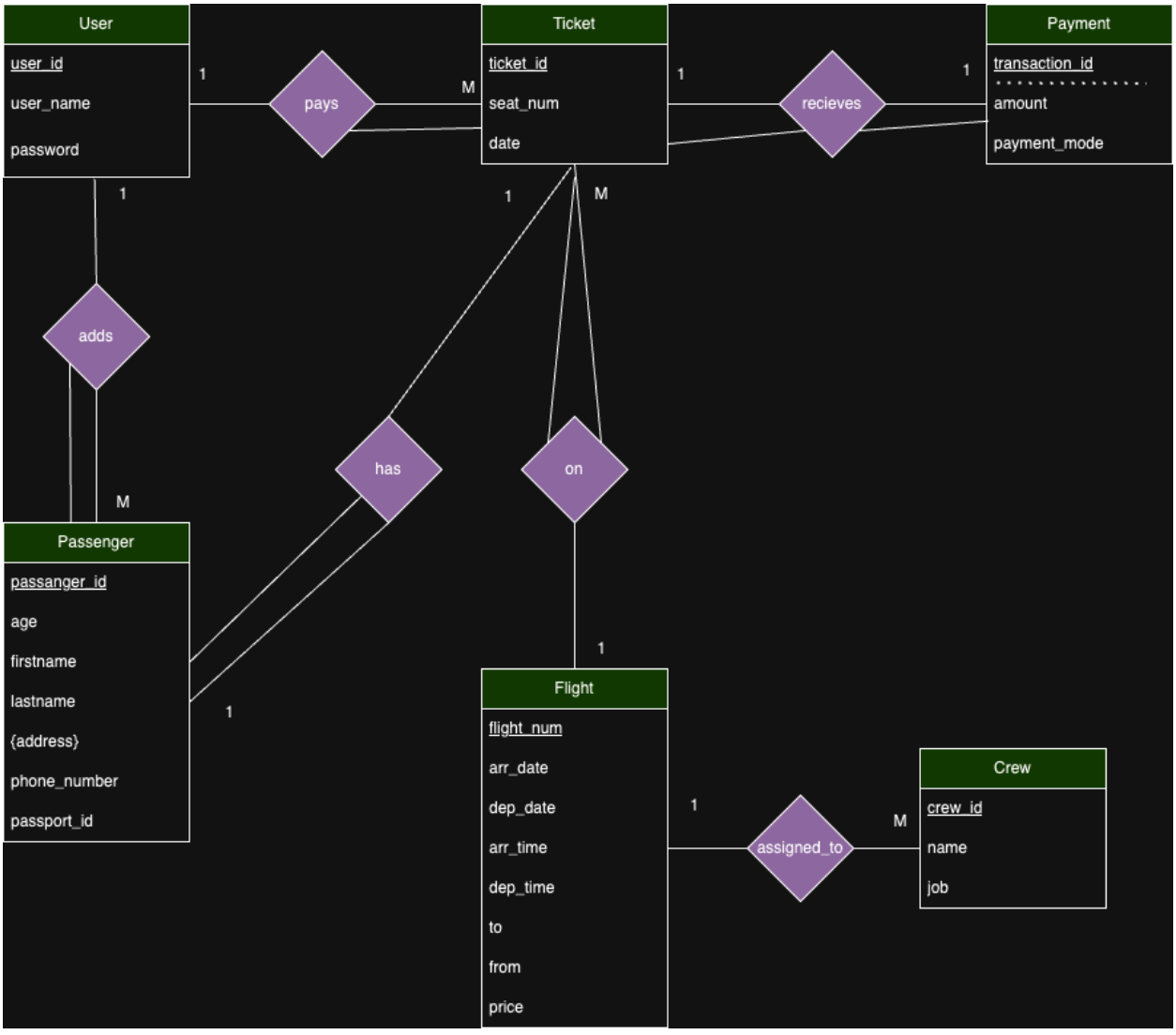
DBMS for Airline Ticket Reservation System

Moina Swaray, Abe Olsson, Venkata Krishna Kandalai



Relational Schema and Explanations

1.1: USER

User (user_id: INT PRIMARY KEY, user_name: VARCHAR(15), password: VARCHAR(15))

The user table stores information regarding the users of the flight booking system. Each user is assigned a unique id (PK), a username, and a password.

1.2: PASSENGER

Passenger (passenger_id: INT PRIMARY KEY, age: INT, firstname: VARCHAR(50), lastname: VARCHAR(50), phone_number: VARCHAR(15), passport_id: VARCHAR(20), user_id: INT)

The passenger table contains information regarding passengers who are travelling. The primary key is passenger_id, and includes other personal information of the passengers. Each passenger is also associated with a user by the attribute user_id foreign key.

1.3: PASSENGER ADDRESS

Passenger_Address (passenger_id, street_num, street_name, city, state, zip, country)

This entity stores the addresses of passengers. The primary key of this table is passenger_id, street_num, street_name.

1.4: FLIGHT

Flight (flight_num: VARCHAR(20) PRIMARY KEY, arr_date: DATE, dep_date: DATE, arr_time: TIME, dep_time: TIME, price: DECIMAL(10, 2) ,to: VARCHAR(20), from: VARCHAR(20))

The flight table stores information about flights. The primary key is the flight_id, and it also includes other information about flight arrival time and city, departure time and city. Price is added here and removed from ticket, because the flight comes before the ticket and must determine the cost of the ticket.

1.5: TICKET

Ticket (ticket_id: INT PRIMARY KEY, seat_num: VARCHAR(3), date: DATE, passenger_id: INT, flight_num: VARCHAR(20))

The ticket table represents the tickets that are sold to passengers. It includes attributes such as price, date, and seat number. The primary key is ticket_id and each ticket is linked to a passanger and a specific flight.

1.6: PAYMENT

Payment (transaction_id: INT PRIMARY KEY, amount: DECIMAL(10, 2), payment_mode: VARCHAR(50), ticket_id: INT)

The payment table has transcation information regarding the reservations. It has information such as ticket price and payment method. The primary key is transaction_id and each payment is associated with a specific ticket.

1.7: CREW

Crew (crew_id: INT PRIMARY KEY, name: VARCHAR(50), job: VARCHAR(50), flight_num VARCHAR(20))

The crew table contains details about the crew members assigned to flights. Each crew member has a crew_id (primary key), name and a job title. Each crew member is assigned to a flight.

Implementation of Schema in MySQL

```
use TravelXDB;

/* Database Table Creation*/

/* Drop any existing tables. Any errors are ignored. */

DROP TABLE IF EXISTS user;

DROP TABLE IF EXISTS passenger;

DROP TABLE IF EXISTS passenger_address;

DROP TABLE IF EXISTS flight;

DROP TABLE IF EXISTS ticket;
```

```
DROP TABLE IF EXISTS payment;

DROP TABLE IF EXISTS crew;


/* Add each table. */


/* user_id must have a not null constraint because each user_id must be unique. */

CREATE TABLE user (

user_id VARCHAR(15),

User_name VARCHAR(15),

password VARCHAR(15),

PRIMARY KEY (user_id)) ENGINE=xxxxdb;


/* The passenger_id must have a not null constraint because it is the primary key. If a
user_id is updated or deleted, the corresponding passenger information should also be
updated or deleted. The user_id must have a not null constraint because it is the primary
key in user table.*/

CREATE TABLE passenger (

passenger_id INT,

age INT,

firstname VARCHAR (50),

lastname VARCHAR (50),

phone_number VARCHAR(15),

passport_id VARCHAR (20),

user_id INT NOT NULL,

PRIMARY KEY (passenger_id),

FOREIGN KEY (user_id) REFERENCES user(user_id)

ON DELETE CASCADE
```

```
ON UPDATE CASCADE) ENGINE=xxxxdb;


/* The primary key needs to be passenger_id, street_num, street_name to make each row
unique in the address table. If a passenger_id is updated or deleted, the passenger
instance must also be updated or deleted . The passenger_id must have a not null
constraint because it is the primary key in passenger table. */

CREATE TABLE passenger_address(

passenger_id INT NOT NULL,

street_num VARCHAR(15),

street_name VARCHAR(30),

city VARCHAR (30),

state VARCHAR (30),

zip INT(10),

country VARCHAR (50),

PRIMARY KEY (passenger_id),

PRIMARY KEY (street_num),

PRIMARY KEY (street_name)

FOREIGN KEY (passenger_id) REFERENCES passenger(passenger_id)

ON DELETE CASCADE

ON UPDATE CASCADE) ENGINE=xxxxdb;


 /* flight_num must have a not null constraint because it needs to be unique. */

CREATE TABLE flight (

flight_num VARCHAR (20),

price DECIMAL (10,2),

arr_date DATE,

dep_date DATE,
```

```
arr_time TIME,

dep_time TIME,

to VARCHAR (5),

from VARCHAR (5),

PRIMARY KEY (flight_num)) ENGINE=xxxxdb;


/*passenger_id is a foreign key that is from passenger and it ties the ticket_id to a specific passenger along with their flight_num. Therefore, passenger_id and flight_num must not be a NULL, the primary key is ticket_id . The foreign keys are both flight and passenger_id, which point to other tables to make the unique ticket_id. */


CREATE TABLE ticket(

ticket_id INT ,

seat_num VARCHAR (3),

date DATE,

passenger_id INT NOT NULL,

flight_num VARCHAR (20) NOT NULL,

PRIMARY KEY (ticket_id),

FOREIGN KEY (passenger_id) REFERENCES passenger(passenger_id)

ON DELETE CASCADE

ON UPDATE CASCADE),

FOREGIN KEY (flight_num) REFERENCES flight(flight_num)

ON DELETE CASCADE

ON UPDATE CASCADE), ENGINE=xxxxdb;


/* the payment is unique to a ticket_id and is represented by the primary key transaction_id and the foreign key references the ticket_id from the ticket, which includes its own
```

information such as flight_num, the other foreign key is the amount which is referenced to the flight price. The amount must not be a null otherwise the payment or transaction is not valid */

```
CREATE TABLE payment(

transaction_id INT ,

amount DECIMAL(10,2) NOT NULL,

payment_mode VARCHAR (50),

ticket_id INT NOT NULL,

PRIMARY KEY (transaction_id),

FOREIGN KEY (ticket_id) REFERENCES ticket(ticket_id)

ON DELETE CASCADE

ON UPDATE CASCADE),

FOREIGN KEY (amount) REFERENCES flight(price)

ON DELETE CASCADE

ON UPDATE CASCADE) ENGINE=xxxxdb;
```

/* the crew_id is the primary key and it makes each crew member unique from each other. Flight assignment cannot be NULL because it is the primary key for the flight table. If a flight assignment is updated or deleted, the crew table must also correspond to the change. * /

```
CREATE TABLE crew(

crew_id INT,

name VARCHAR (50),

job VARCHAR (50),

flight_assignment VARCHAR (20) NOT NULL,

PRIMARY KEY (crew_id),

FOREIGN KEY (flight_assignment) REFERENCES flight(flight_num)
```

ON DELETE CASCADE

ON UPDATE CASCADE) ENGINE=xxxxb;

Loading Data

INSERT INTO user (user_id, user_name, password) VALUES

(1, 'John123', 'pass123'),

(2, 'Alice456', 'securepwd'),

(3, 'David789', 'password123'),

(4, 'SarahDoe', 'p@ssw0rd'),

(5, 'MichaelK', 'letmein'),

(6, 'Emily_23', '12345678'),

(7, 'ChrisSmith', 'abcdefg'),

(8, 'Jennifer', 'qwertyui'),

(9, 'Robert87', 'password!'),

(10, 'Laura1995', 'passw0rd');

INSERT INTO passenger (passenger_id, age, firstname, lastname, phone_number, passport_id, user_id) VALUES

(11, 25, 'John', 'Doe', '123-456-7890', 'ABC123456', 1),

(21, 30, 'Jane', 'Smith', '987-654-3210', 'DEF654321', 2),

(31, 40, 'Michael', 'Johnson', '555-555-5555', 'GHI987654', 3),

(41, 22, 'Emily', 'Williams', '111-222-3333', 'JKL123456', 4),

(51, 35, 'David', 'Brown', '444-444-4444', 'MNO789012', 5),

(61, 28, 'Sarah', 'Miller', '666-777-8888', 'PQR345678', 6),

(71, 45, 'Christopher', 'Davis', '999-999-9999', 'STU234567', 7),

(81, 33, 'Amanda', 'Wilson', '222-333-4444', 'VWX456789', 8),

(91, 29, 'Ryan', 'Anderson', '777-888-9999', 'YZ0AB5678', 9),

(101, 27, 'Michelle', 'Martin', '888-999-0000', 'CDE901234', 10);

INSERT INTO Passenger_Address (passenger_id, street_num, street_name, city, state, zip, country) VALUES

(11, '123', Main St', 'Anytown', 'CA', 12345, 'USA'),

(21, '456', 'Elm St', 'Somewhere', 'NY', 54321, 'USA'),

(31, '789', 'Oak St', 'Nowhere', 'TX', 98765, 'USA'),

(41, '321', 'Maple St', 'Everytown', 'FL', 67890, 'USA'),

(51, '555', 'Pine St', 'Anyville', 'WA', 13579, 'USA'),

(61, '999', 'Oak Ln', 'Somewhere', 'CA', 98765, 'USA'),

(71, '777', 'Elm Ave', 'Nowhere', 'FL', 54321, 'USA'),

(81, '888', 'Maple Blvd', 'Anytown', 'TX', 24680, 'USA'),

(91, '111', 'Pine Rd', 'Everyville', 'WA', 98765, 'USA'),

(101, '444', 'Oak Dr', 'Anyville', 'CA', 54321, 'USA');

INSERT INTO flight (flight_num, arr_date, dep_date, arr_time, dep_time, price, to_location, from_location) VALUES

('FL123', '2024-05-10', '2024-05-09', '15:30:00', '10:00:00', 250.00, 'New York', 'London'),

('FL456', '2024-05-11', '2024-05-10', '12:00:00', '08:30:00', 300.00, 'Paris', 'Tokyo'),

('FL789', '2024-05-12', '2024-05-11', '14:45:00', '09:15:00', 200.00, 'London', 'Beijing'),

('FL101', '2024-05-13', '2024-05-12', '09:00:00', '05:30:00', 350.00, 'Tokyo', 'New York'),

('FL202', '2024-05-14', '2024-05-13', '18:20:00', '14:00:00', 400.00, 'Beijing', 'Paris'),

('FL303', '2024-05-15', '2024-05-14', '10:45:00', '07:20:00', 280.00, 'London', 'Tokyo'),

('FL404', '2024-05-16', '2024-05-15', '13:15:00', '09:45:00', 320.00, 'Paris', 'New York'),

('FL505', '2024-05-17', '2024-05-16', '16:30:00', '12:00:00', 270.00, 'Tokyo', 'London'),

('FL606', '2024-05-18', '2024-05-17', '11:00:00', '06:30:00', 380.00, 'Beijing', 'Paris'),

('FL707', '2024-05-19', '2024-05-18', '19:00:00', '15:30:00', 350.00, 'New York', 'Tokyo');

INSERT INTO ticket (ticket_id, seat_num, date, passenger_id, flight_num) VALUES

(1000000001, '01A', '2024-05-09', 11, 'FL123'),

(1000000002, '02B', '2024-05-10', 21, 'FL456'),

(1000000003, '03C', '2024-05-11', 31, 'FL789'),

(1000000004, '04D', '2024-05-12', 41, 'FL101'),

(1000000005, '05E', '2024-05-13', 51, 'FL202'),

(1000000006, '06F', '2024-05-14', 61, 'FL303'),

(1000000007, '07A', '2024-05-15', 71, 'FL404'),

(1000000008, '08B', '2024-05-16', 81, 'FL505'),

(1000000009, '09C', '2024-05-17', 91, 'FL606'),

(1000000010, '10D', '2024-05-18', 101, 'FL707');

INSERT INTO Payment (transaction_id, amount, payment_mode, ticket_id) VALUES

(1001, 250.00, 'Credit Card', 1000000001),

(1002, 300.00, 'PayPal', 1000000002),

(1003, 200.00, 'Debit Card', 1000000003),

(1004, 350.00, 'Cash', 1000000004),

(1005, 400.00, 'Apple Pay', 1000000005),

(1006, 280.00, 'Google Pay', 1000000006),

(1007, 320.00, 'Credit Card', 1000000007),

(1008, 270.00, 'PayPal', 1000000008),

(1009, 380.00, 'Debit Card', 1000000009),

(1010, 350.00, 'Cash', 1000000010);

INSERT INTO Crew (crew_id, name, job, flight_assignment) VALUES

(1, 'John Smith', 'Pilot', 'FL123'),

(2, 'Sarah Johnson', 'Co-Pilot', ‘FL101'),

(3, 'Michael Brown', 'Flight Attendant', 'FL123'),

(4, 'Emily Davis', 'Flight Attendant', 'FL303'),

(5, 'David Wilson', 'Engineer', 'FL505'),

(6, 'Lisa Taylor', 'Flight Attendant', 'FL707’),

(7, 'Matthew Martinez', 'Ground Crew', 'FL909’),

(8, 'Amanda Anderson', 'Ground Crew', 'FL505'),

(9, 'James Thomas', 'Engineer', 'FL404’),

(10, 'Jessica Garcia', 'Flight Attendant', ‘FL101');