Airline Database Management

Airline industry is one of the largest and ever-growing businesses in the world. It caters to a vast majority of the population. These airlines are committed to provide a wonderful user experience. The aircrafts are equipped with modern day technologies which not only guarantee a safe flight, but also a comfortable journey to the passengers.

The Airlines offers certain kinds of discounts, based on the person- such as discounts for children and senior citizens and early booking of flights. Using a database, it becomes easier to keep track of the discounts.

Also, online reservations can be done for reservation of seats which are secure and allows customers the flexibility of booking seats from the convenience of their homes. Having a computerized database system to manage all the transactions of customers, along with keeping track of all the employees of the organization would be much easier.

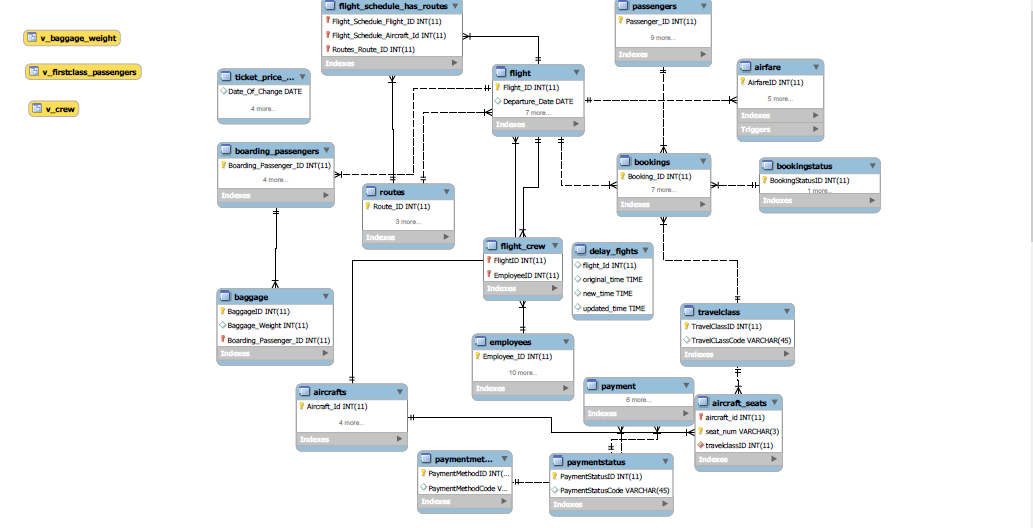
In this project I will be using stored procedures and transactions for adding customers or booking the seats on a flight. In case of cancellations, based on certain factors, the refund would be given to the customer.

Main Entities:

* **Aircrafts**: Contains the details of the aircrafts owned by the airline.
* **Customers**: Contains the details of the customers.
* **Employees**: Contains the details of the employees working for the airline.
* **Routes**: Each flight will have a route, based on the departure and arrival locations.
* **Flight Schedule**: This contains the date time, departure, arrival of each flight.
* **Booking**: Contains the transaction details such as booking a seat by customers.
* **Flight Crew**: Contains the discount details based on certain factors.

Relationships:

* There will be a one-to-one relationship between the flight schedule and aircrafts.
* There will be a many-to-many relationship between customers and transactions.
* There exists a many-to-many relationship between the flight schedule and routes.

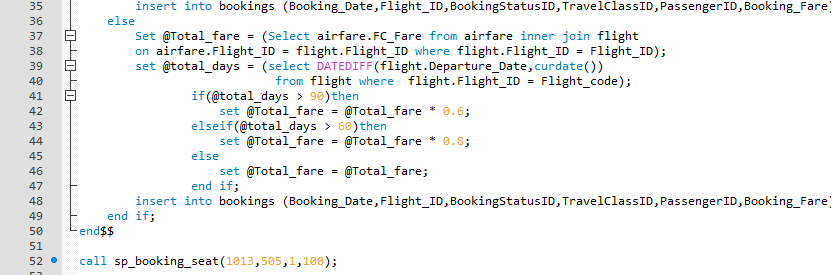


An EER Diagram with connecting the entities and showing the relationships among them. Along with the views.

In this project I have used Stored Procedures, Triggers and Views which are described in detail below:

* Stored Procedures:
* **sp\_booking\_seat:** This stored procedure is used to book a flight ticket for all the passengers. Here I have considered the date of booking and the date of departure of a particular flight, and based on that the passenger will be given a discount on the total fare.The screenshot is attached below.

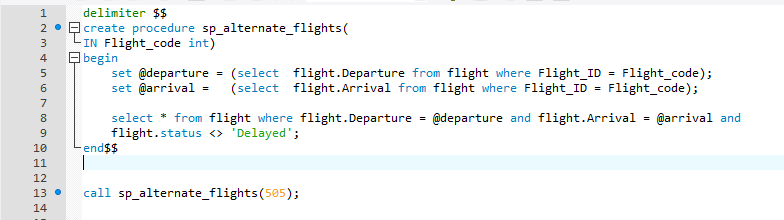




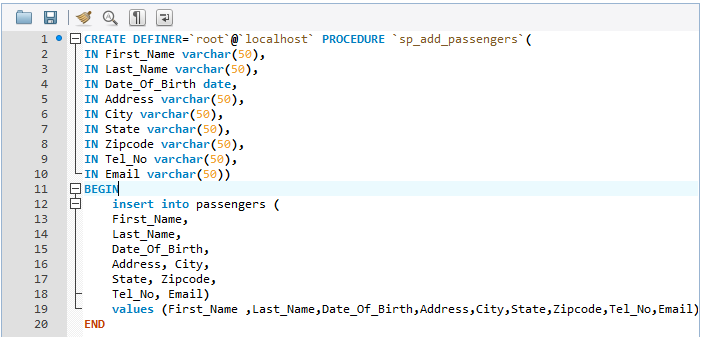
* **sp\_cancel\_Ticket:** Thisstored procedure cancels a ticket booked by customer by providing a refund amount based on the number of days before departure the ticket was cancelled by the passenger.



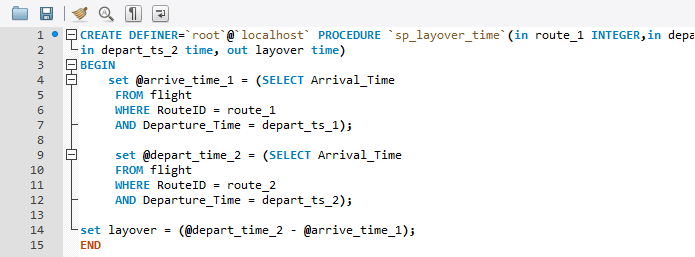
* **sp\_alternate\_flights:** This is a stored procedure to display an alternate list of flights for the same departure and arrival locations, when a flight is delayed or cancelled. This helps to re-allocate passengers from the delayed flight to another.



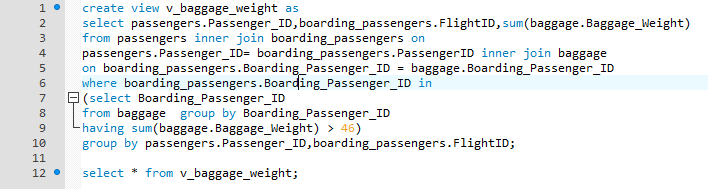
* **sp\_add\_passengers:** This stored procedure is used to add a passenger details onto the database. In case a new passenger is to be added, this stored proc can be called directly by passing all the details. This can be used to control, who all have access to the database.



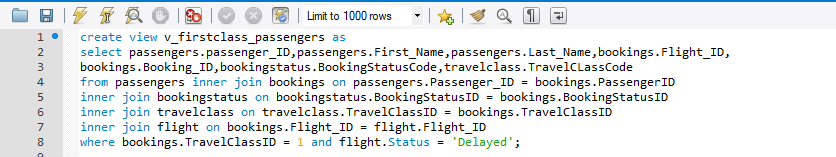
* **sp\_layover\_time:** This displays the layover time between the two flights which are in different routes.



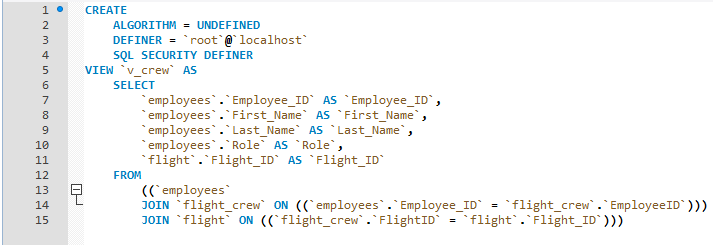
* **Views:**
* **v\_baggage\_weight:** This view is used to display all the passengers who have exceed the allowed baggage weight limit of 46 kilos. It is selected based on the passengers who are boarding the flights.

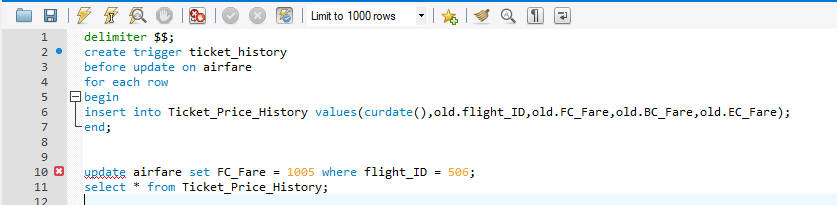


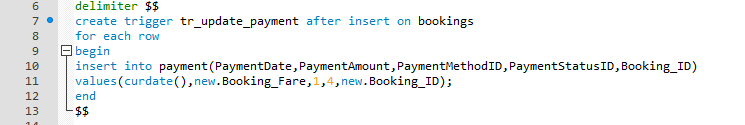
* **v\_firstclass\_passengers:** Generally, first class passengers are given preferential treatment over the other passengers. So, in case there is a delay in the departure of the flights or a flight is cancelled these passengers have to be accommodated elsewhere. This view will give us the details of those passengers whose flight was delayed.

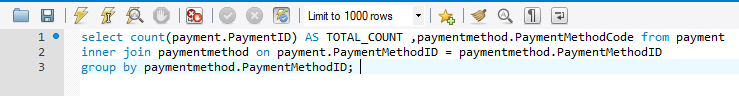


* **v\_crew:** This gives us all the details of the crew members who are on a flight.



* **Trigger:**
* **ticket\_history:** A triggerwhich is fired when there is an update on the airfare. This trigger saves the changes of the price of the ticket in a separate table - **Ticket\_Price\_History** which can be referred in case of any further updates. 
* **tr\_update\_payment:** A trigger is fired whenever a ticket is booked, and the values are inserted to the bookings table. This trigger updates the table payments, which contains the payment details.



* 

Gives the total count of the different payment methods to the airline management.