

# Database Management System Project Topic: Railway Reservation System Subject code: CS593

**Group no. - 05**

**Names: Prithviraj Bhowmick(09) Saptarshi Bose(10)**

**Section:CSE3D**

**Subject: Database Management Systems Laboratory**

**Enrolment no. :**

**Batch: 2017-2021**

**University of Engineering and Management(Kolkata)**

### INDEX:

#### Acknowledgement………………………………………… 3 Introduction to topic ……………………………………... 4 ER Diagram ………………………………………………. 5 Sample SQL queries ……………………………………… 6 Useful queries related to the database ………………….. 14 Relational model …………………………………………. 23 Future scope of project…………………………………… 24 Conclusion ………………………………………………… 25

**Remarks ………………………………………………….. 26**

**Acknowledgement**

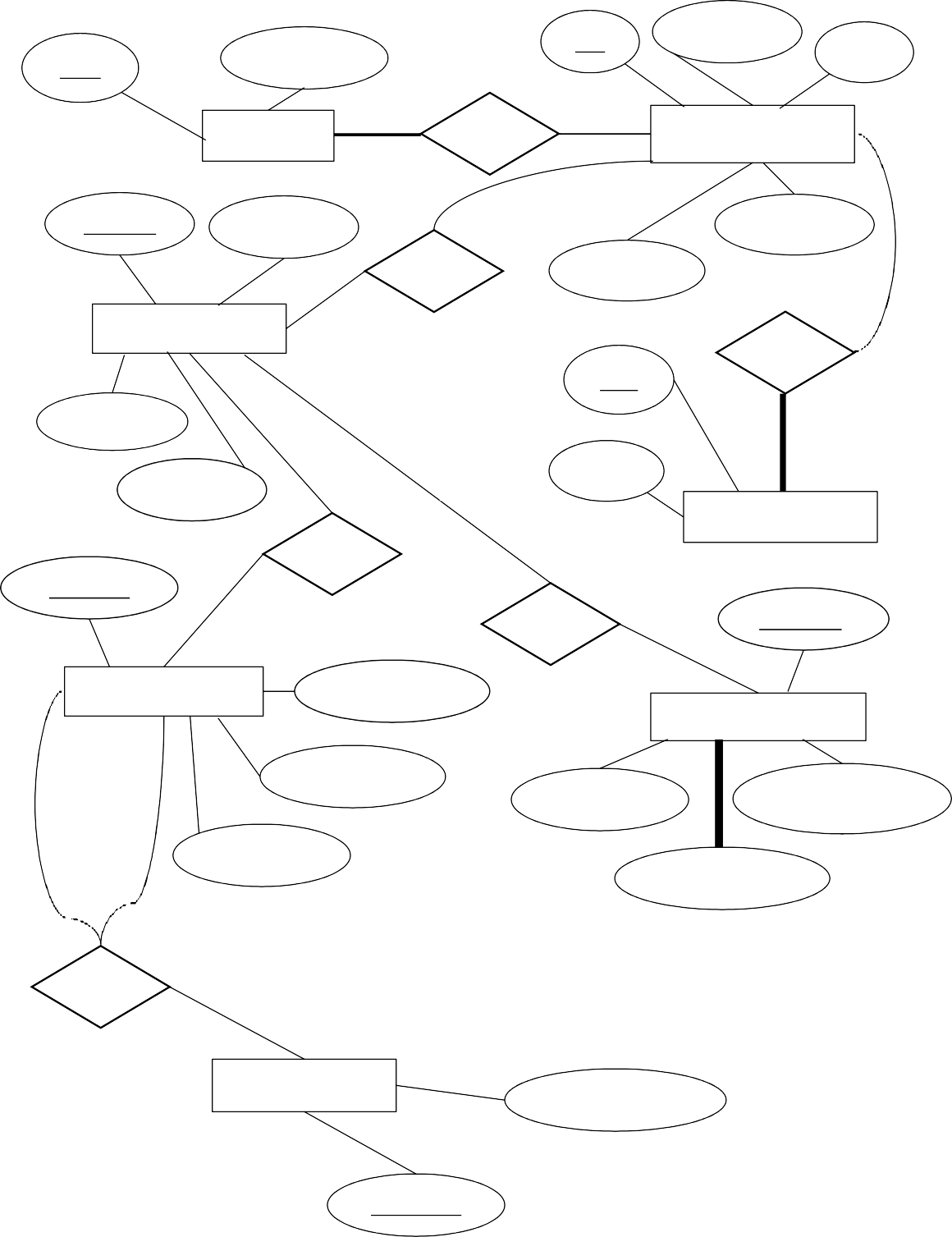
We, Prithviraj Bhowmick and Saptarshi Bose from CSE3D, would firstly like to thank our University for providing us with the infrastructure we used to learn about the subject. Secondly, we would thank Sukalyan Goswami sir, HOD of Computer Science Engineering department, UEM Kolkata. Thirdly we would love to thank our concerned subject teachers Varsha Poddar and Subarna Sen for providing us with the project idea and necessary study material for the project.

Introduction:

Railway Reservation System:

A railway reservation system is required by a booking platform where railway tickets are booked. It must have all information regarding the trains and the passengers. As for the trains, it must include data regarding the status of the train, the route of the train and also the information about the stations. As for the passengers it should have the details of the passengers and details of the ticket. Passengers should be booked under certain users who have access to the database.

This is one of the most widely used system that requires a powerful database acting in the background. Under a reservation system there are thousands of trains which are boarded by millions of passengers. If not stored under a database through a properly programmed GUI, keeping track of trains and passengers is quite impossible. Thus, a database is strongly required in order to add, manipulate and delete passenger and train details effectively and efficiently.



**ER- Diagram:**

Pnr

P\_name

U\_id

Password

Age

User

1

N

Has

Passenger

N

Train\_id

T\_name

Seat no.

Has

Gender

1

Train

1

Has

T\_id

Dept\_d

1

Arr\_d

1

1

Price

Ticket

Has

Route\_id

1

Has

Status\_id

1

Route

No\_of\_stop

Status

Start

Dept\_time

Wait\_seat

Booked\_seat

Arr\_time

Avail\_seat

End

Has

Station

Station\_name

Station\_id

**Sample SQL queries:**

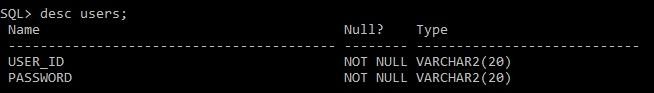
Creations:

#### Users

create table users( user\_id varchar2(20) primary key, password varchar2(20) not null

);

alter table users add constraint chk1 check (lengthb(password)>=8);



### Passenger

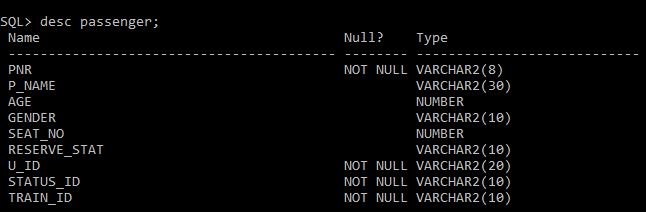
create table passenger( pnr varchar(8) primary key, p\_name varchar2(30),

age number, gender varchar(10), seat\_no number,

reserve\_stat varchar2(10),

u\_id varchar2(20) not null references users(user\_id), status\_id varchar2(10) not null,

train\_id varchar2(10) not null);

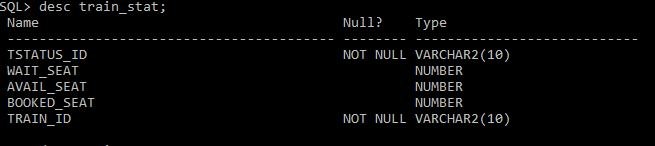


### Train Status

create table train\_stat( tstatus\_id varchar2(10) primary key, wait\_seat number,

avail\_seat number, booked\_seat number,

train\_id varchar2(10) not null);



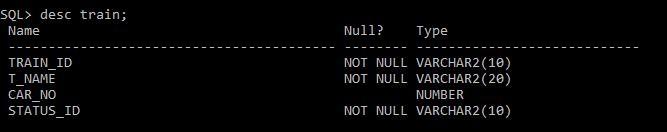
### Train

create table train( train\_id varchar2(10) primary key, t\_name varchar2(20) not null,

car\_no number,

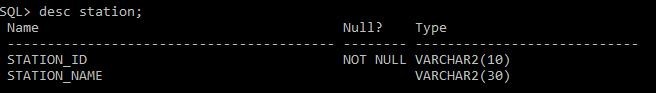
status\_id varchar2(10) not null references train\_stat(tstatus\_id));

alter table train\_stat add constraint train\_fk foreign key(train\_id) references train(train\_id);



### Station

create table station( station\_id varchar2(10) primary key, station\_name varchar2(30));



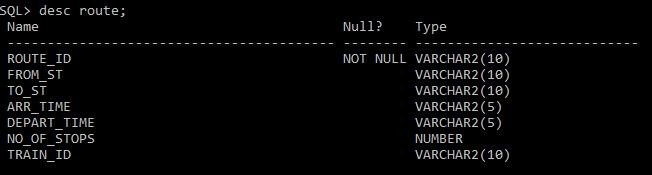
### Route

create table route( route\_id varchar2(10) primary key,

from\_st varchar2(10) references station(station\_id), to\_st varchar2(10) references station(station\_id), arr\_time varchar2(5),

depart\_time varchar2(5), no\_of\_stops number,

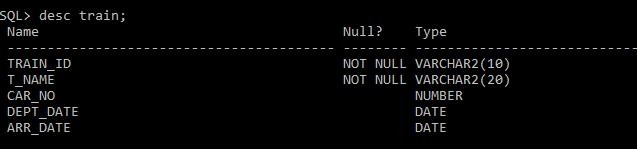
train\_id varchar2(10) references train(train\_id));



Alterations:

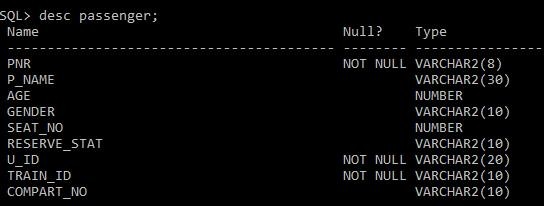
1. SQL>alter table train drop column status\_id; ( had conflicting foreign key)

SQL>alter table train add dept\_date date; SQL> alter table train add arr\_date date;



1. SQL> alter table passenger add constraint cst2 foreign key(train\_id) references train(train\_id);

SQL> alter table passenger add compart\_no varchar(10);



## Insertions:

Table: Users->

insert into users values('145FG','kolkata7'); insert into users values('147L2','prithviraj'); insert into users values('22LO4','dbmsproj');

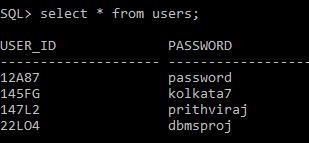


Table: Station->

insert into station values('KOL001', 'Kolkata'); insert into station values('DEL012', 'Delhi'); insert into station values('AGR234', 'Agra');

insert into station values('VSK111', 'Vishakhapatnam'); insert into station values('BOM212', 'Mumbai');

insert into station values('CHH141', 'Chennai'); insert into station values('BAN154', 'Bengaluru');

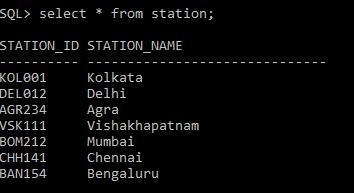


Table: Train->

insert into train values('HWDLJNS10','Janashatabdi',20); insert into train values('AGVKDU11','Duranta',15);

insert into train values('HWBMBE45','Bombay Express',16); insert into train values('CHDLRD14','Rajdhani',20);

insert into train values('HWDLRD10','Rajdhani',20); insert into train values('BMDLRD74','Rajdhani',20); insert into train values('HWDLJNS12','Janashatabdi',20); insert into train values('BMDLRD84','Rajdhani',20);

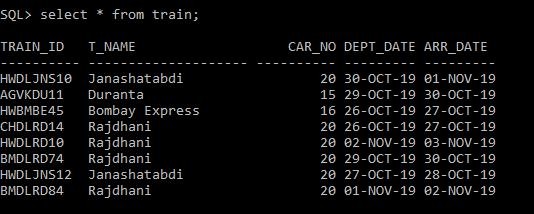


Table: Train\_stat->

insert into train\_stat values('ST100',0,1000,2000,'HWDLJNS10'); insert into train\_stat values('ST101',500,0,2500,'AGVKDU11'); insert into train\_stat values('ST102',0,400,2000,'HWBMBE45'); insert into train\_stat values('ST103',0,40,2000,'CHDLRD14'); insert into train\_stat values('ST104',544,0,2500,'HWDLRD10'); insert into train\_stat values('ST105',40,0,2080,'BMDLRD74'); insert into train\_stat values('ST106',0,200,2080,'HWDLJNS12'); insert into train\_stat values('ST107',40,0,2600,'BMDLRD84');

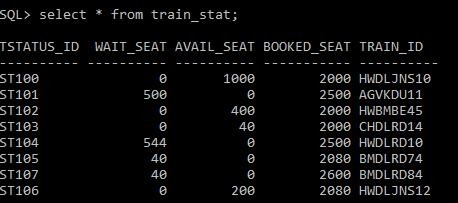


Table: Route->

insert into route values('RT100','KOL001','DEL012','00:45','14:50',26,'HWDLJNS10');

insert into route values('RT101','KOL001','DEL012','17:45','08:15',26,'HWDLJNS12');

insert into route values('RT102','KOL001','DEL012','21:45','06:50',10,'HWDLRD10');

insert into route values('RT103','AGR234','VSK111','17:45','10:15',30,'AGVKDU11');

insert into route values('RT104','KOL001','BOM212','15:10','16:50',33,'HWBMBE45');

insert into route values('RT105','CHH141','DEL012','02:25','21:15',12,'CHDLRD14');

insert into route values('RT106','BOM212','DEL012','17:15','04:50',11,'BMDLRD74');

insert into route values('RT107','BOM212','DEL012','23:45','08:45',10,'BMDLRD84');

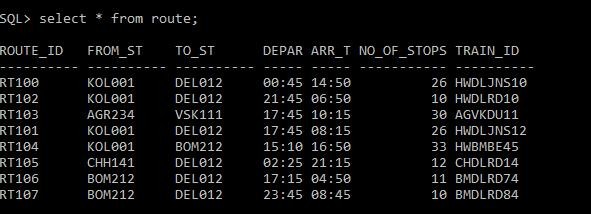


Table: Passenger->

insert into passenger values('ADM101','Prithviraj',20,'Male','140','Booked','145FG','HWDLRD 10');

insert into passenger values('ADM102','Prasad',23,'Male','141','Booked','145FG','HWDLRD10')

;

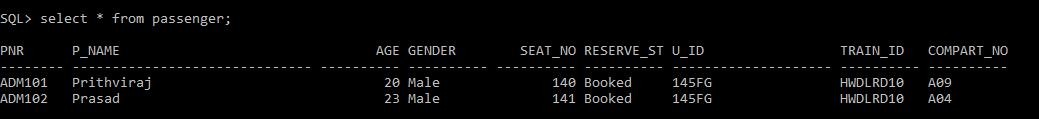
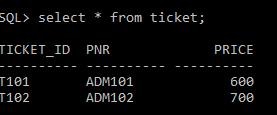


Table: Ticket->

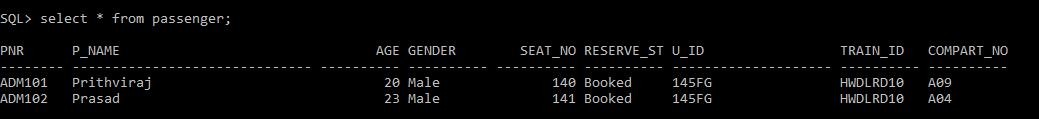
insert into ticket values('T102','ADM102',700); insert into ticket values('T101','ADM101',600);



# Useful SQL queries related to the database:

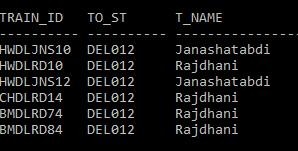
#### Passengers booked by specific user id:

* 1. SQL> select \* from passenger where u\_id = '145FG';



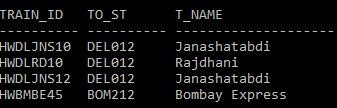
#### Trains going to Delhi:

* 1. SQL> select train\_id,to\_st,t\_name from route natural join train where to\_st = (select station\_id from station where station\_name = 'Delhi')**;**



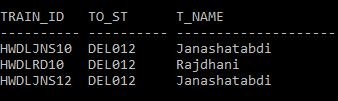
#### Trains starting from Kolkata;

* 1. SQL> select train\_id,to\_st,t\_name from route natural join train where from\_st = (select station\_id from station where station\_name = 'Kolkata');



#### Trains starting from Kolkata going to Delhi:

* 1. SQL> select train\_id,to\_st,t\_name from route natural join train where from\_st = (select station\_id from station where station\_name = 'Kolkata') and to\_st = (select station\_id from station where station\_name = 'Delhi');



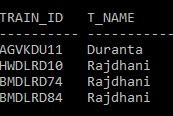
#### Details of passengers going from Kolkata to Delhi:

* 1. SQL> select pnr, p\_name, age , gender , train\_id from passenger natural join train natural join route where from\_st = (select station\_id from station where station\_name = 'Kolkata') and to\_st = (select station\_id from station where station\_name = 'Delhi');



#### Trains having zero vacancy:

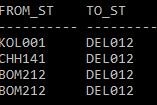
* 1. SQL> select train\_id,t\_name from train natural join train\_stat where wait\_seat > 0;



#### Stations connected via train named Rajdhani:

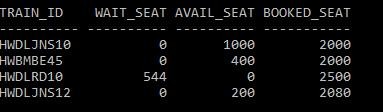
* 1. SQL> select from\_st, to\_st from route natural join train where t\_name

= 'Rajdhani';



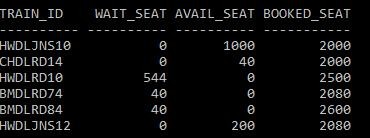
#### status of the trains going from Kolkata:

* 1. SQL> select train\_id, wait\_seat, avail\_seat, booked\_seat from route natural join (train natural join train\_stat) where from\_st = (select station\_id from station where station\_name = 'Kolkata');



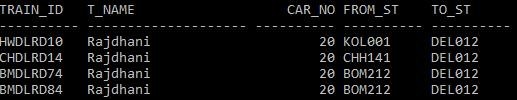
#### Status of trains reaching Delhi:

* 1. SQL> select train\_id, wait\_seat, avail\_seat, booked\_seat from route natural join (train natural join train\_stat) where to\_st = (select station\_id from station where station\_name = 'Delhi');



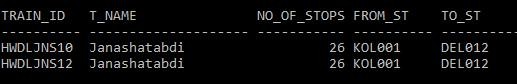
#### No. of train cars in Rajdhanis along with stations :

* 1. SQL> select train\_id, t\_name, car\_no, from\_st , to\_st from train natural join route where t\_name = 'Rajdhani';



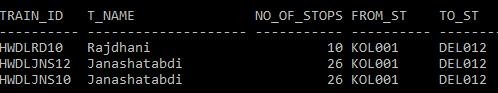
#### No. of stops of trains named Janashatabdi:

* 1. SQL> select train\_id, t\_name, no\_of\_stops , from\_st , to\_st from train natural join route where t\_name = 'Janashatabdi';



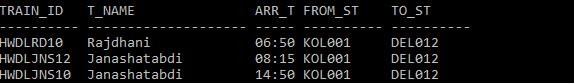
#### No. of stops of trains connecting from Kolkata to Delhi:

* 1. SQL> select train\_id, t\_name, no\_of\_stops , from\_st , to\_st from train natural join route where to\_st = (select station\_id from station where station\_name = 'Delhi') and from\_st = (select station\_id from station where station\_name = 'Kolkata') order by no\_of\_stops asc;



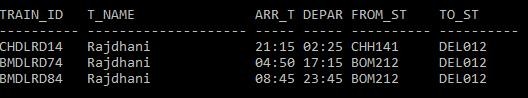
#### Arrival time of trains connect from Kolkata to Delhi:

* 1. SQL> select train\_id, t\_name, arr\_time , from\_st , to\_st from train natural join route where to\_st = (select station\_id from station where station\_name = 'Delhi') and from\_st = (select station\_id from station where station\_name = 'Kolkata') order by arr\_time asc;



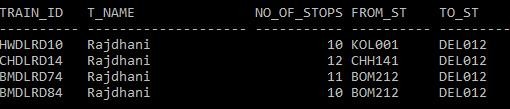
#### Details of trains leaving from Mumbai or from Chennai:

* 1. SQL> select train\_id, t\_name, arr\_time, depart\_time, from\_st , to\_st from train natural join route where from\_st = (select station\_id from station where station\_name = 'Mumbai') or from\_st = (select station\_id from station where station\_name = 'Chennai');



#### Trains which have less than 20 stops:

* 1. SQL> select train\_id, t\_name, no\_of\_stops , from\_st , to\_st from train natural join route where no\_of\_stops < 20;



#### Passengers with ticket price more than 600 (inner join):

* 1. SQL> select passenger.pnr, p\_name , train\_id, ticket\_id from passenger, ticket where passenger.pnr = ticket.pnr and price >600 ;



#### Cities from which Rajdhani leaves (outer join left):

* 1. SQL> select distinct station\_name from train natural join route left join station on route.from\_st = station.station\_id where t\_name = 'Rajdhani';



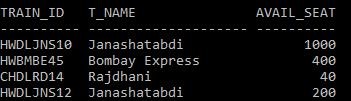
#### Trains with more than 100 on waiting seats:

* 1. SQL> select train\_id, t\_name, wait\_seat from train natural join train\_stat where wait\_seat > 100;



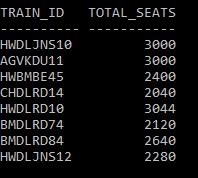
#### Trains which have available seats:

* 1. SQL> select train\_id, t\_name, avail\_seat from train natural join train\_stat where avail\_seat > 0;



#### Total seats in each trains:

* 1. SQL> select train\_id, (wait\_seat+avail\_seat+booked\_seat) TOTAL\_SEATS from train\_stat;



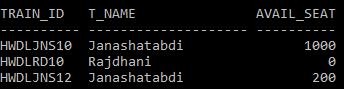
#### Average number of seats in Rajdhani:

* 1. SQL> select avg(wait\_seat+avail\_seat+booked\_seat) AVG\_seats from train\_stat natural join train where t\_name = 'Rajdhani';



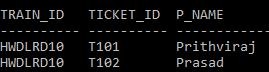
#### Available seats on trains going from Kolkata to Delhi:

* 1. SQL> select train\_id,t\_name, avail\_seat from route natural join train natural join train\_stat where from\_st = (select station\_id from station where station\_name = 'Kolkata') and to\_st = (select station\_id from station where station\_name = 'Delhi');



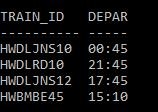
#### Ticket Id and train\_id of passengers whose name starts with P:

* 1. SQL> select Train\_id, Ticket\_id, P\_name from passenger, ticket where passenger.pnr = ticket.pnr and P\_name like 'P%';



#### Departure time of trains that leave from Kolkata:

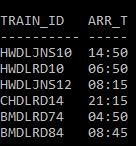
* 1. SQL> select train\_id,depart\_time from train natural join route where from\_st = (select station\_id from station where station\_name = 'Kolkata');



#### Arrival time of trains that reach Delhi:

* 1. SQL> select train\_id,arr\_time from train natural join route where to\_st

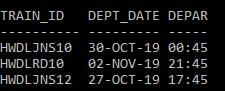
= (select station\_id from station where station\_name = 'Delhi');



#### Departure dates and times of trains leaving for Delhi from Kolkata:

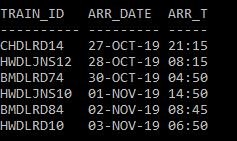
* 1. SQL> select train\_id,dept\_date, depart\_time from train natural join route where from\_st = (select station\_id from station where station\_name

= 'Kolkata') and to\_st = (select station\_id from station where station\_name = 'Delhi');



#### Arrival time of trains in Delhi order by dates:

* 1. SQL> select train\_id, arr\_date, arr\_time from route natural join train where to\_st = (select station\_id from station where station\_name = 'Delhi') order by arr\_date,arr\_time;



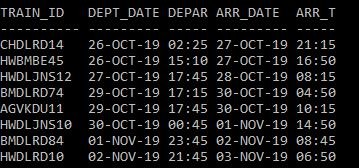
#### Time and date of the ticket booked by a particular passenger:

* 1. SQL> select ticket\_id,pnr,p\_name,train\_id, dept\_date , depart\_time from (passenger natural join ticket) natural join (train natural join route) where p\_name = 'Prithviraj';



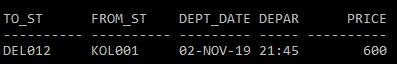
#### Departure and arrival dates and times of all trains registered in the database:

* 1. SQL> select train\_id,dept\_date, depart\_time, arr\_date, arr\_time from train natural join route order by dept\_date;



#### Generate a ticket with necessary details:

* 1. SQL> select ticket\_id, pnr, p\_name, age, gender, train\_id, compart\_no, seat\_no, to\_st, from\_st, dept\_date, depart\_time, price from (passenger natural join ticket) natural join (train natural join route) where p\_name = 'Prithviraj';



# Relational Model:

Arr\_date

Dept\_date

Car\_no

T\_name

Train\_id

**Train**

|  |  |
| --- | --- |
| **User** |  |
| U-id |
|  |
| Password |
|  | |

|  |
| --- |
| **Passenger** |
| Pnr |
| P\_name |
| Age |
| Gender |
| Seat\_no |
| Reserve\_stat |
| U\_id |
| Train\_id |
| Compart\_no |

|  |
| --- |
| **Ticket** |
| Ticket\_id |
| Pnr |
| Price |

|  |
| --- |
| **Train\_stat** |
| Tstatus\_id |
| Wait\_seat |
| Avail\_seat |
| Booked\_seat |
| Train\_id |

|  |
| --- |
| **Route** |
| Route\_id |
| From\_st |
| To\_st |
| Depart\_time |
| Arr\_time |
| No\_of\_stops |
| Train\_id |

|  |
| --- |
| **Station** |
| Station\_id |
| Station\_name |

**Future scope of the Project :**

Railway system is a huge implementation of databases and as time advances new procedures of implementing reservation of railway seats are coming up. Thus, new GUI’s are being developed making it more and more user friendly. If this database is used in making a GUI with or online application using Java Server Pages, this can be devised to be a succesful implementation.

# Conclusion:

Database Management and Data Handling is a big part of the IT industry. Thus proper knowledge of SQL is important while working on databases in order to avoid data redundancy and conflicts. Through this project, we learned to avoid such data conflicts and implement the queries properly.

# REMARKS: