**Database Development**

Airline company

Student ID - 100613103

**SCOPE OF AN AIRLINE COMPANY:**

For this, they use airplanes that they own or rent. This Airline project can provide many facilities to the airline staff. This project allows users to add records when passenger request to reserve a ticket. This project uses a MySQL server as a back-end so it can manage data from a very large customer database and all airline-generated data like flight schedules, reservation ticket details, customer profiles, destination details, etc. This is a very well-structured project so it is very easy to restore old records with just a few clicks. It is universal software that can be applied by any organization for a smooth workflow

**IMPLEMENTATIONS:**

Before there were systems available that only staff members were allowed to use and book a flight but we implement this system by creating now a user-friendly system that allows users to register their account and book their flight, assign a seat number in an airplane, and also to be able to watch the schedule. This system is implemented in a way that the admin can add flights, and airplanes and manage the staff of any airline.

**ACCOMPLISHMENTS:**

The project has achieved its objectives by:

1. Help airlines system in making their business more efficient
2. Complete automation of airlines management
3. Provide accurate information about the addition and modification of records
4. Provide an efficient and reliable structure that can handle a large number of customer data

**KNOWLEDGE USED:**

1. We use MySQL software to create a table and E-R Diagram
2. normalization techniques to normalize the tables
3. For the E-R diagram, we used the **DB schema tool**

**Entity Relationship Diagram:**

“An entity-relationship diagram (ERD), also known as an entity-relationship model, ERD is logical representation to show relationship of object , project . So, Airline Company ERD contains the following tables”

* Staff
* Airplane
* Passenger
* Flight
* Reservation
* User

Too simple to create an ERD first we create tables then assign those tables the entities and then add a relationship between those attributes. We make relationships between tables using the primary key of a table in another table as foreign to connect these tables in attribute relationships.

**Staff table:**

It contains (the staff’s id **(primary key),** full name, surname, address, phone number, salary) and role. It also contains the **primary id** of the flight table (f\_id) as a **foreign id**. This table has a (**many-to-one**) relationship attribute with the **flight table**.

**Airplane table:**

The airplane table contains (the serial number (primary key), manufacturer company name, and model number)

**Passenger table:**

It contains (the passenger’s id **(primary key),** name, surname, address, and phone number). It also contains a **primary id** of the reservation table (r\_id) as a **foreign id**. This table has a (**one-to-one**) relationship attribute with the reservation table.

**Flight table:**

It contains the flight’s id (**primary key**), origin (original place of plane departure), destination, departure time, arrival time, and departure date. It also contains the primary id of the airplane table (ser\_num) as a foreign id. This table has a one-to-one relationship attribute with the airplane table, a one-to-many relationship with the staff table, and a one-to-many relationship with the reservation table.

**Reservation table:**

It contains the reservation’s id (primary key), ticket number, ticket type (business, first class, economy), and seat number (seat num of an airplane).it also contain the primary id of the flight table(f\_id) and user table as foreign id. This table has a many-to-one relationship attribute with the flight table, a many-to-one relationship attribute with the user table, and a one-to-one relationship attribute with the passenger table.

**User table:**

It contains the “user’s id (primary key), user\_name, and user\_password”. This table has a many-to-one relationship attribute with the reservation table.

**ERD from MySQL application**

Diagram

Description automatically generated

**ERD of Airline Company from DB Schema application**

Diagram

Description automatically generated

**UNNORMALIZED TABLES:**

1. **Staff:**

|  |  |
| --- | --- |
| Columns | explanation |
| s\_name | Staff full name |
| s\_surname | Staff surname |
| s\_address | Staff address |
| s\_contact\_info | Staff phone number |
| s\_salary | Staff salary per Annual |
| s\_role | Staff role example pilot or attendant |
| s\_age | Age of staff members |
| s\_gender | Gender of staff members |

1. **Airplane:**

|  |  |
| --- | --- |
| Columns | explanation |
| ser\_num | Serial number of airplane |
| manufac\_name | Manufacturer of airplane |
| model\_number | The model number of the airplane |

1. **Passenger**

|  |  |
| --- | --- |
| Columns | explanation |
| p\_name | Passenger name |
| p\_surname | Passenger surname |
| p\_address | Passenger address |
| p\_contact\_num | Passenger phone number |
| Ticket\_number | Reserved ticket number |
| ticket\_type | Ticket type (business, economy, first class) |
| P\_last\_name | Passenger last name |

1. **Flight**

|  |  |
| --- | --- |
| Columns | explanation |
| f\_origin | Flight departure location |
| f\_dest | Flight destination |
| f\_dep\_time | Flight departure time |
| f\_arriv\_time | Flight arrival time |
| f\_dep\_date | Flight departure date |
| ser\_num | Serial number of airplane |
| model\_number | The model number of the airplane |

1. **Reservation**

|  |  |
| --- | --- |
| Columns | explanation |
| ticket\_num | Reserved ticket number |
| ticket\_type | Ticket type (business, economy, first class) |
| seatnum | Seat number of plane |
| f\_id | The primary key of the flight table |
| f\_dep\_time | Flight departure time |
| F\_origin | Location of flight’s departure |
| p\_id | The primary key of the passenger table |

1. **User**

|  |  |
| --- | --- |
| Columns | explanation |
| User\_name | Username of customer |
| password | Password of customer |

**NORMALIZTION:**

Database normalization is the process of shaping data into tables so that the results of using the database are always clear and as expected. Such normalization is the core of relational database theory.

**Data Normalization Rules:**

**First normal form (1NF):**

This is the "basic" form of database normalization and usually matches the description of any database, specifically:

* It contains a two-dimensional table with rows and columns.
* Each column corresponds to a child object.
* There shouldn’t be any duplicate rows
* All entries in a column must be of the same type

Now we will apply these rules which are mentioned above in our table as follows:

|  |  |
| --- | --- |
| Tables name | Columns |
| Airplane | ser\_num, manufac\_name, model\_number |
| Flight | f\_origin, f\_dest, f\_dep\_time, f\_arriv\_time, f\_dep\_date, ser\_num, model\_number |
| Passenger | p\_name, p\_surname, p\_address, p\_contact\_num, Ticket\_number, ticket\_type |
| Reservation | ticket\_num, ticket\_type, seat\_num, f\_id, f\_dep\_timeF\_origin, p\_id |
| Staff | s\_name, s\_surname, s\_address, s\_contact\_info, s\_salary, s\_role, s\_age, s\_gender |
| User | User\_name, password |

**Second Normal Form (2NF):**

For a table to be in the Second Normal Form,

1. It should be in the First Normal form.
2. And, it should not contain any kind of Partial Dependency.

Now we will apply the above-mentioned rule of 2NF in our tables:

|  |  |
| --- | --- |
| Tables name | Columns |
| Airplane | ser\_num, manufac\_name, model\_number |
| Flight | F\_id, f\_origin, f\_dest, f\_dep\_time, f\_arriv\_time, f\_dep\_date, ser\_num, model\_number |
| Passenger | P\_id, p\_name, p\_surname, p\_address, p\_contact\_num, Ticket\_number, ticket\_type |
| Reservation | R\_id, ticket\_num, ticket\_type, seat\_num, f\_id, f\_dep\_timeF\_origin, p\_id |
| Staff | S\_id, s\_name, s\_surname, s\_address, s\_contact\_info, s\_salary, s\_role, s\_age, s\_gender |
| User | User\_id, User\_name, password |

**Third Normal Form (3NF)**

* “The table should be in the Second Normal Form”.
* “The second condition is that there should be no transitive dependency”
* A Transitive dependency in a database is a secondary relationship between values in the same table.

Now we will apply the rules that are mentioned in our tables:

|  |  |
| --- | --- |
| Tables name | Columns |
| Airplane | ser\_num (primary key), manufac\_name, model\_number |
| Flight | F\_id(primary key), f\_origin, f\_dest, f\_dep\_time, f\_arriv\_time, f\_dep\_date,ser\_num(foreign key) |
| Passenger | P\_id(primary key), p\_name, p\_surname, p\_address, p\_contact\_num,r\_id(foreign key), user\_id(foreign key) |
| Reservation | R\_id(primary key), ticket\_num, ticket\_type, seat\_num, f\_id(foreign key), |
| Staff | S\_id(primary key), s\_name, s\_surname, s\_address, s\_contact\_info, s\_salary, s\_role, s\_age, s\_gender,f\_id(foreign key) |
| User | User\_id(primary key), User\_name, password |

# **Data Definition Language (DDL)**

“A data definition language (DDL) is a computer language used to produce and change the structure of database objects in a database. These database objects include views, schemas, tables, indexes, etc.”

As following:

**Create**

This command is used to make new tables, views, and procedure

**Alter**

Alter give us privilege to modify or rearrange database table . we used alter commend to add new columns and drop columns .we also can change data type of columns from database table.

**Drop**

“A drop command is used to delete objects such as a table, index, or view. A DROP statement cannot be rolled back, so once an object is destroyed, there’s no way to recover it.”

**Truncate**

TRUNCATE statement is intended for quickly removing records from a table.

**Constraints:**

“Constraints are used to limit the data type that can fit into a table. This ensures the precision and consistency of the panel data”

**Not null:**

“This forces a column to always contain a value, which means you cannot insert a new record or update a record without adding a value to this column.”

**Primary key**

“Each entry in a table is given a special identification by the PRIMARY KEY constraint”.

**Foreign key**

As we know primary key is unique , we use we same primary key to another table then it will become foreign key

**Checks**

The value range that may be entered into a column is restricted by the CHECK constraint.

## Data Manipulation Language (DML)

A data manipulation language (DML) , we used DML for manipulate data in database , like if we want to add ,update , select ,delete data from database .it give us lots of authorizing ,like those

**Select**: this command is meant for regaining data from a table.

**Update**: this command is meant for modifying data of one or more databases.

**Insert**: By using insert command we add new columns more then one or just one.

**Delete**: this command is meant for removing one or more data from a table.

**DATA TYPES:**

**“VARCHAR** (size): It isa VARIABLE that contains letters, numbers, and special characters.

**“INT** (size): A medium integer. The range of int is from -2100 to 2100. The unsigned range is from 0 to 42900.

**DATE**: A date. Format: YYYY-MM-DD. That syntax show that the how data we have to enter year , month , day