**INFO 6210 Airline Database**

**Database Specification: Purpose, Business Problems Addressed And Business Rules**

*Heng Wang, Dingting Huang, Yize Wang, Ran Wei*

**Database Purpose:**

Build a secure airline database which will adequately and efficiently support the airline and booking operations. It maintains the information of airplanes, flights and schedule; it also can be used to book flights and securely store the information of customers. The database will be used by the airline staff and customer.

**Business Problems Addressed:**

* Allow airline company, traveling app and customers to generate descriptive reports
* Provide information to adjust airline scheduling (e.g. adjust the schedule according to the off season and peak season)
* Provide convenience to users and implement unified management. Simply and fast query; securely and comprehensively store information; associated database.
* Allow staff to modify the flight schedule and adjust order information (e.g. can help passengers change flights when oversold the tickets)
* Permit staff to analyze and refine sales quarterly objectives.
* Get the frequent flights, travel locations and frequencies and other helpful information of the target passengers, and provide them with more accurate suggestions.

**Business Rules:**

* Each city may have zero or more airports
* Each airport may have zero or more flights
* Each Flight may have zero or more Flight Schedules
* Each Scheduled Airplane will have one and only one flight schedule
* Each Scheduled Airplane will have one and only one Airplane information
* Each Scheduled Flight will have one or more Scheduled Seats
* Each Booking will have one and only one Seat
* Each order will have one or more bookings
* Each order will have one and only one credit card
* Each Employee (Salesman) may have zero or more orders
* Each Employee will have one and only one role
* Each role will belong to one and only one department
* Each department will belong to one and only one company
* Each company may have zero or more company addresses
* Each Employee will have one and only one account
* Each account may have zero or more missions
* Each account may receive zero or more reports
* Each Mission may have zero or more reports
* Each Employee will have one and only one Person information
* Each Person may have zero or more Email Address
* Each address will belong to one and only one Person
* Each Customer will have one and only one Person information
* Each Customer will have one and only one Customer Type
* Each Customer may have zero or more credit cards

**Design Requirements (Credit to Professor Simon Wang):**

* Use Crow’s Foot Notation.
* Specify the primary key fields in each table by specifying PK beside the fields
* Draw a line between the fields of each table to show the relationships between each table. This line should be pointed directly to the fields in each table that are used to form the relationship.
* Specify which table is on the one side of the relationship by placing a one next to the field where the line starts.
* Specify which table is on the many side of the relationship by placing a crow’s feet symbol next to the field where the line ends.

**Design Decisions:**

|  |  |  |
| --- | --- | --- |
| **Entity Name** | **Why Entity Included** | **How Entity is Related to Other Entities** |
| Airport | The important airport data includes airport name as well as the location(city). The Airport, which can be regarded as both departure and destination, like lots of supporting points around the world, propping up all the flights. | The airport entity is directly related to the flight entity as crucial factors(Departure Airport and Arrival Airport) in flight information, and there are more than one flight under a airport. The entity’s foreign key cityid representing the specific city so that insight may be gained about more details about location. |
| City | As an attributes of airport data, includes city name. | The city entity is directly related to the airport entity as a location factor. One city can relate to more than one airports. |
| Flight | As the bridge connecting different airports, it is important to gain information about departure airport, arrival airport and the distance of the whole route. | The flight entity is directly related to the flight schedule entity as a crucial factor. Each flight relates to more than one flight schedules. |
| FlightSchedule | One of the primary purposes of the database is to collect information about flight schedule factors related to sale. The important flight schedule data to collect include sales information as well as flight and airplane information, departure and arrival time(scheduled and actual time). Specific seat information that redeemed at a flight can be determined from seatschedule table data. | As the core entity in the database, the flightschedule entity’s primary key, flightscheduleid, relates it to airplanes, flights and seat schedule so that all information will feed back to booking system. As there is many-to-many relationship between flight schedule and airplanes, an associative entity is created as described for each entity. |
| Airplane | The flight schedule will arrange an airplane to execute a flight mission, and the seat schedule will be subject to the max seat number | The airplane entity is directly related to the flightschedule entity through an associative entity due to many-to-many relationship. Multiple airplanes may be offered for a flight schedule and there are many flight schedules for which a airplane may fly. |
| SeatSchedule | Each customer can choose a seat that he loves, so the booking system can track flight information and airplane model and max seats number, then customer can make a decision about his seats. | The seatschedule entity is directly related to flightschedule. Each seatschedule and its booking is one-to-one relationship. As a part of booking operations, each seat schedule matches one and only one booking. |
| Booking | After customers purchase a ticket, he will need to offer passenger’s ID. At the same time, a special price of that ticket need to be recorded. Booking Entity can help us record these information and help customers to check details about their flights. | The booking entity is directly related to the SeatSchedule entity through an associative entity due to one-to-one relationship. Each seat can only belong to one passenger and has one price. The Booking entity also is related to Order entity by a many-to one relationship. |
| Order | Customers will purchase several tickets in one order, therefore order entity can help to stored all booking records and their purchasing date in each order. Order entity can also record information about employees who handle it and customers who buy it, which will be benefit for companies to assign tasks and for customers to check their orders. | The Order entity is related to Booking entity by an one-to-many relationship. Each order will contain several booking detail. Order entity also is related to Employee entity and Customer entity by a many-to-one relationship.Several Orders are handled by a special employee and bought by one customer. |
| BillingAddress | This help to verify customer, and each booking order must have one and only one billing address, with the same address information as Address entity. | It directly relates to Order entity with one-to-one relationship. And It includes primary BillingAddressID and references Order Entity with OrderID, and details of address. |
| Customer | The key function of out database to allow customers to purchase tickets, therefore this entity will collect all important information of customers such as name, email, billing address, passport number and so on. | The Customer entity is directly related to the Order entity and Credit Card entity through an associative entity due to one-to-many relationship. Each customer can have several credit cards and purchase many orders. Customer Entity also is related to Person entity by an one-to-one relationship and is related to Customer Type by a many-to-one relationship. |
| CustomerType | Airline companies often offer some different promotion activities to different kinds of customers. This entity will store total customer types and their promotions. | The CustomerType entity is related to Customer entity through an one-to-many relationship. Each customer type will have a list of customers. |
| CreditCard | The main billing method of customers is to use credit card, therefore this entity will record customers’ credit card that they have used. It is convenient for customers to pay bills since they do not need to offer the same information if they have offered. | The CreditCard entity is related to Customers entity through a many-to-one relationship. Each customer will have several credit card for them to purchase tickets. |
| Person | Each employee or customer is a person and shares some attributes of person. The person entity is used to collect personal information such as name, gender, date of birth, phone and address. It will also share these information to employees and customers. | The Person entity is related to Employee entity and Customer entity by an one-to-one relationship. It also is related to Email entity through an one-to-many relationship. Each person will have several email address. |
| Address | Each Person should have address, but they can have more than one address. It is very important to fill out the order and also it’s an important factor related to a person. | It directly relates to Person Entity with many-to-one relationship. It includes primary AddressID, references to Person Entity with PersonID, and other details of address information. |
| EmailAddress | Airline companies will send some important message to their customers by email. This entity will collect all email address for all customers so that companies can contact to their customers. | The Email Address is directly related to the Person entity through an associative entity due to many-to-one relationship. |
| Employee | It is important entity for company part. It has the information which person entity does not have. It can tell others who handled the order. In addition, it decides who has the account. Employees have their own roles and they form departments, and the departments constitutes the company. | Employee entity is related to person entity and account entity. Employee is one kind of person and each employee has only one account. So they are one-to-one relationship. It also has one-to-many relationship to order, because one employee can handle many orders. It also has many-to-one relationship to role entity, because many employees may have the same role. |
| Role | It stores the every roles in the company. It can tell employee which role and level they are. It can also be used for the further level management. | Role entity is related to employee entity, which is on-to-many relationship. It also has many-to-one relationship with department entity. Under one department, there is many roles. |
| Department | It can tell employees which department they belong to. It also can divide the company into several departments which is convenient to manage. | Department entity is related to role entity, which is one-to-many relationship. It also has many-to-one relationship with company entity. Each company has many departments. |
| Company | If customers are interested in which company is doing service for them. The entity will tell them the company’s name. Each company has many departments. The employees also want to know which company the work for. | The company entity is related to department entity which is one-to-many relationships. A company can have many departments. |
| Account | Each employee should has their own account which is used for work. Their work can be recorded under the account. It also connect the mission and report part. | Account entity has one-to-one relationship with employee entity. Each employee only has one account. In addition, it is related to mission entity. It is one-to-many relationship, because each account can write many missions and reports. |
| Mission | The leaders will give the mission(s) to employees, so their accounts will have zero or more missions to execute. The unique mission belong to unique account. | It directly relates to Account entity with many-to-one relationship. It includes primary MissionID, references to Account entity with AccoutID, and other details of mission content. |
| Report | Employees can write the report to their leader. The leaders can check the process and result of the mission. It is used for work communication. | It only has many-to-one relationship with Mission entity. Many reports can be from one account. |