**Venkatavihan Devaki 2021A7PS0429P**

**Amandeep Singh 2021A7PS0575P**

**Project 2 : Airline ticketing**

**Project Specifications:**

To create a database system to maintain records of different flights and to create a system to maintain user booking data.

Files included in zip:

1. queries.sql- contains all queries used in the project
2. create\_db.sql- contains queries specifically to create the schema
3. project.py- backend code written in python
4. images of ER diagram and Schema

**Instructions to run backend application:**

1. Run create queries script files to create and populate the database.
2. Open project.py and modify the values for host, user, password and database to reflect your system specifications.
3. Run project.py and use the application via command line.

**Videos explanation:** [**https://drive.google.com/drive/folders/1klpvDALdyQfTfNsRhKaSrvyZG7haLswZ?usp=share\_link**](https://drive.google.com/drive/folders/1klpvDALdyQfTfNsRhKaSrvyZG7haLswZ?usp=share_link)

**Screenshots of queries outputs :**

[**https://drive.google.com/drive/folders/1\_xG\_v9BIuy6vVhJedfYvclkyJa\_Pj9NU?usp=share\_link**](https://drive.google.com/drive/folders/1_xG_v9BIuy6vVhJedfYvclkyJa_Pj9NU?usp=share_link)

**ER Diagram**

The ERD represents the relationships between the entities in an Airline Ticketing System.

The following are the entities in the diagram-

Diagram

Description automatically generated

1. **Users**- Representing all the users using the system to book an airline ticket. It contains the following attributes-
2. UID- It is a string that is used to uniquely identify all users.
3. fname- It is a string that stores the first name of the user.
4. lname- It is a string that stores the last name of the user.
5. Dob- It is a date the stores the Date of Birth of the user.
6. phone- it is a string that is used to store phone number of a user.
7. email- It is a string that stores the user’s email.
8. Password- It is a string that stores the user’s password.
9. gender-It is an enum storing gender of the user.

The Primary key here **UID**.

1. **airlines**-The entity represents the airlines providing flight service to the users. It contains the following attributes-
2. airline\_id- This is an integer used to uniquely identify all different airlines.
3. airline\_name- It is a string that stores the airline’s name.
4. airline\_password- It is a string that stores the airline’s password.

The Primary key here **airline\_id**.

1. **flights**- Contains all the flights available for the users to book from different airlines. It contains the following attributes-
2. flight\_id- This is an integer that is used to uniquely identify all flights. This is set using auto increment function.
3. departure\_date-It is a date that records the date of departure of the flight.
4. departure\_time- It is a time that records the time of departure of the flight.
5. f\_airline\_id-This is a foreign key referencing airlines entity.
6. origin-It is a string that stores the origin of a flight.
7. destination- It is a string that stores the destination of a flight.
8. duration- It is an integer that stores duration of the flight.

The Primary Key here **airline\_id**.

The Foreign Key here **f\_airline\_id.**

1. **Booking**-Contains all the flight bookings made by a customer. It contains the following attributes-
2. bid-This is an integer that stores the unique id of a booking.
3. b\_fid-This is a foreign key referencing flights entity.
4. b\_uid- This is a foreign key referencing users entity.

The Primary Key here **bid**.

The Foreign Keys here **b\_uid** and **b\_fid.**

1. **payment**-The entity maintains the record of payments made for a certain booking. It contains the following attributes-
   1. p\_id- This is an integer used to uniquely identify all different payments.
   2. amount - It is a floating number that stores the amount of transaction.
   3. confirm- It is a string that confirms the transaction’s occurance.
   4. p\_bid-It is a foreign key referencing booking entity.

The Primary key here **p\_id**.

The Foreign key here **p\_bid**.

1. **cancellation**-The entity maintains the record of cancellations made for a certain booking. It contains the following attributes-
   1. c\_id- This is an integer used to uniquely identify all cancellations.
   2. message - It is a string that stores the information about cancellation.
   3. c\_bid-It is a foreign key referencing booking entity.
   4. c\_time- It is stores the time stamp of when the cancellation took place.

The Primary key **b\_id**.

The Foreign key here **c\_bid**.

1. **Seats-** The entity contains all the seats on all the flights that are available for booking.It contains the following attributes-
   1. seat\_no- This is an integer used to uniquely identify all seats on a flight.
   2. s\_fid - This is an integer used to uniquely identify all flights in an airline.
   3. s\_class-It is an enum stating the class of the seat.
   4. s\_type- It is an enum stating the type of the seat.
   5. s\_price-It is a floating value storing the price of booking a specific seat.
   6. s\_bid-It is a foreign key referencing booking entity.

The Primary key here **seat\_no** and **s\_fid**.

The Foreign key here **s\_bid**.

The relationships between the entities are as follows-

1. One user can have multiple bookings whereas a booking must have 1 user (one to many relationship between user and booking)
2. A booking can have a payment whereas a payment must have a booking(one to one relationship between booking and payment)
3. A booking can have a cancellation whereas a cancellation must have a booking(one to one relationship between booking and cancellation)
4. A booking must have a flight whereas a flight can have multiple bookings(many to one relationship between booking and flight )
5. A flight must contain many seats whereas a seat must have a flight(one to many relationship between flight and seats)
6. Every airline must have at least one flight whereas every flight must have an airline(one to many relationship between airline and flight)

**Conversion**

To convert the above ER model to a relational database, we need to create tables for each entity and represent the relationships between entities using foreign keys.

Each entity is converted into a table with columns representing the attributes of the entity. The table schema for each entity is as follows:

1. users
2. airlines
3. flights
4. booking
5. payments
6. books
7. booked\_seats
8. seats
9. cancellation
10. has\_payment

For attribute details refer to the diagram.

The relationships between the entities are represented using foreign keys. The foreign key columns in each table refer to the primary key column of the related table. The relationships are as follows:

* The Booking table has a foreign key column b\_uid that references the primary key column UID in the User table.
* The Booking table has a foreign key column b\_fid that references the primary key column flight\_id in the flight table.
* The flights table has a foreign key column f\_airline\_id that references the primary key column airline\_id in the airline table.
* The Payment table has a foreign key column p\_bid that references the primary key column b\_id in the booking table.
* The Cancellation table has a foreign key column c\_bid that references the primary key column b\_id in the booking table.
* The Seats table has a foreign key column s\_fid that references the primary key column flight\_id in the flights table.
* The Seats table has a foreign key column s\_bid that references the primary key column b\_id in the booking table.

Diagram

Description automatically generated

**Normalization**

Let us first list all the functional dependencies that exist in our schema.

The dependencies are as follows-

1. **In the Users table:**

UID → fname, lname, email, password, dob, phone, gender

Phone → fname, lname, email, password, dob, gender

Email → fname, lname, password, dob, phone, gender

1. **In the booking table:**

B\_id → b\_fid, b\_uid

1. **In the booked\_seats table:**

Bs\_bid → bs\_no, bs\_fid

Bs\_no,bs\_fid → bs\_bid

1. **In the has\_payment table:**

Hp\_bid → hp\_pid, hp\_time

Hp\_pid → hp\_bid, hp\_time

Hp\_time → hp\_pid, hp\_bid

1. **In the payment table:**

Pid → amount, confirm, p\_bid

P\_bid → amount, confirm, pid

1. **In the Cancellation table:**

Cid → c\_bid, c\_time

C\_bid → cid, c\_time

1. **In the books table:**

Bo\_uid → bo\_bid, bo\_time

Bo\_bid → bo\_uid, bo\_time

1. **In the airline table:**

Airline\_name →airline\_id, airline\_password

Airline\_password → airine\_name,airline\_id

Airline\_id → airline\_name, airline\_password

1. **In the booking table:**

S\_fid, seat\_no → s\_class,s\_type,s\_price,s\_bid

S\_bid, seat\_no → s\_class,s\_type,s\_price,s\_fid

To bring the current schema to 1NF, there should be no composite attributes.

As this condition is already satisfied, the schema is in 1NF.

To bring the schema to 2NF, it must satisfy the condition that all non-key attributes in the table are dependent on the entire primary key.

As there are no partial dependencies in any of the tables, the schema is already in 2NF.

To bring the schema in 3NF, it must satisfy the condition that all non-key attributes in the table are independent of each other.

As there are no transitive dependencies in any of the tables, the schema is already in 3nf.