

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное
учреждение высшего образования
“Национальный исследовательский университет ИТМО”

Факультет инфокоммуникационных технологий

ЛАБОРАТОРНАЯ РАБОТА №2

**ЗАПРОСЫ НА ВЫБОРКУ И МОДИФИКАЦИЮ ДАННЫХ,
ПРЕДСТАВЛЕНИЯ И ИНДЕКСЫ В POSTGRESQL.**

по дисциплине:

«Проектирование и реализация баз данных»

Выполнил студент:

Алексеев Павел Алексеевич

Группа №K3243

Проверила:

Говорова Марина Михайловна

Санкт-Петербург
2022

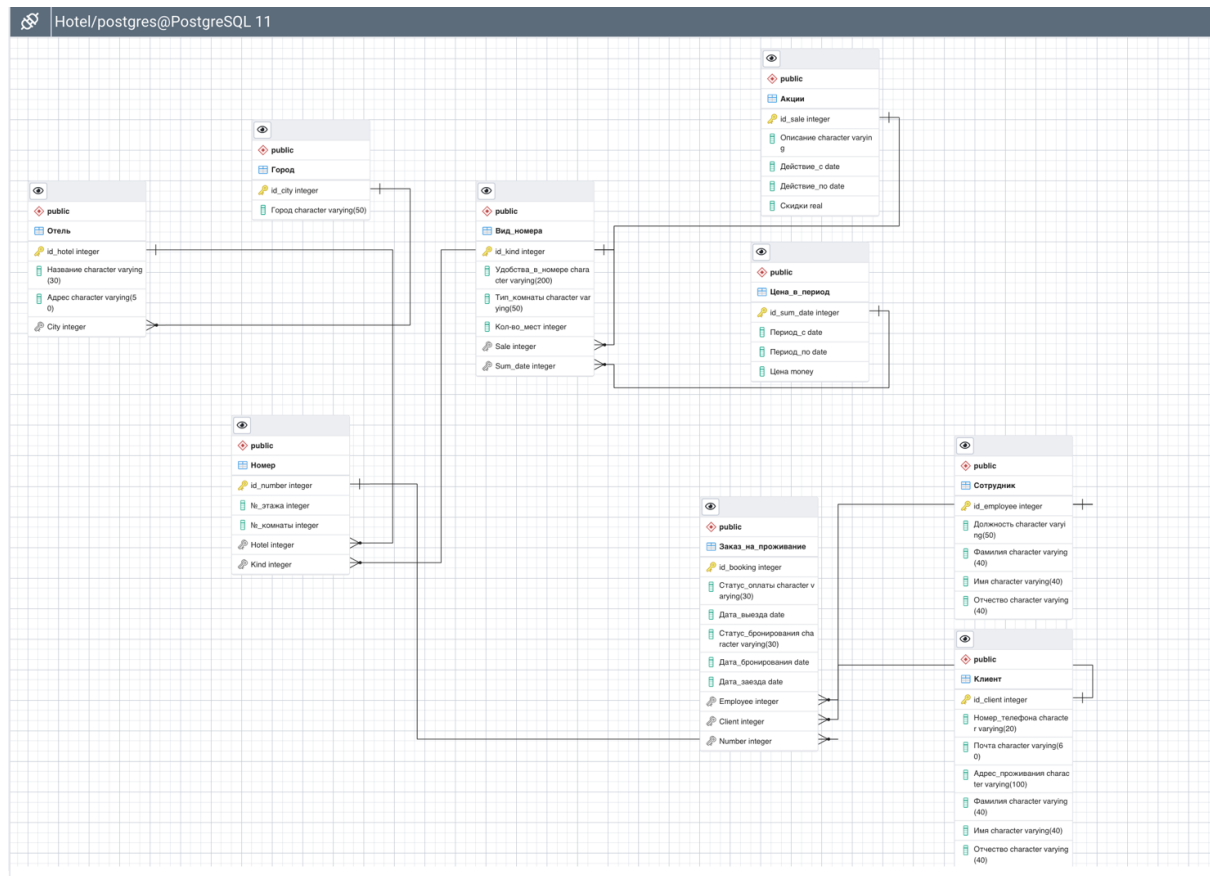
Цель работы:

Овладеть практическими навыками создания представлений и запросов на выборку данных к базе данных PostgreSQL, использования подзапросов при модификации данных и индексов.

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.

Схема базы данных (Отели):



Выполнение:

1. Запросы к базе данных. Выполнить запросы согласно индивидуальному заданию, часть 2.

- Составить список всех 2-местных номеров отелей, с ценой менее 2000\$, упорядочив данные в порядке уменьшения стоимости.

Hotel/postgres@PostgreSQL 11

Query Editor История запросов

```
1 SELECT "Вид_номера"."Тип_комнаты", "Цена_в_период"."Цена" FROM "Вид_номера", "Цена_в_период"
2 WHERE "Цена_в_период"."number_kind" = "Вид_номера"."id_kind" AND "Цена_в_период"."Цена" < '2000' AND "Вид_номера"."Кол-во_мест" = 2
3 AND "Цена_в_период"."Период_с" = '2022-05-01'
4 ORDER BY "Цена_в_период"."Цена" DESC
```

Результат План выполнения Сообщения Notifications

	Тип_комнаты character varying (50)	Цена money
1	двухместный полулюкс	\$1,000.00
2	Люкс	\$800.00
3	полу-люкс	\$600.00
4	стандарт	\$400.00

- Выбрать все записи регистрации постояльцев, которые выехали из отелей в течение двух последних недель.

Hotel/postgres@PostgreSQL 11

Query Editor История запросов

```
1 SELECT * FROM "Заказ_на_проживание" WHERE "Статус_бронирования" = 'Выехал' AND "Дата_выезда" BETWEEN '2022-03-16' AND '2022-03-30'
```

Результат План выполнения Сообщения Notifications

	id_booking [PK] integer	Статус_оплаты character varying (30)	Дата_выезда date	Статус_бронирования character varying (30)	Дата_бронирования date	Дата_заезда date	Employee integer	Client integer	Number integer
1	9	Оплачено	2022-03-20	Выехал	2022-03-10	2022-03-13		4	1
2	10	Оплачено	2022-03-23	Выехал	2022-03-14	2022-03-16		4	2
3	11	Оплачено	2022-03-25	Выехал	2022-03-18	2022-03-20		5	3
4	17	Оплачено	2022-03-27	Выехал	2022-03-11	2022-03-20		5	7
5	18	Оплачено	2022-03-27	Выехал	2022-03-13	2022-03-20		6	7
6	19	Оплачено	2022-03-25	Выехал	2022-03-12	2022-03-19		6	8
7	20	Оплачено	2022-03-26	Выехал	2022-03-15	2022-03-17		6	9
8	21	Оплачено	2022-03-28	Выехал	2022-03-10	2022-03-20		7	10

- Чему равен общий доход с каждого отеля за последний день?

Query Editor

История запросов

1

SELECT "Отель"."Название", SUM("Цена_в_период"."Цена") FROM "Отель"

2

INNER JOIN "Номер" ON "Отель"."id_hotel" = "Номер"."Hotel"

3

LEFT JOIN "Цена_в_период" ON "Номер"."Kind" = "Цена_в_период"."number_kind"

4

LEFT JOIN "Заказ_на_проживание" ON "Заказ_на_проживание"."Number" = "Номер"."id_number"

5

WHERE '2022-03-22' BETWEEN "Заказ_на_проживание"."Дата_заезда" AND "Заказ_на_проживание"."Дата_выезда"

6

GROUP BY "Отель"."Название"

7

8

Результат

План выполнения

Сообщения

Notifications

	Название character varying (30)	sum money
1	Redisson	\$3,400.00
2	Four Seasons	\$3,600.00

- Составить список свободных номеров одного из отелей на текущий день.

Hotel/postgres@PostgreSQL 11																											
Query Editor История запросов																											
<pre> 1 SELECT "№_комнаты" 2 FROM "Номер" LEFT JOIN "Заказ_на_проживание" 3 ON "Номер"."id_number" = "Заказ_на_проживание"."Number" 4 WHERE "Номер"."Hotel" = 1 AND ("Заказ_на_проживание"."Дата_заезда" IS NULL OR '2022-03-22' 5 NOT BETWEEN "Заказ_на_проживание"."Дата_заезда" AND "Заказ_на_проживание"."Дата_выезда") 6 7 </pre>																											
Результат План выполнения Сообщения Notifications																											
<table> <thead> <tr> <th></th><th>№_комнаты integer</th></tr> </thead> <tbody> <tr><td>1</td><td>301</td></tr> <tr><td>2</td><td>403</td></tr> <tr><td>3</td><td>501</td></tr> <tr><td>4</td><td>503</td></tr> <tr><td>5</td><td>602</td></tr> <tr><td>6</td><td>401</td></tr> <tr><td>7</td><td>402</td></tr> <tr><td>8</td><td>701</td></tr> <tr><td>9</td><td>601</td></tr> <tr><td>10</td><td>302</td></tr> <tr><td>11</td><td>502</td></tr> <tr><td>12</td><td>404</td></tr> </tbody> </table>			№_комнаты integer	1	301	2	403	3	501	4	503	5	602	6	401	7	402	8	701	9	601	10	302	11	502	12	404
	№_комнаты integer																										
1	301																										
2	403																										
3	501																										
4	503																										
5	602																										
6	401																										
7	402																										
8	701																										
9	601																										
10	302																										
11	502																										
12	404																										

- Найти общие потери от незанятых номеров за текущий день по всей сети.

Hotel/postgres@PostgreSQL 11

Query Editor История запросов

```

1 SELECT SUM("Цена_в_период"."Цена") FROM "Номер"
2 LEFT JOIN "Заказ_на_проживание" ON "Номер"."id_number" = "Заказ_на_проживание"."Number"
3 LEFT JOIN "Цена_в_период" ON "Цена_в_период"."number_kind" = "Номер"."Kind"
4
5 WHERE "Заказ_на_проживание"."Дата_заезда" IS NULL OR '2022-03-22'
6 NOT BETWEEN "Заказ_на_проживание"."Дата_заезда" AND "Заказ_на_проживание"."Дата_выезда"

```

Query Tool - Hotel/postgres@PostgreSQL 11

Результат План выполнения Сообщения Notifications

	sum money
1	\$31,950.00

- Определить, в каком отеле имеется наибольшее количество незанятых номеров на текущие сутки.

Hotel/postgres@PostgreSQL 11

Query Editor История запросов

```

1 SELECT "Название", COUNT(*) c FROM "Отель"
2 INNER JOIN "Номер" ON "Отель"."id_hotel" = "Номер"."Hotel"
3 LEFT JOIN "Заказ_на_проживание" ON "Заказ_на_проживание"."Number" = "Номер"."id_number"
4 WHERE ("Заказ_на_проживание"."Дата_заезда" IS NULL OR '2022-03-22'
5 NOT BETWEEN "Заказ_на_проживание"."Дата_заезда" AND "Заказ_на_проживание"."Дата_выезда")
6 GROUP BY "Отель"."Название"
7 ORDER BY c DESC LIMIT 1;
8

```

Результат План выполнения Сообщения Notifications

Название character varying (30)	c bigint
1 Hilton	15

- Определить самый популярный тип номеров.

Hotel/postgres@PostgreSQL 11

Query Editor История запросов

```

1 SELECT "Вид_номера"."Тип_комнаты", COUNT(*) c FROM "Заказ_на_проживание", "Вид_номера", "Номер"
2 WHERE "Заказ_на_проживание"."Number" = "Номер"."id_number"
3 AND "Номер"."Kind" = "Вид_номера"."id_kind"
4 GROUP BY "Вид_номера"."Тип_комнаты"
5 ORDER BY c DESC LIMIT 1;
6

```

Результат План выполнения Сообщения Notifications

Тип_комнаты character varying (50)	c bigint
1 Семейный стандарт	4

- Для турагентов (поиск свободных номеров в отелях).

Hotel/postgres@PostgreSQL 11

Query Editor История запросов

```
1 SELECT "№_комнаты"
2 FROM "Номер" LEFT JOIN "Заказ_на_проживание"
3 ON "Номер"."id_number" = "Заказ_на_проживание"."Number"
4 WHERE "Номер"."Hotel" = 1 AND ("Заказ_на_проживание"."Дата_заезда" IS NULL OR '2022-03-22'
5 NOT BETWEEN "Заказ_на_проживание"."Дата_заезда" AND "Заказ_на_проживание"."Дата_выезда")
6
7
```

Результат План выполнения Сообщения Notifications

	№_комнаты	integer
1		301
2		403
3		501
4		503
5		602
6		401
7		402
8		701
9		601
10		302
11		502
12		404

Результат План выполнения Сообщения Notifications

Graphical Analysis Статистика

The diagram illustrates the execution plan for the query. It shows a 'Hash Right Join' operation. The left input is the 'Номер' table, which is first converted into a 'Hash' structure. The right input is the 'Заказ_на_проживание' table. The join operation is represented by a large arrow pointing from the 'Hash' structure and the 'Заказ_на_проживание' table to the 'Hash Right Join' node.

- Для владельца компании (информация о доходах каждого отеля в сети за прошедший месяц)

Query Editor История запросов

```
1 SELECT SUM(("Заказ_на_проживание"."Дата_выезда" - "Заказ_на_проживание"."Дата_заезда") * "Цена_в_период"."Цена") AS DEAL
2 FROM "Заказ_на_проживание", "Цена_в_период", "Номер"
3 WHERE "Цена_в_период"."number_kind" = "Номер"."Kind" AND "Номер"."id_number" = "Заказ_на_проживание"."Number" AND
4 "Заказ_на_проживание"."Дата_заезда" BETWEEN '2022-03-01' AND '2022-03-30'
5
```

Результат План выполнения Сообщения Notifications

	deal	money
1	\$90,650.00	

Результат План выполнения Сообщения Notifications

Graphical Analysis Статистика

```
graph LR
    A[Заказ_на_проживание] --> B[Hash]
    C[Номер] --> D[Hash Inner Join]
    B --> D
    D --> E[Hash]
    F[Цена_в_период] --> G[Hash Inner Join]
    E --> G
    G --> H[Aggregate]
```

The diagram illustrates the execution plan for the SQL query. It starts with three input tables: 'Заказ_на_проживание' (Orders), 'Номер' (Rooms), and 'Цена_в_период' (Prices). 'Заказ_на_проживание' is hashed and joined with 'Номер' using a Hash Inner Join. The result is then hashed and joined with 'Цена_в_период' using another Hash Inner Join. Finally, the data is aggregated to produce the result.

2. Запросы на модификацию данных. Выполнить запросы на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов (составить самостоятельно). В отчете привести формулировку запроса, команду, скриншот до и после выполнения запроса. Учитывается сложность запроса;
- INSERT с использованием подзапросов;

Результат План выполнения Сообщения Notifications										
	id_booking [PK] integer	Статус_оплаты character varying (30)	Дата_выезда date	Статус_бронирования character varying (30)	Дата_бронирования date	Дата_заезда date	Employee integer	Client integer	Number integer	
1	10	Оплачено	2022-03-23	Выехал	2022-03-14	2022-03-16	4	2	9	
2	11	Оплачено	2022-03-25	Выехал	2022-03-18	2022-03-20	5	3	10	
3	15	Оплачено	2022-04-06	Выехал	2022-03-21	2022-03-27	5	6	17	
4	17	Оплачено	2022-03-27	Выехал	2022-03-11	2022-03-20	5	7	18	
5	18	Оплачено	2022-03-27	Выехал	2022-03-13	2022-03-20	6	7	23	
6	19	Оплачено	2022-03-25	Выехал	2022-03-12	2022-03-19	6	8	24	
7	20	Оплачено	2022-03-26	Выехал	2022-03-15	2022-03-17	6	9	27	
8	21	Оплачено	2022-03-28	Выехал	2022-03-10	2022-03-20	7	10	30	
9	13	Оплачено	2022-04-05	Забронировано	2022-03-21	2022-03-27	5	4	13	
10	14	Оплачено	2022-04-06	Забронировано	2022-03-21	2022-03-27	5	5	15	
11	23	Оплачено	2022-04-05	Забронировано	2022-03-20	2022-04-03	7	10	30	
12	25	Оплачено	2022-04-03	Забронировано	2022-03-25	2022-03-25	7	1	32	
13	9	Оплачено	2022-03-20	Забронировано	2022-03-10	2022-03-13	4	1	7	
14	24	Оплачено	2022-04-03	Забронировано	2022-03-27	2022-04-01	7	1	31	

```
INSERT INTO "Заказ_на_проживание"("Статус_оплаты", "Дата_выезда", "Статус_бронирования", "Дата_бронирования", "Дата_заезда", "Employee", "Client", "Number")
VALUES('Оплачено', '2022-03-28', 'Забронировано', '2022-03-21', '2022-03-24', (SELECT id_employee FROM "Сотрудник" WHERE "Фамилия" = 'Сепреев'), 2, 13)
```

	id_booking [PK] integer	Статус_оплаты character varying (30)	Дата_выезда date	Статус_бронирования character varying (30)	Дата_бронирования date	Дата_заезда date	Employee integer	Client integer	Number integer	
1	10	Оплачено	2022-03-23	Выехал	2022-03-14	2022-03-16	4	2	9	
2	11	Оплачено	2022-03-25	Выехал	2022-03-18	2022-03-20	5	3	10	
3	15	Оплачено	2022-04-06	Выехал	2022-03-21	2022-03-27	5	6	17	
4	17	Оплачено	2022-03-27	Выехал	2022-03-11	2022-03-20	5	7	18	
5	18	Оплачено	2022-03-27	Выехал	2022-03-13	2022-03-20	6	7	23	
6	19	Оплачено	2022-03-25	Выехал	2022-03-12	2022-03-19	6	8	24	
7	20	Оплачено	2022-03-26	Выехал	2022-03-15	2022-03-17	6	9	27	
8	21	Оплачено	2022-03-28	Выехал	2022-03-10	2022-03-20	7	10	30	
9	13	Оплачено	2022-04-05	Забронировано	2022-03-21	2022-03-27	5	4	13	
10	14	Оплачено	2022-04-06	Забронировано	2022-03-21	2022-03-27	5	5	15	
11	23	Оплачено	2022-04-05	Забронировано	2022-03-20	2022-04-03	7	10	30	
12	25	Оплачено	2022-04-03	Забронировано	2022-03-25	2022-03-25	7	1	32	
13	9	Оплачено	2022-03-20	Забронировано	2022-03-10	2022-03-13	4	1	7	
14	24	Оплачено	2022-04-03	Забронировано	2022-03-27	2022-04-01	7	1	31	
15	27	Оплачено	2022-03-28	Забронировано	2022-03-21	2022-03-24	5	2	13	

– UPDATE с использованием подзапросов;

Результат									
	id_booking [PK] integer	Статус_оплаты character varying (30)	Дата_выезда date	Статус_бронирования character varying (30)	Дата_бронирования date	Дата_заезда date	Employee integer	Client integer	Number integer
1	25	Оплачено	2022-04-03	Заехал	2022-03-25	2022-03-25	7	1	32
2	9	Оплачено	2022-03-20	Выехал	2022-03-10	2022-03-13	4	1	7
3	10	Оплачено	2022-03-23	Выехал	2022-03-14	2022-03-16	4	2	9
4	11	Оплачено	2022-03-25	Выехал	2022-03-18	2022-03-20	5	3	10
5	15	Оплачено	2022-04-06	Выехал	2022-03-21	2022-03-27	5	6	17
6	17	Оплачено	2022-03-27	Выехал	2022-03-11	2022-03-20	5	7	18
7	18	Оплачено	2022-03-27	Выехал	2022-03-13	2022-03-20	6	7	23
8	19	Оплачено	2022-03-25	Выехал	2022-03-12	2022-03-19	6	8	24
9	20	Оплачено	2022-03-26	Выехал	2022-03-15	2022-03-17	6	9	27
10	21	Оплачено	2022-03-28	Выехал	2022-03-10	2022-03-20	7	10	30
11	13	Оплачено	2022-04-05	Забронировано	2022-03-21	2022-03-27	5	4	13
12	14	Оплачено	2022-04-06	Забронировано	2022-03-21	2022-03-27	5	5	15
13	23	Оплачено	2022-04-05	Забронировано	2022-03-20	2022-04-03	7		
14	24	Оплачено	2022-04-03	Забронировано	2022-03-27	2022-04-01	7		

```

1 UPDATE "Заказ_на_проживание" SET "Статус_бронирования" = 'Забронировано'
2 WHERE "Client" IN (SELECT "id_client" FROM "Клиент" WHERE "Имя" = 'Вячеслав')

```

Результат									
	id_booking [PK] integer	Статус_оплаты character varying (30)	Дата_выезда date	Статус_бронирования character varying (30)	Дата_бронирования date	Дата_заезда date	Employee integer	Client integer	Number integer
1	10	Оплачено	2022-03-23	Выехал	2022-03-14	2022-03-16	4	2	9
2	11	Оплачено	2022-03-25	Выехал	2022-03-18	2022-03-20	5	3	10
3	15	Оплачено	2022-04-06	Выехал	2022-03-21	2022-03-27	5	6	17
4	17	Оплачено	2022-03-27	Выехал	2022-03-11	2022-03-20	5	7	18
5	18	Оплачено	2022-03-27	Выехал	2022-03-13	2022-03-20	6	7	23
6	19	Оплачено	2022-03-25	Выехал	2022-03-12	2022-03-19	6	8	24
7	20	Оплачено	2022-03-26	Выехал	2022-03-15	2022-03-17	6	9	27
8	21	Оплачено	2022-03-28	Выехал	2022-03-10	2022-03-20	7	10	30
9	13	Оплачено	2022-04-05	Забронировано	2022-03-21	2022-03-27	5	4	13
10	14	Оплачено	2022-04-06	Забронировано	2022-03-21	2022-03-27	5	5	15
11	23	Оплачено	2022-04-05	Забронировано	2022-03-20	2022-04-03	7	10	30
12	25	Оплачено	2022-04-03	Забронировано	2022-03-25	2022-03-25	7	1	32
13	9	Оплачено	2022-03-20	Забронировано	2022-03-10	2022-03-13	4	1	7
14	24	Оплачено	2022-04-03	Забронировано	2022-03-27	2022-04-01	7	1	31

– DELETE с использованием подзапросов.

Результат	План выполнения		Сообщения		Notifications				
1	10	Оплачено	2022-03-23	Выехал	2022-03-14	2022-03-16	4	2	9
2	11	Оплачено	2022-03-25	Выехал	2022-03-18	2022-03-20	5	3	10
3	15	Оплачено	2022-04-06	Выехал	2022-03-21	2022-03-27	5	6	17
4	17	Оплачено	2022-03-27	Выехал	2022-03-11	2022-03-20	5	7	18
5	18	Оплачено	2022-03-27	Выехал	2022-03-13	2022-03-20	6	7	23
6	19	Оплачено	2022-03-25	Выехал	2022-03-12	2022-03-19	6	8	24
7	20	Оплачено	2022-03-26	Выехал	2022-03-15	2022-03-17	6	9	27
8	21	Оплачено	2022-03-28	Выехал	2022-03-10	2022-03-20	7	10	30
9	13	Оплачено	2022-04-05	Забронировано	2022-03-21	2022-03-27	5	4	13
10	14	Оплачено	2022-04-06	Забронировано	2022-03-21	2022-03-27	5	5	15
11	23	Оплачено	2022-04-05	Забронировано	2022-03-20	2022-04-03	7	10	30
12	25	Оплачено	2022-04-03	Забронировано	2022-03-25	2022-03-25	7	1	32
13	9	Оплачено	2022-03-20	Забронировано	2022-03-10	2022-03-13	4	1	7
14	24	Оплачено	2022-04-03	Забронировано	2022-03-27	2022-04-01	7	1	31

DELETE FROM "Заказ_на_проживание"

WHERE "Number" = (SELECT id_number FROM "Номер" WHERE "№_комнаты" = 304 AND "Hotel" = 1)

1	10	Оплачено	2022-03-23	Выехал	2022-03-14	2022-03-16	4	2	9
2	15	Оплачено	2022-04-06	Выехал	2022-03-21	2022-03-27	5	6	17
3	17	Оплачено	2022-03-27	Выехал	2022-03-11	2022-03-20	5	7	18
4	18	Оплачено	2022-03-27	Выехал	2022-03-13	2022-03-20	6	7	23
5	19	Оплачено	2022-03-25	Выехал	2022-03-12	2022-03-19	6	8	24
6	20	Оплачено	2022-03-26	Выехал	2022-03-15	2022-03-17	6	9	27
7	21	Оплачено	2022-03-28	Выехал	2022-03-10	2022-03-20	7	10	30
8	13	Оплачено	2022-04-05	Забронировано	2022-03-21	2022-03-27	5	4	13
9	14	Оплачено	2022-04-06	Забронировано	2022-03-21	2022-03-27	5	5	15
10	23	Оплачено	2022-04-05	Забронировано	2022-03-20	2022-04-03	7	10	30
11	25	Оплачено	2022-04-03	Забронировано	2022-03-25	2022-03-25	7	1	32
12	9	Оплачено	2022-03-20	Забронировано	2022-03-10	2022-03-13	4	1	7
13	24	Оплачено	2022-04-03	Забронировано	2022-03-27	2022-04-01	7	1	31
14	27	Оплачено	2022-03-28	Забронировано	2022-03-21	2022-03-24	5	2	13

3. Создание индексов. Выполнить запросы без индекса и создать планы запросов. Выполнить создание индексов. Выполнить запросы с индексами и создать планы запросов. Сравнить время выполнения запросов. Удалить индексы.


I. Простой запрос


Время выполнения запроса в таблицу без индекса – 69 мсек:

```
1 SELECT "Тип_комнаты" FROM "Вид_номера"
2 WHERE "Кол-во_мест" = 2
```

Результат **План выполнения** Сообщения Notifications

Graphical Analysis Статистика





Вид_номера

Создание индекса:

```
1 CREATE INDEX typenum_idx
2 ON "Вид_номера" ("Кол-во_мест")
3 WHERE "Кол-во_мест" = 2
```

План выполнения **Сообщения** Notifications

CREATE INDEX

Запрос завершён успешно, время выполнения: 192 msec.

Время выполнения запроса в таблицу с индексом – 43 мсек:

```
1 SELECT "Тип_комнаты" FROM "Вид_номера"
2 WHERE "Кол-во_мест" = 2
```

План выполнения


Сообщения


Notifications


Graphical


Analysis

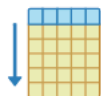
Статистика











Вид_номера

Удаление индекса:

```
1 DROP INDEX typenum_idx
```

План выполнения

Сообщения

Notifications

DROP INDEX

Запрос завершён успешно, время выполнения: 38 мсек.

II. Составной запрос

Время выполнения запроса в таблицу без индекса – 39 мсек:

```
1 SELECT "№_комнаты"
2 FROM "Номер" LEFT JOIN "Заказ_на_проживание"
3 ON "Номер"."id_number" = "Заказ_на_проживание"."Number"
4 WHERE "Hotel" = 1
```

Результат План выполнения Сообщения Notifications

Graphical Analysis Статистика

The diagram illustrates the execution plan for the query. It shows two input tables: 'Номер' (Rooms) and 'Заказ_на_проживание' (Reservations). The 'Заказ_на_проживание' table is first processed by a 'Hash' operation, creating a hash table. This hash table is then used in a 'Hash Left Join' operation with the 'Номер' table. The final output is the result of the join.

Создание индекса:

```
1 CREATE INDEX num_room_idx
2 ON "Номер" ("№_комнаты")
```

Результат План выполнения Сообщения Notifications

CREATE INDEX

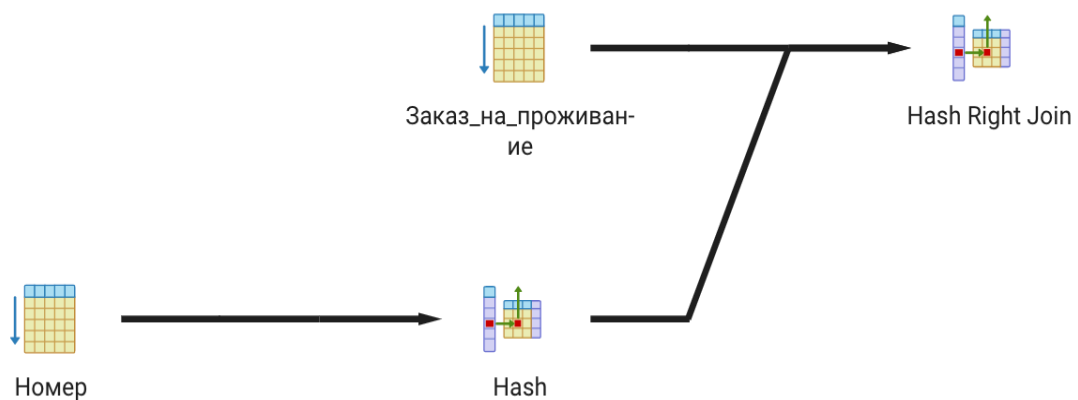
Запрос завершён успешно, время выполнения: 34 мсек.

Время выполнения запроса в таблицу с индексом – 35 мсек:

```
1 SELECT "№_комнаты"
2 FROM "Номер" LEFT JOIN "Заказ_на_проживание"
3 ON "Номер"."id_number" = "Заказ_на_проживание"."Number"
4 WHERE "Hotel" = 1
```

Результат План выполнения Сообщения Notifications

Graphical Analysis Статистика



Удаление индекса:

Query Editor История запросов

```
1 DROP INDEX num_room_idx
```

Результат План выполнения Сообщения Notifications

DROP INDEX

Запрос завершён успешно, время выполнения: 35 msec.

Выводы:

В результате выполненной работы были выполнены запросы и представления. Изучены запросы на модификацию данных, а также реализация индексов.