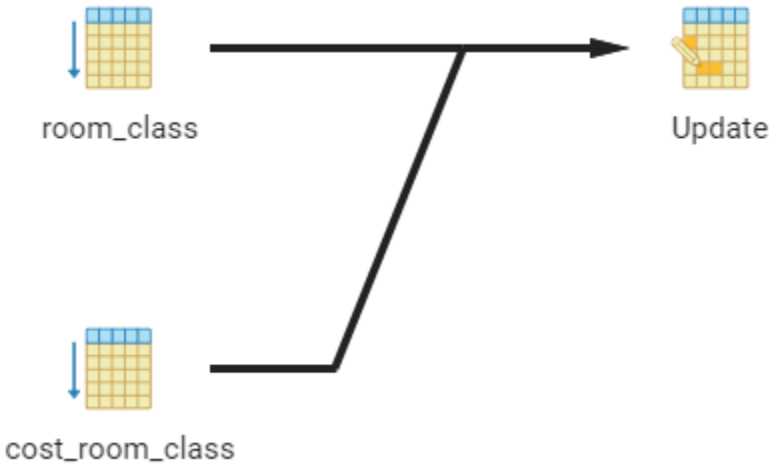
```
DELETE FROM cleaning
WHERE scheduled_date = CURRENT_DATE - INTERVAL '1 week';
```

3. Изучение графического представления запросов и просмотр истории запросов

1)INSERT



2)UPDATE



3)DELETE

4. Создание простого и составного индексов для двух произвольных запросов и сравнение времени выполнения запросов без индексов и с индексами с использованием EXPLAIN.

Простой индекс:
Запрос без индекса:

```
EXPLAIN SELECT * FROM room WHERE is_free = true;
```

Время выполнения: 78 msec

Создание простого индекса:

```
CREATE INDEX idx_room_is_free ON room(is_free);
```

Запрос с индексом:

```
    EXPLAIN SELECT * FROM room WHERE is_free = true;
```

Ответ: Время выполнения: 58 msec

Составной индекс:
Запрос без индекса:

```
EXPLAIN SELECT r.*, rc.class
FROM room r
JOIN room_class rc ON r.ID_class = rc.ID_class
WHERE r.is_free = true
AND rc.class = 'Class 1';
```

Время выполнения: 63 msec
Создание двойного индекса:

```
CREATE INDEX idx_room_class_free ON room(is_free) INCLUDE (ID_class)
   ;
CREATE INDEX idx_room_class_class ON room_class(class);
```

Запрос с индексом:

```
EXPLAIN SELECT r.*, rc.class
FROM room r
JOIN room_class rc ON r.ID_class = rc.ID_class
WHERE r.is_free = true
AND rc.class = 'Class 1';
```

Время выполнения: 49 msec

однако, запрос с составным индексом прироста в скорости не дает:

```
    CREATE INDEX idx_room_is_free_class ON room(is_free, ID_class);
```

время выполнение того же запроса увеличивается до 99 msec