**ЛАБОРАТОРНАЯ РАБОТА №1**

**«Язык SQL. Манипулирование базой данных. Запросы на основе нескольких таблиц»**

**1.1 Цель работы**

Научиться анализировать предметную область с целью создания схемы БД, учитывая ссылочную целостность набора. Изучить способы получения информации из нескольких таблиц, способы выполнения и принцип действия рекурсивных запросов и научится использовать вложенные подзапросы.

**1.2 Индивидуальный вариант**

Вариант 9: База данных туристического агентства:

Направления (id\_направления, название, страна, описание, популярность);

Отели (id\_отеля, название, адрес, звездность, рейтинг, услуги);

Полеты (id\_полета, номер\_рейса, дата\_отправления, дата\_прибытия, цена, авиакомпания);

Бронирования (id\_бронирования, id\_направления, id\_отеля, id\_полета, имя, фамилия, дата\_бронирования, статус);

Отзывы (id\_отзыва, id\_бронирования, текст\_отзыва, оценка, дата\_отзыва);

**1.3 Ход выполнения работы**

База данных туристического агентства состоит из 5 сущностей, таких как: направление, отель, полёт, бронирование, отзыв. Свойства сущностей были даны в задании по варианту.

Далее были выделены и классифицированы существующие связи:

1. Связь Направления – Бронирования

Связь типа 1:М, так как много бронирований могут быть на одно направление, связь является не идентифицирующей, так как сущности независимы между собой. Отношение Бронирования связано с отношением Направления внешним ключом (id\_направления).

2. Связь Отели – Бронирования

Связь типа 1:М, так как много бронирований могут быть на один и тот же отель, связь является не идентифицирующей, так как сущности независимы между собой. Отношение Бронирования связано с отношением Отели внешним ключом (id\_отеля).

3. Связь Полёты – Бронирования

Связь типа 1:М, так как много бронирований могут быть на один и тот же полёт, связь является не идентифицирующей, так как сущности независимы между собой. Отношение Бронирования связано с отношением Полёты внешним ключом (id\_полёта).

4. Связь Бронирования – Отзывы

Связь типа 1:М, так как у одного бронирования может быть много отзывов, связь является идентифицирующей, так как сущность отзыва зависит от сущности бронирования. Отношение Отзывы связано с отношением Бронирования внешним ключом (id\_бронирования), который входит в состав первичного ключа сущности отзыва.

Далее были выделены необходимые ограничения целостности. Их первичных ключей были выделены следующие атрибуты: id в таблице Направления, id в таблице Отели, id в таблице Полёты, id в таблице Бронирования, id и booking\_id в таблице Отзывы.

Ключи, которые было решено сделать внешними: id\_отеля в таблице Бронирования ссылается на id в таблице Отели, id\_направления в таблице Бронирования ссылается на id в таблице Направления, id\_полёта в таблице Бронирования ссылается на id в таблице Полёты, id\_бронирования в таблице Отзывы ссылается на id в таблице Бронирования и входит в состав составного ключа сущности отзыва.

Для атрибута стоимость в таблице Полёты была сделана проверка на ввод числа больше 0, также для таблицы Отзывы была добавлена проверка оценки, которая может быть от 0 до 10. Для таблицы Бронирования были установлены правила SET NULL при удалении записи и CASCADE при обновлении записи из таблиц Направления, Отели и Полёты. Также для таблицы были установлены правила каскадного обновления и удаления записей при изменении в таблице Бронирования.

На основе анализа описания БД была построена схема в нотации IDEF1X, что показано на рисунке 1.1.

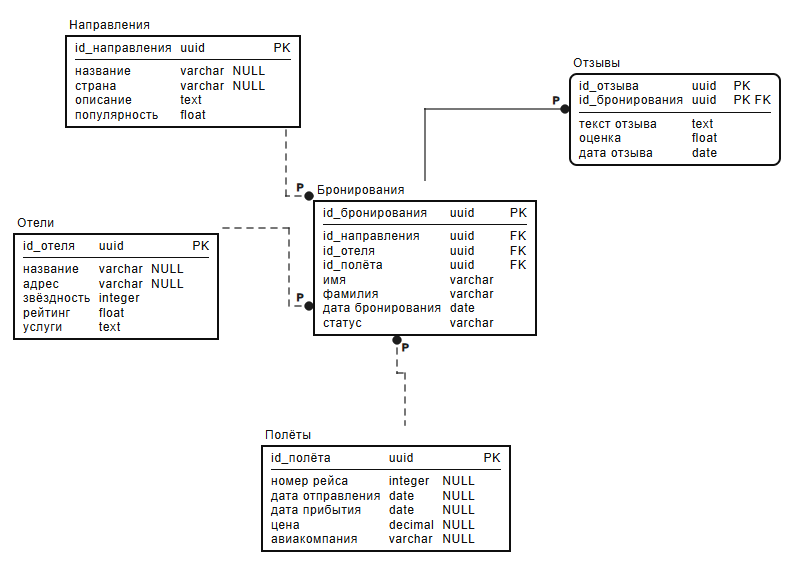


Рисунок 1.1 – Схема базы данных растений и их ухода

1.3.2 Далее с использованием PostgreSQL были созданы все таблицы, с необходимыми ограничениями целостности.

Для того чтобы работать с uuid было добавлено расширение "uuid-ossp" (Листинг 1.1)

Листинг 1.1 – Добавление расширения "uuid-ossp"

CREATE EXTENSION "uuid-ossp";

Запрос на создание таблицы Отели отображен в листинге 1.2.

Листинг 1.2 – Запрос для создания таблицы Отели

CREATE TABLE HOTELS(

id uuid DEFAULT uuid\_generate\_v4() PRIMARY KEY,

name varchar(30),

address varchar(50),

stars integer,

rating float,

services text

);

Создание таблицы Направления показано в листинге 1.3.

Листинг 1.3 – Запрос для создания таблицы Направления

CREATE TABLE directions(

id uuid DEFAULT uuid\_generate\_v4() PRIMARY KEY,

name varchar(30) NOT NULL,

country varchar(30) NOT NULL,

description text,

popularity float

);

Листинг 1.4 содержит запрос для создания таблицы Полёты.

Листинг 1.4 – Запрос для создания таблицы Полёты

CREATE TABLE flights(

id uuid DEFAULT uuid\_generate\_v4() PRIMARY KEY,

number int NOT NULL,

departure\_date date NOT NULL,

arrival\_date date NOT NULL,

cost decimal NOT NULL,

company varchar(30) NOT NULL,

CONSTRAINT cost\_above\_zero CHECK (cost > 0)

);

Запрос для создания таблицы Бронирования показан на листинге 1.5.

Листинг 1.5 – Запрос для создания таблицы Бронирования

CREATE TABLE bookings(

id uuid DEFAULT uuid\_generate\_v4() PRIMARY KEY,

direction\_id uuid,

hotel\_id uuid,

flight\_id uuid,

name varchar(30),

surname varchar(30),

booking\_date date,

status varchar(15),

FOREIGN KEY (direction\_id) REFERENCES directions(id) ON DELETE SET NULL ON UPDATE CASCADE,

FOREIGN KEY (hotel\_id) REFERENCES hotels(id) ON DELETE SET NULL ON UPDATE CASCADE,

FOREIGN KEY (flight\_id) REFERENCES flights(id) ON DELETE SET NULL ON UPDATE CASCADE

);

Запрос для создания таблицы Отзывы показан на листинге 1.6.

Листинг 1.6 – Запрос для создания таблицы Бронирования

CREATE TABLE reviews(

id uuid DEFAULT uuid\_generate\_v4(),

booking\_id uuid,

review\_text text,

grade float,

review\_date date,

PRIMARY KEY (id, booking\_id),

FOREIGN KEY (booking\_id) REFERENCES bookings(id) ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT grade\_constraint CHECk (grade >= 0 and grade <= 10)

);

1.3.3 Созданные ранее таблицы были заполнены данными. Листинг 1.7 содержит запрос на заполнение данными таблицы Отели 15 записями. На рисунке 1.2 показана итоговая заполненная таблица.

Листинг 1.7 – Запрос на заполнение данными таблицы Отели

INSERT INTO HOTELS (name, address, stars, rating, services) VALUES

('Hotel Sunshine', '123 Beach Ave', 5, 9.1, 'WiFi, Pool, Spa'),

('Mountain Retreat', '456 Hill St', 4, 8.5, 'WiFi, Breakfast, Parking'),

('City Central', '789 Main Rd', 3, 7.8, 'WiFi, Gym'),

('Ocean View', '321 Ocean Dr', 5, 9.3, 'WiFi, Pool, Gym, Spa'),

('Budget Inn', '654 Budget Ln', 2, 6.4, 'WiFi, Parking'),

('Luxury Suites', '987 Luxury Blvd', 5, 9.7, 'WiFi, Pool, Spa, Gym, Breakfast'),

('Country Lodge', '111 Country Rd', 4, 8.2, 'WiFi, Breakfast, Parking'),

('Urban Stay', '222 City St', 3, 7.5, 'WiFi, Gym, Parking'),

('Seaside Hotel', '333 Seaside Ave', 4, 8.9, 'WiFi, Pool, Breakfast'),

('Downtown Hotel', '444 Downtown St', 3, 7.2, 'WiFi, Parking'),

('Airport Hotel', '555 Airport Rd', 4, 8.0, 'WiFi, Breakfast, Parking'),

('Historic Hotel', '666 Historic Ln', 5, 9.0, 'WiFi, Spa, Breakfast'),

('Business Hotel', '777 Business Blvd', 4, 8.3, 'WiFi, Gym, Parking'),

('Family Hotel', '888 Family St', 3, 7.6, 'WiFi, Pool, Breakfast'),

('Boutique Hotel', '999 Boutique Ave', 5, 9.5, 'WiFi, Spa, Gym, Breakfast');

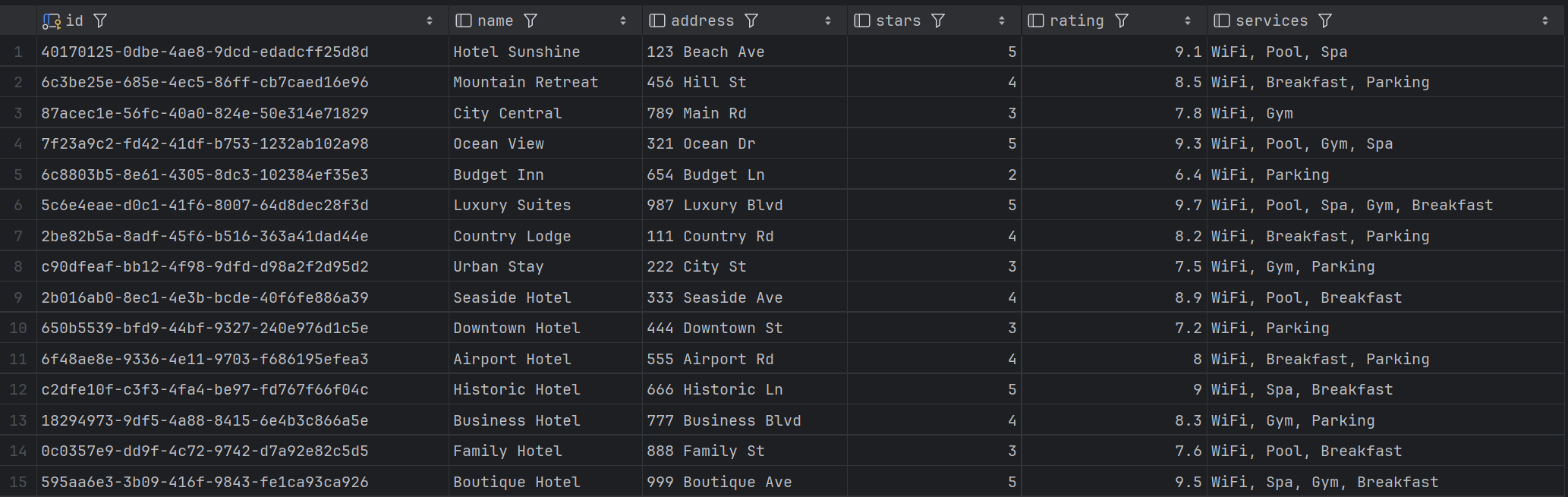


Рисунок 1.2 – Таблица Отели

Запрос на заполнение данными таблицы 15 записями Направления показан в листинге 1.8. Рисунок 1.3 показывает заполненную таблицу Направления.

Листинг 1.8 – Запрос на заполнение данными таблицы Нарпавления

INSERT INTO directions (name, country, description, popularity) VALUES

('Paris', 'France', 'City of Light', 9.8),

('Tokyo', 'Japan', 'Land of the Rising Sun', 9.5),

('New York', 'USA', 'The Big Apple', 9.7),

('London', 'UK', 'Capital of England', 9.6),

('Sydney', 'Australia', 'Harbour City', 9.4),

('Rome', 'Italy', 'The Eternal City', 9.3),

('Berlin', 'Germany', 'Capital of Germany', 9.2),

('Bangkok', 'Thailand', 'City of Angels', 9.1),

('Dubai', 'UAE', 'City of Gold', 9.0),

('Barcelona', 'Spain', 'City of Gaudi', 8.9),

('Moscow', 'Russia', 'Capital of Russia', 8.8),

('Istanbul', 'Turkey', 'City of Two Continents', 8.7),

('Rio de Janeiro', 'Brazil', 'Marvelous City', 8.6),

('Cape Town', 'South Africa', 'Mother City', 8.5),

('Hong Kong', 'China', 'Asia’s World City', 8.4);

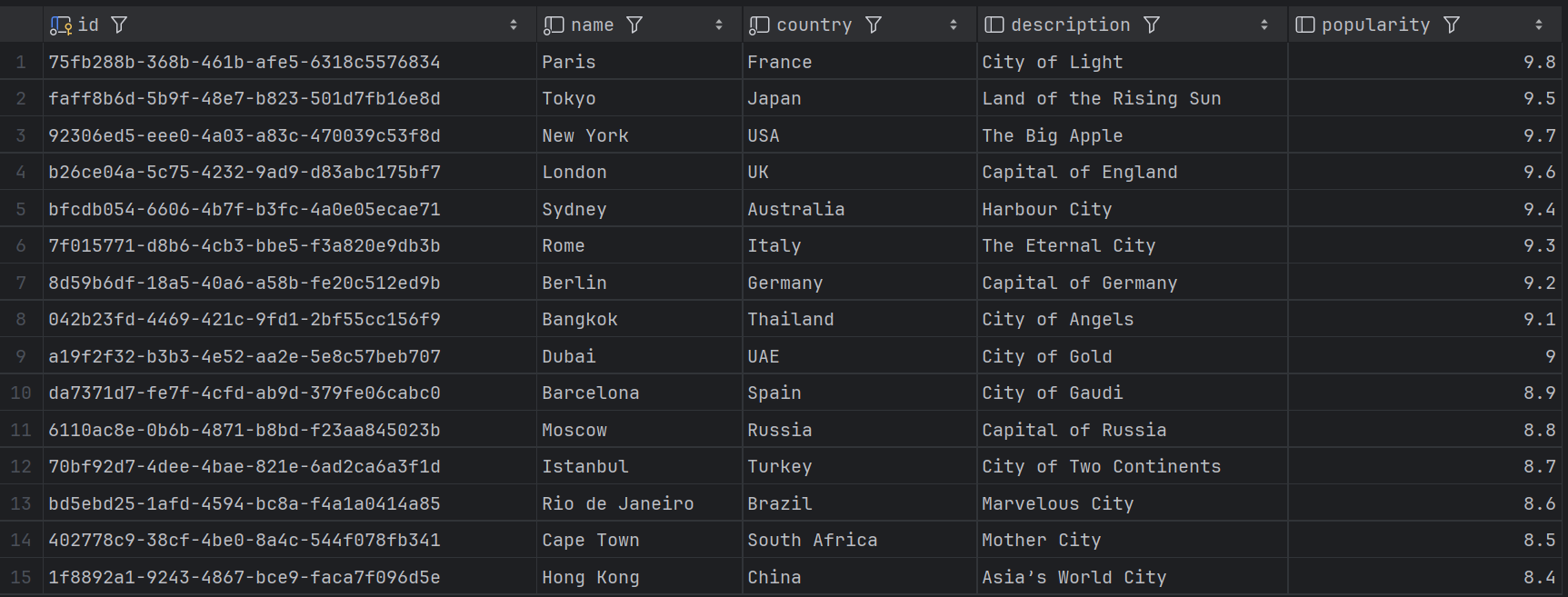


Рисунок 1.3 – Таблица Направления

Листинг 1.9 содержит запрос на запись 15 строк в таблицу Полёты. На рисунке 1.4 видно заполненную запросом таблицу.

Листинг 1.9 – Запрос на заполнение данными таблицы Полёты

INSERT INTO flights (number, departure\_date, arrival\_date, cost, company) VALUES

(101, '2024-10-01', '2024-10-02', 500.00, 'Airline A'),

(102, '2024-10-03', '2024-10-04', 450.00, 'Airline B'),

(103, '2024-10-05', '2024-10-06', 600.00, 'Airline C'),

(104, '2024-10-07', '2024-10-08', 550.00, 'Airline D'),

(105, '2024-10-09', '2024-10-10', 700.00, 'Airline E'),

(106, '2024-10-11', '2024-10-12', 650.00, 'Airline F'),

(107, '2024-10-13', '2024-10-14', 400.00, 'Airline G'),

(108, '2024-10-15', '2024-10-16', 750.00, 'Airline H'),

(109, '2024-10-17', '2024-10-18', 800.00, 'Airline I'),

(110, '2024-10-19', '2024-10-20', 850.00, 'Airline J'),

(111, '2024-10-21', '2024-10-22', 900.00, 'Airline K'),

(112, '2024-10-23', '2024-10-24', 950.00, 'Airline L'),

(113, '2024-10-25', '2024-10-26', 1000.00, 'Airline M'),

(114, '2024-10-27', '2024-10-28', 1050.00, 'Airline N'),

(115, '2024-10-29', '2024-10-30', 1100.00, 'Airline O');

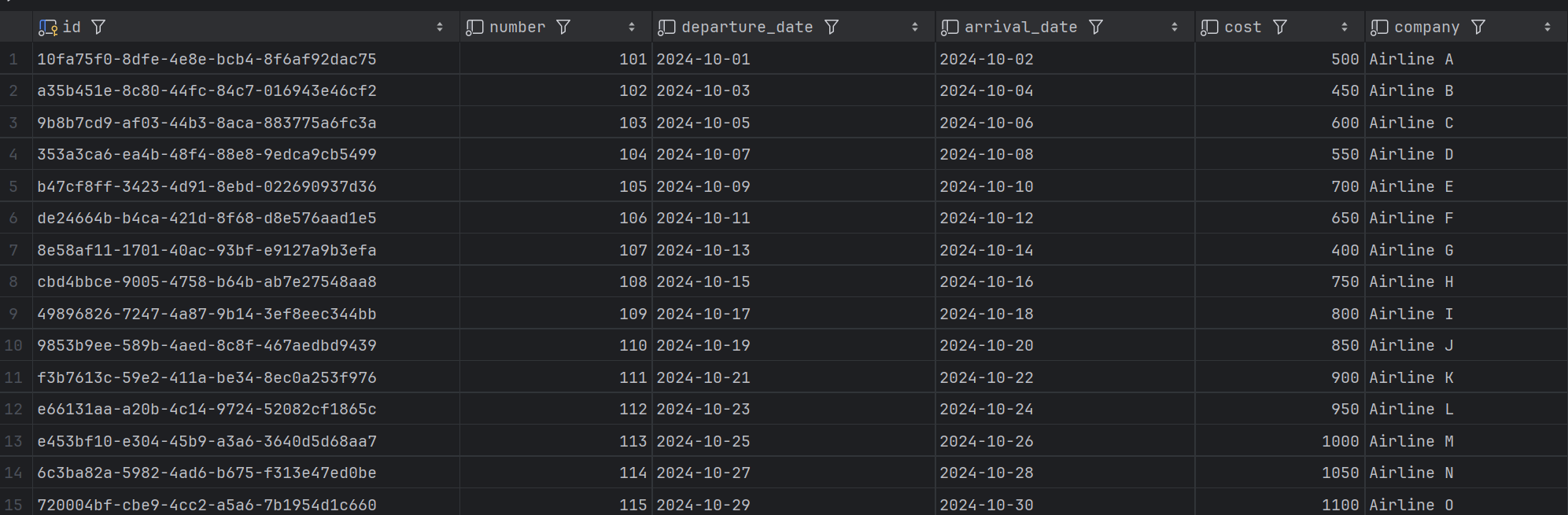


Рисунок 1.4 – Таблица Полёты

Внесение 15 записей в таблицу Бронирования показано в листинге 1.10. Рисунок 1.5 демонстрирует заполненную запросом таблицу.

Листинг 1.10 – Запрос на заполнение данными таблицы Бронирования

INSERT INTO bookings (direction\_id, hotel\_id, flight\_id, name, surname, booking\_date, status) VALUES

('75fb288b-368b-461b-afe5-6318c5576834', '40170125-0dbe-4ae8-9dcd-edadcff25d8d', '10fa75f0-8dfe-4e8e-bcb4-8f6af92dac75', 'John', 'Doe', '2024-09-01', 'Confirmed'),

('faff8b6d-5b9f-48e7-b823-501d7fb16e8d', '6c3be25e-685e-4ec5-86ff-cb7caed16e96', 'a35b451e-8c80-44fc-84c7-016943e46cf2', 'Jane', 'Smith', '2024-09-02', 'Pending'),

('92306ed5-eee0-4a03-a83c-470039c53f8d', '87acec1e-56fc-40a0-824e-50e314e71829', '9b8b7cd9-af03-44b3-8aca-883775a6fc3a', 'Alice', 'Johnson', '2024-09-03', 'Cancelled'),

('b26ce04a-5c75-4232-9ad9-d83abc175bf7', '7f23a9c2-fd42-41df-b753-1232ab102a98', '353a3ca6-ea4b-48f4-88e8-9edca9cb5499', 'Bob', 'Brown', '2024-09-04', 'Confirmed'),

('bfcdb054-6606-4b7f-b3fc-4a0e05ecae71', '6c8803b5-8e61-4305-8dc3-102384ef35e3', 'b47cf8ff-3423-4d91-8ebd-022690937d36', 'Charlie', 'Davis', '2024-09-05', 'Pending'),

('7f015771-d8b6-4cb3-bbe5-f3a820e9db3b', '5c6e4eae-d0c1-41f6-8007-64d8dec28f3d', 'de24664b-b4ca-421d-8f68-d8e576aad1e5', 'David', 'Wilson', '2024-09-06', 'Cancelled'),

('8d59b6df-18a5-40a6-a58b-fe20c512ed9b', '2be82b5a-8adf-45f6-b516-363a41dad44e', '8e58af11-1701-40ac-93bf-e9127a9b3efa', 'Eve', 'Taylor', '2024-09-07', 'Confirmed'),

('042b23fd-4469-421c-9fd1-2bf55cc156f9', 'c90dfeaf-bb12-4f98-9dfd-d98a2f2d95d2', 'cbd4bbce-9005-4758-b64b-ab7e27548aa8', 'Frank', 'Anderson', '2024-09-08', 'Pending'),

('a19f2f32-b3b3-4e52-aa2e-5e8c57beb707', '2b016ab0-8ec1-4e3b-bcde-40f6fe886a39', '49896826-7247-4a87-9b14-3ef8eec344bb', 'Grace', 'Thomas', '2024-09-09', 'Cancelled'),

('da7371d7-fe7f-4cfd-ab9d-379fe06cabc0', '650b5539-bfd9-44bf-9327-240e976d1c5e', '9853b9ee-589b-4aed-8c8f-467aedbd9439', 'Hank', 'Jackson', '2024-09-10', 'Confirmed'),

('6110ac8e-0b6b-4871-b8bd-f23aa845023b', '6f48ae8e-9336-4e11-9703-f686195efea3', 'f3b7613c-59e2-411a-be34-8ec0a253f976', 'Ivy', 'White', '2024-09-11', 'Pending'),

('70bf92d7-4dee-4bae-821e-6ad2ca6a3f1d', 'c2dfe10f-c3f3-4fa4-be97-fd767f66f04c', 'e66131aa-a20b-4c14-9724-52082cf1865c', 'Jack', 'Harris', '2024-09-12', 'Cancelled'),

('bd5ebd25-1afd-4594-bc8a-f4a1a0414a85', '18294973-9df5-4a88-8415-6e4b3c866a5e', 'e453bf10-e304-45b9-a3a6-3640d5d68aa7', 'Karen', 'Martin', '2024-09-13', 'Confirmed'),

('402778c9-38cf-4be0-8a4c-544f078fb341', '0c0357e9-dd9f-4c72-9742-d7a92e82c5d5', '6c3ba82a-5982-4ad6-b675-f313e47ed0be', 'Leo', 'Garcia', '2024-09-14', 'Pending'),

('1f8892a1-9243-4867-bce9-faca7f096d5e', '595aa6e3-3b09-416f-9843-fe1ca93ca926', '720004bf-cbe9-4cc2-a5a6-7b1954d1c660', 'Mia', 'Martinez', '2024-09-15', 'Cancelled');

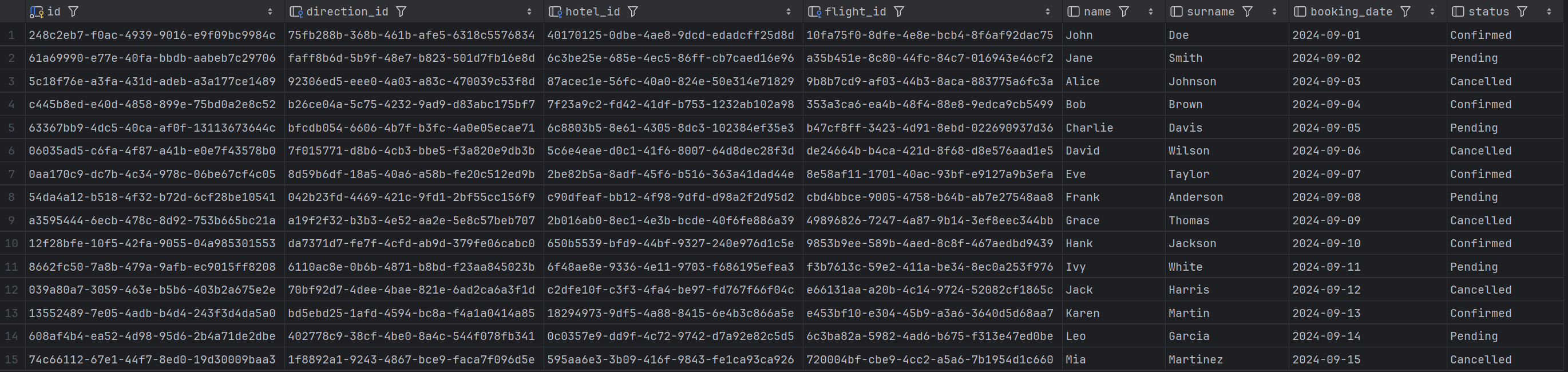


Рисунок 1.5 – Таблица Бронирования

Далее было внесено 15 записей в таблицу Отзывы, запрос показан в листинге 1.11. На рисунке 1.6 показан результат внесенных данных в виде заполненной таблицы.

Листинг 1.11 – Запрос на заполнение данными таблицы Отзывы

INSERT INTO reviews (booking\_id, review\_text, grade, review\_date) VALUES

('248c2eb7-f0ac-4939-9016-e9f09bc9984c', 'Great experience!', 9.0, '2024-09-16'),

('61a69990-e77e-40fa-bbdb-aabeb7c29706', 'Good service.', 8.0, '2024-09-17'),

('5c18f76e-a3fa-431d-adeb-a3a177ce1489', 'Average stay.', 6.0, '2024-09-18'),

('c445b8ed-e40d-4858-899e-75bd0a2e8c52', 'Not satisfied.', 4.0, '2024-09-19'),

('63367bb9-4dc5-40ca-af0f-13113673644c', 'Excellent!', 10.0, '2024-09-20'),

('06035ad5-c6fa-4f87-a41b-e0e7f43578b0', 'Could be better.', 5.0, '2024-09-21'),

('0aa170c9-dc7b-4c34-978c-06be67cf4c05', 'Loved it!', 9.5, '2024-09-22'),

('54da4a12-b518-4f32-b72d-6cf28be10541', 'Just okay.', 7.0, '2024-09-23'),

('a3595444-6ecb-478c-8d92-753b665bc21a', 'Terrible experience.', 2.0, '2024-09-24'),

('12f28bfe-10f5-42fa-9055-04a985301553', 'Fantastic!', 9.8, '2024-09-25'),

('8662fc50-7a8b-479a-9afb-ec9015ff8208', 'Good value.', 8.5, '2024-09-26'),

('039a80a7-3059-463e-b5b6-403b2a675e2e', 'Not worth it.', 3.0, '2024-09-27'),

('13552489-7e05-4adb-b4d4-243f3d4da5a0', 'Very nice.', 8.8, '2024-09-28'),

('608af4b4-ea52-4d98-95d6-2b4a71de2dbe', 'Disappointing.', 4.5, '2024-09-29'),

('74c66112-67e1-44f7-8ed0-19d30009baa3', 'Highly recommend!', 9, '2024-09-30');

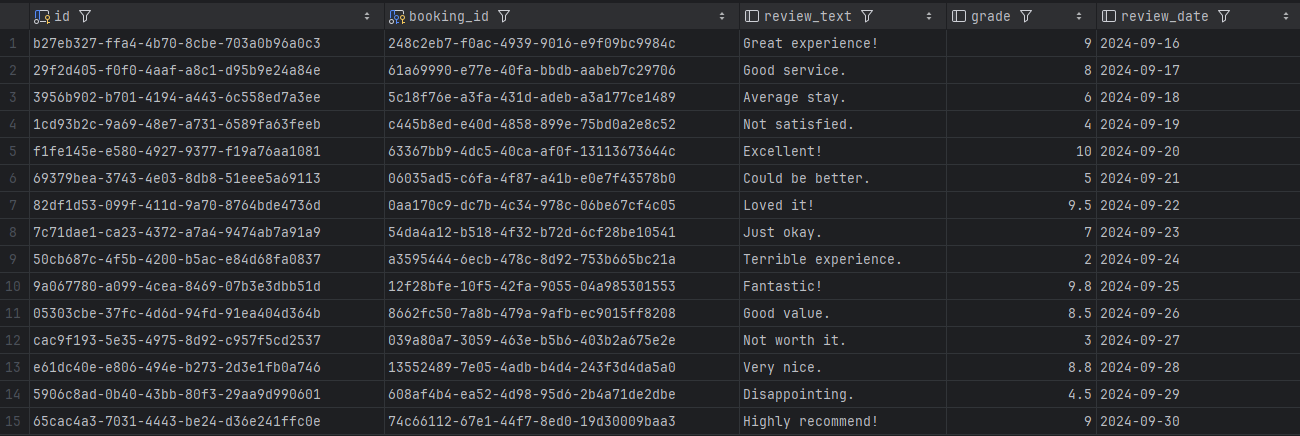


Рисунок 1.6 – Таблица Отзывы

1.3.4 Была выполнена проверка ограничений целостности, добавленных при создании таблиц. При попытке добавить в таблицу Полёты строку, содержащую отрицательное или нулевое значение атрибута цена, появляется ошибка, связанная с ограничением CHECK cost\_above\_zero (рисунок 1.7).



Рисунок 1.7 – Ошибка при проверке ограничения cost\_above\_zero таблицы Полёты

При попытке добавить в таблицу Отзывы строку с неверным значением поля grade, тоже появляется ошибка из-за ограничения grade\_constraint, что показано на рисунке 1.8.

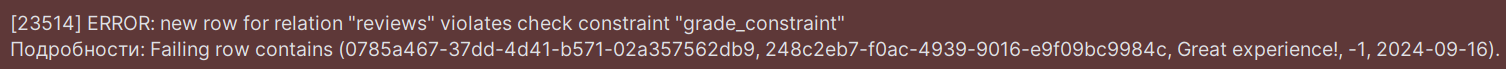


Рисунок 1.8 – Ошибка при проверке ограничения grade\_constraint таблицы Отзывы

При модификации атрибута id в таблицах Направления, Полёты, Отели происходят соответствующие модификации в таблице Бронирования, связанные с ограничением ON UPDATE CASCADE. Строки таблицы Бронирования до изменения показаны на рисунке 1.9, а после изменения – на рисунке 1.10. При удалении строк из таблиц Направления, Полёты, Отели соответствующие им поля записи из таблицы Бронирования будут установлены в значение NULL из-за ограничения ON DELETE SET NULL. Также записи таблицы Отзывы будут каскадно обновляться и удаляться.

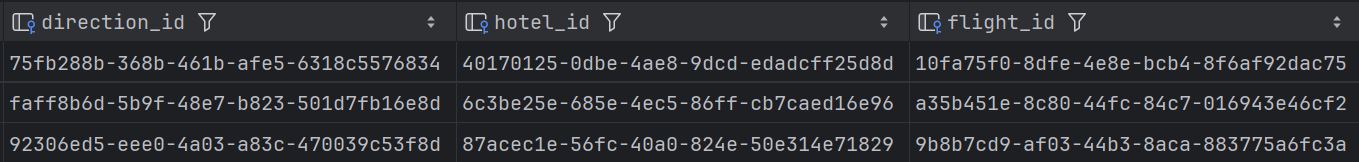


Рисунок 1.9 – Строки таблицы Бронирования до изменения

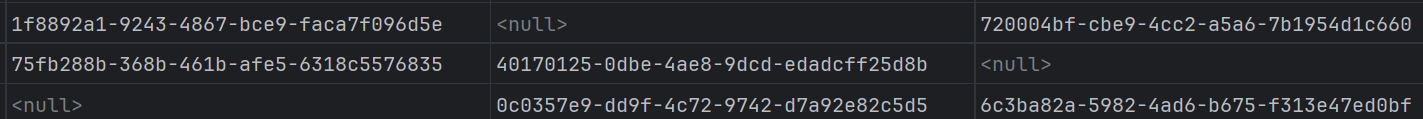


Рисунок 1.10 – Строки таблицы Бронирования после изменения

1.3.5 Далее представлены запросы на выборку из базы данных. Запрос, содержащий простые вычисления: необходимо вывести список всех полётов с ценой в тенге. Текст запроса показан в листинге 1.12, результат выполнения на рисунке 1.12.

Листинг 1.12 ­– Запрос, содержащий простые вычисления

SELECT number, company, cost\*5.09 from flights;



Рисунок 1.11 – Результат вывода запроса с простыми вычислениями

Запрос, содержащий простые вычисления как параметр агрегатной функции: вычислить среднюю продолжительность рейсов. Текст запроса показан в листинге 1.13, результат выполнения на рисунке 1.12.

Листинг 1.13 ­– Запрос, содержащий простые вычисления в агрегатной функции

SELECT AVG(arrival\_date - departure\_date) AS average\_duration FROM flights;

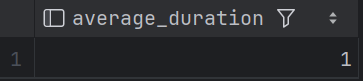


Рисунок 1.12 – Результат вывода запроса с простыми вычислениями в агрегатной функции

Запрос, содержащий предложение GROUP BY: узнать количество отзывов с одинаковой оценкой. Текст запроса показан в листинге 1.14, результат выполнения на рисунке 1.13.

Листинг 1.14 ­– Запрос, содержащий предложение GROUP BY

SELECT grade, count(\*) from reviews GROUP BY grade;

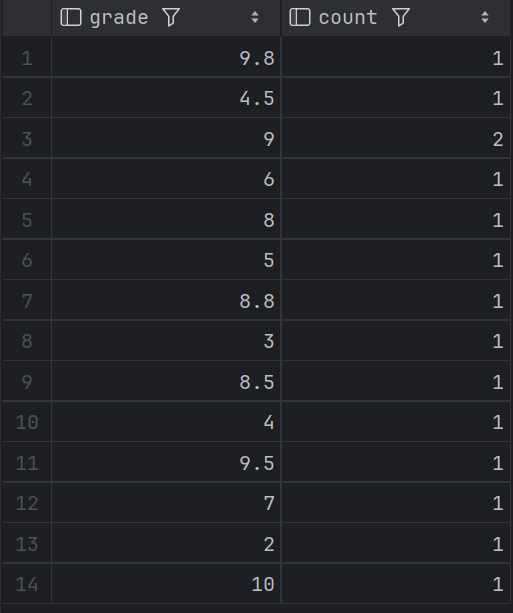


Рисунок 1.13 – Результат вывода запроса с предложением GROUP BY

Запрос, содержащий предложение HAVING: вывести какую оценку ставили в отзыве более одного раза. Текст запроса показан в листинге 1.15, результат выполнения на рисунке 1.14.

Листинг 1.15 ­– Запрос, содержащий предложение HAVING

SELECT grade, count(\*) from reviews GROUP BY grade HAVING count(\*) > 1;

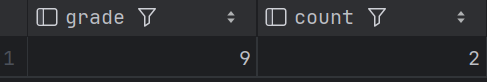


Рисунок 1.14 – Результат вывода запроса с предложением HAVING

Запрос, соединяющий две таблицы с помощью JOIN: вывести все бронирования на рейс номер 105. Текст запроса показан в листинге 1.16, результат выполнения на рисунке 1.15.

Листинг 1.16 ­– Запрос, соединяющий две таблицы с помощью JOIN

SELECT flights.number, bookings.name, flight\_id, bookings.id as booking\_id from flights JOIN bookings ON flights.id = bookings.flight\_id WHERE flights.number = 105;

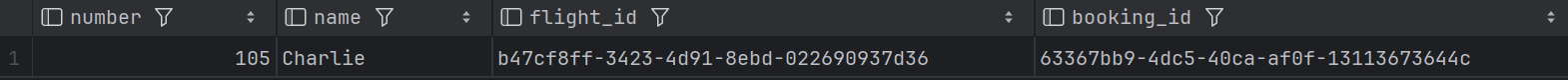


Рисунок 1.15 – Результат вывода запроса, соединяющего две таблицы с помощью JOIN

Тот же запрос без использования JOIN показан в листинге 1.17, результат выполнения на рисунке 1.16.

Листинг 1.17 ­– Запрос, соединяющий две таблицы без JOIN

SELECT flights.number, bookings.name, flight\_id, bookings.id as booking\_id from flights, bookings WHERE flights.number = 105 AND flights.id = bookings.flight\_id;

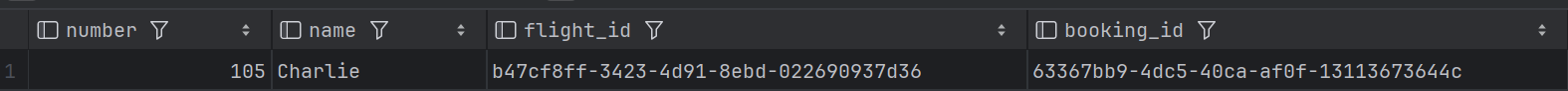


Рисунок 1.16 – Результат вывода запроса, соединяющего две таблицы без JOIN

Запрос, соединяющий больше, чем две таблицы с помощью JOIN: вывести все отзывы на бронирования на рейс номер 105. Текст запроса показан в листинге 1.18, результат выполнения на рисунке 1.17.

Листинг 1.18 ­– Запрос, соединяющий больше, чем две таблицы с помощью JOIN

SELECT flights.number, bookings.name, review\_text, grade from flights JOIN bookings ON flights.id = bookings.flight\_id JOIN reviews ON reviews.booking\_id = bookings.id WHERE flights.number = 105;

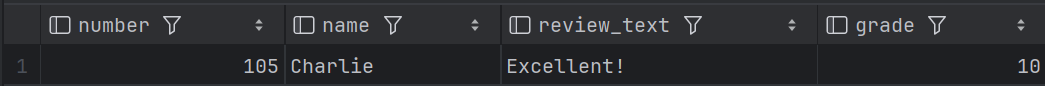


Рисунок 1.17 – Результат вывода запроса, соединяющего больше, чем две таблицы с помощью JOIN

Тот же запрос без использования JOIN показан в листинге 1.19, результат выполнения на рисунке 1.18.

Листинг 1.19 ­– Запрос, соединяющий больше, чем две таблицы без JOIN

SELECT flights.number, bookings.name, review\_text, grade from flights, bookings, reviews WHERE flights.id = bookings.flight\_id AND reviews.booking\_id = bookings.id AND flights.number = 105;

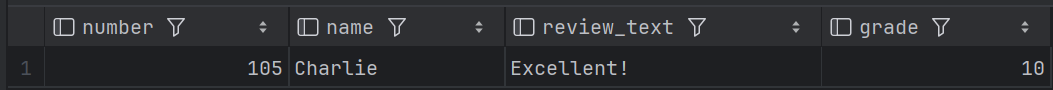


Рисунок 1.18 – Результат вывода запроса, соединяющего больше, чем две таблицы без JOIN

**1.4 Вывод**

В ходе выполнения работы было проанализировано описание базы данных туристического агенства. Были выделены и классифицированы все существующие связи, определены необходимые ограничения целостности. Также была построена схема базы данных в нотации IDEF1X. С помощью PostrgreSQL были написаны запросы на создание необходимых таблиц, создание определенных ранее ограничений. Таблицы были заполнены с использованием соотвествующего оператора. Далее была протестирована работа ограничений целостности. Были написаны запросы на выборку данных с использованием простых вычислений, агрегатных функций, предложений GROUP BY и HAVING. Также были рассмотрены способы соединения двух и более таблиц с использованием JOIN и без него.