**Airline Reservation Database Design Document**

Project Group 3: Duanhao Zhong, Yueming Zhang, Yuxin Shen

December 4, 2017

## Database Purpose

The purpose of the database is to maintain the data used to search all airline flights information, purchase or cancel tickets and check order status. It also can be used by administrative staff only to browse all users’ information and travel history. This database is designed for a travel agent use.

## Business Problems Addressed

* Allow customers to search and compare different airline flights information.
* Provide customers access to purchase, cancel tickets and check their order status.
* Allow administrative staff to maintain the customer information.
* Permit administrative staff to update flight information in time (e.g., when any flight gets delayed or canceled, they can send notifications to customers immediately).
* Supply insight for administrative staff to analysis flight marketing initiatives (e.g., generate information from history data to help administrative staff learn which is the popular season and what type of customers they have).
* Permit administrative staff knowledge the income and analyze their profit.

## Assumptions/Constraints/Risks

### Assumptions

* Microsoft Visio is used to create the entity relationship diagram model.
* Crow’s foot notation is considered in the ERD model.
* Microsoft SQL Server and SQL Server Management Studio are used to design and maintain the database.

### Constraints

* All customers’ information can only be seen or changed by administrators.
* One flight ticket can only be booked by one customer, one customer can buy more than one flight tickets either for themselves or for others.
* Customer can only cancel flight tickets twice in one day.
* Flight information can only be changed by one administrator at one time.

### Risks

* There may be mistake input in flight information because only one administrator is needed to change the information. Once an administrator change the information, it must be approved by at least one other administrator.
* Customers may buy tickets over the amount of the residual tickets. Warning or error should be generated when this situation happens.

## Design Decisions

### Identify Entity Types

|  |  |  |
| --- | --- | --- |
| Entity Name | Description | Why included |
| User | All general information needed as a user, such as userID, address and telephone number. | The main function of this database is the interaction with users. Therefore, we want to use this entity to record basic users’ information. |
| Customer | As a subclass of User, besides all general user information, extra information is needed as a registered customer. Such as order history and payment method. | In this database, some functions must be operated by customers. And some extra attributes are needed for that. |
| Administrator | As a subclass of User, besides all general information, extra information is needed as an administrator. Such as working schedule and salary. | In this database, some functions must be operated by administrator. And some extra attributes are needed for that. |
| Booking | This entity is used as a record for each booking processed by customers to buy flight tickets. | We need to use this entity to connect passenger and customer information with ticket. In database, we also pass each booking information to order history entity. |
| Passenger | This entity is used to record all passenger information provided by customers when they book the tickets. | Since customers are likely to buy tickets for more than one passengers, we need this entity to record all the passenger information which might be different from customer information. |
| Ticket | Information like specific flight, seats and price. | We need an entity functioning as a ticket to present all information that customers want to see. |
| Seat | Seat information like location, type and price. | We want to distinguish seats. Different seats should have different price and service. |
| Flight | Specific flight information like direction, airplane and flight departure & arrival time. | We can’t include all detailed information in ticket entity, so we need to use this entity to present more detailed flight information. |
| Airplane | Airplane information like capacity, model. | We want to record airplane information like capacity and model for each flight. |
| FlightStatus | Users can check specific flight status like on-time, early or delayed. | Customers may want to check their flight status and administrators also need this entity to update the status of each flight. |
| Direction | Record departure airport and arrival airport for different flights. | We can’t include all airports’ detailed information in the Flight entity, so we use this entity to connect airport entity with flight entity. |
| DepartureAirport | Information about departure airport like location, terminal and departure gate. | A particular flight needs departure airport. |
| ArrivalAirport | Information about arrival airport like location, terminal and arrival gate. | A particular flight needs arrival airport. |
| AirlineCompany | Airline company information like address and contact number. | Through this entity, we can connect aircrew entity and airplane entity. |
| Staff | All staff’s information for every airline company. | As an airline company, they should have lots of staff with different positions. |
| Aircrew | A group of staff working as an aircrew team to serve one flight. | We want to group several staff as an aircrew team, so it’ll be easier to connect one group to one flight. |

### Identify Relationship Types

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Entity name | Multuplicity | Relationship | Multiplicity | Entity name |
| User | 1..1  1..1 | ContainsC  ContainsA | 1..1  1..1 | Customer  Administrator |
| Customer | 1..1 | Processes | 0..\* | Booking |
| Booking | 1..1 | Generates | 1..\* | Ticket |
| Ticket | 1..1  0..\*  0..\* | Shows  Corresponds  Indicates | 1..1  1..1  1..1 | Passenger  Seat  Flight |
| Flight | 1..1  1..\*  1..\* | Refers  Towards  Servedby | 1..1  1..1  1..1 | Flight Status  Direction  Aircrew |
| Airplane | 1..1  0..\*  1..1 | Equipedwith  Ownedby  Flies | 1..\*  1..1  0..\* | Seat  AirlineCompany  Flight |
| Direction | 1..\*  1..\* | Leaves  Arrives | 1..1  1..1 | DepartureAirport  ArrivalAirport |
| AirlineCompany | 1..1 | Employs | 1..\* | Staff |
| Staff | 1..\* | forms | 0..1 | Aircrew |

### Identify Attributes

|  |  |  |
| --- | --- | --- |
| Entity name | Attributes | DataType |
| User | UserID  UserType  FirstName  LastName  Phone  Street  City  State  ZipCode  Country  Email  Gender  DOB | int  varchar(20)  varchar(20)  varchar(20)  varchar(15)  varchar(20)  varchar(20)  varchar(10)  varchar(10)  varchar(20)  varchar(30)  varchar(10)  date |
| Customer | UserID  Payment | int  varchar(15) |
| Administrator | UserID  WorkingShift  Salary | int  int  money |
| Booking | BookingID  UserID  PassengerID  TicketNumber  OrderTime | int  int  int  char(15)  datetime |
| Passenger | PassengerID  FirstName  LastName  DOB  Phone  Email  Street  City  State  ZipCode  Country | int  varchar(20)  varchar(20)  date  varchar(15)  varchar(30)  varchar(20)  varchar(20)  varchar(10)  varchar(10)  varchar(20) |
| Ticket | TicketNumber  PassengerID  SeatID  FlightNumber  Price  ServiceFee | char(15)  int  int  varchar(10)  money  money |
| Seat | SeatID  AirplaneID  SeatNumber  Type | int  int  char(3)  varchar(10) |
| Flight | FlightNumber  DirectionID  AirplaneID | varchar(10)  varchar(5)  int |
| FlightStatus | FlightNumber  FlightStatus  IfApproved | varchar(10)  varchar(10)  varchar(20) |
| Airplane | AirplaneID  Model  Capacity  CompanyID | int  varchar(10)  int  int |
| Direction | DirectionID  DepartureTime  DepartureAirportCode  DepartureTerminal  DepartureGate  ArrivalTime  ArrivalAirportCode  ArrivalTerminal  ArrivalGate  Duration | varchar(5)  datetime  varchar(5)  varchar(5)  varchar(5)  datetime  varchar(5)  varchar(5)  varchar(5)  time |
| DepartureAirport | DepartureAirportCode  Street  City  State  ZipCode  Country | varchar(5)  varchar(20)  varchar(20)  varchar(10)  varchar(10)  varchar(20) |
| ArrivalAirport | ArrivalAirportCode  Street  City  State  ZipCode  Country | varchar(5)  varchar(20)  varchar(20)  varchar(10)  varchar(10)  varchar(20) |
| AirlineCompany | CompanyID  CompanyName  Phone  Street  City  State  ZipCode  Country | int  varchar(20)  varchar(15)  varchar(20)  varchar(20)  varchar(10)  varchar(10)  varchar(20) |
| Staff | StaffNo  CompanyID  FirsrName  LastName  Phone  Street  City  State  ZipCode  Country  Email  Gender  DOB  Title  Department  Salary/Hour  WorkingHours  SalaryTotal | int  int  varchar(20)  varchar(20)  varchar(15)  varchar(20)  varchar(20)  varchar(10)  varchar(10)  varchar(20)  varchar(30)  varchar(10)  date  varchar(15)  varchar(15)  money  int  money |
| Aircrew | AircrewID  StaffNo  FlightNumber | int  int  varchar(10) |