



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.03 (Прикладная информатика)

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К КУРСОВОЙ РАБОТЕ

по дисциплине «Базы Данных»

НА ТЕМУ:

Гостиница

Студент

ИУ6-44Б

(Группа)

(Подпись, дата)

А.И. Гарифуллин

(И.О. Фамилия)

Руководитель

(Подпись, дата)

Д. А. Миков

(И.О. Фамилия)

2023 г.

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

УТВЕРЖДАЮ

Заведующий кафедрой ИУ6

А.В. Пролетарский
« » 2023 г.

ЗАДАНИЕ на выполнение курсовой работы

по дисциплине Базы данных

Студент группы ИУ6-44Б

Гарифуллин Амир Ильнурович
(Фамилия, имя, отчество)

Тема курсовой работы «Гостиница»

Направленность КР (учебная, исследовательская, практическая, производственная, др.)
учебная

Источник тематики (кафедра, предприятие, НИР) кафедра

График выполнения работы: 25% к 4 нед., 50% к 7 нед., 75% к 11 нед., 100% к 14 нед.

Задание: разработать базу данных «Гостиница», содержащую не менее 7 таблиц. В основной таблице должно содержаться не менее 100 тыс. записей, в других таблицах должно содержаться не менее 10 записей. В БД должно быть разработано не менее 5 сложных запросов и не менее 2 сложных отчетов.

Оформление курсовой работы:

1. Расчетно-пояснительная записка на 25-30 листах формата А4.

Дата выдачи задания «8» февраля 2023 г.

Руководитель

Студент

(Подпись,
дата)

(Подпись,
дата)

Д. А. Миков
(И.О. Фамилия)

А.И.Гарифуллин
(И.О. Фамилия)

Примечание: Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре.

РЕФЕРАТ

Расчётно-пояснительная записка состоит из 33 страниц и включает в себя 19 рисунков, 2 источника и 16 листингов.

Объектом разработки является база данных гостиницы. Цель работы — создание базы данных для описания номеров, бронирования номеров обычными посетителями и турфирмами. Система предполагает хранение информации о датах начала и завершения каждого пребывания в гостинице, учета и архива постояльцев.

Для реализации используется реляционная база данных PostgreSQL 15, графический клиент DBeaver.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	5
1 ПРОЕКТИРОВАНИЕ	6
1.1 Анализ предметной области	6
1.2 Выделение сущностей	6
1.3 Проектирование инфологической модели базы данных	8
1.4 Проектирование даталогической модели базы данных	9
2 РЕАЛИЗАЦИЯ	13
2.1 Написание скрипта создания базы данных	13
2.2 Заполнение базы данных	15
2.3 Сложные запросы	25
Запрос 1	25
Запрос 2	26
Запрос 3	27
Запрос 4	28
Запрос 5	29
2.4 Сложные отчёты	30
Отчёт 1	30
Отчёт 2	31
ЗАКЛЮЧЕНИЕ	32
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ	33

ВВЕДЕНИЕ

Система предназначена для обеспечения согласованного хранения данных предметной области. База данных обеспечивает доступность информации о каждом номере гостиницы, текущих и прошлых пребываниях в гостинице, постояльцах и компаниях, которые бронировали номера.

Актуальность разработки данной системы состоит в том, что она позволяет отслеживать и анализировать занятость гостиницы и пребывания, которые уже были и текущие, система масштабируема.

Для реализации используется реляционная база данных PostgreSQL 15, графический клиент DBeaver.

1 ПРОЕКТИРОВАНИЕ

1.1 Анализ предметной области

Предметная область базы данных связана с хранением информации о номерах гостиницы, текущих и прошлых пребываниях и бронированиях в них, постояльцах и турфирмах, видах полупансионов.

1.2 Выделение сущностей

Для проектирования базы данных необходимо выделить элементы предметной области. [1]

Сущности:

1. Номера

Содержит информацию о номерах. Содержит уникальный идентификатор, номер, тип и цену комнаты.

2. Бронирования

Содержит информацию о бронированиях обычных посетителей. Содержит уникальный идентификатор, идентификатор комнаты, которую забронировали, дату брони, заезда и выезда, а также ФИ посетителя.

3. Бронирования турфирм

Содержит информацию о бронированиях турфирмами. Содержит уникальный идентификатор, идентификатор комнаты, которую забронировали, идентификатор типа питания, дату брони, заезда и выезда, а также идентификатор турфирмы.

4. Турфирмы

Данная сущность содержит информацию о турфирмах. Содержит уникальный идентификатор турфирмы и ее название.

5. Питание

Содержит информацию о видах полупансиона. Имеет уникальный идентификатор типа питания и название типа.

6. Оплата

Содержит информацию об оплатах бронирований. Имеет уникальный идентификатор операции и идентификатор оплаченного бронирования обычным посетителем.

7. Оплата турфирмой

Содержит информацию об оплатах бронирований. Имеет уникальный идентификатор операции и идентификатор оплаченного бронирования турфирмой.

8. Пребывания

Содержит информацию о текущих и прошлых пребываниях в гостинице. Имеет уникальный идентификатор, идентификатор постояльца, дату заезда и выезда и уникальный идентификатор номера.

9. Постояльцы

Содержит информацию о всех постояльцах гостиницы. Имеет уникальный идентификатор и ФИ постояльца.

10. Заказы услуг

Содержит информацию о заказах дополнительных услуг. Имеет уникальный идентификатор заказа, идентификатор дополнительной услуги, идентификатор пребывания и дату заказа.

11. Услуги

Данная сущность содержит информацию о всех дополнительных услугах, предлагаемых гостиницей. Имеет уникальный идентификатор услуги, наименование и цену.

Так как объектом разработки является база данных описания и бронирования номеров, то мы можем однозначно выделить главную сущность в данной схеме. Сущность «Номера» является главной в схеме.

1.3 Проектирование инфологической модели базы данных

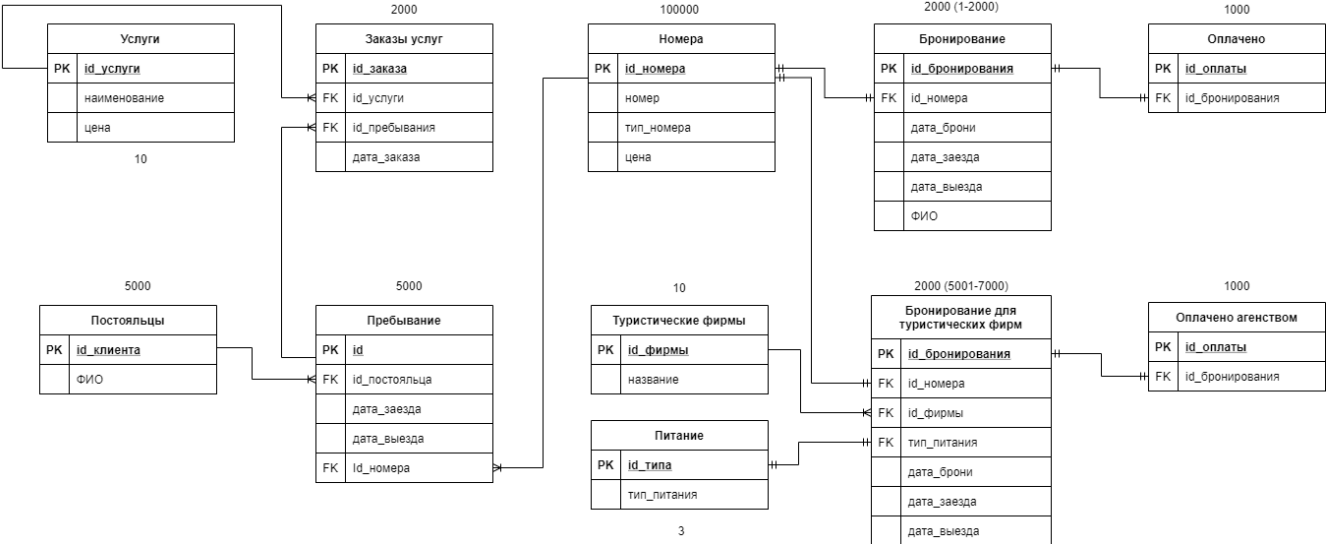


Рисунок 1 – Инфологическая модель базы данных

1.4 Проектирование даталогической модели базы данных

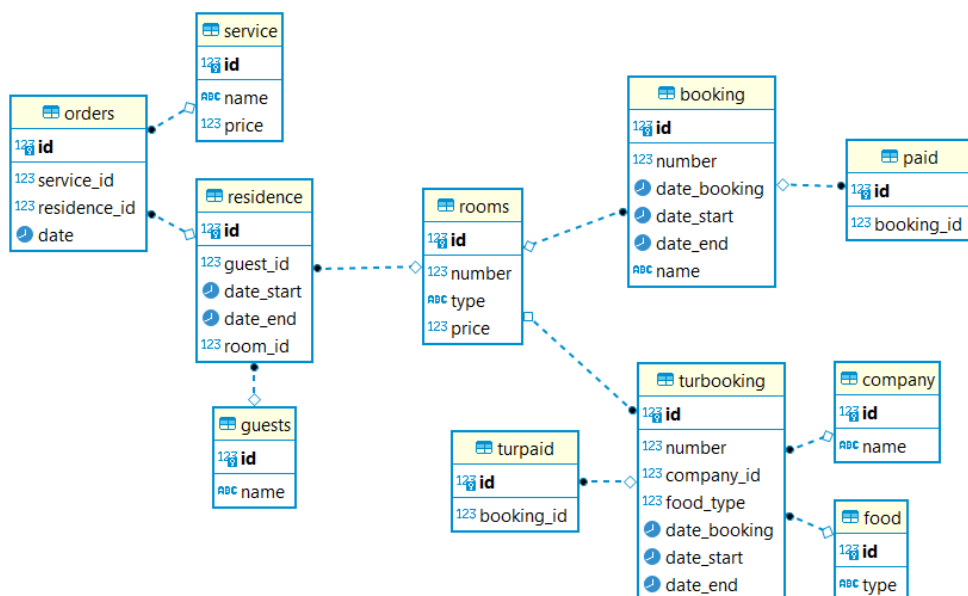


Рисунок 2 – Даталогическая модель базы данных

Ниже приведено описание каждой из таблиц и её полей.

Таблица 1 – поля таблицы rooms

Название поля	Предназначение	Тип
id	Уникальный номер	INTEGER
number	Номер	INTEGER
type	Тип	TEXT
price	Цена	INTEGER

Таблица 2 – поля таблицы booking

Название поля	Предназначение	Тип
id	Уникальный номер	INTEGER
number	Указатель на rooms (таблица 1)	INTEGER
date_booking	Дата бронирования	DATE
date_start	Дата заезда	DATE
date_end	Дата выезда	DATE

name	Имя забронировавшего	TEXT
------	-------------------------	------

Таблица 3 – поля таблицы paid

Название поля	Предназначение	Тип
id	Уникальный номер	INTEGER
group_id	Указатель на booking (таблица 2)	INTEGER

Таблица 4 – поля таблицы turbooking

Название поля	Предназначение	Тип
id	Уникальный номер	INTEGER
number	Указатель на rooms (таблица 1)	INTEGER
company_id	Указатель на company (таблица 5)	INTEGER
food_type	Указатель на food (таблица 6)	INTEGER
date_booking	Дата бронирования	DATE
date_start	Дата заезда	DATE
date_end	Дата выезда	DATE

Таблица 5 – поля таблицы company

Название поля	Предназначение	Тип
id	Уникальный номер	INTEGER
name	Название турфирмы	TEXT

Таблица 6 – поля таблицы food

Название поля	Предназначение	Тип
id	Уникальный номер	INTEGER
type	Тип полупансиона	TEXT

Таблица 7 – поля таблицы turpaid

Название поля	Предназначение	Тип
id	Уникальный номер	INTEGER
booking_id	Указатель на turbooking (таблица 4)	INTEGER

Таблица 8 – поля таблицы residence

Название поля	Предназначение	Тип
id	Уникальный номер	INTEGER
guest_id	Указатель на guests (таблица 9)	INTEGER
date_start	Дата заезда	DATE
date_end	Дата выезда	DATE
room_id	Указатель на rooms (таблица 1)	INTEGR

Таблица 9 – поля таблицы guests

Название поля	Предназначение	Тип
id	Уникальный номер	INTEGER
name	Имя посетителя	TEXT

Таблица 10 – поля таблицы orders

Название поля	Предназначение	Тип
id	Уникальный номер	INTEGER

service_id	Указатель на service (таблица 11)	INTEGER
residence_id	Указатель на residence (таблица 8)	INTEGER
date	Дата заказа	DATE

Таблица 11 – поля таблицы service

Название поля	Предназначение	Тип
id	Уникальный номер	INTEGER
name	Название услуги	TEXT
price	Цена	INTEGER

2 РЕАЛИЗАЦИЯ

2.1 Написание скрипта создания базы данных

Создадим таблицы, описанные на этапе проектирования. [2,3]

Первичные и внешние ключи были установлены с помощью функционала графического клиента DBeaver. [4]

Листинг 1 – Скрипт создания таблиц, первичных и внешних ключей.

```
-- Н о м е р а
CREATE TABLE rooms (
    id SERIAL PRIMARY KEY,
    number INTEGER NOT NULL,
    type TEXT NOT NULL,
    price INTEGER NOT NULL
);

-- Б р о н и р о в а н и я
CREATE TABLE booking (
    id SERIAL PRIMARY KEY,
    number INTEGER NOT NULL,
    date_booking DATE NOT NULL,
    date_start DATE NOT NULL,
    date_end DATE NOT NULL
    name TEXT NOT NULL
);

-- О п л а т ы
CREATE TABLE paid (
    id SERIAL PRIMARY KEY,
    booking_id INTEGER NOT NULL
);

-- Б р о н и р о в а н и я т у р ф и р м о й
CREATE TABLE turbooking (
    id SERIAL PRIMARY KEY,
    number INTEGER NOT NULL,
    company_id INTEGER NOT NULL,
    food_type INTEGER NOT NULL,
    date_booking DATE NOT NULL,
    date_start DATE NOT NULL,
    date_end Date NOT NULL
);

-- Т у р ф и р м ы
CREATE TABLE company (
    id SERIAL PRIMARY KEY,
    number INTEGER NOT NULL
```

```

);

-- Тип полупансиона
CREATE TABLE food (
    id SERIAL PRIMARY KEY,
    type TEXT NOT NULL
);

-- Оплаты турфирмами
CREATE TABLE turpaid (
    id SERIAL PRIMARY KEY,
    booking_id INTEGER NOT NULL
);

-- Пребывания
CREATE TABLE residence (
    id SERIAL PRIMARY KEY,
    guest_id INTEGER NOT NULL,
    date_start DATE NOT NULL,
    date_end Date NOT NULL,
    room_id INTEGER NOT NULL
);

-- Постояльцы
CREATE TABLE guests (
    id SERIAL PRIMARY KEY,
    name TEXT NOT NULL
);

-- Заказы доп услуг
CREATE TABLE orders (
    id SERIAL PRIMARY KEY,
    service_id INTEGER NOT NULL,
    residence_id INTEGER NOT NULL,
    date DATE NOT NULL
);

-- Доп услуги
CREATE TABLE service (
    id SERIAL PRIMARY KEY,
    name TEXT NOT NULL,
    price INTEGER NOT NULL
);

```

2.2 Заполнение базы данных

Заполнение некоторых сущностей, таких как: Компании, Допуслуги и Типы питания, были заполнены вручную, а остальные с помощью python-скрипта: Листинг 2 – Листинг 9.

Код заполнения был написан на языке Python [5]

Листинг 2 – python скрипт генерации данных сущности booking

```
from mimesis import Person
from mimesis import Text
from faker import Faker
import pandas as pd
import random
import json

person = Person('ru')
rand_text = Text()
faker = Faker()

roles = ['econom', 'standard', 'luxе']

def booking(start, number_of_rows):
    return [{
        'id': x + 1,
        'number': random.randint(1, 100000),
        'date_booking': faker.date(),
        'date_start': faker.date(),
        'date_end': faker.date(),
        'name': person.first_name() + ' ' + person.last_name()
```

```
        } for x in range(start, start + number_of_rows, 1)
    ]
```

```
pd.DataFrame(booking(0, 2000)).to_csv('booking.csv', index=False)
```

Листинг 3 – python скрипт генерации данных сущности company

```
from mimesis import Person
from mimesis import Text
from mimesis import Finance
from faker import Faker
import pandas as pd
import random
import json

finance = Finance('ru')
person = Person('ru')
rand_text = Text()
faker = Faker()

roles = ['econom', 'standard', 'luxе']

def company(start, number_of_rows):
    return [{
        'id': x + 1,
        'name': finance.company()
    } for x in range(start, start + number_of_rows, 1)
    ]

pd.DataFrame(company(0, 10)).to_csv('company.csv', index=False)
```

Листинг 4 - python скрипт генерации данных сущности guests

```
from mimesis import Person
from mimesis import Text
from faker import Faker
import pandas as pd
import random
import json
```



```

person = Person('ru')
rand_text = Text()
faker = Faker()

roles = ['econom', 'standard', 'luxе']

def guests(start, number_of_rows):
    return [{
        'id': x + 1,
        'name_id': person.first_name() + ' ' + person.last_name()
    } for x in range(start, start + number_of_rows, 1)
    ]

pd.DataFrame(guests(0, 5000)).to_csv('guests.csv', index=False)

```

Листинг 5 - python скрипт генерации данных сущности orders

```

from mimesis import Person
from mimesis import Text
from faker import Faker
import pandas as pd
import random
import json

person = Person('ru')
rand_text = Text()
faker = Faker()

roles = ['econom', 'standard', 'luxе']

def orders(start, number_of_rows):
    return [{
        'id': x + 1,
        'service_id': random.randint(1,10),
        'residence_id': random.randint(1,5000),
        'date': faker.date()
    } for x in range(start, start + number_of_rows, 1)
    ]

pd.DataFrame(orders(0, 2000)).to_csv('orders.csv', index=False)

```

Листинг 6 - python скрипт генерации данных сущности paid

```
from mimesis import Person
from mimesis import Text
from faker import Faker
import pandas as pd
import random
import json

person = Person('ru')
rand_text = Text()
faker = Faker()

roles = ['econom', 'standard', 'luxе']

def paid(start, number_of_rows):
    return [{
        'id': x + 1,
        'booking_id': random.randint(1, 2000)
    } for x in range(start, start + number_of_rows, 1)
    ]

pd.DataFrame(paid(0, 1000)).to_csv('paid.csv', index=False)
```

Листинг 7 - python скрипт генерации данных сущности residence

```
from mimesis import Person
from mimesis import Text
from faker import Faker
import pandas as pd
import random
import json

person = Person('ru')
rand_text = Text()
faker = Faker()

roles = ['econom', 'standard', 'luxе']

def residence(start, number_of_rows):
    return [{
```

```

        'id': x + 1,
        'guest_id': random.randint(1,5000),
        'date_start': faker.date(),
        'date_end': faker.date(),
        'room_id': random.randint(1,100001)
    } for x in range(start, start + number_of_rows, 1)
]

pd.DataFrame(residence(0, 5000)).to_csv('residence.csv', index=False)

```

Листинг 8 - python скрипт генерации данных сущности rooms

```

from mimesis import Person
from mimesis import Text
from faker import Faker
import pandas as pd
import random
import json

person = Person('ru')
rand_text = Text()
faker = Faker()

roles = ['econom', 'standard', 'luxе']

def rooms(start, number_of_rows):
    return [{
        'id': x + 1,
        'number': random.randint(1, 999999),
        'type': random.choices(roles)[0],
        'price': random.randint(1000, 20000)
    } for x in range(start, start + number_of_rows, 1)
    ]

pd.DataFrame(rooms(0, 100001)).to_csv('rooms.csv', index=False)

```

Листинг 9 - python скрипт генерации данных сущности turbooking

```

from mimesis import Person
from mimesis import Text
from faker import Faker
import pandas as pd
import random

```

```

import json

person = Person('ru')
rand_text = Text()
faker = Faker()

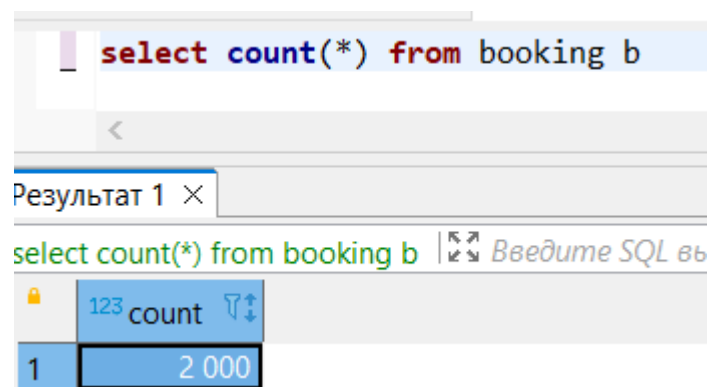
roles = ['econom', 'standard', 'luxе']

def turbooking(start, number_of_rows):
    return [{
        'id': x + 1,
        'number': random.randint(1, 100000),
        'company_id': random.randint(1, 10),
        'food_type': random.randint(1, 3),
        'date_booking': faker.date(),
        'date_start': faker.date(),
        'date_end': faker.date()
    } for x in range(start, start + number_of_rows, 1)
    ]

pd.DataFrame(turbooking(5000, 7000)).to_csv('turbooking.csv', index=False)

```

Проверим, что таблицы заполнены: [4]



Результат 1 ×

	count
1	2 000

Рисунок 3 – Число записей в таблице booking

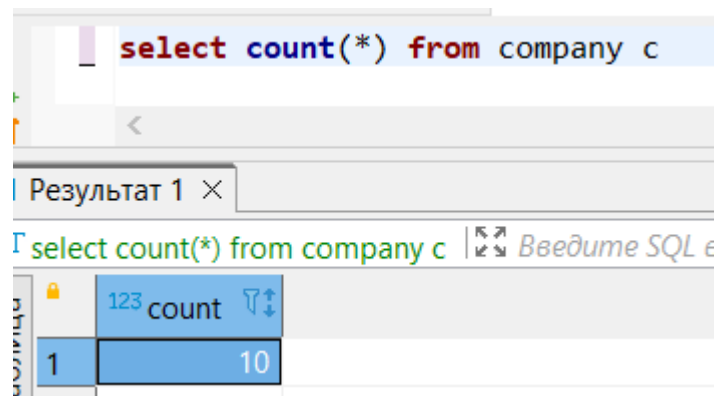


Рисунок 4 – Число записей в таблице company

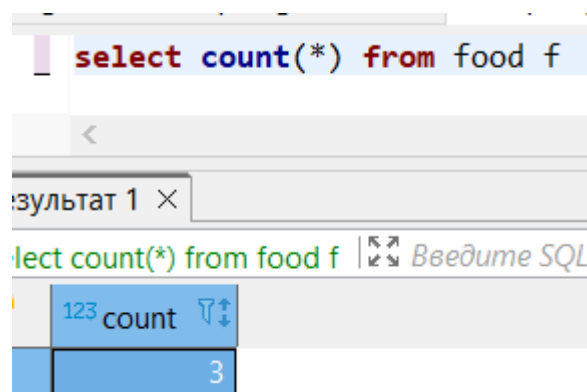


Рисунок 5 – Число записей в таблице food

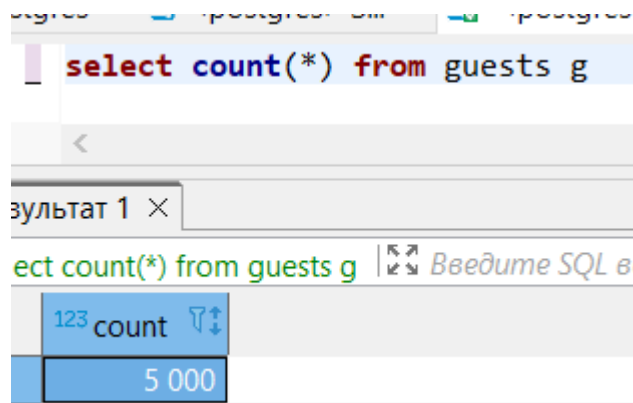


Рисунок 6 – Число записей в таблице guests

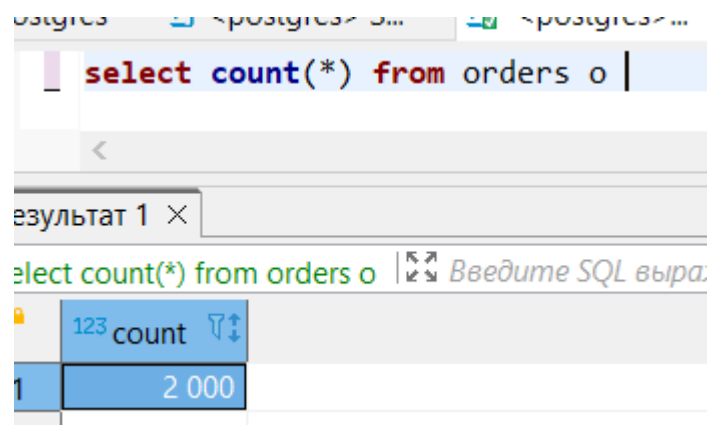


Рисунок 7 – Число записей в таблице orders

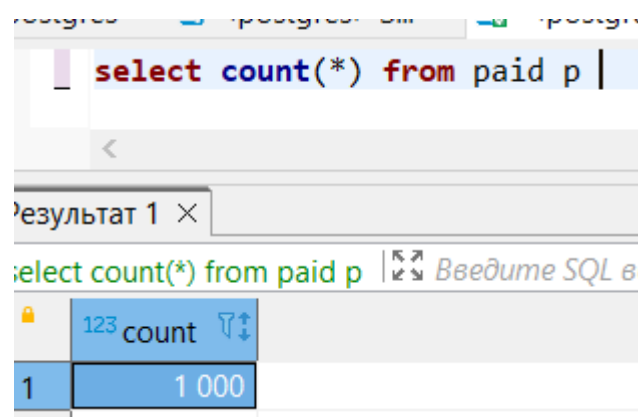


Рисунок 8 – Число записей в таблице paid

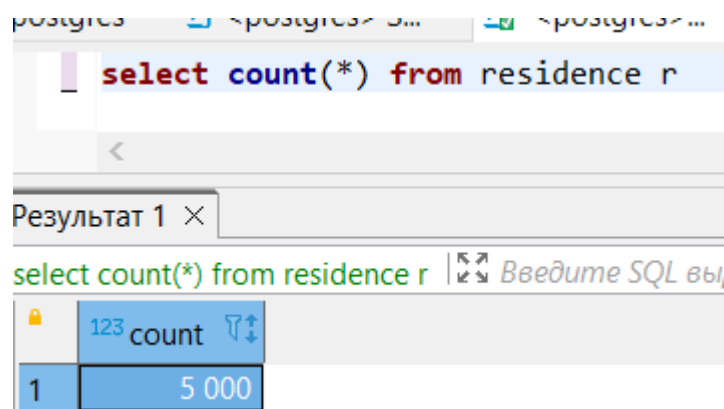


Рисунок 9 – Число записей в таблице residence

<code>select count(*) from rooms r2</code>	
<	
Результат 1 ×	
<code>select count(*) from rooms r2</code>	Введите SQL
count	123
	100 001

Рисунок 9 – Число записей в таблице rooms

<code>select count(*) from service s</code>	
<	
Результат 1 ×	
<code>select count(*) from service s</code>	Введите SQL
count	123
	10

Рисунок 10 – Число записей в таблице service

<code>select count(*) from turbooking t</code>	
<	
Результат 1 ×	
<code>t count(*) from turbooking t</code>	Введите SQL
count	123
	7 000

Рисунок 11 – Число записей в таблице turbooking

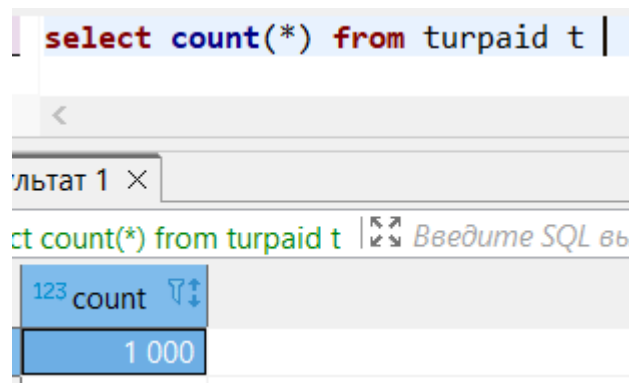


Рисунок 12 – Число записей в таблице turpaid

2.3 Сложные запросы

Для получения различной статистики, был разработан ряд SQL-запросов, приведенных в листинге ниже.

Запрос 1

Задача: Получить список всех постояльцев, которые хоть раз покупали доп услугу "Сувенир".

Листинг 10 – Первый SQL-запрос

```
SELECT DISTINCT g.*  
FROM guests g  
JOIN residence r ON g.id = r.guest_id  
JOIN orders o ON r.id = o.residence_id  
JOIN service s ON o.service_id = s.id  
WHERE s."name" = 'С у в е н и р';
```



	id	name
1	3 482	Алиса Винтилова
2	2 856	Фаддей Карпенков
3	3 613	Юнона Боченков
4	2 436	Акакий Дубинянский
5	4 024	Юстин Вакулов
6	1 464	Павлина Зудин
7	3 246	Паула Бакланова
8	4 341	Стефания Арцыбушева
9	4 717	Екатерина Римский
10	2 205	Юнона Трудолюбова
11	1 975	Пантелеймон Малецкий
12	4 887	Иулиан Орловский
13	4 144	Евдоким Новосельцева
14	459	Лолита Кудинов
15	1 342	Далида Ковалева
16	2 909	Лукьян Крюков
17	1 794	Джамиля Грин
18	3 143	Виталия Палицин
19	4 529	Нонна Мартынов
20	3 380	Ждан Купреянова
21	222	Анфиса Ельшина
22	591	Аиша Пушкина
23	4 413	Еремей Филипов
24	779	Зита Полуектова
25	4 487	Мария Елизарова

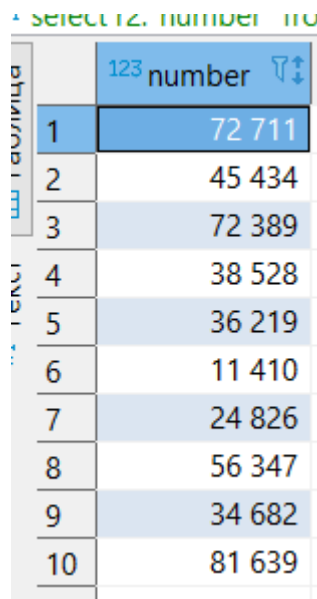
Рисунок 13 – Результат первого SQL-запроса

Запрос 2

Задача: Вывести 10 комнат, которые заняты на дату 1975-05-13

Листинг 11 – Второй SQL-запрос

```
select r2."number"  
from booking b, turbooking t, residence r, rooms r2  
where  
    ((b.date_start < '1975-05-13' and b.date_end > '1975-05-13') and b."number" =  
r2."number") or  
    ((t.date_start < '1975-05-13' and t.date_end > '1975-05-13') and t."number" =  
r2."number") or  
    ((r.date_start < '1975-05-13' and r.date_end > '1975-05-13') and r.room_id =  
r2."number")  
limit 10
```



The screenshot shows a query result with 10 rows. The first row is highlighted in blue. The column header is 'number' with a sort icon. The values are room numbers: 72 711, 45 434, 72 389, 38 528, 36 219, 11 410, 24 826, 56 347, 34 682, and 81 639.

	number
1	72 711
2	45 434
3	72 389
4	38 528
5	36 219
6	11 410
7	24 826
8	56 347
9	34 682
10	81 639

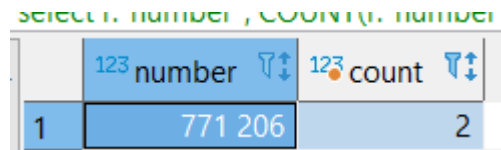
Рисунок 14 – Результат второго SQL-запроса

Запрос 3

Задача: Номер, который больше всего раз бронировался с типом питания завтрак-ужин.

Листинг 12 – Третий SQL-запрос

```
select r."number", COUNT(r."number")
from turbooking t, food f, rooms r
where
    r.id = t."number" and
    t.food_type = (select f.id where f."type" = 'з а в т р а к - у ж и н')
group by r."number"
order by COUNT(r."number") desc
limit 1
```



	123 number	123 count
1	771 206	2

Рисунок 15 – Результат третьего SQL-запроса

Запрос 4

Задача: Номера, которые были оплачены турфирмой ТагАз.

Листинг 13 – Четвертый SQL-запрос

```
select r."number"  
from rooms r, turbooking t, turpaid t2  
where  
    r.id = t."number" and  
    t.id = t2.booking_id and  
    t.company_id = (select c.id from company c where c."name"='Т а г А з')
```

	123 number
1	422 624
2	526 286
3	526 286
4	502 485
5	876 707
6	519 163
7	519 557
8	266 609
9	915 804
10	667 670
11	985 582
12	946 459
13	931 260
14	687 263
15	741 998
16	967 835
17	654 537
18	116 630
19	295 144
20	224 508
21	109 883
22	109 883
23	536 158

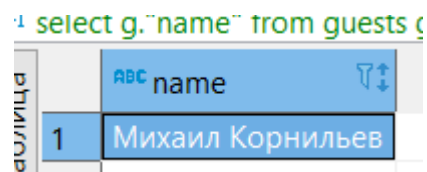
Рисунок 16 – Результат четвертого SQL-запроса

Запрос 5

Задача: Вывести последнего постояльца жившего в номере 659207

Листинг 14 – Пятый SQL-запрос

```
select g."name"  
from guests g, residence r , rooms r2  
where  
    r.room_id = (select r3.id from rooms r3 where r3."number"=659207) and  
    r.guest_id = g.id  
order by r.date_end desc  
limit 1
```



The screenshot shows a SQL query editor with the query: `select g."name" from guests g`. Below the query, a table with two columns is displayed. The first column is labeled 'id' and the second column is labeled 'name'. The first row of the table contains the value '1' in the 'id' column and 'Михаил Корнильев' in the 'name' column.

id	name
1	Михаил Корнильев

Рисунок 17 – Результат пятого SQL-запроса

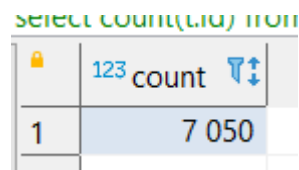
2.4 Сложные отчёты

Отчёт 1

Задача: Вывести кол-во бронирований ТагАЗ'ом с типом питания завтрак-ужин.

Листинг 15 – Шестой SQL-запрос

```
select count(t.id)
from company c, turbooking t, food f
where
    t.food_type = (select f2.id from food f2 where f2."type"='з а в т р а к -
у ж и н') and
    t.company_id = (select c2.id from company c2 where c2."name"='Т а г А З')
```



The image shows a screenshot of a database query result. At the top, the SQL query is partially visible: `select count(t.id) from`. Below the query, there is a table with two columns. The first column contains the value '1', and the second column contains the value '7 050'. The table has a light blue header row and a white body row. The text '123 count' is visible in the header row, and '7 050' is in the body row.

123 count
7 050

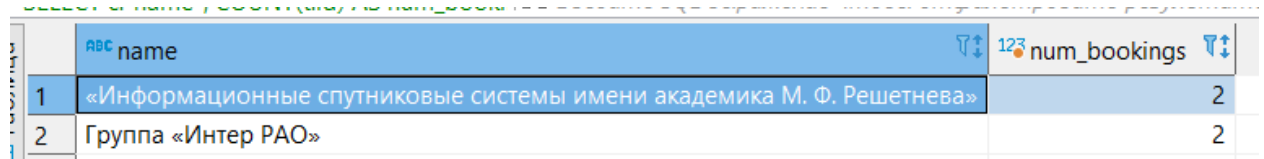
Рисунок 18 – Результат отчёта 1

Отчёт 2

Задача: Сколько раз каждая турфирма бронировала комнату 67119.

Листинг 16 – Седьмой SQL-запрос

```
SELECT c."name", COUNT(t.id) AS num_bookings
FROM company c
JOIN turbooking t ON c.id = t.company_id
JOIN rooms ON t."number" = rooms."number"
WHERE rooms."number" = '67119'
GROUP BY c."name";
```



	name	num_bookings
1	«Информационные спутниковые системы имени академика М. Ф. Решетнева»	2
2	Группа «Интер РАО»	2

Рисунок 19 – Результат седьмого SQL-запроса

ЗАКЛЮЧЕНИЕ

Во время выполнения работы выполнено следующее:

Проведен анализ предметной области, выделены основные сущности и процессы;

Спроектирована база данных, разработаны инфологическая, даталогическая модели базы данных;

Написан скрипт создания таблиц базы данных;

Разработаны и написаны сложные SQL-запросы и отчёты для получения статистической информации;

В результате курсовой работы, была реализована система, позволяющая обеспечить доступность информации о номерах гостиницы, пребываниях в них, их бронированиях и оплат, постояльцах и турфирмах, доп услугах. Система позволяет отслеживать и анализировать брони, пребывания и номера гостиницы.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. "Базы данных: проектирование, реализация, использование" К. Дейт, П. Коллинс. Издательство "Вильямс", 2015 г.
2. "Базы данных. Полное руководство" У. Коннолли, К. Бегг. Издательство "Вильямс", 2018 г.
3. "Реляционные базы данных и SQL" К. Бегг, Т. Криннин. Издательство "Вильямс", 2016 г.
4. "DBeaver Documentation" Команда разработчиков DBeaver, 2023 г.
5. Лутц М. Изучаем Python / М. Лутц. - М.: ДМК Пресс, 2013. - 1344 с.