# Semester Project Documentation

# Railways Management System

****

Fa20-BCS-009

Maryam Amjad

Fa20-BCS-053

Sofia Ahmed

Submitted To:

Mr.Waseem Khan

17th JUNE, 2022

Contents

[Semester Project Documentation 1](#_Toc106311034)

[Railways Management System 1](#_Toc106311035)

[1. Abstract 2](#_Toc106311036)

[2. Introduction 2](#_Toc106311037)

[3. Objectives: 2](#_Toc106311038)

[4. Project Benefits: 3](#_Toc106311039)

[5. Scope: 3](#_Toc106311040)

[6. Outcomes: 4](#_Toc106311041)

[7. Database Diagram 5](#_Toc106311042)

[8. ER Diagram 6](#_Toc106311043)

[9. Quries 7](#_Toc106311044)

[1.1 Create table 7](#_Toc106311045)

[1.1.1. Train Table 7](#_Toc106311046)

[1.1.2. Tickett 7](#_Toc106311047)

[1.1.3. Passanger 7](#_Toc106311048)

[1.1.4. Staff Table 8](#_Toc106311049)

[1.1.5. Sign IN Table 8](#_Toc106311050)

[1.1.6. Booking Table 9](#_Toc106311051)

[1.1.7. Complain Table 9](#_Toc106311052)

[1.1.8. Audit Table 9](#_Toc106311053)

[1.1.9. AdminLog Table 9](#_Toc106311054)

[1.2 Insert 10](#_Toc106311055)

[1.3 Select 13](#_Toc106311056)

[1.4 Drop 13](#_Toc106311057)

[1.5 Triggers 14](#_Toc106311058)

[1.6 **Procedure** 20](#_Toc106311059)

[1.7 update 20](#_Toc106311060)

[1.8 alter 21](#_Toc106311061)

[1.9 Delete 21](#_Toc106311062)

[1.10 Join 22](#_Toc106311063)

# Abstract

The Railway Reservation System facilitates the passengers to enquire about the trains , Booking of tickets, enquire about the status of the booked ticket, etc. The aim of case study is to design and develop a database maintaining the records of different trains, train status, and passengers and railway staff.

This project contains Introduction to the Railways reservation system It is the computerized system of reserving the seats of train seats in advanced. Online reservation has made the process for the reservation of seats very much easier than ever before.

In our country , there are number of counters for the reservation of the seats and one can easily make reservations and get tickets. Then this project contains entity relationship model diagram based on railway reservation system and introduction to relation model .

# Introduction

Database is an organized collection of data. A DBMS makes it possible for end users to create, read, update and delete data in a database. It provide concurrency, security, data integrity and uniform administration procedures.

The main purpose of maintaining database for Railway Reservation System is to reduce the manual errors involved in the booking of tickets and make it convenient for the customers and providers to maintain the data about their passenger and also about the seats available at them Due to automation many loopholes that exist in the manual maintenance of the records can be removed. The speed of obtaining and processing the data will be fast. For future expansion the proposed system can be web enabled so that clients can make various enquiries about trains between stations. Due to this, sometimes a lot of problems occur and they are facing many disputes with customers. To solve the above problem, we design a data base which includes passenger details, ticket detail, Train record, Staff Record ,availability of seats in trains, no of trains and their details.

# Objectives:

The main objective of Railway Management System is to provide many facilities to the user or passengers like online booking of tickets , they can check availability of train and seats and confirmed their seats for various places if seats are available .This system will provide best facilities to the passenger. This system will keep the record of passengers through admin. Admin can insert, delete and update records. The Railway Management System is an implementation of managing Railway system which helps the passengers to check the availability of train for different places.

# Project Benefits:

Multiple advantages of the Railway Management System are:

* One of the most important benefits of the system is that not everyone can access to the data.
* Only Admin can access to the records of train, staff, and passenger and complain. Similarly if passenger wants book ticket He/She should have an sign in to the system and if he has no account he/she can sign up without sign in or sign up then he can only see the train timings.
* Another benefit of the system is that the user can complain about the trains and their complain record is saved in database.
* Database provide complete security to the user’s data .No one can access the data except admin.
* User or passenger is not known about that how the data is stored in the database he can only view the data in the form of tables.
* All information stored in the database can easily be accessed, updated and organized as well through admin.

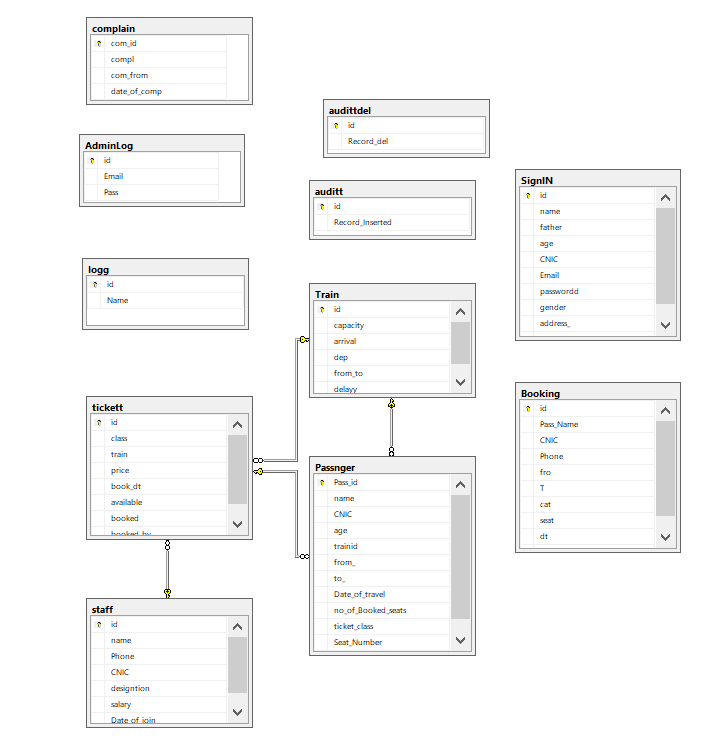
# Scope:

The main purpose of the system is to help people to manage their booking of trains online and save their time. And also all data is stored in database and admin can easily change and delete any data. The whole system can be managed by simple queries and is user friendly. Different modules have been unified in the system that helps the administration and passengers to use it in a very simple way. One of the main purposes of this system is to provide details about the operation performed by it. Customer can book tickets online from the comfort of their home or Admin can Book ticket for them .

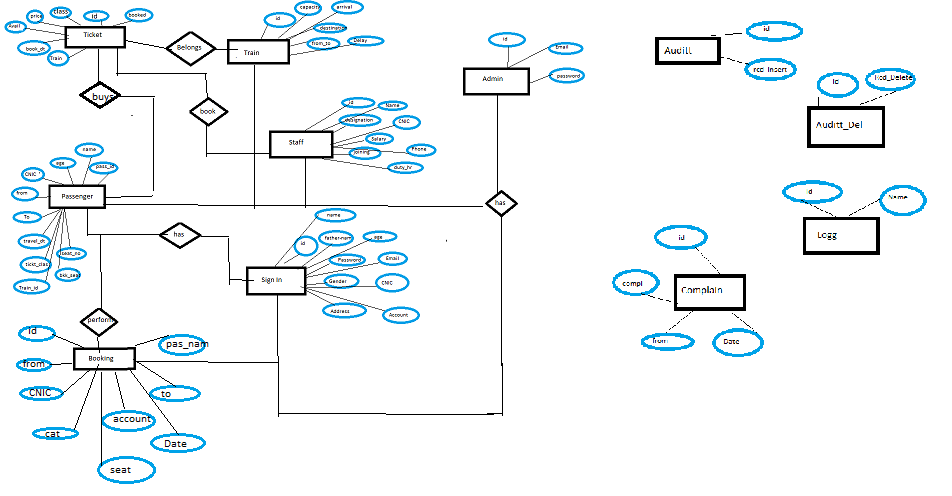
# Outcomes:

* Passengers and administration can interact with each other through internet so internet connection is necessary.
* Passengers can search for different trains and can also book tickets.
* Passenger can view timings of different trains
* Passenger Can Sign Up
* Passenger can Sign In.
* Passenger can file complains
* Admin has authorized Access
* Only Admin can View Record of Passenger ,Trains ,Staff ,Tickets ,Booking ,Complain and Some Database Info.
* Admin Can Insert ,Update ,Delete ,Show Passenger Record
* Admin Can Insert , Update ,Delete ,Show Ticket Record
* Admin Can Insert Update ,Delete ,Show Booking Record
* Admin Can Insert Update ,Delete ,Show Train Record
* Admin Can Insert Update ,Delete ,Show Staff Record
* Admin Can View Complain Record
* Admin can view Database info
* This system maintains user’s data and provides security.
* This system will provide details about train and train stops.

# Database Diagram



# ER Diagram



# Quries

## Create table

### Train Table

create table Train(

id intprimarykeyidentity(1,1),

capacity int,

arrival varchar(30),

dep varchar(30),

from\_tovarchar(200),

delayyvarchar(30))

### Tickett

createtabletickett(

id intprimarykeyidentity,

class varchar(30),

train intforeignkeyreferences Train(id),

price int,

book\_dtvarchar(30),

available int,

booked int)

altertabletickett

altercolumnbook\_dtdate

### Passanger

createtablePassnger(

Pass\_idintprimarykeyidentity,

name varchar(30),

CNIC bigint,

age int,

trainidintforeignkeyreferences Train(id),

from\_ varchar(30),

to\_ varchar(30),

Date\_of\_traveldate,

no\_of\_Booked\_seatsint,

ticket\_classvarchar(30),

Seat\_Numberint,

ticket intforeignkeyreferencestickett(id))

### Staff Table

createtablestaff(

id intprimarykeyidentity,

name varchar(30),

Phone int,

CNIC int,

designtionvarchar(30),

salary int,

Date\_of\_joinvarchar(30))

### Sign IN Table

createtableSignIN(

id intprimarykeyidentity,

name varchar(30),

father varchar(30),

age int,

CNIC int,

Email varchar(50),

passworddvarchar(30),

gender varchar(20),

address\_ varchar(30),

account int)

### Booking Table

createtableBooking(

id intprimarykeyidentity,

Pass\_Namevarchar(30),

CNIC int,

Phone int,

frovarchar(30),

T varchar(30),

cat varchar(30),

seat int,

dt varchar(30),

account int)

### Complain Table

createtable complain(

com\_idintprimarykeyidentity,

complvarchar(500),

com\_fromvarchar(30),

date\_of\_compDateTimenotnulldefault (getdate()))

insertinto complain values('Water shortage','Maryam',0)

### Audit Table

createtableauditt(

id intprimaryKeyidentity,

Record\_Insertedvarchar(500))

### AdminLog Table

createtableAdminLog(

id intprimarykeyidentity,

Email varchar(30),

Pass varchar(10))

## Insert

#### Insert into Train

insertinto Train Values(200,'4;20 pm','5:00','Lahore-Karachi','20 min')

insertinto Train Values(700,'2:00 pm','2:30','Karachi-Islamabad','5 min')

insertinto Train Values(400,'4;00 pm','4:20','Lahore-Peshawar','40 min')

insertinto Train Values(300,'3;40 pm','4:00','Peshawar-Karachi','30 min')

insertinto Train Values(200,'9;20 pm','9:54','Lahore-Karachi','00 min')

insertinto Train Values(200,'4;20 pm','5:00','Lahore-Karachi','20 min')

insertinto Train Values(700,'2:00 pm','2:30','Karachi-Islamabad','5 min')

insertinto Train Values(400,'4;00 pm','4:20','Lahore-Peshawar','40 min')

insertinto Train Values(300,'3;40 pm','4:00','Peshawar-Karachi','30 min')

insertinto Train Values(200,'9;20 pm','9:54','Lahore-Karachi','00 min')

#### Insert into tickett

insertintotickettValues('Economy',8,'200','2022-06-23',600,200,4)

insertintotickettValues('Bussiness',8,'900','2022-06-23',200,100,3)

insertintotickettValues('FirstClass',8,'200','2022-06-23',600,200,3)

insertintotickettValues('Economy',9,'200','2022-04-03',600,200,3)

insertintotickettValues('Bussiness',9,'900','2022-03-23',200,100,6)

insertintotickettValues('FirstClass',9,'200','2022-06-23',600,200,3)

insertintotickettValues('Economy',10,'200','2022-06-23',600,200,4)

insertintotickettValues('Bussiness',10,'900','2022-06-23',200,100,4)

insertintotickettValues('FirstClass',10,'200','2022-06-23',600,200,3)

#### Insert into Passnger

insertintoPassngerValues('Maryam',030000,20,2,'Attock','Quetta','2022-06-09',5,'Economy',202,2)

insertintoPassngerValues('Maha',030000,20,2,'Attock','Quetta','2022-06-09',5,'Economy',202,1)

insertintoPassngerValues('Sarah',030000,20,2,'Attock','Quetta','2022-06-09',5,'Economy',202,3)

insertintoPassngerValues('Ali',030000,20,2,'Attock','Quetta','2022-06-09',5,'Economy',202,5)

insertintoPassngerValues('Hassan',030000,20,2,'Attock','Quetta','2022-06-09',5,'Economy',202,2)

insertintoPassngerValues('Rahul',030000,20,2,'Attock','Quetta','2022-06-09',5,'Economy',202,1)

insertintoPassngerValues('Rehman',030000,20,2,'Attock','Quetta','2022-06-09',5,'Economy',202,4)

insertintoPassngerValues('Aish',030000000,20,2,'Attock','Quetta','2022-06-09',5,'Economy',202,5)

#### insert into Staff

insertinto staff Values('Rehman',09090,9000,'CEO',90000,'2002-03-03')

insertinto staff Values('Maryam',09090,9000,'Manager',50000,'2002-03-03')

insertinto staff Values('Rahul',09090,9000,'Employee',9000,'2002-03-03')

insertinto staff Values('Sofia',09090,9000,'Employee',9000,'2002-03-03')

insertinto staff Values('Sarah',09090,9000,'intern',5000,'2002-03-03')

insertinto staff Values('Ali',09090,9000,'intern',5000,'2002-03-03')

insertinto staff Values('Ahmed',09090,9000,'Employee',9000,'2002-03-03')

insertinto staff Values('Hassan',09090,9000,'intern',5000,'2002-03-03')

insertinto staff Values('Haider',09090,9000,'Employee',9000,'2002-03-03')

#### insert into SignIN

insertintoSignINValues('Rehman','Ahmed',30,9000,'7787@gmail.com','56565','M','Attock',45454)

insertintoSignINValues('Maryam','Amjad',30,9030,'7787@gmail.com','56565','M','Attock',45454)

insertintoSignINValues('Rahul''Mahnutra',30,96700,'7787@gmail.com','56565','M','Attock',45454)

insertintoSignINValues('Sofia','hmed',30,90880,'7787@gmail.com','56565','M','Attock',45454)

insertintoSignINValues('Sarah','Ali',30,909090,'7787@gmail.com','56565','M','Attock',45454)

insertintoSignINValues('Ali','Khuram',30,945400,'7787@gmail.com','56565','M','Attock',45454)

insertintoSignINValues('Ahmed''Khan',,30,906565,'7787@gmail.com','56565','M','Attock',45454)

insertintoSignINValues('Hassan','Amjad',30,905450,'7787@gmail.com','56565','M','Attock',45454)

insertintoSignINValues('Haider','Amjad',30,90054,'7787@gmail.com','56565','M','Attock',45454)

select\*fromPassnger

#### insert into complain

insertinto complain Values('Bad behaviour of staff and poor managment','Maryam','2022-09-22')

insertinto complain Values(' poor managment','Maryam','2022-09-22')

insertinto complain Values('Water Shortage','Maryam','2022-09-22')

insertinto complain Values('Bad behaviour of staff and poor managment','Maryam','2022-09-22')

insertinto complain Values('Bad behaviour of staff and poor managment','Maryam','2022-09-22')

insertinto complain Values('Water Shortage','Maryam','2022-09-22')

insertinto complain Values('Bad behaviour of staff and poor managment','Maryam','2022-09-22')

insertinto complain Values(' poor managment','Maryam','2022-09-22')

insertinto complain Values('Bad behaviour of managment','Maryam','2022-09-22')

#### insert into Booking

insertinto Booking Values('Maryam',030000,200900,'Attock','Quetta','Economy',202,'2022-09-22',2)

insertinto Booking Values('Maha',030000,20090,'Attock','Quetta''Economy',202,'2022-09-22',1)

insertinto Booking Values('Sarah',030000,209090,'Attock','Quetta','Economy',202,'2022-09-22',3)

insertinto Booking Values('Ali',030000,278780,'Attock','Quetta''Economy',202,'2022-09-22',5)

insertinto Booking Values('Hassan',030000,267670,'Attock','Quetta','Economy',202,'2022-09-22',2)

insertinto Booking Values('Rahul',030000,207767,'Attock','Quetta','Economy',202,'2022-09-22',1)

insertinto Booking Values('Rehman',030000,208782,'Attock','Quetta','Economy',202,'2022-09-22',4)

insertinto Booking Values('Aish',030000000,207672,'Attock','Quetta','Economy',202,'2022-09-22',5)

## Select

select\*fromAdminLog

select\*fromauditt

select\*fromauditt

select\*from Booking

select\*from complain

select\*fromlogg

select\*fromPassnger

select\*from staff

select\*from Train

select\*fromticket

## Drop

droptableAdminLog

droptableauditt

droptableaudittdel

droptable Booking

droptable complain

droptablelogg

droptablePassnger

droptable staff

droptable Train

droptabletickett

## Triggers

##### Insert Trigger

createtriggertr\_insert\_pss

onPassnger

forinsert

asbegin

declare @id int

select @id =Pass\_idfrom inserted

insertintoaudittvalues('In Passnger New id is added '+cast(@id asnvarchar(20))+' at '+cast(getdate()asnvarchar(200)))

end

droptriggertr\_insert\_pss

createtriggertr\_insert\_Booking

on Booking

forinsert

asbegin

declare @id int

select @id =id from inserted

insertintoaudittvalues('In Booking New id is added '+cast(@id asnvarchar(20))+' at '+cast(getdate()asnvarchar(200)))

end

droptriggertr\_insert\_Booking

createtriggertr\_insert\_train-

on Train

forinsert

asbegin

declare @id int

select @id =id from inserted

insertintoaudittvalues('In Train New id is added '+cast(@id asnvarchar(20))+' at '+cast(getdate()asnvarchar(200)))

end

droptriggertr\_insert\_train

createtriggertr\_insert\_tickett

ontickett

forinsert

asbegin

declare @id int

select @id =id from inserted

insertintoaudittvalues('IN tickett id is added '+cast(@id asnvarchar(20))+' at '+cast(getdate()asnvarchar(200)))

end

droptriggertr\_insert\_tickett

createtriggertr\_insert\_staff

on staff

forinsert

asbegin

declare @id int

select @id =id from inserted

insertintoaudittvalues('In staff New id is added '+cast(@id asnvarchar(20))+' at '+cast(getdate()asnvarchar(200)))

end

droptriggertr\_insert\_staff

##### Delete Trigger

createtriggertr\_del\_pss

onPassnger

fordelete

asbegin

declare @id int

select @id =Pass\_idfrom deleted

insertintoaudittdelvalues(' From Passnger id '+cast(@id asnvarchar(20))+' is Deleted at '+cast(getdate()asnvarchar(200)))

end

createtriggertr\_del\_Booking

on Booking

fordelete

asbegin

declare @id int

select @id =id from deleted

insertintoaudittdelvalues('From Booking id '+cast(@id asnvarchar(20))+' is Deleted at '+cast(getdate()asnvarchar(200)))

end

droptriggertr\_del\_Booking

createtriggertr\_del\_staff

on staff

fordelete

asbegin

declare @id int

select @id =id from deleted

insertintoaudittdelvalues('From staff id '+cast(@id asnvarchar(20))+' is Deleted at '+cast(getdate()asnvarchar(200)))

end

droptriggertr\_del\_staff

createtriggertr\_del\_train

on Train

fordelete

asbegin

declare @id int

select @id =id from deleted

insertintoaudittdelvalues(' From Train id '+cast(@id asnvarchar(20))+' is Deleted at '+cast(getdate()asnvarchar(200)))

end

createtriggertr\_del\_tickett

ontickett

fordelete

asbegin

declare @id int

select @id =id from deleted

insertintoaudittdelvalues('From tickett id '+cast(@id asnvarchar(20))+' is Deleted at '+cast(getdate()asnvarchar(200)))

end

##### Update Trigger

createtriggerTrainUp

on Train

forupdate

asbegin

declare @id int

declare @oldCap int

declare @oldarri varchar(30), @olddep varchar(30),@ofrotovarchar(200),@odelvarchar(30)

declare @nCap int,@narrivarchar(30), @ndep varchar(30),@nfrotovarchar(200),@ndelvarchar(30)

declare @audit varchar(300)

begin

set @audit=''

select @id=id ,@nCap=Capacity,@narri=arrival,@ndep=dep, @nfroto=from\_to ,@ndel=delayyfrom inserted

select @oldCap=Capacity,@oldarri=arrival,@olddep=dep, @ofroto=from\_to ,@odel=delayyfrom deleted where id=@id

set @audit=cast(@id asnvarchar(20))+'Changed'

if(@oldCap<>@nCap)

set @audit=@audit+' capacity changed form'+cast(@oldCap asnvarchar(20))+'to'+cast(@nCap asnvarchar(20))

if(@oldarri<>@narri)

set @audit=@audit+' changed form'+@oldarri+'to'+@narri

if(@ofroto<>@nfroto)

set @audit=@audit+' changed form'+@ofroto+'to'+@nfroto

if(@odel<>@ndel)

set @audit=@audit+' changed form'+@odel+'to'+@ndel

insertintologgvalues(@audit)

end

end

droptriggerTrainUp

createtriggerupdtTicket

ontickett

forupdate

asbegin

declare @id int

declare @oc varchar(30),@op int,@obddate,@oaint,@obint

declare @nc varchar(30),@np int,@nbddate,@naint,@nbint

declare @audit varchar(300)

begin

set @audit=''

select @id=id, @nc=class,@np=price,@nbd=book\_dt,@na=available,@nb=booked from inserted

select @oc=class,@op=price,@obd=book\_dt,@oa=available,@ob=booked from deleted where id=@id

set @audit=cast(@id asnvarchar(20))

if(@oc<>@nc)

set @audit=@audit+'changed from'+cast(@oc asnvarchar(20))+'to'+cast(@nc asnvarchar(20))

if(@op<>@np)

set @audit=@audit+' changed from'+cast(@op asnvarchar(20))+'to'+cast(@op asnvarchar(20))

if(@obd<>@nbd)

set @audit=@audit+'Class changed from'+cast(@obd asnvarchar(20))+'to'+cast(@nbd asnvarchar(20))

if(@ob<>@nb)

set @audit=@audit+'Class changed from'+@ob+'to'+@nb

insertintologgvalues(@audit)

end

end

droptriggerupdtTicket

select\*fromtickett

updatetickettset price=2000 where id =3

select\*fromlogg

createtriggerStaffUp

on staff

forupdate

asbegin

declare @id int

declare @on varchar(30),@op int,@ocint,@od varchar(30),@osint,@odjvarchar(30)

declare @nn varchar(30),@np int,@ncint,@ndvarchar(30),@ns int,@ndjvarchar(30)

declare @audit varchar(300)

begin

select @id=id, @on=name,@op=Phone,@oc=CNIC ,@od=designtion,@os=salary ,@odj=Date\_of\_joinfrom inserted

select @nn=name,@np=Phone,@nc=CNIC ,@nd=designtion,@ns=salary ,@ndj=Date\_of\_joinfrom deleted where id=@id

set @audit=CAST(@id asvarchar(20))+' Changed '

if(@on<>@nn)

set @audit=@audit+@on +'to'+ @nn

if(@op<>@np)

set @audit=@audit+CAST(@op asvarchar(20))+'from'+CAST(@np asvarchar(20))

if(@oc<>@nc)

set @audit=@audit+CAST(@oc asvarchar(20))+'from'+CAST(@nc asvarchar(20))

if(@od<>@nd)

set @audit=@audit+@on +'from'+ @nn

if(@os<>@ns)

set @audit=@audit+CAST(@os asvarchar(20))+'from'+CAST(@ns asvarchar(20))

if(@odj<>@ndj)

set @audit=@audit+@odj +'from'+ @ndj

insertintologgvalues(@audit)

end

end

## Procedure

createprocedure age

as

Begin

select\*fromPassngerwhere age<15

End

Executeagee

createprocedureagee

as

Begin

select\*fromPassngerwhere age>15

End

sp\_columns Train

sp\_columns staff

sp\_tables

createprocedure Bus

as

Begin

select\*fromtickettwhere class='Bussiness'

End

Execute Bus

## update

update Train set capacity=100 where id =2

select\*fromlogg

updatetickettset price=2000 where id =3

select\*fromlogg

update staff set salary=5 where id=2

update complain setcom\_from='Aisha'where id=3

## alter

altertablePassnger

altercolumn CNIC bigint

altertable Booking

altercolumn CNIC bigint

altertable Booking

altercolumn Phone bigint

altertablePassnger

addticket\_classvarchar(30)

altertabletickett

altercolumnbook\_dtdate

altertable Booking

altercolumn dt date

## Delete

deleteauditt

delete Train

deletePassnger

delete Train

delete ticket

delete complain

delete Booking

delete staff

deletefrom staff where id=1

deletefrom complain where id=1

deletefromPassngerwherePass\_id=1

deletefrom Booking where id=1

deletefrom staff where id=1

## Join

SELECTname,CNIC,age,from\_,to\_,Date\_of\_travel,ticket\_class,Seat\_Number,class,price,book\_dtfromPassngerJointicketton tickett.id=Passnger.ticket

selectname,from\_,to\_,class,pricefromPassngerjointickettonPassnger.Pass\_id= tickett.id