This project is going to span through various procedures and step that will be exhaustively highlighted in this note

**Step 1:** Collecting and cleaning of data- this step involves gathering of data that is related to aviation from various source which included internet, journals etc. it should be noted that this data are only for testing purposes as real-time data will be used when the project is completed, tested and deployed.

**Step 2 :** This step encompasses creation of relevant tables that will house data for the aviation company i.e., it involves the use of programming language to create a backend database models in the form of relevant tables. For the sake of this project we will be using the SQL developer on an ORACLE database to create our objects, indicating the different procedures and scripts for it, thus, procedures and script given below.

For this project we are going to utilize 13 relational different objects that will be created with SQL script in their preceding order. Viz:

1. **Booking table**
2. **Ticket table**
3. **Payment table**
4. **Department table**
5. **Employee table**
6. **Flight\_Crew table**
7. **Meal table**
8. **Booking Agency table**
9. **Passenger table**
10. **Flight table**
11. **Aircraft procurement table**
12. **Flight Service table**
13. **Airport table**

It should be noted that creating table in their preceding order is mainly due to the relationships between this table (i.e. the primary and foreign key relationships). **Preceding order:** there will be no child without a parent in other words you cannot create a table referencing a primary key without first creating the table that have the primary key values been referenced.

Having said that, we can now proceed to creating the aforementioned tables in their preceding order.

The first batch of tables we are to create in no particular order are:

Booking Agency, Airport, Aircraft procurement, Meal, Department, Flight service and Flight crew tables. This first batch tables in the aviation project are created first because they are all parent tables that other tables are yet to create reference to, in other words, they do not reference any other parent tables, thus creating them first is advisable.

Below is are the sql script for the table creation:

***--BOOKING AGENCY***

***Create table Booking\_Agency (***

***Agency\_ID number (7) null not primary key,***

***Agency\_Name varchar2 (18) not null,***

***Years\_of\_Contract varchar2 (15) not null);***

***-- CREW TABLE***

***Create table Crew(***

***Crew\_ID number (7) not null primary key,***

***Crew\_Name varchar2 (10) not null,***

***Crew\_Captain varchar2 (20) not null);***

***--AIRPORT***

***Create table Airport(***

***Airport\_ID varchar2 (10) not null primary key,***

***Airport\_Name varchar2 (60) not null,***

***City\_Country varchar2 (20) not null,***

***Continent varchar2 (15) not null);***

***-- AIRCRAFT PROCUREMENT***

***Create table Aircraft\_Procurement(***

***Aircraft\_Code varchar2 (15) primary key,***

***Aircraft\_Name varchar2 (20) not null,***

***Man\_Com varchar2 (20) not null,***

***Man\_Date date,***

***Expected\_LifeSpan varchar2 (15) not null,***

***Cost number (8) not null,***

***Procure\_Officer varchar2 (25) not null);***

***--MEAL***

***Create table Meal(***

***Meal\_ID number (7) primary key,***

***Menu varchar2 (75) not null,***

***Cheff varchar2 (30) not null);***

***--DEPARTMENT***

***Create table Department(***

***Dept\_ID number (7) primary key,***

***Department\_Name varchar2 (20) not null,***

***Department\_Head varchar2 (18) null default 'No HOD Currently',***

***Manager\_ID number (7) null default 'Manager not Assigned');***

Having created this tables, we can then proceed to creating the remaining tables which have reference directly to this first batch of tables. And the first of such to create is Flight Service Table. The following SQL scripts creates them.

***-- FLIGHT SERVICE***

***Create table Flight\_Service(***

***SerVice\_ID varchar2 (7) primary key,***

***Aircraft\_Code varchar2 (15) not null references Aircraft\_procurement(Aircraft\_Code),***

***Service\_Engineer varchar2 (18) not null,***

***Man\_Date date,***

***last\_Service\_Date date,***

***Next\_Service\_Date date,***

***Service\_Category varchar2 (7) not null,***

***Cost\_of\_Servcing number (9) not null);***

**Note:** Aircraft\_code here in Flight Service Table is directly reference Aircraft\_code in Aircraft\_procurement Table. This references is what gives rise to Parent and Child Table, in this example Flight\_Service Table is the Child Table.

***--Booking Table***

***create table Booking(***

***Booking\_ID varchar2(15) primary key,***

***Trip\_Round varchar2 (3) not null,***

***Booking\_Date date,***

***Return\_Date date,***

***Booking\_Time varchar2 (15) not null,***

***Adult number (7) not null,***

***Children number (7)null,***

***Infant number(7) null,***

***Agency\_ID varchar2 (18) default 'Direct Booking' references Booking\_Agency(Agency\_ID));***

**Note:** In the Booking Table, Agency\_ID is a foreign key which reference Agency\_ID a primary key in the Booking\_Agency Table. Parent-Child relationship.

***-- PAYMENT TABLE***

***Create table Payment(***

***P\_REF varchar2 (7) not null,***

***B\_Cat varchar2(15) not null check(B\_Cat in ('Adult','Children','Infant')),***

***P\_ID varchar2 (7) primary key,***

***Amount\_Payable number (9) not null,***

***Grand\_Total number (9) not null,***

***Booking\_ID varchar2 (15) not null references Booking(Booking\_ID));***

**Note:** Payment Table (Booking\_ID) is foreign key to Booking Table whose primary key is Booking\_ID that is been referenced.

***--TICKET TABLE***

***Create table Ticket(***

***Ticket\_No varchar2 (15) primary key,***

***P\_ID varchar2 (7) not null references Payment(P\_ID),***

***Booking\_ID varchar2 (15) not null references Booking(Booking\_ID),***

***Seat varchar2 (10) not null unique,***

***Transit\_Seat varchar2 (18) default ' Not Applicable',***

***Sec\_T\_Seat varchar2 (18) default 'Not Applicable',***

***Departure\_Date date,***

***Departure\_Time varchar2 (15) not null,***

***Arrival\_Date date,***

***Arrival\_Time varchar2 (15) not null);***

**Note:** In the Ticket Table there are two references or Foreign keys referencing Payment Table and Booking Table respectively. i.e there are two parent tables ( Payment and Booking) that the Ticket have referenced.

***-- PASSENGER TABLE***

***Create table Passenger(***

***Passenger\_ID varchar2 (15) primary key,***

***Ticket\_No varchar2 (15) not null references Ticket(Ticket\_No),***

***First\_Name varchar2 (15) not null,***

***Last\_Name varchar2 (15) not null,***

***Age number (7) not null check( Age>=2),***

***Num\_of\_Lugg varchar2 (12) default 'No Entry',***

***Lugg\_Wght varchar2 (18) default 'Not Registered',***

***Gender varchar2 (3) not null check(Gender in ('M','F')),***

***Phone varchar2(15) not null unique,***

***Email varchar2(15) not null unique,***

***P\_Means\_of\_Id varchar2 (25) not null check(P\_Means\_of\_Id in ('Passport','Drivers license', 'Student ID')),***

***ID\_No varchar2 (18) not null unique,***

***ID\_Ex\_Date date);***

**Note:** The Passenger Table Ticket\_No is a foreign key to Ticket Table i.e Values in the Ticket\_No in Passenger Table are first to be found in Ticket\_No in Ticket table, else Oracle will raise an error of integrity constraint

***--FLIGHT TABLE***

***Create table Flight(***

***Flight\_ID varchar2 (25) primary key,***

***Passenger\_ID varchar2 (15) not null references Passenger(Passenger\_ID),***

***Flight\_No varchar2 (15) not null references Aircraft\_Procurement(Aircraft\_Code),***

***T\_Flight\_No varchar2 (18) default 'Not Applicable',***

***T2\_Flight\_No varchar2 (18) default 'Not Applicable' ,***

***flight\_Type varchar2 (20) not null check( flight\_Type in('International', 'Domestic')),***

***Departure\_Day varchar2 (15) not null,***

***Departure\_Airport varchar2 (75) not null references Airport(Airport\_Name),***

***D\_Terminal varchar2 (5) not null,***

***Arrival\_Day varchar2 (15) not null,***

***Arrival\_Airport varchar2 (75) not null,***

***A\_Terminal varchar2 (5) not null,***

***Airport\_Code varchar2 (15) not null,***

***Transit\_Airport varchar2 (18) default 'Not Applicable',***

***Transit\_Airport2 varchar2 (18) default 'Not Applicable',***

***Flight\_Crew\_ID varchar2 (7) not null references Crew(Crew\_ID),***

***SerVice\_ID varchar2 (7) not null references Flight\_Service(SerVice\_ID),***

***Meal\_ID varchar2 (7) not null references Meal(Meal\_ID));***

**Note:** The flight table have six foreign keys which references the primary key in Passenger Table, Aircraft\_Procurement Table , Airport table , Crew Table, Flight\_Serice table and Meal Table respectively.

***-- EMPLOYEES TABLE***

***Create table EmployeesA(***

***Emp\_ID varchar2 (7) primary key,***

***First\_Name varchar2 (15) not null,***

***Last\_Name varchar2 (15) not null,***

***DOB date,***

***Hire\_Date date,***

***Manager\_ID varchar2(25) default 'No Manager Assigned' references EmployeesA(Emp\_ID),***

***HOD\_ID varchar2 (30) default ' No HOD At The Moment',***

***Job\_Title varchar2 (25) not null,***

***Gender varchar2 (5) not null check( Gender in ('M','F')),***

***phone varchar2 (20) not null unique,***

***Email varchar2 (40) not null unique,***

***Salary number (9) not null,***

***Bonus number (7) null,***

***Crew\_ID varchar2 (18) default 'Not Applicable' references Crew(Crew\_ID),***

***Department\_ID number (7) not null references Department(Dept\_ID),***

***EMP\_Status varchar2 (15) not null check ( EMP\_Status in ('Active','Inactive')));***

**Note:** In the EmployeesA table there are three reference, one is an in-table reference which is the Manager\_ID referencing Emp\_ID in the same table EmployeesA what this means is before any individual can be a manager he/she must first be an employee of that establishment. In a nutshell you cannot bring in a person out of the blue to be a manager, except on contracts basis where the said contract have a specified expiring date. The other two foreign keys are Crew\_ID and Department\_ID, they both reference to Crew and Department Table respectively.

Terminology

**Primary key:** This is the unique identifier that defines a particular row or record in a table.

**Foreign Key:** It is an identifier that relates one or more tables together by linking records of one table to another using a unique precedent (Primary key).

**Check Constraints:** Check Constraints is a way of maintaining the integrity of any inputs into a column in a particular table. When a check constraint is on a column it means no other variable inputs will be allowed into the column if it was not checked from creation.

**Default Constraint:** It is a variable either number or character that should be in a record with no value instead of a null.