CP 363 Assignment 7

Tourism Agency Booking System DBMS

Group 37

March 14, 2025

Hibah Hibah

Donil Patel

Normalization Process & Explanation

`Customer (customer_id, first_name, last_name, email, phone, address)`

- `customer_id → (first_name, last_name, email, phone, address)`
- `phone → (customer_id, first_name, last_name, email, address)`
- **Transitive Dependency:** `phone` determines customer details.

After

- `Customer (customer_id, first_name, last_name, email, address)`
- `Customer_Phone (phone, customer_id)`

`Payment (payment_id, booking_id, payment_date, payment_amount, payment_mode)`

- `payment_id → (booking_id, payment_date, payment_amount, payment_mode)`
- `booking_id → (payment_id, payment_date, payment_amount, payment_mode)`
- **Redundant Identifier:** `payment_id` is unnecessary as `booking_id` already determines payment details.

After

- `Payment (booking_id, payment_date, payment_amount, payment_mode)`

`Guide (guide_id, guide_name, contact_number, language, availability_status)`

- `guide_id → (guide_name, contact_number, language, availability_status)`
- `contact_number → (guide_id, guide_name, language, availability_status)`
- **Transitive Dependency:** `contact_number` determines `guide_id`.

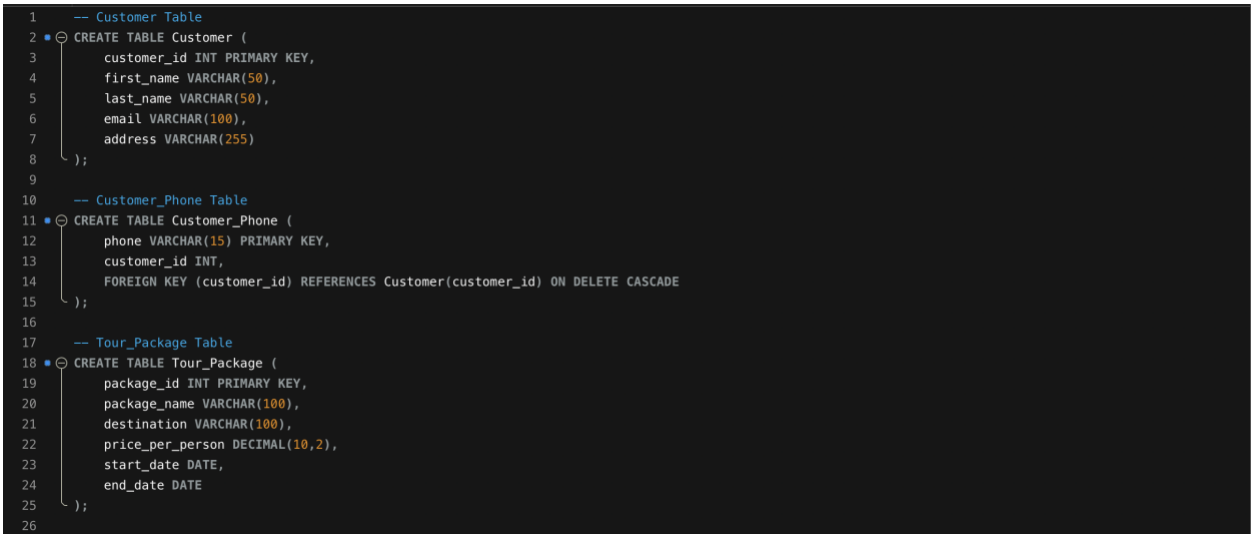
After

- `Guide (guide_id, guide_name, language, availability_status)`
- `Guide_Contact (contact_number, guide_id)`

Functional Dependancies

Table Name	Primary Key	Functional Dependencies
Customer	`customer_id`	`customer_id → (first_name, last_name, email, address)`
Customer_Phone	`phone`	`phone → customer_id`
Tour_Package	`package_id`	`package_id → (package_name, destination, price_per_person, start_date, end_date)`
Booking	`booking_id`	`booking_id → (booking_date, customer_id, package_id, number_of_people, total_cost)`

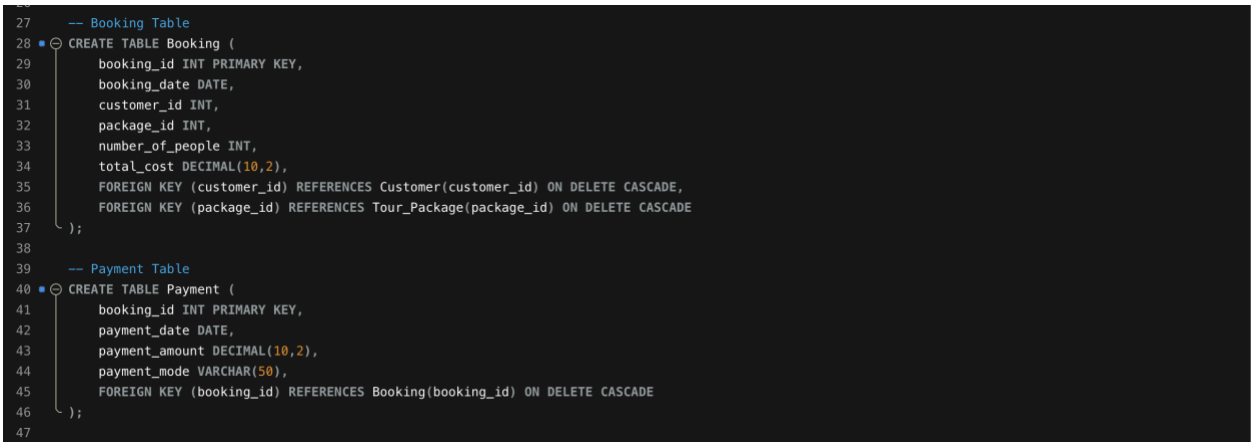
Payment	`booking_id`	`booking_id → (payment_date, payment_amount, payment_mode)`
Guide	`guide_id`	`guide_id → (guide_name, language, availability_status)`
Guide_Contact	`contact_number`	`contact_number → guide_id`



```sql
-- Customer Table
CREATE TABLE Customer (
    customer_id INT PRIMARY KEY,
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    email VARCHAR(100),
    address VARCHAR(255)
);

-- Customer_Phone Table
CREATE TABLE Customer_Phone (
    phone VARCHAR(15) PRIMARY KEY,
    customer_id INT,
    FOREIGN KEY (customer_id) REFERENCES Customer(customer_id) ON DELETE CASCADE
);

-- Tour_Package Table
CREATE TABLE Tour_Package (
    package_id INT PRIMARY KEY,
    package_name VARCHAR(100),
    destination VARCHAR(100),
    price_per_person DECIMAL(10,2),
    start_date DATE,
    end_date DATE
);
```



```sql
-- Booking Table
CREATE TABLE Booking (
    booking_id INT PRIMARY KEY,
    booking_date DATE,
    customer_id INT,
    package_id INT,
    number_of_people INT,
    total_cost DECIMAL(10,2),
    FOREIGN KEY (customer_id) REFERENCES Customer(customer_id) ON DELETE CASCADE,
    FOREIGN KEY (package_id) REFERENCES Tour_Package(package_id) ON DELETE CASCADE
);

-- Payment Table
CREATE TABLE Payment (
    booking_id INT PRIMARY KEY,
    payment_date DATE,
    payment_amount DECIMAL(10,2),
    payment_mode VARCHAR(50),
    FOREIGN KEY (booking_id) REFERENCES Booking(booking_id) ON DELETE CASCADE
);
```

```sql
-- Guide Table
CREATE TABLE Guide (
    guide_id INT PRIMARY KEY,
    guide_name VARCHAR(100),
    language VARCHAR(50),
    availability_status VARCHAR(50)
);

-- Guide_Contact Table
CREATE TABLE Guide_Contact (
    contact_number VARCHAR(15) PRIMARY KEY,
    guide_id INT,
    FOREIGN KEY (guide_id) REFERENCES Guide(guide_id) ON DELETE CASCADE
);
```

We effectively structured our Tourism Agency Booking System using the normalization process. We achieved 3NF by ensuring data integrity, reducing redundancy, and improving query performance, which then increased our systems reliability and scalability.