University of Derby

The School of Computing and mathematics

0

Module Code and Title : 7CS082 Database development

Name :

Submit To :

Project Title : Airline Company

Report of database on MySQL workshop and Oracle database 10g server

**AIRLINE COMPANY SCOPE**

We develop this system to create a easy user interference for the customers and airline companies that have old systems. By using this system, they can do a lot of difficult task easily and will be able automate a lot work which we have been doing all manually and on paper. This system will allow us to store a lot of data about staff, customers and flights

**IMPLEMENTATIONS:**

This project was all manually functioned before after thing from booking ticket to managing flights and staff member were manually so we implemented this system to automated system which can manage staff, book flights, manage airplanes and do all kind of functions that airline company required.

**ACHIEVEMENTS OF THIS PROJECT:**

The project has achieved its objectives by:

1. providing technical assistance for our customers and passengers
2. Providing easy system to book a flight or manage our staff or flights
3. Provided the best system to store data about staff, airplane, flights and passenger.

**KNOWLEDGE AND TOOL USE:**

* We used MySQL to create an ERD and tables
* We used Oracle Database 10g Server tool where we perform **SQL(DDL ,DML)**
* We also used Moon modular to create another ERD for Oracledatabase 10g server
* We use normalization techniques toReduce data redundancy of tables

**ER model and definition of key concepts**

**Short discussion**

**ERD** (**entity relationship diagram**) has three kind of relationships (**one-to-one, one-to-many, many-to-oneand many-to-many**). We made eight tables through that relationship.the table are given blow.

|  |
| --- |
| **Tables** |
| User |
| Booking |
| Ticket |
| Passenger |
| Flight |
| Aero plane |
| Staff |
| Staff Rating |

**Why you have chosen a particular entity, attributes and relationships?**

The logicof making entity is simple .like create ERD and start the work with base table like **User,** Then **Booking, Ticket ,Passenger** etc. The System of Airline Company works like that.

Therefore

We first create **User Table**where he login through his account and starts to **booking tickets**.he also can watch the remain seat and full detail of **flights, aero plane**, etc. So we made it easierthan the waiting in long line.thefurther details and ERD concept given blow.

**How we made aattributesrelationship?**

**User table**

As we take the first table **(User table**) whose column are (User**\_id (Primary key), User\_name, User\_password**),the primary key of **(User table**) is the foreign key of (**Booking table**) theirrelationship with each other is (**one to one**).

**Likewise**

**Booking table**  
The Column of **Booking table** are(**Booking\_id , (Primary key),User\_id(Foreign key), B\_amount)**.The **Booking table** primary key is foreign key of Ticket table ,their relationship with each other is (one to one).

**Passenger table**

The Column of **Passenger table** are (**Passenger\_id(Primary key), P\_name, P\_surname, P\_gender, P\_contact\_no, P\_Gmail**) The **Passenger table** primary key is foreign key of Ticket table , their relationship with each other is (one to one).

**Ticket table**

The Column of Ticket table are (Ticket\_id ,T\_price, T\_date, F\_taskoff\_time , F\_landing\_time,booking\_id(**Foreign key**),Flight\_id (**Foreign key**),Passenger\_id(**Foreign key**)). The Ticket table has three **Foreign key and the** relationship of each on has (one to one).

**Flight Table**

The Column of **Flight table** are (**Flight\_id (Primary key),F\_Seat\_number,F\_seat\_amount,F\_seat\_available ,F\_takeoff\_time,F\_landing\_time,Aeroplane\_id (foreign key),,staff\_id (foreign key),,C\_flight\_takeoff,C\_flight\_land**). The **flight table,Booking table, passenger table**  primary key is foreign key of **Ticket table**, their relationship is only with ticket table (one to many).

**NOTE:**F\_taskoff\_time, F\_landing\_time is present in two table (Ticket and Flight) table the time and date both same in these tables because it is used as a information in ticket for passenger

**Aero Plane Table**

The Column of **aeroplane table** are (**Aeroplane\_id (Primary key), A\_NAME,A\_Manufacturer, A\_model\_no ,Company\_id** ) The **aeroplane table**primary key is foreign key of **Flight**, their relationship with each other is (one to one).

**Staff table**

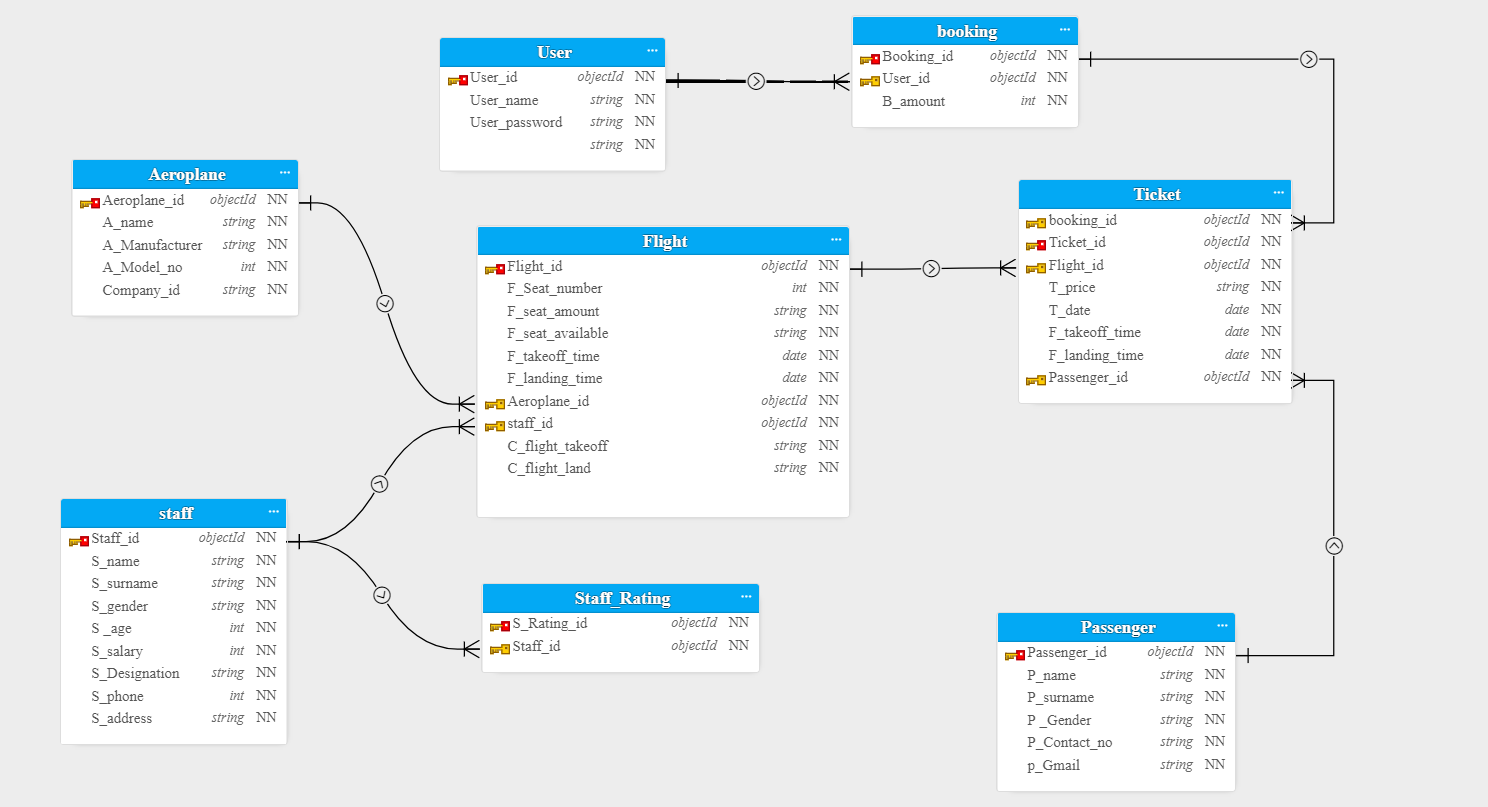
The Column of Staff table are (**Staff\_id (Primary key),S\_name,S\_surname,S\_gender,S \_age,S\_salary,S\_Designation,S\_phone,S\_address**) The **Staff table** primary key is foreign key of **Flight table** and **Staff Rating table**, their relationship with each other is (many to one).

**Staff ratingTable**

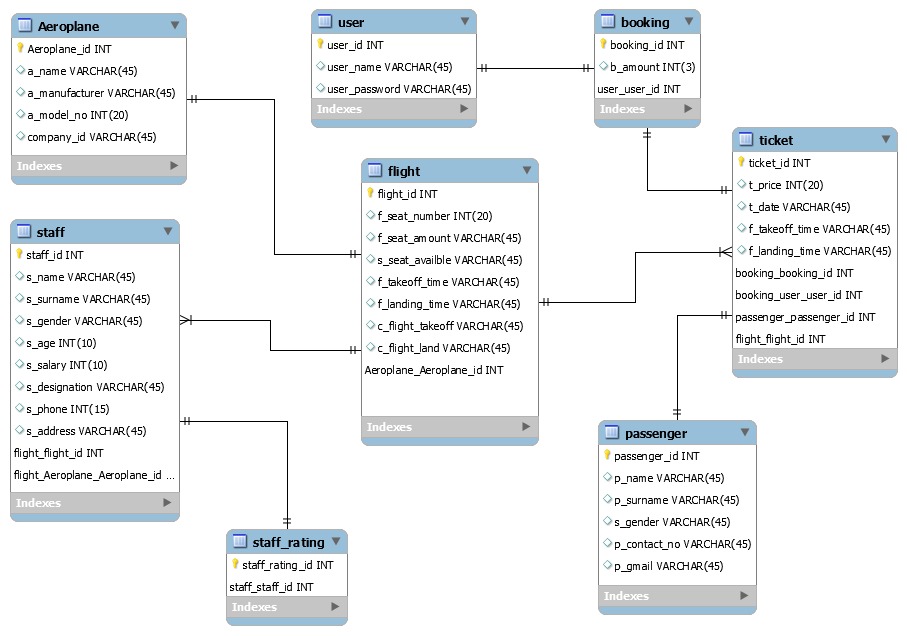
The Column of **Staff table** are (**S\_Rating\_id (Primary key), Staff\_id (foreign key)** ) The **Staff rating table** primary key is foreign key of **Flight table**, their relationship with each other is (one to one).

|  |  |  |
| --- | --- | --- |
| **TABLES** | **Primary key** | **Foreign key** |
| User | User id | none |
| booking | Booking id | User id |
| Ticket | Ticket id | Booking id , Flight id ,Passenger id |
| Flight | Flight id | Aeroplane id ,Staff id |
| Aeroplane | Aeroplane id | None |
| Staff | Staff id | None |
| Staff rating | S\_Rating\_id | Staff id |

**E-R Diagram from Moon Moduler Application**



**E-R Diagram from MySQL**

****

**Unnormalized tables**

The1st Unnormalized Table Booking

**Booking table**

|  |  |
| --- | --- |
| **Columns of table** | **Definition section** |
| Booking\_Id | Booking Id ( Primary key ) |
| F\_seat\_number | Number of the seat which is booked |
| S\_designation | Staff scale or staff duty like(Pilot,Air hostess etc . ) |
| T\_date | Date of booking ticket |
| Passenger\_Id | Passenger Id for identify of passenger |
| User\_Id | User Id of who want to book a flight |
| Flight\_Id | Flight Id for identify flight |
| Aeroplane\_id | Id of aero plane is to recognizeaero plane |
| User\_Name | Name of User |
| Ticket\_ID | Ticket ID is assigned to passenger every passenger has it’s unique ticket Id |
| S\_name | Name of Staff |
| C\_Name | Name of City |
| P\_age | Age of Passenger |
| User\_Password | Password of User |
| T\_price | Price of Ticket |
| P\_Name | Name of Passenger |
| B\_amount | Total amount of this booking |

The 2nd Unnormalized table Ticket

**Ticket table**

|  |  |
| --- | --- |
| **Columns of table** | **Definition section** |
| Ticket\_id | Ticket ID is assigned to passenger every passenger has it’s unique ticket Id (primary key) |
| Flight\_id | Flight Id for identify flight |
| A\_Manufacturer | The name of country which build the aero plane |
| Staff\_id | Staff id to identify every employees or crew |
| A Module\_no | Model number of aero plane like (ML-201) |
| T\_price | Price of ticket |
| F\_takeoff\_time | Departure date and time |
| P\_gender | Gender |
| P\_contact\_no | Phone number of passenger |
| S\_rating\_id | Rating id of staff to give a credit |
| F\_seat\_number | Number of seat which is booked |
| Aeroplane\_id | Id of aero plane is to recognize aero plane |
| Booking\_Id | Id of the booking from users |
| T\_date | Date of order ticket /conform ticket |
| C\_flight\_takeoff | City of Departure |
| C\_flight\_land | Arrive (City where aero plane is arrive) |
| F\_takeoff\_time | Departure date and time of flight |
| F\_landing\_time | Arrive Departure date and time flight |

The 3rd Unnormalized table Passenger

**Passenger table**

|  |  |
| --- | --- |
| **Columns of table** | **Definition section** |
| Passenger\_id | Passenger id (primary key) |
| P\_name | Name of Passenger |
| P\_surname | Last name of Passenger |
| P\_Gender | Gender of Passenger |
| P\_contact\_no | Phone number of Passenger |
| P\_Gmail | Gmail of Passenger |

The 4th Unnormalized table Flight

**Flighttable**

|  |  |
| --- | --- |
| **Columns of table** | **Definition section** |
| Flight\_id | Flight Id (primary key) |
| F\_Seat\_number | Number of seat which is booked |
| S\_Rating\_id | Rating id of staff to give a (credit or star) |
| S\_age | Age of Staff |
| F\_seat\_amount | Total seat in flight |
| F\_seat\_available | Remaining seat which is open of booking |
| A\_name | Name of Aero plane |
| C\_flight\_takeoff | City of departure flight |
| C\_flight\_land | City of arrival flight |

The 5th Unnormalized table Staff

**Staff table**

|  |  |
| --- | --- |
| **Columns of table** | **Definition section** |
| Staff\_id | Staff id (primary key) |
| S\_name | Name of staff |
| S\_surname | Last\_name of staff |
| S\_gender | Gender of staff |
| S\_age | Age of staff |
| S\_Designation | Staff scale or staff duty like(Pilot ,Air hostess etc . ) |
| S\_Salary | Salary of staff |
| S\_phone | Phone number of staff |
| S\_Address | Address of staff |
| S\_Rating id | Rating id of staff to give a credit |

**Discussing the normalisation process**

**NORMAL FORM (NF)**

**1 NF follow this instruction**

In 1st NF we make no repeating values. And no repeating group

Like. (user\_name ,user\_name), then we solve it through 1st NF (user\_name).

As we know first we go to 1NF and after then we can go 2NF and 3NF. we can't go directly 1NF to 3NF or we start from 2NF

|  |  |
| --- | --- |
| Table | Columns of table |
| Booking | (Booking Id, F\_seat\_number, S\_designation ,T\_date ,Passenger\_Id ,User\_Id ,Flight\_Id ,Aeroplane\_id ,User\_Name ,Ticket\_ID, S\_name, C\_Name ,P\_age ,User\_Password, T\_price, P\_Name ,B\_amount) |
| Ticket | (Ticket\_id, Flight\_id, A\_Manufacturer, C\_flight\_takeoff, Staff\_id, A Module\_no, T\_price, F\_takeoff\_time, P\_gender,, P\_contact\_no, S\_rating\_id, F\_seat\_number, Aeroplane\_id, Booking\_Id, T\_date, F\_landing\_time) |
| Passenger | (Passenger\_id, P\_name, P\_surname, P\_Gender, P\_contact\_no, P\_Gmail) |
| Flight | (Flight\_id, F\_Seat\_number, S\_Rating\_id, S\_age, F\_seat\_amount, F\_seat\_available, A\_name) |
| Staff | (Staff\_id, S\_name, S\_surname, S\_gender, S\_age, S\_Designation, S\_Salary, S\_phone, S\_Address, S\_Rating id) |

**2NF Any non-key field should entirely depend on its primary key follow this instruction**

* We can perform 2NF after the result of 1st NF
* We have to sure about no partial dependency

To perform 2NF we must have 1NF solution and every non-primary keycolumn must be dependent on all parts of the primary key.

|  |  |
| --- | --- |
| Table | Columns of table |
| Booking | (Booking Id ,B\_amount, User\_id,user\_name,user\_password) |
| Ticket | (Ticket\_id, T\_price, T\_date, F\_Takeoff\_time, F\_landing\_time) |
| Passenger | (Passenger\_id, P\_name, P\_surname, P\_Gender, P\_contact\_no, P\_Gmail) |
| Flight | (Flight\_id, F\_Seat\_number, F\_seat\_amount, F\_seat\_available, F\_Takeoff\_time, F\_landing\_time) |
| Staff | (Staff\_id, S\_name, S\_surname, S\_gender, S\_age, S\_Designation, S\_Salary, S\_phone, S\_Address, S\_Rating id) |
| Aeroplane | ( Aeroplane\_id , A\_name,A\_maunfacturer,A\_model\_no,Company\_id) |

**3NF Any non\_key field dependent on other key field**

To perform 3NF we must have 2NF solution and every no non-primary key column may be dependent on all parts of the non-primary key either we can say, the primary key and it column will spilt in another table and make a new table from **Booking table** from whom we made User table .( User\_id,user\_name user\_password) .the spilted table primary key must be another table foreign key .

|  |  |
| --- | --- |
| Table | Columns of table |
| User | ( User\_id(**PK**),user\_name,user\_password ) |
| Booking | (Booking Id(**PK**) ,B\_amount, User\_id(**FK**)) |
| Ticket | (Ticket\_id(**PK**), T\_price, T\_date, F\_Takeoff\_time, F\_landing\_time, Booking Id, Flight\_id(**FK**), Passenger\_id(**FK**)) |
| Passenger | (Passenger\_id(**PK**), P\_name, P\_surname, P\_Gender, P\_contact\_no, P\_Gmail) |
| Flight | (Flight\_id(**PK**), F\_Seat\_number, F\_seat\_amount, F\_seat\_available, F\_Takeoff\_time, F\_landing\_time, Staff\_id(**FK**), Aeroplane\_id(**FK**)) |
| Aeroplane | ( Aeroplane\_id(**PK**) , A\_name,A\_maunfacturer,A\_model\_no,Company\_id) |
| Staff | (Staff\_id(**PK**), S\_name, S\_surname, S\_gender, S\_age, S\_Designation, S\_Salary, S\_phone, S\_Address) |
| Staff rating | (S\_Rating id(**PK**), Staff\_id(**FK**)) |

|  |  |  |
| --- | --- | --- |
| Tables | Primary key | Foreign key |
| User | User id | None |
| booking | Booking id | User id |
| Ticket | Ticket id | Booking id , Flight id ,Passenger id |
| Flight | Flight id | Aeroplane id ,Staff id |
| Aeroplane | Aeroplane id | None |
| Staff | Staff id | None |
| Staff rating | S\_Rating\_id | Staff id |

**Data Definition Language**

“DDL perform many institutions like create a table, drop table, alter table, rename table or column name, truncate table, we build a table by creating table privileges”.

**NOTE**

**Create table:**Create table give authority to make a table.

**Alter table:**Alter table give authority of modify column, add column, drop column.

**Rename table:** for changing table name we use Rename. i.e (Rename Airplane to aircraft).

**Constraints. “**Constraints enforce rules at the table level”,

**Unique: “**it makes sure there is no same data in rows like a primary key does”.

**Primary key:** “it enforces the uniqueness of the column data”.

**Foreign key:** “foreign key makes a relationship between a primary key or a unique key in the different table”.

**Data Manipulation Language**

**As DDL performs its task, we perform DML to insert, update, delete, merge**

**NOTE**

**Insert:** By Insert key we add row and column in table

**Update:** If we want to change our data, we use update.

**Delete:** if we want to delete any data from table we use Delete.