DBMS PROJECT

METRO SYSTEM MANAGEMENT SYSTEM DATABASE

BY:

YASHSVINI KATARE (21CSB0A67)

SAVVY JAIN (21CSB0F15)

ACKNOWLEDGEMENT

- Inspiration and motivation play a key role in the success of any venture. Successful completion of any project requires proper guidance and help. We express our gratitude to the following people who have made this possible.

- We thank DBMS professors, **Dr. T Ramakrishnudu** Sir and **Dr. RBV Subramanyam** Sir for giving us the opportunity to develop deep into database concepts. Their lessons have been really helpful in learning the fundamental DBMS concepts like relations, E-R diagrams, Structured Query Language, functional dependencies, normalization, indexing, etc. in detail.

- We extend our thanks to the whole **CSE Department and lab faculty** for being so helpful and available whenever we require any kind of guidance..

- We thank the NITW management, Dean - Academic and Director - NITW, for giving us this opportunity to discover this subjects in the course, and enter new avenues in the field of database management. We also thank our parents, elders and well-wishers for being there with us and giving us all kinds of technical and moral support.

TABLE OF CONTENTS

TOPIC	PAGE NO.
PROBLEM STATEMENT	4
ER DIAGRAM	6
RELATIONAL MODEL DIAGRAM	7
RELATIONS	8-31
FUNCTIONAL DEPENDENCY AND NORMALIZATION	32-37
STANDARD QUERIES	38-43

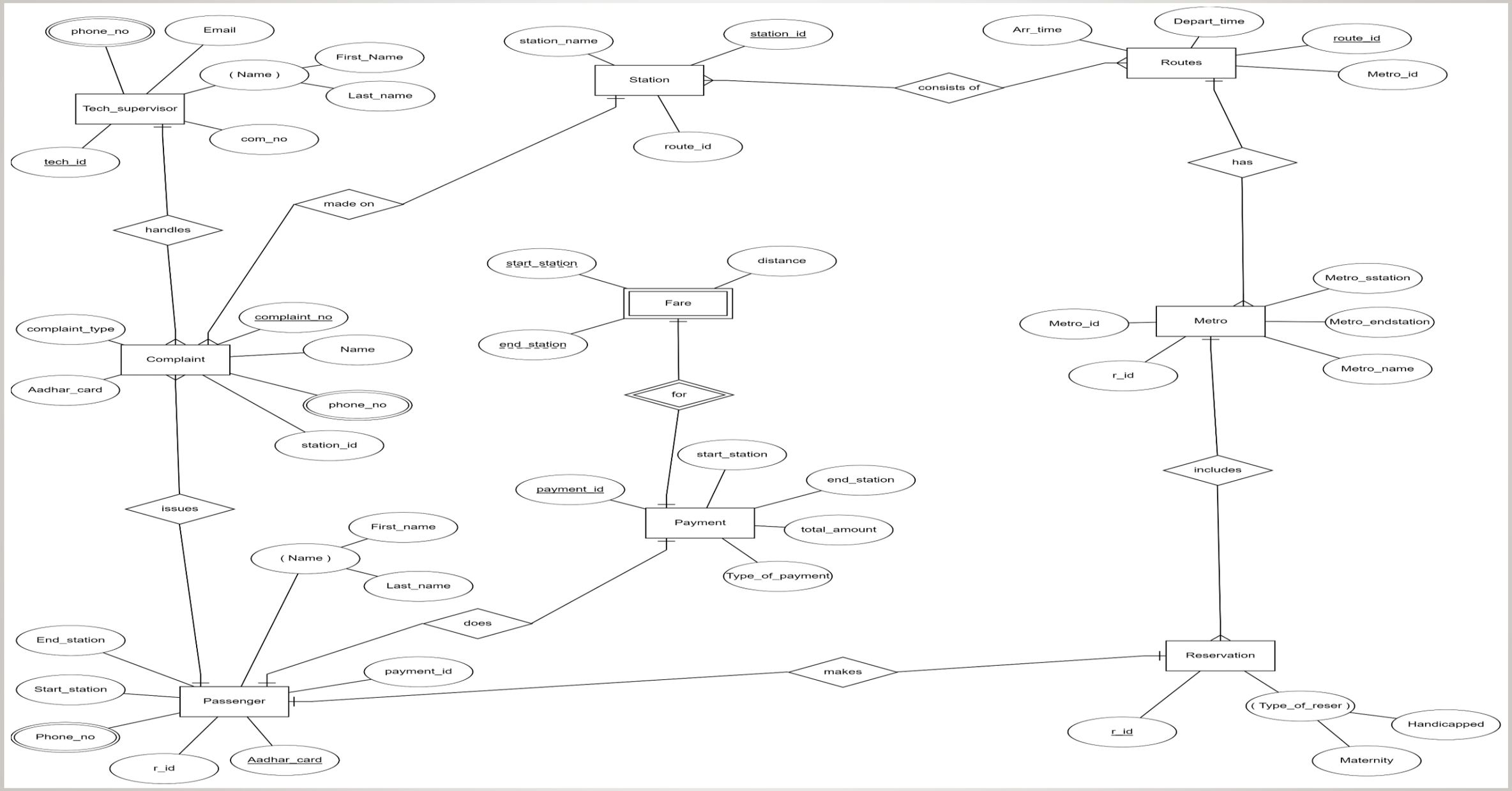
PROBLEM STATEMENT:

- A Metro System is one which helps us to maintain an organisational data regarding a metro system. Since it is very difficult to maintain a record of data items and this information manually or in file system, we are creating a database for the complete management of resources and tasks of metro system.For main functionalities, we have to maintain many records such as train numbers, train route, departure time, arrival time, train type, fare, passengers, etc. When people travel ,some reserve seat (Additional feature for Metro System) for them,we have maintained a record of reservations. To avoid any hassles,we have gathered all the information in a structured manner in our project.

ASSUMPTIONS OF ER MODEL

- One Tech_supervisor can handle many complaints at a time raised by passengers on the inconvenience caused while travelling in Metro.
- One Passenger can file many complaints at a time.
- One passenger makes a reservation at a time.
- One Passenger makes a payment at a time either by cash or online
- Each metro has a different route
- We can make many reservations on a Metro.

ER DIAGRAM


phone_no
Email
(Name)
First_Name
Last_name
Tech_supervisor
com_no
tech_id
handles
made on
station_name
station_id
Station
route_id
consists of
Arr_time
Depart_time
route_id
Routes
Metro_id
has
start_station
distance
Fare
end_station
for
complaint_no
complaint_type
Name
Complaint
Aadhar_card
phone_no
station_id
issues
Metro_sstation
Metro_id
Metro
Metro_endstation
r_id
Metro_name
includes
start_station
payment_id
end_station
Payment
total_amount
Type_of_payment
First_name
(Name)
Last_name
does
End_station
Start_station
Phone_no
r_id
Aadhar_card
Passenger
payment_id
makes
Reservation
(Type_of_reser)
r_id
Handicapped
Maternity

RELATIONAL MODEL

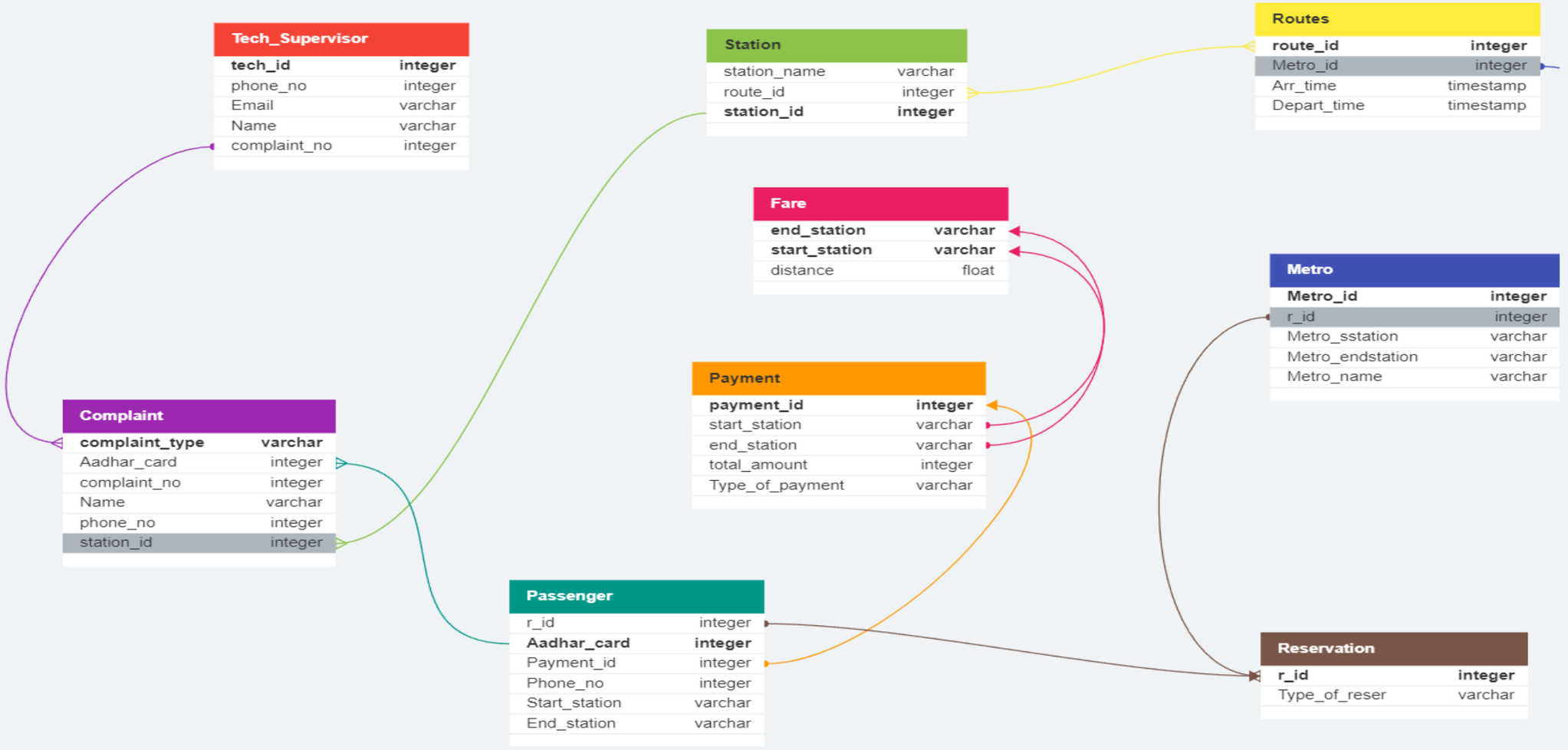


TABLE CREATION

TABLE TECH_SUPERVISOR

- INFO ABOUT TABLE

Tech_supervisor: The tech_supervisor relation is the collection of all the tech_supervisors who manage all the issues related to complaints by passengers and other transactions It contains the information of those Tech_supervisors . Tech_id is the primary key here.

- CREATE TABLE TECH_SUPERVISOR
- (
- TECH_ID INTEGER,
- EMAIL VARCHAR(50),
- FIRST_NAME VARCHAR(50),

```
LAST_NAME VARCHAR(50),
COMPLAINT_NO INTEGER,
PHONE_NO INTEGER,
PRIMARY KEY(TECH_ID),
FOREIGN KEY(COMPLAINT_NO) REFERENCES complaint
);
DESC TECH_SUPERVISOR;
```

```
Name          Null?    Type
------------- -------- ------------
TECH_ID       NOT NULL NUMBER(38)
EMAIL                  VARCHAR2(50)
FIRST_NAME             VARCHAR2(50)
LAST_NAME              VARCHAR2(50)
COMPLAINT_NO           NUMBER(38)
PHONE_NO               NUMBER(38)
```

```
INSERT INTO TECH_SUPERVISOR VALUES(1,'k@gmail.com','Vinita,'Jain',123,9668710234);
INSERT INTO TECH_SUPERVISOR VALUES(2,'t@gmail.com','Radha','Gupta',124,9772345634);
INSERT INTO TECH_SUPERVISOR VALUES(3,'r@gmail.com','Mohan','Reddy',125,9890123234);
INSERT INTO TECH_SUPERVISOR VALUES(4,'p@gmail.com','Krish’,'Singh',126,9980345671);
INSERT INTO TECH_SUPERVISOR VALUES(5,'c@gmail.com','Meenal','Singh',127,773826534);
```

TECH_ID	EMAIL	FIRST_NAME	LAST_NAME	COMPLAINT_NO	PHONE_NO
11	k@gmail.com	Kirti	Jain	123	9668710234
2	t@gmail.com	Trisha	Gupta	124	9772345634
3	r@gmail.com	Rohan	Reddy	125	9890123234
4	p@gmail.com	Paul	Singh	126	9980345671
5	c@gmail.com	Chetan	Singh	127	773826534

TABLE COMPLAINT

- INFORMATION ABOUT TABLE

- Complaint:The Complaint relation is a record of all complaints filed by the passengers boarding the metro for issues like crime,lost items etc. The Complaints table also consists about the details of passengers filing those Complaints.

```
CREATE TABLE COMPLAINT
(
COMPLAINT_NO INTEGER NOT NULL,
COMPLAINT_TYPE VARCHAR(50) NOT NULL,
STATION_ID INTEGER NOT NULL,
NAME VARCHAR(50) NOT NULL,
PHONE_NO INTEGER NOT NULL,
```

```
AADHAR_CARD INTEGER NOT NULL,
PRIMARY KEY(COMPLAINT_NO),
FOREIGN KEY(STATION_ID) REFERENCES STATION,
FOREIGN KEY(AADHAR_CARD) REFERENCES PASSENGER
);
```

DESC COMPLAINT;

```
Name           Null?    Type
-------------- -------- ------------
COMPLAINT_NO   NOT NULL NUMBER(38)
COMPLAINT_TYPE NOT NULL VARCHAR2(50)
STATION_ID     NOT NULL NUMBER(38)
NAME           NOT NULL VARCHAR2(50)
PHONE_NO       NOT NULL NUMBER(38)
AADHAR_CARD    NOT NULL NUMBER(38)
```

```
--INSERT INTO COMPLAINT VALUES(123,'Robbery',11001,'Kirti',9668710234,11111);
--INSERT INTO COMPLAINT VALUES(124,'Hygiene',11002,'Trisha',9772345634,22222);
--INSERT INTO COMPLAINT VALUES(125,'Robbery',11001,'Rohan',9890123234,33333);
--INSERT INTO COMPLAINT VALUES(126,'Hygiene',11002,'Paul',9980345671,44444);
--INSERT INTO COMPLAINT VALUES(127,'LostItems',11003,'Chetan',773826534,55555);
```

COMPLAINT_NO	COMPLAINT_TYPE	STATION_ID	NAME	PHONE_NO	AADHAR_CARD
123	Robbery	11001	Kirti	9668710234	11111
124	Hygiene	11002	Trisha	9772345634	22222
125	Robbery	11001	Rohan	9890123234	33333
126	Hygiene	11002	Paul	9980345671	44444
127	LostItems	11003	Chetan	773826534	55555

TABLE PASSENGER

The Passengers table is a collection of all Passengers and their details like phone_no,aadhar card number,and most importantly their start and end stations.

Aadhar_card is the primary key here.

```
CREATE TABLE PASSENGER
(
 NAME VARCHAR(50) NOT NULL,
PHONE_NO INTEGER NOT NULL,
START_STATION VARCHAR(50) NOT NULL,
END_STATION VARCHAR(50) NOT NULL,
PAYMENT_ID INTEGER NOT NULL,
AADHAR_CARD INTEGER NOT NULL,
R_ID INTEGER NOT NULL,
```

```
PRIMARY KEY(AADHAR_CARD),
FOREIGN KEY(PAYMENT_ID) REFERENCES PAYMENT,
FOREIGN KEY(R_ID) REFERENCES RESERVATION
);
desc PASSENGER;
```

Name	Null?	Type
NAME	NOT NULL	VARCHAR2(50)
PHONE_NO	NOT NULL	NUMBER(38)
START_STATION	NOT NULL	VARCHAR2(50)
END_STATION	NOT NULL	VARCHAR2(50)
PAYMENT_ID	NOT NULL	NUMBER(38)
AADHAR_CARD	NOT NULL	NUMBER(38)
R_ID	NOT NULL	NUMBER(38)

```
--INSERT INTO PASSENGER
VALUES('Kirti',9668710234,'Secunderabad','BharatNagar',12345,11111,1);

--INSERT INTO PASSENGER VALUES('Trisha',9772345634,'Erragadda','Begumpet',13456,22222,2);

--INSERT INTO PASSENGER VALUES('Rohan',9890123234,'Paradise','Begumpet',14567,33333,3);

--INSERT INTO PASSENGER
VALUES('Paul',9980345671,'Secunderabad','Madhapur',15678,44444,4);

--INSERT INTO PASSENGER
VALUES('Chetan',773826534,'BharatNagar','Kukatpally',16789,55555,5);

SELECT * FROM PASSENGER;
```

NAME	PHONE_NO	START_STATION	END_STATION
Kirti	9668710234	Secunderabad	BharatNagar
Trisha	9772345634	Erragadda	Begumpet
Rohan	9890123234	Paradise	Begumpet
Paul	9980345671	Secunderabad	Madhapur
Chetan	773826534	BharatNagar	Kukatpally

PAYMENT_ID	AADHAR_CARD	R_ID
12345	11111	1
13456	22222	2
14567	33333	3
15678	44444	4
16789	55555	5

TABLE PAYMENT

Payment:The Payment table is a record of the fare involved in travelling from one station to another and also the type of payment (cash,online),etc

Payment_id is the primary key here.

```
CREATE TABLE PAYMENT
(
PAYMENT_ID INTEGER NOT NULL,
START_STATION VARCHAR(50) ,
END_STATION VARCHAR(50) ,
TOTAL_AMOUNT INTEGER,
```

```
TYPE_OF_PAYMENT VARCHAR(50),
PRIMARY KEY(PAYMENT_ID),
FOREIGN KEY(START_STATION,END_STATION) REFERENCES FARE
);

Desc PAYMENT;
```

Name	Null?	Type
PAYMENT_ID	NOT NULL	NUMBER(38)
START_STATION		VARCHAR2(50)
END_STATION		VARCHAR2(50)
TOTAL_AMOUNT		NUMBER(38)
TYPE_OF_PAYMENT		VARCHAR2(50)

--INSERT INTO PAYMENT VALUES(12345,'Secunderabad','BharatNagar',70,'Cash');

--INSERT INTO PAYMENT VALUES(13456,'Erragadda','Begumpet',80,'Cash');

--INSERT INTO PAYMENT VALUES(14567,'Paradise','Begumpet',90,'Online');

--INSERT INTO PAYMENT VALUES(15678,'Secunderabad','Madhapur',100,'Cash');

--INSERT INTO PAYMENT VALUES(16789,'BharatNagar','Kukatpally',110,'Online’);

PAYMENT_ID	START_STATION	END_STATION	TOTAL_AMOUNT	TYPE_OF_PAYMENT
12345	Secunderabad	BharatNagar	70	Cash
13456	Erragadda	Begumpet	80	Cash
14567	Paradise	Begumpet	90	Online
15678	Secunderabad	Madhapur	100	Cash
16789	BharatNagar	Kukatpally	110	Online

TABLE FARE

```
CREATE TABLE FARE
(
DISTANCE INTEGER NOT NULL,
START_STATION VARCHAR(50) NOT NULL,
END_STATION VARCHAR(50) NOT NULL,
PRIMARY KEY(START_STATION,END_STATION)

);
```

INFORMATION ABOUT TABLE:

Fare is a weak entity table here.It contains the distances and start,end stations and keeps track of the distance according to which the passengers pay.

Desc FARE;

Name	Null?	Type
DISTANCE	NOT NULL	NUMBER(38)
START_STATION	NOT NULL	VARCHAR2(50)
END_STATION	NOT NULL	VARCHAR2(50)

--INSERT INTO FARE VALUES(30,'Secunderabad','BharatNagar');

--INSERT INTO FARE VALUES(45,'Erragadda','Begumpet');

--INSERT INTO FARE VALUES(50,'Paradise','Begumpet');

--INSERT INTO FARE VALUES(55,'Secunderabad','Madhapur');

--INSERT INTO FARE VALUES(65,'BharatNagar','Kukatpally');

DISTANCE	START_STATION	END_STATION
30	Secunderabad	BharatNagar
45	Erragadda	Begumpet
50	Paradise	Begumpet
55	Secunderabad	Madhapur
65	BharatNagar	Kukatpally

TABLE STATION

```
CREATE TABLE STATION
(
STATION_ID INTEGER NOT NULL,
ROUTE_ID INTEGER NOT NULL,
STATION_NAME VARCHAR(50) NOT NULL,
PRIMARY KEY(STATION_ID),
FOREIGN KEY(ROUTE_ID) REFERENCES ROUTES
);
```

TABLE INFORMATION:

The station table maintains the stations id s and the routes along which metro travels.

Station_id is the primary key here.

Desc STATION;

Name	Null?	Type
STATION_ID	NOT NULL	NUMBER(38)
ROUTE_ID	NOT NULL	NUMBER(38)
STATION_NAME	NOT NULL	VARCHAR2(50)

```
INSERT INTO station VALUES(11001,1,'S1');
INSERT INTO STATION VALUES(11002,2,'S2');
INSERT INTO STATION VALUES(11003,3,'S3');
INSERT INTO STATION VALUES(11004,4,'S4');
INSERT INTO STATION VALUES(11005,5,'S5');
SELECT * FROM STATION;
```

STATION_ID	ROUTE_ID	STATION_NAME
11001	1	S1
11002	2	S2
11003	3	S3
11004	4	S4
11005	5	S5

TABLE ROUTES

```
CREATE TABLE ROUTES
(
ARR_TIME TIMESTAMP(0) NOT NULL,
DEPART_TIME TIMESTAMP(0) NOT NULL,
ROUTE_ID INTEGER NOT NULL,
METRO_ID INTEGER NOT NULL,
PRIMARY KEY(ROUTE_ID),
FOREIGN KEY(METRO_ID) REFERENCES METRO
);
```

TABLE INFORMATION:

The routes table maintains the collection of metros ,their ids and arrival times and departure times.

R_id is the primary key.

DESC ROUTES;

Name	Null?	Type
ARR_TIME	NOT NULL	TIMESTAMP(0)
DEPART_TIME	NOT NULL	TIMESTAMP(0)
ROUTE_ID	NOT NULL	NUMBER(38)
METRO_ID	NOT NULL	NUMBER(38)

ROUTES

--INSERT INTO ROUTES VALUES('01-01-20 11:08:54','01-01-20 11:10:54',1,1122);

--INSERT INTO ROUTES VALUES('01-01-20 06:02:34','01-01-20 06:08:36',2,1133);

--INSERT INTO ROUTES VALUES('01-01-20 10:15:20','01-01-20 10:22:54',3,1144);

--INSERT INTO ROUTES VALUES('01-01-20 04:10:54','01-01-20 04:11:54',4,1155);

--INSERT INTO ROUTES VALUES('01-01-20 07:05:54','01-01-20 07:10:54',5,1166);

ARR_TIME	DEPART_TIME	ROUTE_ID	METRO_ID
01-01-20 11:08:54.000000000 AM	01-01-20 11:10:54.000000000 AM	1	1122
01-01-20 6:02:34.000000000 AM	01-01-20 6:08:36.000000000 AM	2	1133
01-01-20 10:15:20.000000000 AM	01-01-20 10:22:54.000000000 AM	3	1144
01-01-20 4:10:54.000000000 AM	01-01-20 4:11:54.000000000 AM	4	1155
01-01-20 7:05:54.000000000 AM	01-01-20 7:10:54.000000000 AM	5	1166

TABLE RESERVATION

```
CREATE TABLE RESERVATION
(
  R_ID INTEGER NOT NULL,
 TYPE_OF_RESER VARCHAR(50) NOT NULL,
 PRIMARY KEY(R_ID)
);
```

TABLE INFORMATION:

The reservation table maintains the reservations like Handicapped,Maternity along with their reservation ids.

DESC RESERVATION;

Name	Null?	Type
R_ID	NOT NULL	NUMBER(38)
TYPE_OF_RESER	NOT NULL	VARCHAR2(50)

```
--INSERT INTO RESERVATION VALUES(1,'Maternity');
--INSERT INTO RESERVATION VALUES(2,'Maternity');
--INSERT INTO RESERVATION VALUES(3,'Handicapped');
--INSERT INTO RESERVATION VALUES(4,'Handicapped');
--INSERT INTO RESERVATION VALUES(5,'Handicapped');
SELECT * FROM RESERVATION;
```

R_ID	TYPE_OF_RESER
1	Handicapped
2	Maternity
3	Handicapped
4	Handicapped
5	Handicapped

FUNCTIONAL DEPENDENCIES AND NORMALISATION

1)Tech_supervisor:

PRIMARY KEY:tech_id

F.D :tech_id->R

- As there are no multivalued dependencies, it is in 1NF.
- Here, there is no partial dependency as there is no non-prime attribute depending on the proper subset of a candidate key so it is in 2NF.
- As all the functional dependencies have determinants as super keys it is both in 3NF and BCNF.
- Therefore, the table is in BCNF.

2)Complaint:

PRIMARY KEY: Complaint_no

F.D: Complaint_no - > R

- As there are no multivalued dependencies, it is in 1NF.
- Here, there is no partial dependency as there is no non-prime attribute depending on the proper subset of a candidate key so it is in 2NF.
- As all the functional dependencies have determinants as super keys it is both in 3NF and BCNF.
- Therefore, the table is in BCNF.

3)Passenger:

PRIMARY KEY:Aadhar_card

F.D : Aadhar_card - > R

- As there are no multivalued dependencies,it is in 1NF.
- Here, there is no partial dependency as there is no non-prime attribute depending on the proper subset of a candidate key so it is in 2NF.
- As, all the functional dependencies have determinants as super keys it is both in 3NF and BCNF.
- Therefore,the table is in BCNF.

4)Payment:

PRIMARY KEY: Payment_id

F.D :Payment_id - > R

- As there are no multivalued dependencies,it is in 1NF.
- Here, there is no partial dependency as there is no non-prime attribute depending on the proper subset of a candidate key so it is in 2NF.
- As, all the functional dependencies have determinants as super keys it is both in 3NF and BCNF.
- Therefore,the table is in BCNF.

5)Fare:

PRIMARY KEY:ticket_id

F.D :ticket_id - > R

- As there are no multivalued dependencies,it is in 1NF.
- Here, there is no partial dependency as there is no non-prime attribute depending on the proper subset of a candidate key so it is in 2NF.
- As, all the functional dependencies have determinants as super keys it is both in 3NF and BCNF.
- Therefore,the table is in BCNF.

6)Station:

PRIMARY KEY:Station_id

F.D :Station_id - > R

- As there are no multivalued dependencies,it is in 1NF.
- Here, there is no partial dependency as there is no non-prime attribute depending on the proper subset of a candidate key so it is in 2NF.
- As, all the functional dependencies have determinants as super keys it is both in 3NF and BCNF.
- Therefore,the table is in BCNF.

7)Reservation:

PRIMARY KEY: r_id

F.D: r_id - >R

- As there are no multivalued dependencies,it is in 1NF.

- Here, there is no partial dependency as there is no non-prime attribute depending on the proper subset of a candidate key so it is in 2NF.
- As, all the functional dependencies have determinants as super keys it is both in 3NF and BCNF.
- Therefore,the table is in BCNF.

8)Routes:

PRIMARY KEY: route_id

F.D: route_id - >R

- As there are no multivalued dependencies,it is in 1NF.
- Here, there is no partial dependency as there is no non-prime attribute depending on the proper subset of a candidate key so it is in 2NF.
- As, all the functional dependencies have determinants as super keys it is both in 3NF and BCNF.
- Therefore,the table is in BCNF.

9)Metro:

PRIMARY KEY: seat_no

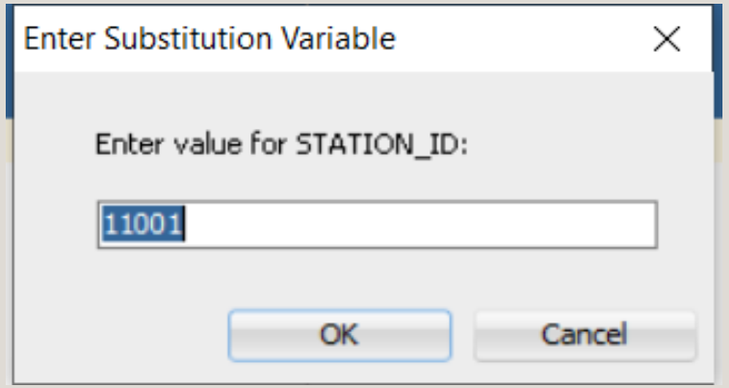
F.D: seat_no - >R

- As there are no multivalued dependencies,it is in 1NF.
- Here, there is no partial dependency as there is no non-prime attribute depending on the proper subset of a candidate key so it is in 2NF.
- As, all the functional dependencies have determinants as super keys it is both in 3NF and BCNF.
- Therefore,the table is in BCNF.

QUERIES

1. Find the complaints of a station_id inputted from user;

```
SELECT COMPLAINT_TYPE
FROM COMPLAINT
WHERE STATION_ID='&STATION_ID';
```




```
old:SELECT COMPLAINT_TYPE
FROM COMPLAINT
WHERE STATION_ID='&STATION_ID'
new:SELECT COMPLAINT_TYPE
FROM COMPLAINT
WHERE STATION_ID='11001'

COMPLAINT_TYPE
--------------------------------
Robbery
LostItems
```

2. Find the start station where the end station starts with B and person paying in cash

```
SELECT START_STATION
FROM PAYMENT
WHERE END_STATION LIKE 'B%' AND TYPE_OF_PAYMENT='Cash’;
```

```
START_STATION
-------------------------
Secunderabad
Erragadda
```

3.Find minimum wait time of a metro train

```
SELECT  MIN(DEPART_TIME-ARR_TIME) AS WAIT_TIME
FROM ROUTES;
```

```
WAIT_TIME
-------------------
+00 00:01:00.000000
```

4.Find the email_id of the Tech_supervisor filing the complaint entered by the user.

```
SELECT EMAIL,FIRST_NAME
FROM TECH_SUPERVISOR
WHERE COMPLAINT_NO='&COMPLAINT_NO';
```

EMAIL	FIRST_NAME
p@gmail.com	Krish

5.Find the Payment_ID’s of those Passengers who are boarding from station 11001.

```
SELECT COMPLAINT.NAME,PAYMENT_ID
FROM COMPLAINT,PASSENGER
WHERE STATION_ID=11001 AND
COMPLAINT.AADHAR_CARD=PASSENGER.AADHAR_CARD;
```

NAME	PAYMENT_ID
Kirti	12345
Rohan	14567

6.Find the TECH_SUPERVISORS solving Hygiene issue.

```
SELECT TECH_SUPERVISOR.FIRST_NAME
FROM TECH_SUPERVISOR,COMPLAINT
WHERE COMPLAINT.COMPLAINT_TYPE='Hygiene'
AND TECH_SUPERVISOR.COMPLAINT_NO=COMPLAINT.COMPLAINT_NO;
```

```
FIRST_NAME
------------------------------
Radha
Krish
```

7.Find the Metros and their arrival time which are arriving before 8 AM

```
SELECT METRO.METRO_NAME,ROUTES.ARR_TIME
from METRO,ROUTES
WHERE ROUTES.ARR_TIME<('01-01-20 8:00:00') AND METRO.METRO_ID=ROUTES.METRO_ID;
```

METRO_NAME	ARR_TIME
MM2	01-01-20 6:02:34.000000000 AM
MM4	01-01-20 4:10:54.000000000 AM
MM5	01-01-20 7:05:54.000000000 AM

THANK YOU

Happy Travelling…