

گزارش کار دیتابیس سیستم رزرو سفر

تهیه کنندگان : علی پوراسگندر - مهدی سلمان

1. مقدمه

این گزارش شامل تحلیل و طراحی دیتابیس سیستم رزرو سفر است. این سیستم به کاربران امکان می‌دهد سفرهای خود را برنامهریزی کنند، بلیط تهیه کنند، پرداخت انجام دهند و گزارشات خود را ثبت کنند. در این گزارش، ساختار جداول، روابط بین آنها، نرمال‌سازی و نحوه عملکرد دیتابیس بررسی خواهد شد.

2. جداول و روابط

در این سیستم، چندین جدول وجود دارد که اطلاعات مختلفی از جمله کاربران، پرداخت‌ها، سفرها، بلیط‌ها، رزروها و وسایل نقلیه را ذخیره می‌کند. این جداول از طریق کلیدهای اصلی و خارجی به هم مرتبط شده‌اند. در ادامه، توضیحات مربوط به هر جدول آورده شده است.

User

جدول کاربران شامل اطلاعات شخصی کاربران و نوع حساب آنها است.

(N 1 به) ثبت می‌کند (Report) هر کاربر یک یا چندین گزارش

(N 1 به) انجام می‌دهد (Payment) هر کاربر یک یا چند پرداخت

(N 1 به) دارد (Reservation) هر کاربر یک یا چندین رزرو

Report

جدول گزارش‌ها برای ثبت مشکلات و مسائل مرتبط با سفرها است.

(N 1 به) هر گزارش توسط یک کاربر ارسال می‌شود

(N 1 به) است (Travel) هر گزارش مربوط به یک سفر

Payment

جدول پرداخت‌ها شامل اطلاعات پرداخت‌های کاربران برای بلیط‌ها است.

(N 1 به) هر پرداخت توسط یک کاربر انجام می‌شود

(N 1 به) مربوط است (Reservation) هر پرداخت به یک رزرو

Reservation

جدول رزروها اطلاعات مربوط به رزرو بلیطها را ذخیره می‌کند

(N 1 به) هر رزرو توسط یک کاربر ایجاد می‌شود

(N 1 به) مربوط است (Ticket) هر رزرو به یک بلیط

Ticket

جدول بلیطها اطلاعات مربوط به قیمت و ظرفیت باقی‌مانده را ذخیره می‌کند

(N 1 به) مرتبط است (Travel) هر بلیط به یک سفر

در ارتباط است (→) 1 به 1 (VehicleDetail) هر بلیط با یک وسیله نقلیه

Travel

جدول سفرها شامل اطلاعات مقصد، ظرفیت، کلاس سفر و شرکت حمل‌ونقل است

(N 1 به) هر سفر یک یا چندین بلیط دارد

(N 1 به) است (VehicleDetail) هر سفر مربوط به یک وسیله نقلیه

VehicleDetail

:این جدول به سه نوع وسیله نقلیه تقسیم شده است

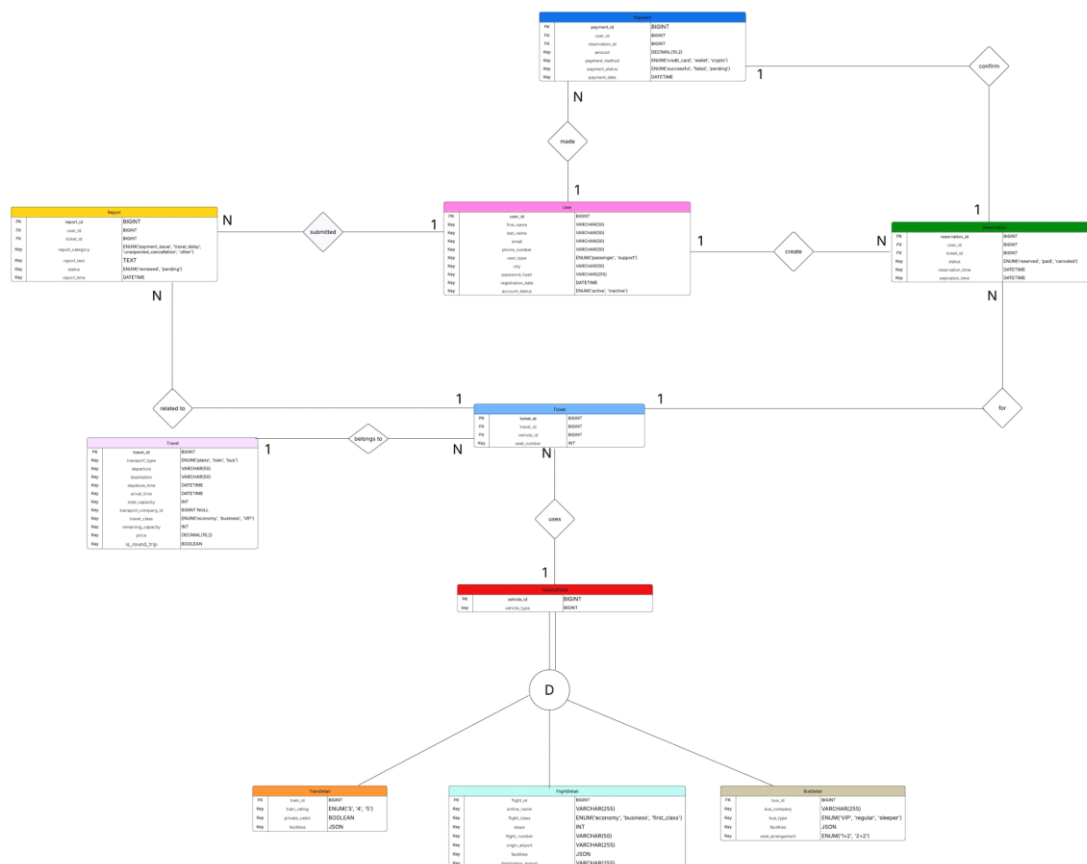
FlightDetail (پروازها)

BusDetail (اتوبوس‌ها)

TrainDetail (قطارها)

هر وسیله نقلیه به یک نوع مشخص مرتبط است (ارتباط ۱ به ۱)

ERD 3. نمودار



کد های ساخت جدول 4.

USER:

CREATE TABLE USER (

USER_ID BIGINT PRIMARY KEY AUTO_INCREMENT,

FIRST_NAME VARCHAR(100) NOT NULL,

LAST_NAME VARCHAR(100) NOT NULL,

EMAIL VARCHAR(255) UNIQUE,

PHONE_NUMBER VARCHAR(20) UNIQUE,

USER_TYPE ENUM('CUSTOMER', 'SUPPORT', 'ADMIN') NOT NULL,

CITY VARCHAR(100),

PASSWORD_HASH VARCHAR(255) NOT NULL,

REGISTRATION_DATE DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,

ACCOUNT_STATUS ENUM('ACTIVE', 'INACTIVE') NOT NULL DEFAULT 'ACTIVE',

```
CHECK (  
    EMAIL IS NOT NULL OR PHONE_NUMBER IS NOT NULL  
)  
  
CHECK (  
    EMAIL REGEXP '^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}$'  
    OR EMAIL IS NULL  
)  
  
CHECK (  
    PHONE_NUMBER REGEXP '^[0-9]{10,15}$'  
    OR PHONE_NUMBER IS NULL  
)  
);
```

TRAVEL

```
CREATE TABLE Travel (  
    travel_id BIGINT PRIMARY KEY AUTO_INCREMENT,  
    transport_type ENUM('plane', 'train', 'bus') NOT NULL,  
    departure VARCHAR(50) NOT NULL,  
    destination VARCHAR(50) NOT NULL,  
    departure_time DATETIME NOT NULL,  
    arrival_time DATETIME NOT NULL,  
    total_capacity INT NOT NULL,  
    remaining_capacity INT NOT NULL,  
    transport_company_id BIGINT NULL,  
    price INT NOT NULL,  
    is_round_trip BOOLEAN,  
    CHECK(price >= 0),  
    travel_class ENUM('economy', 'business', 'VIP') NOT NULL  
);
```

TICKET

```
CREATE TABLE Ticket (  
    ticket_id BIGINT PRIMARY KEY AUTO_INCREMENT,  
    travel_id BIGINT NOT NULL,  
    vehicle_id BIGINT NOT NULL,  
    seat_number INT NOT NULL,  
    FOREIGN KEY (travel_id) REFERENCES Travel(travel_id) ON DELETE CASCADE  
    FOREIGN KEY (vehicle_id) REFERENCES Travel(vehicle_id) ON DELETE CASCADE  
);
```

RESERVATION

```
CREATE TABLE Reservation (  
    reservation_id BIGINT PRIMARY KEY AUTO_INCREMENT,  
    user_id BIGINT NOT NULL,  
    ticket_id BIGINT NOT NULL,  
    status ENUM('reserved', 'paid', 'canceled') NOT NULL,  
    reservation_time DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,  
    expiration_time DATETIME NOT NULL,  
    CHECK (expiration_time >= reservation_time),  
    FOREIGN KEY (user_id) REFERENCES User(user_id) ON DELETE CASCADE,  
    FOREIGN KEY (ticket_id) REFERENCES Ticket(ticket_id) ON DELETE CASCADE  
);
```

PAYMENT

```
CREATE TABLE Payment (  
    payment_id BIGINT PRIMARY KEY AUTO_INCREMENT,  
    user_id BIGINT NOT NULL,  
    reservation_id BIGINT UNIQUE NOT NULL,  
    amount DECIMAL(10,2) NOT NULL,
```

```
payment_method ENUM('credit_card', 'wallet', 'crypto') NOT NULL,  
payment_status ENUM('failed', 'pending', 'completed') NOT NULL,  
payment_date DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,  
FOREIGN KEY (user_id) REFERENCES User(user_id) ON DELETE CASCADE,  
FOREIGN KEY (reservation_id) REFERENCES Reservation(reservation_id) ON DELETE  
CASCADE  
);
```

REPORT

```
CREATE TABLE Report (  
    report_id BIGINT PRIMARY KEY AUTO_INCREMENT,  
    user_id BIGINT NOT NULL,  
    ticket_id BIGINT NOT NULL,  
    report_category ENUM('payment_issue', 'travel_delay', 'unexpected_cancellation', 'other') NOT  
NULL,  
    report_text TEXT NOT NULL,  
    status ENUM('reviewed', 'pending') NOT NULL,  
    report_time DATETIME NOT NULL,  
    FOREIGN KEY (user_id) REFERENCES User(user_id) ON DELETE CASCADE,  
    FOREIGN KEY (ticket_id) REFERENCES Ticket(ticket_id) ON DELETE CASCADE  
);
```

VEHICLE_DETAIL

```
CREATE TABLE VehicleDetail (  
    vehicle_id BIGINT PRIMARY KEY AUTO_INCREMENT,  
    vehicle_type ENUM('train', 'flight', 'bus') NOT NULL,  
);
```

FLIGHT_DETAIL

```
CREATE TABLE FlightDetail (  

```

```
flight_id BIGINT PRIMARY KEY,  
airline_name VARCHAR(255) NOT NULL,  
flight_class ENUM('economy', 'business', 'first_class') NOT NULL,  
stops INT NOT NULL DEFAULT 0,  
flight_number VARCHAR(50) UNIQUE NOT NULL,  
origin_airport VARCHAR(255) NOT NULL,  
destination_airport VARCHAR(255) NOT NULL,  
facilities JSON,  
FOREIGN KEY (flight_id) REFERENCES VehicleDetail(vehicle_id) ON DELETE CASCADE  
);
```

TRAIN_DETAIL

```
CREATE TABLE TrainDetail (  
    train_id BIGINT PRIMARY KEY,  
    train_rating ENUM('3', '4', '5') NOT NULL,  
    private_cabin BOOLEAN,  
    facilities JSON,  
    FOREIGN KEY (train_id) REFERENCES VehicleDetail(vehicle_id) ON DELETE CASCADE  
);
```

BUS_DETAIL

```
CREATE TABLE BusDetail (  
    bus_id BIGINT PRIMARY KEY,  
    bus_company VARCHAR(255) NOT NULL,  
    bus_type ENUM('VIP', 'regular', 'sleeper') NOT NULL,  
    facilities JSON,  
    seat_arrangement ENUM('1+2', '2+2') NOT NULL,  
    FOREIGN KEY (bus_id) REFERENCES VehicleDetail(vehicle_id) ON DELETE CASCADE  
);
```

نتیجه گیری:

این گزارش ساختار کلی دیتابیس را توضیح داد و روابط بین جداول را مشخص کرد. با استفاده از این طراحی، اطلاعات به صورت کارآمد ذخیره شده و عملیات رزرو، پرداخت و گزارش دهی به صورت بهینه انجام خواهد شد.