**Scope of this project:**

The Airline System is management software intended to regulate all tasks of an airline. Airlines offer transportation facilities to passengers. For this purpose, they transport or rent an aeroplane. All airline processes are managed by the Airline Management System. This system can schedule flights, provide air ticket reservations, and staff administration. Daily flight updates can also be retrieved by using the system.

**How did you implement it?**

We can implement this system by using cutting-edge technologies to make software that will allow possible airline passengers of “Overland Airways” to find and reserve available flights, select seats for their flights, pay for the reservation, print their boarding pass only after payment, access aeroplane maintenance reports and be notified of flight cancellation or delays.

**Achievement**

This high-tech system was developed to eliminate the disadvantages of the existing labour-intensive system. The planned system will have many compensations over the old manual labour-intensive system. This project will be modified for users. It will be premeditated in such a way that all new users can use all the options very easily like booking a flight or rescheduling the flight. It is implemented in a simple and fast referencing way. The benefit of the planned system is that security is preserved in the new system. The whole system is planned in such a way that users can make entries easily also quickly.

Knowledge and tool

* We used Oracle 10g server tool
* We also used Moon modular for E-R Diagram
* We use normalization to Reduce the data redundancy of tables and Partial Dependency.
* We use MS ACCESS to create table and E-R Diagram

**DISCUSSION OF THE E-R DIAGRAM**

We build ERD (entity relationship diagram) to make a connection and logical picture like the shape of a database plan. It helps us to recognize what we have to work on and like our plan Airline company, we describe our whole project at our ERD.

**ERD Relationships**

One to one

One to many

Many to one

Many to many

**We define our ERD connection table to table.**

|  |
| --- |
| **Passenger Table** |
| **Ticket Table** |
| **Aeroplane Table** |
| **Flight Table** |
| **Crew Table** |

**Why you have chosen a particular entity**

We made a small E-R diagram to decrease the complexity of the Airline company project more table and more column means more complexity. That is why we use a particular entity, so it makes it easier to understand even for the non-technical person. Therefore, Our E-R Diagram starts from (the passenger table).

**Attributes and relationships?**

**Passenger**

Our first table is (**PASSENGER TABLE)** and the column of (passenger table) are (Passenger\_id (**PK**), p\_name, p-surname, p\_Sex, p\_contact\_no)  **passenger table** has a relationship with the ticket table where passenger\_id is **(FK)** in the **ticket table**. The Relationship of the passenger table (one-to-one) with the **ticket table**.

**Ticket**

Our Second table is (**Ticket table**), and the columns of (**Ticket table**) are (ticket-id**(PK**), t\_price,passenger-id(**FK**), Info\_F\_departure, Info\_F\_arrival, Info\_F\_departure\_date, Info\_F\_arrival\_date) **Ticket Table** has a relationship with FLIGHT TABLE and with **PASSENGER TABLE** where Passenger-id is **(FK**) in **TICKET TABLE** and Ticket\_id is **(FK**) in **FLIGHT TABLE**. The Relationship of the **Ticket Table**(one to one) with the **passenger table** and **ticket table**

(One-to-one) with **FLIGHT TABLE**

**Aeroplane**

Our third table is (the **aeroplane Table**) and the column of (the **aeroplane Table**) are (Aeroplane\_id(PK), A\_name, A\_Model\_no, p\_manufacture, A\_company\_id). **Aeroplane Table** has a relationship between **Flight table** one (one to one).

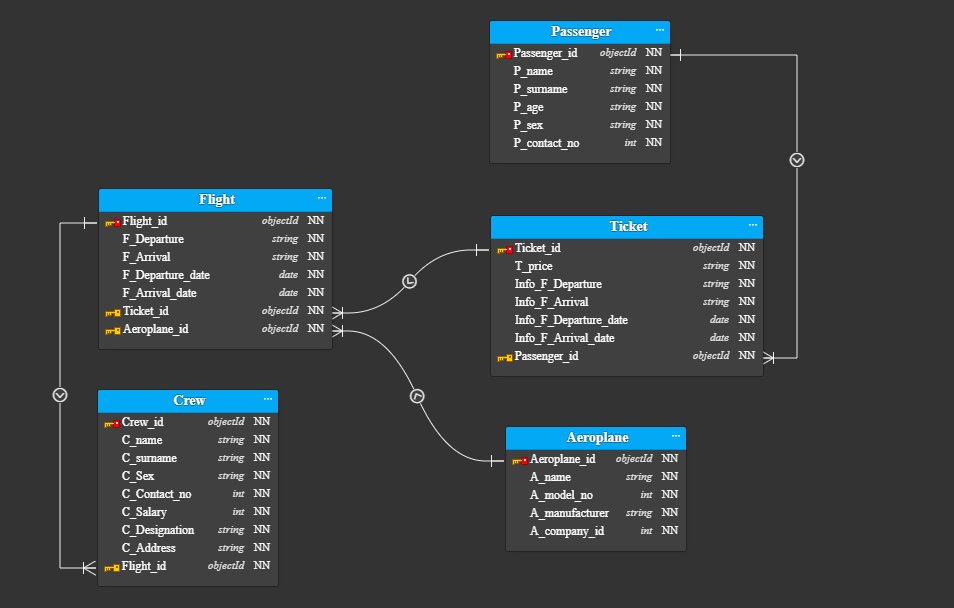
**Flight**

Our fourth table is (**The flight table**) and the column of **(Flight table**) are (Flight\_id (**PK**), Ticket\_id (FK), Info\_F\_departure, Info\_F\_arrival, Info\_F\_departure\_date, Info\_F\_arrival\_date, Ticket\_id (**FK**), Aeroplane\_id(**FK**)). **The flight table** has two (**FK**) Ticket\_id and **Aeroplane\_id.** The relationship between the **flight table** and with **Aeroplane table** is (one to many) and with the Ticket, the table is (one to one)

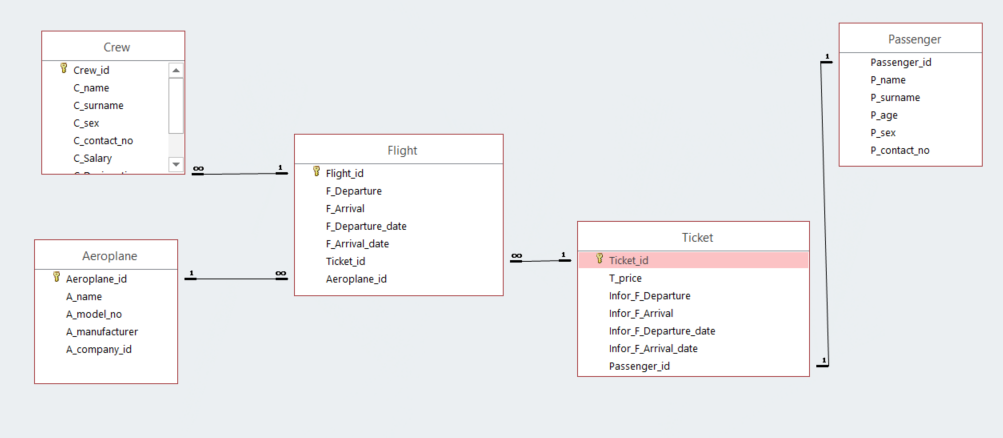
**Crew**

Our Fifth table is (the **Crew table**) and the column of (the **Crew table**) are (Crew\_id(**PK**), Flight\_id (**Fk**), C\_name, c\_surname,c\_sex, c\_Contact\_no,c\_SalaryC\_Designation, C\_address,) **Crew table** has one (FK) **Flight\_id** the relationship is (one to many)

**E-R diagram oracle database 10g expression edition**



**E-R diagram MS Access**



**Unnormalized table**

**Passenger**

|  |  |
| --- | --- |
| **Element of table** | **Definition of Element** |
| Passenger\_Id | Passenger id **(PK)** |
| P\_name | Name of Passenger |
| F\_Departure | Flight Departure |
| Ticket\_id | Ticket\_id |
| T\_price | Ticket price |
| P\_surname | Passenger last\_name |
| P\_age | Passenger age |
| P\_sex | Passenger Gender |
| P\_contact\_no | Passenger contact number |
| A\_name | Aeroplane name |
| A\_company\_id | Company id who made aeroplane |
| F\_Arrival | Flight arrival |
| Info\_F\_arrival | Information about Flight arrival on the ticket |
| Infor\_F\_Departure | Information about Flight Departure on the ticket |
| F\_Departure | Flight Departure |
| C\_designation | Duty of Crew |
| T\_price | Ticket price |
| F\_arrival | Flight arrival |
| Ticket\_id | Ticket\_id |

**Flight**

|  |  |
| --- | --- |
| **Element of table** | **Definition of Element** |
| Flight\_id | Flight\_id **(PK)** |
| F\_Departure | Flight Departure |
| F\_Arrival | Flight arrival |
| F\_Departure\_date | Flight departure date |
| F\_Arrival\_date | Flight arrival date |
| Info\_F\_Arrival\_date | Information about Flight arrival date and time on the ticket |
| Info\_F\_Departure\_date | Information about Flight departure date and time on the ticket |
| Ticket\_id | Ticket\_id |
| Aeroplane\_id | Aeroplane\_id |
|  |  |

**Crew**

|  |  |
| --- | --- |
| **Element of table** | **Definition of Element** |
| Crew\_id | Crew\_id**(PK)** |
| C\_name | Name of crew |
| C\_surname | Last\_name of Crew |
| C\_Sex | Gender of Crew |
| C\_Contact\_no | The contact number of Crew |
| C\_Salary | Salary |
| C\_Designation | Duty of Crew |
| C\_Address | Residence of crew |
| Flight\_id | Flight\_id |

**Definition of Normalization and Rules**

Normalization is the process of eliminating all anomalies of data in our tables. This includes creating tables and creating associations between them according to rules intended both to protect data and to make databases extra resilient by removing redundancies, dependencies, and varying familiarity.

**Instructions of 1NF:**

* Reduce data redundancy
* Organized database
* Flexible database design

|  |  |
| --- | --- |
| **Tables** | **Element of table** |
| Passenger | (Passenger\_Id,P\_name,T\_price,A\_name,A\_company\_id,F\_Arrival,Info\_F\_arrival,Infor\_F\_Departure,F\_Departure,C\_designation,T\_price P\_surname,P\_age,P\_contact\_no, Ticket\_id,P\_sex) |
| Flight | (Flight\_id,F\_Departure,F\_Arrival,F\_Departure\_date,F\_Arrival\_date,Ticket\_id,A\_company\_id,A\_model\_no,A\_manufacturer,A\_name,Aeroplane\_id,Info\_F\_Departure\_date, Info\_F\_Departure\_date) |
| Crew | (Crew\_id,C\_name,C\_surname,C\_Sex,C\_Contact\_no,C\_Salary,C\_Designation,C\_Address,Flight\_id) |

**Instruction of 2nd normal form**

* The Prerequisite of 2nd Normal Form tables needs to be transferred from 1st normal form into 2nd normal form
* it should not have Partial Dependency.

**Partial dependency:**

Partial dependency is when fields that are indirectly reliant on the primary key or partly reliant on nonetheless also depend on additional keys that be contingent on the primary key such that if the field on which another field is contingent is removed, the field still exists because of the dependency on the primary key.

|  |  |
| --- | --- |
| **Tables** | **Element of table** |
| Passenger | (Passenger\_Id,P\_name, P\_surname,P\_age,P\_contact\_no,P\_sex) |
| Ticket | (Ticket\_id,T\_price,T\_price,Info\_F\_arrival,Infor\_F\_Departure,Info\_F\_Departure\_dateInfo\_F\_Departure\_date,Info\_F\_Departure\_date) |
| Flight | (Flight\_id,F\_Departure,F\_Arrival,F\_Departure\_date,F\_Arrival\_date, Ticket\_id,) |
| Aeroplane | (Aeroplane\_id,A\_company\_id,A\_model\_no,A\_manufacturer,A\_name) |
| Crew | (Crew\_id,C\_name,C\_surname,C\_Sex,C\_Contact\_no,C\_Salary, C\_Designation,C\_Address,Flight\_id) |

**Instruction of 3rd normal form**

Instructions of 3NF are:

* Should be in 1NF,2NF
* There shouldn’t be any kind of transitive dependency
* This happens when we can conjecture the value of any column from a non-key column

We Normalize our project till 3NF because it doesn’t have any kind of partial dependency. Here is a dummy data Normalization till 3NF.

|  |  |
| --- | --- |
| **Tables** | **Element of table** |
| Passenger | (Passenger\_Id (**PK**),P\_name, P\_surname,P\_age,P\_contact\_no,P\_sex) |
| Ticket | (Ticket\_id (**PK**)  ,T\_price,T\_price,Info\_F\_arrival,Infor\_F\_Departure,Info\_F\_Departure\_dateInfo\_F\_Departure\_date,Info\_F\_Departure\_date,Passenger\_id(**FK**)) |
| Flight | (Aeroplane\_id (**PK**),A\_company\_id,A\_model\_no,A\_manufacturer,A\_name) |
| Aeroplane | (Flight\_id (**PK**),F\_Departure,F\_Arrival,F\_Departure\_date,F\_Arrival\_date, Ticket\_id (**FK**),Aeroplane\_id (**FK**)) |
| Crew | (Crew\_id (**PK**),C\_name,C\_surname,C\_Sex,C\_Contact\_no,C\_Salary, C\_Designation,C\_Address,Flight\_id (**FK**)) |

**Data Definition Language**

DDL or Data Definition Language consists of SQL instructions that we use to describe the database schema. It resolves database schema descriptions and is used to make and adjust the structure of database objects in the database. It is a set of SQL instructions used to create, adjust, and remove database structures, but not the data. These instructions are not usually used by a typical user that needs to access the database through software.

**List of DDL key**

**CREATE:** we use this command to create tables and other objects of the database.

**DROP:** we use this command to remove tables and other objects from the database.

**ALTER:** we use this command to change the structure of the database.

**TRUNCATE:** we use this command to delete all records from a table.

**COMMENT**: we use this command to add comments to the data dictionary.

**RENAME:** we use this command to rename an object present in the database.

**Primary key:** A primary key is a constraint that uniquely pinpoints a record in a table.

**Foreign key:** A foreign key is a field in one table that references a primary key in another table**.**

**Checks:** The check constraint is used for limiting the variety of values ​​that can be positioned in a field.

**Data Manipulation Language**

SQL instructions that pact with manipulating information residing in databases belong to DML or Data Manipulation Language, to which most SQL statements belong. It is a module of SQL commands that control access to data and databases. Fundamentally, DCL commands are gathered with DML commands.

**List of DML keys**:

**INSERT:** we use this command for inserting data into a table.

**UPDAT**E: we use this command for updating present data in a table.

**DELETE:** we use this command to delete records from a table.

**DATATYPES:**

**VARCHAR (length)**: it is a variable that specifies the length of the string which may contain letters, numbers, and special characters. Its size varies from 0 to 65535 characters.

**INTEGER or INT**: it is a variable that is used for integer values. Its size varies from -2147483648 to 2147483647.

**FLOAT:** it is a variable that is used for specifying the float point number which means it can store decimal points.

**DATE:** it is a variable that is used for specifying the date values in a table using a format of the date (DD-MM-YYYY). we can use different formats that are provided to us.

**CHAR:** it is a variable that specifies a length of string that contains the letter, Unicode.

**REFERENCES:**

**Book:** Database System Concepts

**Author:** Avi Silberschatz, Henry F. Korth, S. Sudarshan

**Year:** 2011

**Publisher:** McGraw-Hill Education

Link: <https://www.db-book.com/db6/slide-dir/>