Федеральное государственное автономное образовательное учреждение высшего образования национальный исследовательский университет ИТМО

Факультет программной инженерии и компьютерной техники

Дисциплина «базы данных»

Отчёт

По лабораторной работе №4

Исполнитель:
Назирджонов Некруз
группа: P33211

Санкт-Петербург
2023

Для выполнения лабораторной работы №4 необходимо:

- Реализовать разработанную в рамках лабораторной работы №3 даталогическую модель в реляционной СУБД PostgreSQL.
- Заполнить созданные таблицы данными.
- Обеспечить целостность данных при помощи средств языка DDL.
- В рамках лабораторной работы должны быть разработаны скрипты для создания/удаления требуемых объектов базы данных, заполнения/удаления содержимого созданных таблиц.

Отчёт по лабораторной работе должен содержать:

- титульный лист;
- текст задания;
- описание предметной области;
- DDL-скрипты, часть DML-скриптов;
- выводы по работе;

```
-- Создание таблицы для путешественников
CREATE TABLE Traveler (
traveler_id SERIAL PRIMARY KEY,
first_name VARCHAR(50) NOT NULL,
last_name VARCHAR(50) NOT NULL,
contact_info TEXT CHECK (contact_info ~* '^[A-Za-z0-9._%-]+@[A-Za-z0-9.-]+\\.[A-Za-z]{2,4}$')
);

-- Создание таблицы для отелей
CREATE TABLE Hotel (
hotel_id SERIAL PRIMARY KEY,
name VARCHAR(100) NOT NULL,
address TEXT,
cost_per_night DECIMAL(10, 2)
);

-- Создание таблицы для музеев временных путешествий
CREATE TABLE TimeMuseum (
museum_id SERIAL PRIMARY KEY,
name VARCHAR(100) NOT NULL,
description TEXT,
address TEXT
);

CREATE TABLE TimeRoute (
route_id SERIAL PRIMARY KEY,
name VARCHAR(100) NOT NULL,
duration INT,
availability BOOLEAN

);

-- Создание таблицы для экскурсий
```

```
CREATE TABLE Excursion (
    excursion_id SERIAL PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    description TEXT,
    start_datetime TIMESTAMP,
    time_museum_id INT REFERENCES TimeMuseum(museum_id)
);


CREATE TABLE Tour (
tour_id SERIAL PRIMARY KEY,
name VARCHAR(100) NOT NULL,
description TEXT,
time_route_id INT,
excursion_id INT,
hotel_id INT,
FOREIGN KEY (time_route_id) REFERENCES TimeRoute(route_id),
FOREIGN KEY (excursion_id) REFERENCES Excursion(excursion_id),
FOREIGN KEY (hotel_id) REFERENCES Hotel(hotel_id)
);









-- Создание таблицы для гидов
CREATE TABLE Guide (
guide_id SERIAL PRIMARY KEY,
first_name VARCHAR(50) NOT NULL,
last_name VARCHAR(50) NOT NULL,
experience INT CHEK (experience > 0),
specialization TEXT,
traveler_id INT REFERENCES Traveler(traveler_id)
);




-- Создание таблицы для эпох
CREATE TABLE Era ( era_id SERIAL PRIMARY KEY, name VARCHAR(100) NOT NULL, description TEXT, time_range TEXT, time_route_id INT REFERENCES TimeRoute(route_id) );

-- Создание таблицы для билетов
CREATE TABLE Ticket ( ticket_id SERIAL PRIMARY KEY, ticket_type VARCHAR(20), cost DECIMAL(10, 2), start_datetime TIMESTAMP, tour_id INT, FOREIGN KEY (tour_id) REFERENCES Tour(tour_id) );

-- Создание таблицы для бронирований
CREATE TABLE Booking (
booking_id SERIAL PRIMARY KEY,
booking_datetime TIMESTAMP,
start_datetime TIMESTAMP,
end_datetime TIMESTAMP,
status VARCHAR(20),
```

```
traveler_id INT,
ticket_id INT,
FOREIGN KEY (traveler_id) REFERENCES Traveler(traveler_id),
FOREIGN KEY (ticket_id) REFERENCES Ticket(ticket_id),
   CHECK (start_datetime < end_datetime),
   CHECK (booking_datetime <= start_datetime)
);

ALTER TABLE Traveler
ADD COLUMN guide_id INT,
ADD FOREIGN KEY (guide_id) REFERENCES Guide(guide_id);

-- Изменение таблицы "Tour" для связи с "Booking" вместо "Hotel"
ALTER TABLE Tour
DROP COLUMN hotel_id, -- удаляем связь с отелем
ADD COLUMN booking_id INT, -- добавляем связь с бронированием
ADD FOREIGN KEY (booking_id) REFERENCES Booking(booking_id); --
устанавливаем внешний ключ на бронирование

-- Добавление столбца hotel_id в таблицу Booking
ALTER TABLE Booking
ADD COLUMN hotel_id INT,
ADD FOREIGN KEY (hotel_id) REFERENCES Hotel(hotel_id);
```

```
-- Добавление столбца booking_id в таблицу Guide
ALTER TABLE Guide
ADD COLUMN booking_id INT,
ADD FOREIGN KEY (booking_id) REFERENCES Booking(booking_id);
```

```

INSERT INTO TimeRoute (name, duration, availability)
VALUES
('Маршрут 1', 2, true),
('Маршрут 2', 3, false),
('Маршрут 3', 4, true);

INSERT INTO Era (name, description, time_range, time_route_id)
 VALUES
('Эпоха 1', 'Древняя цивилизация', '1000 до н.э. - 500 н.э.', 1),
('Эпоха 2', 'Возрождение', '14-17 века', 2),
 ('Эпоха 3', 'Современная наука', '19-21 век', 3);

insert INTO TimeMuseum (name, description, address)
VALUES
('Museum X', 'Исторический музей', '456 Park Ave'),
('Museum Y', 'Художественный музей', '789 Museum Dr'),
('Museum Z', 'Научно-технический музей', '123 Science St');

```

```
INSERT INTO Excursion (name, description, start_datetime,
time_museum_id)
VALUES
('Экскурсия 1', 'Историческая экскурсия', '2023-11-10 10:00:00', 1),
('Экскурсия 2', 'Художественная экскурсия', '2023-11-12 14:30:00',
2),
('Экскурсия 3', 'Научно-техническая экскурсия', '2023-11-15
11:15:00', 3);
```

```
ALTER TABLE ticket
DROP COLUMN tour_id;
```

```
уд карам чунки дарокр не
```

```
INSERT INTO Hotel (name, address, cost_per_night)
VALUES
('Hotel A', '123 Main St', 150.00),
('Hotel B', '456 Elm St', 120.00),
('Hotel C', '789 Oak St', 180.00);
```

```
INSERT INTO Ticket (tour_id, cost)
VALUES
    ('Стандартный', 50.00),
    ('VIP', 80.00),
    ('Обычный', 40.00;
```

```
-- Заполнение таблицы "Booking"
INSERT INTO Booking (booking_datetime, start_datetime, end_datetime,
status, traveler_id, ticket_id, hotel_id)
VALUES
('2023-11-20 08:00:00', '2023-11-22 10:00:00', '2023-11-24 12:00:00',
'Забронировано', 10, 1, 1),
('2023-11-25 09:30:00', '2023-11-27 11:30:00', '2023-11-29 13:30:00',
'Забронировано', 19, 2, 2),
('2023-11-30 11:00:00', '2023-12-02 13:00:00', '2023-12-04 15:00:00',
'Забронировано', 20, 3, 3);
-- Заполнение таблицы "Tour" с реалистичными данными
INSERT INTO Tour (name, description, time_route_id, excursion_id,
booking_id)
VALUES
('Исторический тур по Риму', 'Изучение античной истории в вечном городе',
1, 1, 4),
```

```
('Искусство и культура в Париже', 'Погружение в мир искусства и культуры Франции', 2, 2, 5),
('Природное путешествие в Национальном парке Yellowstone', 'Исследование природных красот США', 3, 3, 6);
```

Выборка всех туров с их описанием и информацией о бронировании:

```
SELECT Tour.name, Tour.description, Booking.booking_datetime, Booking.status
FROM Tour
JOIN Booking ON Tour.booking_id = Booking.booking_id;
```

Выборка всех музеев и адресов:

```
SELECT name, address
FROM TimeMuseum;
```

3. Выборка информации о всех бронированиях и связанных с ними путешественниках:

```
SELECT Booking.booking_datetime, Booking.status, Traveler.first_name, Traveler.last_name
FROM Booking
JOIN Traveler ON Booking.traveler_id = Traveler.traveler_id;
```

Выборка доступных временных маршрутов:

```
SELECT name, duration
FROM TimeRoute
WHERE availability = true;
```

Выборка билетов с их типами и ценами:

```
SELECT ticket_type, cost
FROM Ticket;
```

Количество забронированных билетов

```
SELECT COUNT(*) AS total_bookings
FROM Booking
WHERE status = 'Забронировано';
```

Получение информации о доступных временных маршрутах и соответствующих экскурсиях:

```
SELECT TimeRoute.name AS time_route_name, TimeRoute.duration, Excursion.name AS excursion_name
FROM TimeRoute
LEFT JOIN Excursion ON TimeRoute.route_id = Excursion.time_museum_id;
```

Получение информации об эпохах и связанных с ними временных

маршрутах:
SELECT Era.name AS era_name, Era.description, TimeRoute.name AS time_route_name
FROM Era
LEFT JOIN TimeRoute ON Era.time_route_id = TimeRoute.route_id;

Получение информации о музеях и соответствующих экскурсиях:
SELECT TimeMuseum.name AS museum_name, TimeMuseum.description, Excursion.name AS excursion_name
FROM TimeMuseum
LEFT JOIN Excursion ON TimeMuseum.museum_id = Excursion.time_museum_id;
SELECT TimeRoute.name AS time_route_name, COUNT(Booking.booking_id) AS num_bookings
FROM TimeRoute
LEFT JOIN Tour ON TimeRoute.route_id = Tour.time_route_id
LEFT JOIN Booking ON Tour.booking_id = Booking.booking_id
GROUP BY TimeRoute.name;

```
-- Удаление таблицы Booking
DROP TABLE IF EXISTS Booking;

-- Удаление таблицы Ticket
DROP TABLE IF EXISTS Ticket;

-- Удаление таблицы Era
DROP TABLE IF EXISTS Era;

-- Удаление таблицы Guide
DROP TABLE IF EXISTS Guide;

-- Удаление таблицы Tour
DROP TABLE IF EXISTS Tour;

-- Удаление таблицы Excursion
DROP TABLE IF EXISTS Excursion;

-- Удаление таблицы TimeRoute
DROP TABLE IF EXISTS TimeRoute;

-- Удаление таблицы TimeMuseum
DROP TABLE IF EXISTS TimeMuseum;
```

```
-- Удаление таблицы Hotel
DROP TABLE IF EXISTS Hotel;

-- Удаление таблицы Traveler
DROP TABLE IF EXISTS Traveler;
```

Вывод

В ходе выполнения лабораторной работы №3, я углубил свои знания в области ER-диаграмм и их роли в моделировании информационных систем и баз данных. Сначала я ознакомился с основными понятиями, связанными с ER-диаграммами, включая сущности, атрибуты и связи. Затем, следуя инструкциям лабораторной работы, я научился создавать ER-диаграммы, идентифицировать сущности и их атрибуты, а также определять связи между сущностями. Это позволило мне более полно и точно моделировать предметную область. Кроме того, я изучил различные типы связей в базах данных, такие как один-к-одному, один-ко-многим и многие-ко-многим, и научился их использовать в ER-диаграммах. В процессе выполнения лабораторной работы, я также преобразовал инфологическую модель, созданную на ER-диаграмме, в даталогическую модель. Это практическое знание будет полезным для меня при разработке и проектировании баз данных в будущем.