

BUS RESERVATION BACKEND SYSTEM

DBMS MINI PROJECT

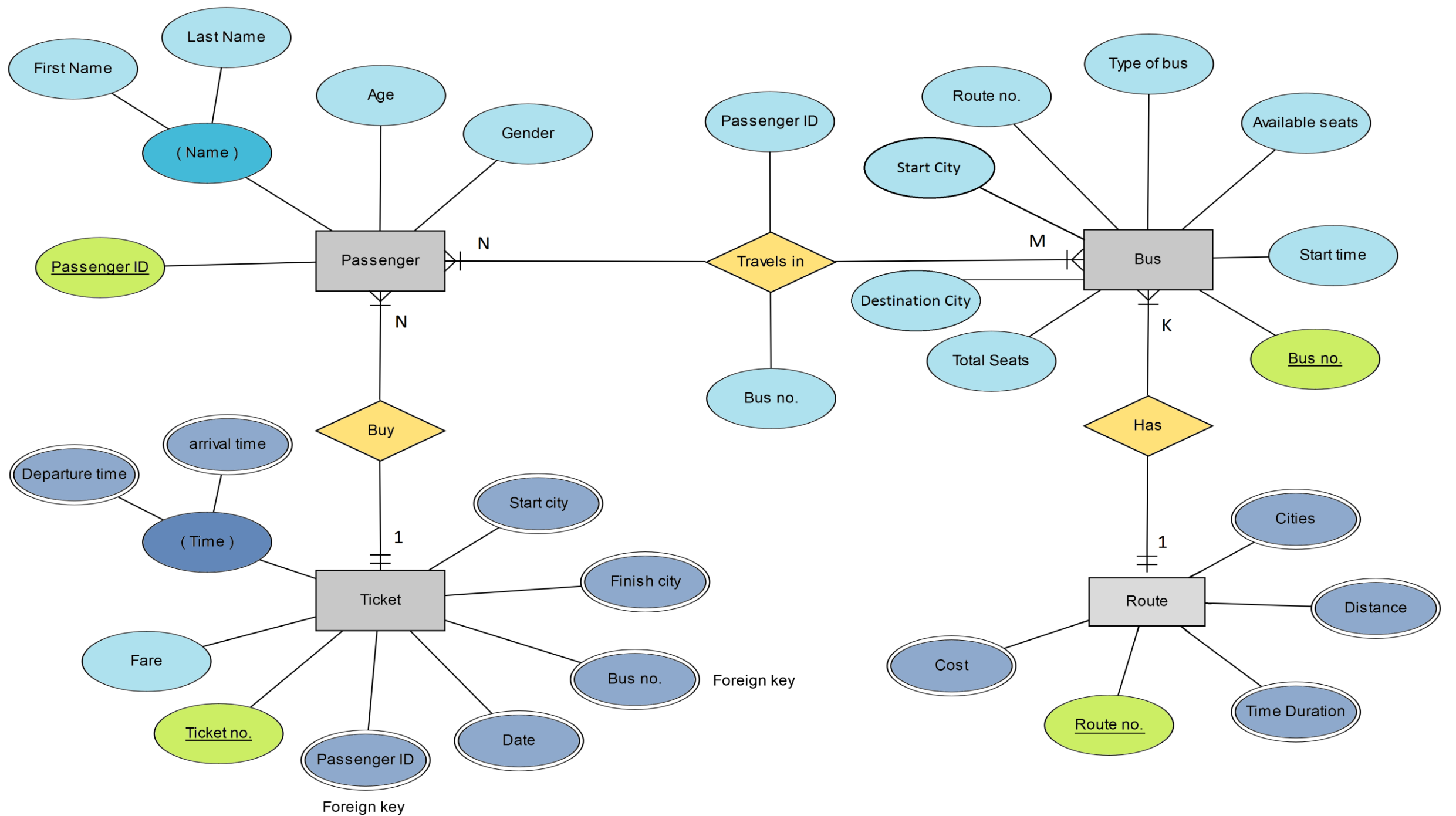
PROJECT PARTNERS

- I) JIGNESH JINJALA (UI8CO073)

DEFINITION

- We make a bus reservation backend system used by many government transportation commissions.
- In this project, first of all we make ER-Diagram such that it gives basic structure of database.
- Then we convert ER-Diagram into relational database.
- And final part is Normalization such that we can take away the disadvantages such as redundancy and inconsistency in database so that we can make our process fast and save space for database such that we can store much data.

ER - DIAGRAM



EXPLANATION OF ER-DIAGRAM

- ER-Diagram basically consists of 4 main Entities-sets and 3 binary relationships among them.
- Each entity-sets consists of many attributes.
- Primary and Foreign keys are presented by notation.
- Some attributes are multivalued , derived etc.
- Also Travels-In relationship consists attributes.

EXPLANATION OF ER-DIAGRAM

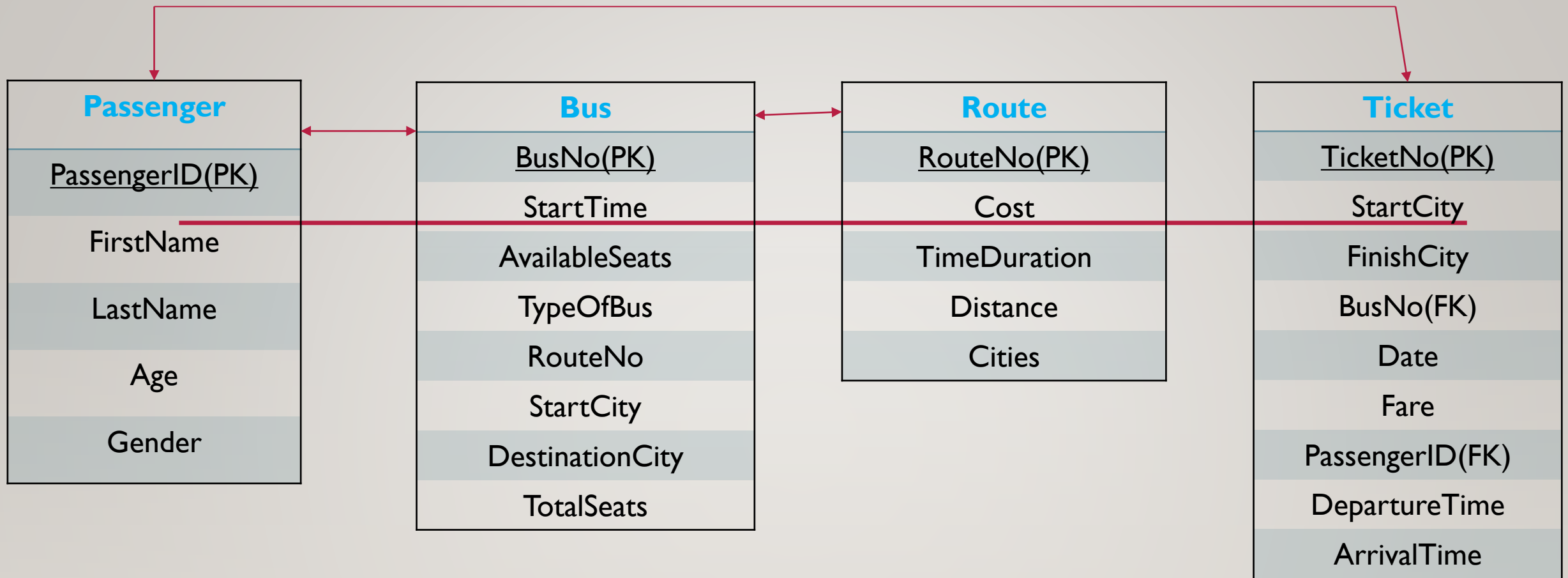
- Entity-sets:
 - 1) BUS
 - 2) TICKET
 - 3) PASSENGER
 - 4) ROUTE

EXPLANATION OF ER-DIAGRAM

- Relationships:
 - 1)TRAVELS-IN(MANY-MANY RELATIONSHIPS BETWEEN PASSENGER AND BUS ENTITY-SETS)
 - 2)BUY(MANY-ONE RELATIONSHIP BETWEEN PASSENGER AND TICKET ENTITY-SETS)
 - 3)HAS(MANY-ONE RELATIONSHIP BETWEEN BUS AND ROUTE ENTITY SETS)

CONVERSION OF ER-DIAGRAM INTO RELATIONAL MODEL





PK – PRIMARY KEY
FK – FOREIGN KEY

NORMALIZATION



RULES FOR NORMALIZATION

- 1NF
 - No multivalued attributes
- 2NF
 - No partial dependencies
- 3NF
 - Either satisfy first or second or both for FD ($X \rightarrow Y$) [No transitive dependencies]
 - 1) X is a super key
 - 2) Y is a prime attribute

RULES FOR NORMALIZATION

- BCNF
 - FOR $(X \twoheadrightarrow Y)$, X must be a super key
- 4NF
 - Has no multivalued dependency
- 5NF
 - Not contains any join dependency & joining should be lossless
- For the succeeding ones , they must first satisfy preceding ones.

NORMALIZATION CONCEPTS FOR ROUTE(ROUTENO , COST , DISTANCE , TIME DURATION , CLASS)

- Let take this relation as R(A , B , C , D , E) respectively for all attributes of Route.
- Here the functional dependency is from (A \rightarrow BCDE).
- The table is in 1NF as there is no multivalued attribute.
- The table is in 2NF as there is no partial dependencies.
- The table is in 3NF as there is no transitive dependencies.
- The table is in BCNF because A is a super key.

NORMALIZATION CONCEPTS FOR BUS(BUSNO.,STARTTIME ,AVAILABLESEATS ,TYPEOFBUS , ROUTENO.,STARTCITY , DESTINATIONCITY,TOTALSEATS)

- Let take this relation as R(A , B , C , D , E , F , G , H) respectively for all attributes of Bus.
- Here the functional dependency is from (A \longrightarrow BCDEFGH).
- The table is in 1NF as there is no multivalued attribute.
- The table is in 2NF as there is no partial dependencies.
- The table is in 3NF as there is no transitive dependencies.
- The table is in BCNF because A is a super key.

NORMALIZATION CONCEPTS FOR PASSENGER(PASSENGERID , FIRSTNAME , LASTNAME , AGE , GENDER)

- Let take this relation as $R(A, B, C, D, E)$ respectively for all attributes of Bus.
- Here the functional dependency is from $(A \rightarrow BCDE)$.
- The table is in 1NF as there is no multivalued attribute.
- The table is in 2NF as there is no partial dependencies.
- The table is in 3NF as there is no transitive dependencies.
- The table is in BCNF because A is a super key.

NORMALIZATION CONCEPTS FOR TICKET(TICKETNO., PASSENGERID , STARTCITY , FINISHCITY , BUSNO. , DATE , FARE , DEPARTURETIME , ARRIVALTIME)

- Let take this relation as $R(A, B, C, D, E, F, G, H, I)$ respectively for all attributes of Bus.
- Here the functional dependency is from $(A \rightarrow BCDEFGHI)$.
- The table is in 1NF as there is no multivalued attribute.
- The table is in 2NF as there is no partial dependencies.
- The table is in 3NF as there is no transitive dependencies.
- The table is in BCNF because A is a super key.

CREATION OF TABLES



CODE

- CREATE TABLE PASSENGER
 - (PASSENGER_ID NUMBER PRIMARY KEY,
 - FIRST_NAME VARCHAR(20) NOT NULL,
 - LAST_NAME VARCHAR(20),
 - AGE NUMBER,
 - GENDER VARCHAR(1));
- CREATE TABLE TICKET(
 - TICKET_NO NUMBER PRIMARY KEY,
 - FARE NUMBER NOT NULL);

-
- CREATE TABLE TICKET_P(

 - TICKET_NO NUMBER,

 - PASSENGER_ID NUMBER,

 - CONSTRAINT TICKET_P_PK
PRIMARY KEY(TICKET_NO ,
PASSENGER_ID));

- CREATE TABLE TICKET_B(

- BUS_NO NUMBER,

- TICKET_NO NUMBER,

- DATE_B DATE,

- BOARDING_CITY VARCHAR(20),

- DROP_CITY VARCHAR(20),

- START_J_TIME TIMESTAMP,

- FINISH_J_TIME TIMESTAMP,

- CONSTRAINT TICKET_B_PK PRIMARY
KEY(BUS_NO ,TICKET_NO));

-
- CREATE TABLE BUS(
 - BUS_NO NUMBER PRIMARY KEY,
 - TYPE VARCHAR(10),
 - START_TIME DATE,
 - TOTAL_SEAT NUMBER,
 - RATE_PKM NUMBER);

- CREATE TABLE BUS_C(
 - BUS_NO NUMBER,
 - CITY VARCHAR(20),
 - TIMEC DATE,
 - SEAT_AV NUMBER,
 - DISTANCE NUMBER,
 - CONSTRAINT BUS_C_PK PRIMARY KEY(BUS_NO , CITY));

BASIC OPERATIONS



INSERT INTO PASSENGER

- INSERT INTO PASSENGER VALUES(1 , 'JIGNESH' , 'JINJALA' , 19 , 'M');
- INSERT INTO PASSENGER VALUES(2 , 'JITENDRA' , 'JAT' , 20 , 'M');
- INSERT INTO PASSENGER VALUES(3 , 'DHARMIK' , 'JINJALA' , 16 , 'M');
- INSERT INTO PASSENGER VALUES(4 , 'ROHIT' , 'BAKOLIYA' , 19 , 'M');
- INSERT INTO PASSENGER VALUES(5 , 'ABHI' , 'LUNAGARIYA' , 18 , 'M');
- SELECT * FROM PASSENGER

PASSENGER_ID	FIRST_NAME	LAST_NAME	AGE	GENDER
1	JIGNESH	JINJALA	19	M
2	JITENDRA	JAT	20	M
3	DHARMIK	JINJALA	16	M
4	ROHIT	PATEL	19	M
5	ABHI	LUNAGARIYA	18	M

INSERT INTO TICKET

- INSERT INTO TICKET VALUES(101 , 450);
- INSERT INTO TICKET VALUES(102 , 500);
- INSERT INTO TICKET VALUES(103 , 150);
- INSERT INTO TICKET VALUES(104 , 520);
- INSERT INTO TICKET VALUES(105 , 250);
- SELECT * FROM TICKET;

TICKET_NO	FARE
102	500
103	150
104	520
105	250
101	450

INSERT INTO TICKET_B

- INSERT INTO TICKET_B VALUES(1001 , 101 , DATE '2020-06-01' , 'SURAT' , 'BHAVNAGAR' , (CURRENT_TIMESTAMP) , (CURRENT_TIMESTAMP));
- INSERT INTO TICKET_B VALUES(1002 , 102 , DATE '2020-06-02' , 'JAMNAGAR' , 'MAHESANA' , (CURRENT_TIMESTAMP) , (CURRENT_TIMESTAMP));
- INSERT INTO TICKET_B VALUES(1003 , 103 , DATE '2020-06-03' , 'BHARUCH' , 'VALSAD' , (CURRENT_TIMESTAMP) , (CURRENT_TIMESTAMP));
- INSERT INTO TICKET_B VALUES(1004 , 104 , DATE '2020-06-04' , 'SURAT' , 'RAJKOT' , (CURRENT_TIMESTAMP) , (CURRENT_TIMESTAMP));
- INSERT INTO TICKET_B VALUES(1005 , 105 , DATE '2020-06-05' , 'BARODA' , 'AHMEDABAD' , (CURRENT_TIMESTAMP) , (CURRENT_TIMESTAMP));
- SELECT * FROM TICKET_B;

BUS_NO	TICKET_NO	DATE_B	BOARDING_CITY	DROP_CITY	START_J_TIME	FINISH_J_TIME
1001	101	01-JUN-20	SURAT	BHAVNAGAR	01-JUN-20 03.25.21.358707 AM	01-JUN-20 03.25.21.358707 AM
1002	102	02-JUN-20	JAMNAGAR	MAHESANA	01-JUN-20 03.25.21.363457 AM	01-JUN-20 03.25.21.363457 AM
1003	103	03-JUN-20	BHARUCH	VALSAD	01-JUN-20 03.25.21.367663 AM	01-JUN-20 03.25.21.367663 AM
1004	104	04-JUN-20	SURAT	RAJKOT	01-JUN-20 03.25.21.371652 AM	01-JUN-20 03.25.21.371652 AM
1005	105	05-JUN-20	BARODA	AHMEDABAD	01-JUN-20 03.25.21.375534 AM	01-JUN-20 03.25.21.375534 AM

INSERT INTO TICKET_P

- INSERT INTO TICKET_PVALUES(101 , 1);
- INSERT INTO TICKET_PVALUES(102 , 2);
- INSERT INTO TICKET_PVALUES(103 , 3);
- INSERT INTO TICKET_PVALUES(104 , 4);
- INSERT INTO TICKET_PVALUES(105 , 5);
- SELECT * FROM TICKET_P;

TICKET_NO	PASSENGER_ID
101	1
102	2
103	3
104	4
105	5

INSERT INTO BUS

- INSERT INTO BUS VALUES(5001 , 'AC' ,
CURRENT_TIMESTAMP , 45 , 6);
- INSERT INTO BUS VALUES(5002 , 'NON-AC' ,
CURRENT_TIMESTAMP , 40 , 8);
- INSERT INTO BUS VALUES(5003 , 'NON-AC' ,
CURRENT_TIMESTAMP , 50 , 5);
- INSERT INTO BUS VALUES(5004 , 'AC' ,
CURRENT_TIMESTAMP , 48 , 6);
- INSERT INTO BUS VALUES(5005 , 'AC' ,
CURRENT_TIMESTAMP , 45 , 7);
- SELECT * FROM BUS;

BUS_NO	TYPE	START_TIME	TOTAL_SEAT	RATE_PKM
5002	NON-AC	01-JUN-20	40	1.68
5003	NON-AC	01-JUN-20	50	.05
5001	AC	01-JUN-20	45	6
5004	AC	01-JUN-20	48	6
5005	AC	01-JUN-20	45	7

INSERT INTO BUS_C

- INSERT INTO BUS_CVALUES(5001 , 'SURAT' ,
CURRENT_TIMESTAMP , 20 , 450);
- INSERT INTO BUS_CVALUES(5002 , 'BHARUCH' ,
CURRENT_TIMESTAMP , 22 , 410);
- INSERT INTO BUS_CVALUES(5003 , 'AHMEDABAD' ,
CURRENT_TIMESTAMP , 4 , 250);
- INSERT INTO BUS_CVALUES(5004 , 'RAJKOT' ,
CURRENT_TIMESTAMP , 10 , 65);
- INSERT INTO BUS_CVALUES(5005 , 'BHAVNAGAR' ,
CURRENT_TIMESTAMP , 0 , 210);
- SELECT * FROM BUS_C;

BUS_NO	CITY	TIMEC	SEAT_AV	DISTANCE
5001	SURAT	01-JUN-20	20	450
5002	BHARUCH	01-JUN-20	22	410
5003	AHMEDABAD	01-JUN-20	4	250
5004	RAJKOT	01-JUN-20	10	65
5005	BHAVNAGAR	01-JUN-20	0	210

DELELTE FROM BUS

DELETE FROM BUS

WHERE TYPE = 'AC';

SELECT *

FROM BUS;

BUS_NO	TYPE	START_TIME	TOTAL_SEAT	RATE_PKM
5002	NON-AC	01-JUN-20	40	8
5003	NON-AC	01-JUN-20	50	5

MODIFY RATE_PKM IN BUS

- UPDATE BUS
- SET RATE_PKM = RATE_PKM +
10*RATE_PKM/100;
- SELECT *
- FROM BUS;

BUS_NO	TYPE	START_TIME	TOTAL_SEAT	RATE_PKM
5002	NON-AC	01-JUN-20	40	8.8
5003	NON-AC	01-JUN-20	50	5.5

UPDATE LAST_NAME IN PASSENGER

- UPDATE PASSENGER
- SET LAST_NAME = 'PATEL'
- WHERE PASSENGER_ID = 4;
- SELECT *
- FROM PASSENGER;

PASSENGER_ID	FIRST_NAME	LAST_NAME	AGE	GENDER
1	JIGNESH	JINJALA	19	M
2	JITENDRA	JAT	20	M
3	DHARMIK	JINJALA	16	M
4	ROHIT	PATEL	19	M
5	ABHI	LUNAGARIYA	18	M

NATURAL JOIN

- SELECT *
- FROM PASSENGER NATURAL JOIN
TICKET_P NATURAL JOIN TICKET;

TICKET_NO	PASSENGER_ID	FIRST_NAME	LAST_NAME	AGE	GENDER	FARE
101	1	JIGNESH	JINJALA	19	M	450
102	2	JITENDRA	JAT	20	M	500
103	3	DHARMIK	JINJALA	16	M	150
104	4	ROHIT	PATEL	19	M	520
105	5	ABHI	LUNAGARIYA	18	M	250

CURSOR

```
DECLARE
C_PASSENGER_ID PASSENGER.PASSENGER_ID%TYPE;
C_FIRST_NAME PASSENGER.FIRST_NAME%TYPE;
C_LAST_NAME PASSENGER.LAST_NAME%TYPE;
C_AGE PASSENGER.AGE%TYPE;
C_GENDER PASSENGER.GENDER%TYPE;

CURSOR C_PASSENGER IS
    SELECT PASSENGER_ID , FIRST_NAME , LAST_NAME , AGE , GENDER FROM PASSENGER;

BEGIN
    OPEN C_PASSENGER;
    LOOP
        FETCH C_PASSENGER INTO C_PASSENGER_ID , C_FIRST_NAME , C_LAST_NAME , C_AGE , C_GENDER;
        EXIT WHEN C_PASSENGER%NOTFOUND;
        IF (C_AGE > 18)
        THEN
            DBMS_OUTPUT.PUT_LINE(C_PASSENGER_ID || ' ' || C_FIRST_NAME || ' ' ||
            C_LAST_NAME || ' ' || C_AGE || ' ' || C_GENDER);
        END IF;
    END LOOP;
    CLOSE C_PASSENGER;
END;
```

Statement processed.

```
1 JIGNESH JINJALA 19 M
2 JITENDRA JAT 20 M
4 ROHIT PATEL 19 M
```


TRIGGER

```
1 CREATE OR REPLACE TRIGGER PRICE_CHANGE
2 AFTER DELETE OR INSERT OR UPDATE ON BUS
3 FOR EACH ROW
4
5 DECLARE
6 PRICE_CHANGE NUMBER;
7 BEGIN
8 PRICE_CHANGE := :OLD.RATE_PKM - :NEW.RATE_PKM;
9 DBMS_OUTPUT.PUT_LINE('OLD VALUE IS ' || :OLD.RATE_PKM || CHR(10) || 'NEW VALUE IS ' || :NEW.RATE_PKM);
10 DBMS_OUTPUT.PUT_LINE(PRICE_CHANGE);
11 END;
12 /
13
14 UPDATE BUS
15 SET RATE_PKM = RATE_PKM - 2
16 WHERE RATE_PKM > 2;
17
18 SELECT *
19 FROM BUS;
```

Trigger created.

2 row(s) updated.
OLD VALUE IS 8.8
NEW VALUE IS 6.8
2
OLD VALUE IS 5.5
NEW VALUE IS 3.5
2

BUS_NO	TYPE	START_TIME	TOTAL_SEAT	RATE_PKM
5002	NON-AC	01-JUN-20	40	6.8
5003	NON-AC	01-JUN-20	50	3.5

PROCEDURE

```
1 CREATE OR REPLACE PROCEDURE FIND_PASSENGER(ID1 IN NUMBER)
2 IS
3 NAME1 VARCHAR(20);
4
5 BEGIN
6 SELECT FIRST_NAME INTO NAME1 FROM PASSENGER WHERE PASSENGER_ID = ID1;
7 DBMS_OUTPUT.PUT_LINE('THE PASSENGER NAME IS ' || NAME1 || ' WHOSE ID IS ' || ID1);
8 END;
9 /
10
11 BEGIN
12 FIND_PASSENGER(1);
13
14 END;
15 /
```

Procedure created.

Statement processed.

THE PASSENGER NAME IS JIGNESH WHOSE ID IS 1

THE END

