

## Создание таблиц «Типы номеров», «Номера», «Клиенты», «Бронирования»

Query	Query History
5	-- Таблица Типы номеров
6	CREATE TABLE Типы_номеров (
7	TypeHotelRoomID SERIAL PRIMARY KEY,
8	Название VARCHAR(50) NOT NULL,
9	Описание TEXT,
10	Вместимость INTEGER NOT NULL,
11	Базовая_цена DECIMAL(10,2) NOT NULL,
12	Количество INTEGER NOT NULL
13	);
14	
15	-- Таблица Номера
16	CREATE TABLE Номера (
17	HotelRoomID SERIAL PRIMARY KEY,
18	TypeHotelRoomID INTEGER NOT NULL,
19	Номер_комнаты VARCHAR(10) NOT NULL UNIQUE,
20	Статус room_status NOT NULL,
21	Доп информация TEXT.

Data Output	Messages	Notifications
CREATE TABLE		
Query returned successfully in 145 msec.		

Query	Query History
15	-- Таблица Номера
16	CREATE TABLE Номера (
17	HotelRoomID SERIAL PRIMARY KEY,
18	TypeHotelRoomID INTEGER NOT NULL,
19	Номер_комнаты VARCHAR(10) NOT NULL UNIQUE,
20	Статус room_status NOT NULL,
21	Доп информация TEXT,
22	FOREIGN KEY (TypeHotelRoomID) REFERENCES Типы_номеров(TypeHotelRoomID)
23	);
24	
25	-- Таблица Клиенты
26	CREATE TABLE Клиенты (
27	ClientID SERIAL PRIMARY KEY,
28	ФИО VARCHAR(100) NOT NULL,
29	Дата_рождения DATE NOT NULL,
30	Телефон VARCHAR(20) NOT NULL,
31	Email VARCHAR(100) UNIQUE,

Data Output	Messages	Notifications
CREATE TABLE		
Query returned successfully in 145 msec.		

```

24
25 -- Таблица Клиенты
26 ✓ CREATE TABLE Клиенты (
27     ClientID SERIAL PRIMARY KEY,
28     ФИО VARCHAR(100) NOT NULL,
29     Дата_рождения DATE NOT NULL,
30     Телефон VARCHAR(20) NOT NULL,
31     Email VARCHAR(100) UNIQUE,
32     Паспортные_данные VARCHAR(100) NOT NULL UNIQUE
33 );
34
35 -- Таблица Бронирования
36 ✓ CREATE TABLE Бронирования (
37     BookingID SERIAL PRIMARY KEY,
38     ClientID INTEGER NOT NULL,
39     HotelRoomID INTEGER NOT NULL,
40     Дата_заезда DATE NOT NULL,
41     Дата_выезда DATE NOT NULL,
42     Статус_бронирования booking_status NOT NULL,
43     Общая_стоимость DECIMAL(10,2) NOT NULL,
44     Дата_создания_брони TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
45     Доп_информация TEXT,
46     FOREIGN KEY (ClientID) REFERENCES Клиенты(ClientID),
47     FOREIGN KEY (HotelRoomID) REFERENCES Номера(HotelRoomID)
48 );

```

Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 145 msec.

```

-- Создаем пользовательские типы для статусов
CREATE TYPE room_status AS ENUM ('свободен', 'занят', 'бронирование', 'ремонт');
CREATE TYPE booking_status AS ENUM ('подтверждено', 'ожидание', 'отменено');

```

Заполнение таблиц:

```
1 -- Заполняем Типы номеров
2 v INSERT INTO Типы_номеров (Название, Описание, Вместимость, Базовая_цена, Количество) VALUES
3 ('Стандарт', 'Комфортабельный номер с одной двухспальной кроватью', 2, 3500.00, 10),
4 ('Люкс', 'Просторный номер с гостиной зоной и видом на море', 3, 6500.00, 5),
5 ('Семейный', 'Большой номер с двумя отдельными кроватями', 4, 5500.00, 8),
6 ('Президентский', 'Роскошный апартамент с отдельной гостиной и кухней', 2, 12000.00, 2);
7
8 -- Заполняем Номера
9 v INSERT INTO Номера (TypeHotelRoomID, Номер_комнаты, Статус, Доп_информация) VALUES
10 (1, '101', 'свободен', 'Окна во двор'),
11 (1, '102', 'занят', 'Окна на улицу'),
12 (2, '201', 'свободен', 'Вид на море, балкон'),
13 (2, '202', 'бронирование', 'Вид на парк'),
14 (3, '301', 'свободен', 'Смежные комнаты'),
15 (3, '302', 'ремонт', 'Требуется замена сантехники'),
16 (4, '401', 'свободен', 'Панорамные окна, джакузи');
17
18 -- Заполняем Клиенты
19 v INSERT INTO Клиенты (ФИО, Дата_рождения, Телефон, Email, Паспортные_данные) VALUES
20 ('Иванов Петр Сергеевич', '1985-03-15', '+7-912-345-67-89', 'ivanov@mail.ru', '4510 123456'),
21 ('Смирнова Анна Владимировна', '1990-07-22', '+7-923-456-78-90', 'smirnova@gmail.com', '4511 654321'),
22 ('Козлов Дмитрий Игоревич', '1978-11-30', '+7-934-567-89-01', 'kozlov@yandex.ru', '4512 789012'),
23 ('Петрова Мария Александровна', '1995-05-18', '+7-945-678-90-12', 'petrova@mail.ru', '4513 345678');
24
25 -- Заполняем Бронирования
26 v INSERT INTO Бронирования (ClientID, HotelRoomID, Дата_заезда, Дата_выезда, Статус_бронирования, Общая_стоимость, Доп_инф
27 (1, 2, '2024-02-15', '2024-02-20', 'подтверждено', 17500.00, 'Завтрак включен'),
28 (2, 3, '2024-02-18', '2024-02-25', 'ожидание', 45500.00, 'Трансфер из аэропорта'),
29 (3, 1, '2024-02-20', '2024-02-22', 'подтверждено', 7000.00, 'Бизнес-поездка'),
30 (1, 4, '2024-02-01', '2024-02-05', 'отменено', 25000.00, 'Отмена во бонусы');
```

Data Output

Messages

Notifications

INSERT 0 5

Query returned successfully in 50 msec.

## Перечисления (enums):

- room\_status — определяет возможные состояния номеров: свободен, занят, бронирование, ремонт.
- booking\_status — определяет статусы бронирований: подтверждено, ожидание, отменено.

## Таблица Типы\_номеров

Содержит описание категорий номеров отеля.

Поля:

- TypeHotelRoomID — уникальный идентификатор типа номера (автоинкремент).
- Название — название категории (например, "Стандарт", "Люкс").
- Описание — подробное описание удобств и особенностей.
- Вместимость — максимальное количество гостей.
- Базовая\_цена — стоимость за ночь в формате DECIMAL.
- Количество — общее количество номеров этой категории.

## Таблица Номера

Содержит информацию о конкретных комнатах отеля.

### Поля:

- HotelRoomID — уникальный идентификатор номера.
- TypeHotelRoomID — ссылка на тип номера (внешний ключ).
- Номер\_комнаты — уникальный номер комнаты.
- Статус — текущий статус из перечисления room\_status.
- Доп\_информация — дополнительные сведения о номере.

## Таблица Клиенты

Хранит данные о гостях отеля.

### Поля:

- ClientID — уникальный идентификатор клиента.
- ФИО — полное имя клиента.
- Дата\_рождения — дата рождения.
- Телефон — контактный телефон.
- Email — уникальный email-адрес.
- Паспортные\_данные — уникальные паспортные данные.

## Таблица Бронирования

Управляет бронированиями номеров клиентами.

### Поля:

- BookingID — уникальный идентификатор бронирования.
- ClientID — ссылка на клиента (внешний ключ).
- HotelRoomID — ссылка на номер (внешний ключ).
- Дата\_заезда — дата начала проживания.
- Дата\_выезда — дата окончания проживания.
- Статус\_бронирования — статус из перечисления booking\_status.

- Общая\_стоимость — расчётная стоимость бронирования.
- Дата\_создания\_брони — автоматически фиксируемая дата создания.

Доп\_информация — комментарии к бронированию

### Агрегатная функция:

**Результат:** Таблица с ФИО клиента и статистикой по его бронированиям.

```

1 SELECT
2     k.ФИО,
3     COUNT(b.BookingID) as Количество_бронирований,
4     COALESCE(SUM(b.Общая_стоимость), 0) as Общая_сумма_бронирований,
5     ROUND(COALESCE(AVG(b.Общая_стоимость), 0)::numeric, 2) as Средняя_стоимость_бронирования
6 FROM Клиенты k
7 LEFT JOIN Бронирования b ON k.ClientID = b.ClientID
8 GROUP BY k.ClientID, k.ФИО;

```

Data Output Messages Notifications

	ФИО character varying (100)	Количество_бронирований bigint	Общая_сумма_бронирований numeric	Средняя_стоимость_бронирования numeric
1	Петрова Мария Александровна	1	84000.00	84000.00
2	Смирнова Анна Владимировна	1	45500.00	45500.00
3	Козлов Дмитрий Игоревич	1	7000.00	7000.00
4	Иванов Петр Сергеевич	2	43500.00	21750.00

### JOIN между тремя таблицами:

**Результат:** Детальная информация о всех подтверждённых бронированиях с данными клиента, номера и типа номера.
















Результат: вывод клиентов с бронью стоимостью выше среднего. Находит всех клиентов, которые бронировали номера дороже среднего, фильтрует только подтверждённые бронирования

Query Query History

```
1  SELECT
2      ФИО,
3      Телефон,
4      Email
5  FROM Клиенты
6  WHERE ClientID IN (
7      SELECT DISTINCT b.ClientID
8      FROM Бронирования b
9      JOIN Номера n ON b.HotelRoomID = n.HotelRoomID
10     JOIN Типы_номеров tn ON n.TypeHotelRoomID = tn.TypeHotelRoomID
11     WHERE tn.Базовая_цена > (
12         SELECT AVG(Базовая_цена)
13         FROM Типы_номеров
14     )
15     AND b.Статус_бронирования = 'подтверждено'
16 );
```

Data Output Messages Notifications



	ФИО character varying (100) 	Телефон character varying (20) 	email character varying (100) 
1	Петрова Мария Александровна	+7-945-678-90-12	petrova@mail.ru

## Условный оператор CASE

Результат: Удобочитаемая таблица с категориями и статусами.

[Query](#) [Query History](#)

```
1  SELECT
2      k.ФИО,
3      b.Общая_стоимость,
4      CASE
5          WHEN b.Общая_стоимость > 20000 THEN 'Высокобюджетное'
6          WHEN b.Общая_стоимость BETWEEN 10000 AND 20000 THEN 'Среднебюджетное'
7          ELSE 'Эконом'
8      END as Категория_бронирования,
9      CASE
10         WHEN b.Статус_бронирования = 'подтверждено' THEN 'Активно'
11         WHEN b.Статус_бронирования = 'ожидание' THEN 'На рассмотрении'
12         ELSE 'Неактивно'
13     END as Статус
14 FROM Бронирования b
15 JOIN Клиенты k ON b.ClientID = k.ClientID
16 ORDER BY b.Общая_стоимость DESC;
```

[Data Output](#) [Messages](#) [Notifications](#)

<div><div><div><div>≡+</div><div>📄</div><div>▼</div><div>📋</div><div>▼</div><div>🗑️</div><div>🔄</div><div>⬇️</div><div>📈</div><div>SQL</div></div></div></div>				
	<div>ФИО</div> <div>character varying (100)</div> <div>🔒</div>	<div>Общая_стоимость</div> <div>numeric (10,2)</div> <div>🔒</div>	<div>Категория_бронирования</div> <div>text</div> <div>🔒</div>	<div>Статус</div> <div>text</div> <div>🔒</div>
1	Петрова Мария Александровна	84000.00	Высокобюджетное	Активно
2	Смирнова Анна Владимировна	45500.00	Высокобюджетное	На рассмотрении
3	Иванов Петр Сергеевич	26000.00	Высокобюджетное	Неактивно
4	Иванов Петр Сергеевич	17500.00	Среднебюджетное	Активно
5	Козлов Дмитрий Игоревич	7000.00	Эконом	Активно