TABLE OF CONTENTS

ANNEXURE 1: INTRODUCTION

I am working on Railway Management System.

Background

For developing the Schema for RDBMS, we first make the Entity-Relationship Model (ER Model) and then map it to the Relational Model.

Entity-Relationship Model

The Entity-Relationship Model (ER Model) is used to give the outline or the overall structure of the Database. It considers the following:

Entities and Entity Sets

The Attributes of those Entities

Relationships among those Entities

ER Diagram

The most important part of the ER Model is the Entity-Relationship Diagram (ER Diagram).

It is a graphical representation of the ER model.

In simple words, it is the blueprint of a database.

ANNEXURE 2: DESIGN OF THE PROJECT

To make an ER Diagram, we first identify the main Entities of the System.

The main Entities associated with the Railway Management System are:

1. Passenger
2. Train
3. Station
4. Class
5. Admin
6. Schedule
7. Route
8. Reservation
9. Payment

Attributes

The Properties of an entity are called Attributes. The attributes that can uniquely identify any Entities are the Primary Keys.

The Entities and Attributes in this project are as follows.

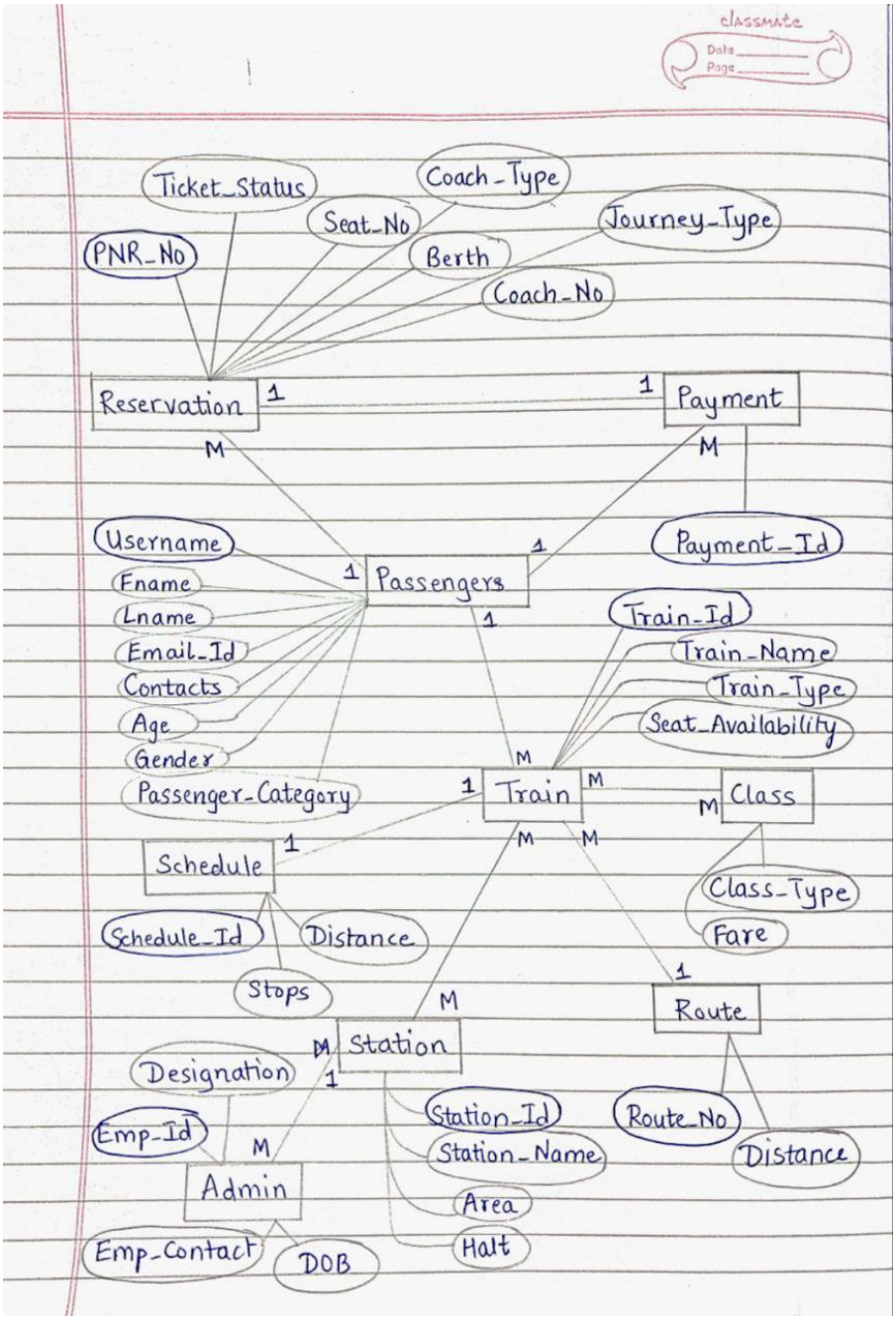
The Primary Keys are those that are underlined.

Entity	**Attributes**
Passenger	<u>Username</u> Fname Lname Email_Id

	Contacts Age Gender Passenger_Category
Train	Train_Id Train_Name Train_Type Seat_Availability
Station	Station_Id Station_Name Area Halt
Class (Weak entity – no pk)	Class_Type Fare
Admin	Emp_Id Designation Emp_Contact Emp_DOB
Schedule	Start_Time End_Time Source Stops Destination
Routes	Route_no Distance
Reservation	PNR_No Ticket_Status

	Seat_No Berth Coach_no Coach_Type Enquiry Journey_Type
Payment	Payment_Id Username (FK) Ticket_Id (FK)

This is the ER Diagram I made for the Railway Management System:


Ticket_Status
Coach_Type
Seat_No
Journey_Type
PNR_No
Berth
Coach_No
Reservation
1
1
Payment
M
M
Username
Payment_Id
1
Passengers
1
Fname
Lname
Email_Id
Contacts
Age
Gender
Passenger_Category
Train_Id
Train_Name
Train_Type
Seat_Availability
M
1
Train
M
M
Class
Schedule
1
M
M
Class_Type
Fare
Schedule_Id
Distance
Stops
1
Route
M
M
Station
Designation
1
Station_Id
Route_No
Emp_Id
M
Station_Name
Distance
Admin
Area
Emp_Contact
DOB
Halt

ANNEXURE 3: SCREENSHOTS

create table Passenger

```
create table Passenger
(
Username varchar2(20) primary key,
Fname varchar2(20),
Lname varchar2(20),
Email_Id varchar2(20),
DOB date,
Age number,
Gender char,
Passenger_Category varchar2(20)
)
```

create table Train

```
create table Train
(
Username varchar2(20),
constraint ck1 foreign key(Username) references Passenger(Username),
Train_Id number primary key,
Train_Name varchar2(20),
Train_Type varchar2(20),
Seat_Availability number
)
```

create table Contacts

```
create table Contacts
(
Username varchar2(20),
constraint c3 foreign key(Username) references Passenger(Username),
Contacts number
)
```

create table Station

```
create table Station
(
Station_Id number primary key,
Station_Name varchar2(20),
Area varchar2(20),
Halt varchar2(20)
)
```

create table Goes_To

```
create table Goes_To
(
Station_Id number,
constraint c4 foreign key(Station_Id) references Station(Station_Id),
Train_Id number,
constraint c5 foreign key(Train_Id) references Train(Train_Id),
Station_Name varchar2(20),
Area varchar2(20),
Halt varchar2(20)
)
```

create table Class

```
create table Class
(
Train_Id number,
constraint c6 foreign key(Train_Id) references Train(Train_Id),
Class_Type varchar2(20),
Fare number
)
```

create table Admin

```
create table Admin
(
Station_Id number,
constraint c7 foreign key(Station_Id) references Station(Station_Id),
Emp_Id number primary key,
Designation varchar2(20),
Emp_Contact number,
Emp_DOB date
)
```

create table Schedule

```
create table Schedule
(
Train_Id number,
constraint c1 primary key(Train_Id),
constraint c2 foreign key(Train_Id) references Train(Train_Id),
Route_no number,
Distance number,
Stops varchar2(20)
)
```

create table Reservation

```
create table Reservation
(
Username varchar2(20),
constraint c8 foreign key(Username) references Passenger(Username)
PNR_No number primary key not null,
Ticket_Status varchar2(20),
Seat_No varchar2(20),
Berth varchar2(20),
Coach_no number,
Coach_Type varchar2(20),
Journey_Type varchar2(20),
Enquiry varchar2(20)
)
```

create table Payment

```
create table Payment
(
Payment_Id number primary key,
Username varchar2(20),
constraint c9 foreign key(Username) references Passenger(Username),
PNR_No number,
constraint c99 foreign key(PNR_No) references Reservation(PNR_No)
)
```

ANNEXURE 4: CODE

Trains and **Passengers** relation is **M-1**.

So, we use Username from Passengers as a Foreign Key in the Trains table.

Passengers table:

```
create table Passenger
(
Username varchar2(20) primary key,
Fname varchar2(20),
Lname varchar2(20),
Email_Id varchar2(20),
DOB date,
Age number,
Gender char,
Passenger_Category varchar2(20)
)
```

Train table:

```
create table Train
(
Username varchar2(20),
constraint ck1 foreign key(Username) references Passenger(Username),
```

Train_Id number primary key,

Train_Name varchar2(20),

Train_Type varchar2(20),

Seat_Availability number

)

Contacts is a multi-valued attribute. Thus, normalizing the relations, we create a different table for Contacts.

Contacts table:

create table Contacts

(

Username varchar2(20),

constraint c3 foreign key(Username) references Passenger(Username),

Contacts number

)

Now, we create the Station table with Station_Id as the primary key.

Station table:

create table Station

(

Station_Id number primary key,

Station_Name varchar2(20),

Area varchar2(20),

Halt varchar2(20)

)

Train and **Station** relation is **M-M**.

So, we create a third table ‘Goes_To’ table, with the Username from Passenger and Station_Id from Station table as Foreign Keys in the Goes_To table.

Goes_To table:

create table Goes_To

(

Station_Id number,

constraint c4 foreign key(Station_Id) references Station(Station_Id),

Train_Id number,

constraint c5 foreign key(Train_Id) references Train(Train_Id),

Station_Name varchar2(20),

Area varchar2(20),

Halt varchar2(20)

)

Class is a weak entity. It is associated with the Train entity. Thus, Train_Id is taken as a foreign key in the Class table.

Class table:

create table Class

(

Train_Id number,

constraint c6 foreign key(Train_Id) references Train(Train_Id),

Class_Type varchar2(20),

Fare number

)

Station and **Admin** relation is **1-M**.

So, we use Station_Id from Station Table as a Foreign Key in the Admin table.

Admin table:

create table Admin

(

Station_Id number,

constraint c7 foreign key(Station_Id) references Station(Station_Id),

Emp_Id number primary key,

Designation varchar2(20),

Emp_Contact number,

Emp_DOB date

)

Employee_Contacts is a multi-valued attribute. Thus, normalizing the relations, we create a different table for Employee_Contacts too.

Employee_Contacts table:

```
create table Employee_Contacts
(
Emp_Id number,
constraint c98 foreign key(Emp_Id) references Admin(Emp_Id),
Employee_Contacts number
)
```

Train and **Schedule** relation is **1-1**.

So, we use Train_Id from rain Table as a Foreign Key in the Train_Schedules table.

Schedule table:

```
create table Schedule
(
Train_Id number,
constraint c1 primary key(Train_Id),
constraint c2 foreign key(Train_Id) references Train(Train_Id),
Route_no number,
Distance number,
Stops varchar2(20)
)
```

Reservation and **Passenger** relation is **M-1**.

So, we use Username from Passenger Table as a Foreign Key in the Reservation table.

Reservation table:

```
create table Reservation
(
Username varchar2(20),
constraint c8 foreign key(Username) references Passenger(Username),
PNR_No number primary key not null,
Ticket_Status varchar2(20),
Seat_No varchar2(20),
Berth varchar2(20),
Coach_no number,
Coach_Type varchar2(20),
Journey_Type varchar2(20),
Enquiry varchar2(20)
)
```

Payment and **Passenger** relation is **M-1**.

So, we use Username from Passenger Table as a Foreign Key in the Payment table.

Payment table:

```
create table Payment
(
```

```
Payment_Id number primary key,
Username varchar2(20),
constraint c9 foreign key(Username) references Passenger(Username),
PNR_No number,
constraint c99 foreign key(PNR_No) references Reservation(PNR_No)
)
```

ANNEXURE 5: QUERIES

Create table Passenger

```
create table Passenger
(
Username varchar2(20) primary key,
Fname varchar2(20),
Lname varchar2(20),
Email_Id varchar2(20),
DOB date,
Age number,
Gender char,
Passenger_Category varchar2(20)
)
```

create table Train

```
create table Train
(
Username varchar2(20),
constraint ck1 foreign key(Username) references Passenger(Username),
Train_Id number primary key,
Train_Name varchar2(20),
Train_Type varchar2(20),
Seat_Availability number
)
```

create table Contacts

```
create table Contacts
(
Username varchar2(20),
constraint c3 foreign key(Username) references Passenger(Username),
Contacts number
)
```

create table Station

```
create table Station
(
Station_Id number primary key,
Station_Name varchar2(20),
Area varchar2(20),
Halt varchar2(20)
)
```

create table Goes_To

```
create table Goes_To
(
Station_Id number,
constraint c4 foreign key(Station_Id) references Station(Station_Id),
Train_Id number,
constraint c5 foreign key(Train_Id) references Train(Train_Id),
Station_Name varchar2(20),
Area varchar2(20),
Halt varchar2(20)
)
```

create table Class

```
create table Class
(
Train_Id number,
constraint c6 foreign key(Train_Id) references Train(Train_Id),
Class_Type varchar2(20),
Fare number
)
```

create table Admin

```
create table Admin
(
Station_Id number,
constraint c7 foreign key(Station_Id) references Station(Station_Id),
Emp_Id number primary key,
Designation varchar2(20),
Emp_Contact number,
Emp_DOB date
)
```

create table Schedule

```
create table Schedule
(
Train_Id number,
constraint c1 primary key(Train_Id),
constraint c2 foreign key(Train_Id) references Train(Train_Id),
Route_no number,
Distance number,
Stops varchar2(20)
)
```

create table Reservation

```
create table Reservation
(
Username varchar2(20),
constraint c8 foreign key(Username) references Passenger(Username)
PNR_No number primary key not null,
Ticket_Status varchar2(20),
Seat_No varchar2(20),
Berth varchar2(20),
Coach_no number,
Coach_Type varchar2(20),
Journey_Type varchar2(20),
Enquiry varchar2(20)
)
```

create table Payment

```
create table Payment
(
Payment_Id number primary key,
Username varchar2(20),
constraint c9 foreign key(Username) references Passenger(Username),
PNR_No number,
constraint c99 foreign key(PNR_No) references Reservation(PNR_No)
)
```

Insert into Passenger (Username,Age,Gender) values ('Aditi01',19,'F')

desc Passenger

select * from Passenger

```
Insert into Passenger (Username,Age,Gender) values ('Aditi01',19,'F')

desc Passenger

select * from Passenger
```

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **PASSENGER**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PASSENGER	USERNAME	Varchar2	20	-	-	1	-	-	-
	FNAME	Varchar2	20	-	-	-	✓	-	-
	LNAME	Varchar2	20	-	-	-	✓	-	-
	EMAIL_ID	Varchar2	20	-	-	-	✓	-	-
	DOB	Date	7	-	-	-	✓	-	-
	AGE	Number	-	-	-	-	✓	-	-
	GENDER	Char	1	-	-	-	✓	-	-
	PASSENGER_CATEGORY	Varchar2	20	-	-	-	✓	-	-

1 - 8

USERNAME	FNAME	LNAME	EMAIL_ID	DOB	AGE	GENDER	PASSENGER_CATEGORY
Aditi01	-	-	-	-	19	F	-

1 rows returned in 0.00 seconds CSV Export

update Passenger set Age=20 where Username='Aditi01'

Autocommit Display 10

```
update Passenger set Age=20 where Username='Aditi01'

select * from Passenger
```

Results Explain Describe Saved SQL History

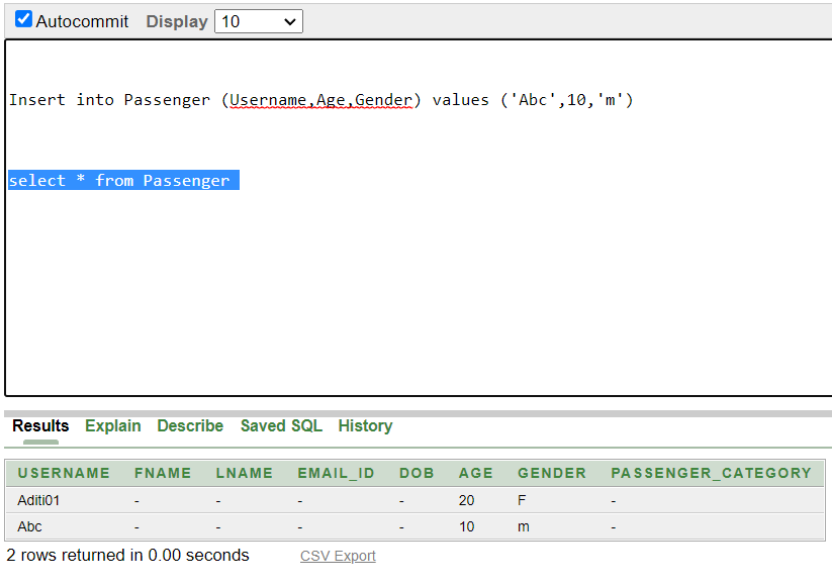
USERNAME	FNAME	LNAME	EMAIL_ID	DOB	AGE	GENDER	PASSENGER_CATEGORY
Aditi01	-	-	-	-	20	F	-

1 rows returned in 0.00 seconds CSV Export

Insert into Passenger (Username,Age,Gender) values ('Abc',10,'m')

Insert into Passenger (Username,Age,Gender) values ('Srushtee',’F’)

Insert into Passenger (Username,Age,Gender) values ('Riya',21,'F')



Summary

We developed a Database Management System for Railway Management System. After identifying the main Entities and the Relationship among them, we created an ER Model with the help of an ER Diagram. Then we mapped the ER Model to the Relational Model.

Then we implemented the DBMS using Oracle. We created 10 tables.

Now the government can use the DBMS for Storing, Managing, and Updating their data on a regular basis. This will help them get rid of the Data Duplication, Inconsistency and many other issues they were facing due to the use of traditional Methods.