

A Mini-Project Report
on
Railway Ticket Reservation System

Submitted for Course
Database Management System

by

| | |
|--------------------------|--------------------|
| Yash Kathe | Roll No. 06 |
| Kaushik Khadilkar | Roll No. 07 |
| Chirag Kunder | Roll No. 08 |
| Tejas Tawde | Roll No. 15 |

TE EXTC A

Under the guidance of
Prof. Santosh Chapaneri

Department of Electronics and Telecommunication Engineering
St. Francis Institute of Technology

2021-2022

Contents

| | | |
|----------|---|----------|
| 1 | Storyline | 2 |
| 1.1 | Requirements | 2 |
| 1.2 | Assumptions | 2 |
| 2 | Entity Relationship Diagram | 3 |
| 2.1 | Entities | 3 |
| 2.2 | Relationships | 3 |
| 2.3 | Entity-Relationship Diagram | 3 |
| 3 | Relational Model | 5 |
| 3.1 | Rules for conversion from ERD to relational model | 5 |
| 3.2 | Relational Model | 5 |
| 4 | SQL Queries | 6 |

List of Figures

| | | |
|------|--|----|
| 2.1 | Entity-Relationship Diagram for the Database | 4 |
| 4.1 | Creating Tables | 7 |
| 4.2 | Data Insertion | 7 |
| 4.3 | Query 1 | 8 |
| 4.4 | Query 2 | 8 |
| 4.5 | Query 3 | 8 |
| 4.6 | Query 4 | 9 |
| 4.7 | Query 5 | 9 |
| 4.8 | Query 6 | 9 |
| 4.9 | Query 7 | 10 |
| 4.10 | Query 8 | 10 |
| 4.11 | Query 9 | 10 |
| 4.12 | Query 10 | 11 |
| 4.13 | Query 11 - Before Execution | 11 |
| 4.14 | Query 11 - After Execution | 11 |
| 4.15 | Query 12 | 12 |

Chapter 1

Storyline

This section should describe the requirements for the chosen database topic. Form a storyline and describe in detail.

1.1 Requirements

List the requirements of your topic here.

1.2 Assumptions

List the assumptions that you make for your database design here.

Chapter 2

Entity Relationship Diagram

2.1 Entities

The entities for our database are as follows:

- Passenger(PNR, first_name, last_name, age, gender, phone_num, reserve_status)
- Station(Stn_code, Stn_name, Stn_type)
- Train(Train_num, train_src, train_dstn, coach_qty)
- Ticket(Ticket-num, arrivalTime, departureTime, ticket_type)
- Fare(Receipt-num, Amount)
- Class (Class-num, Seat_qty)

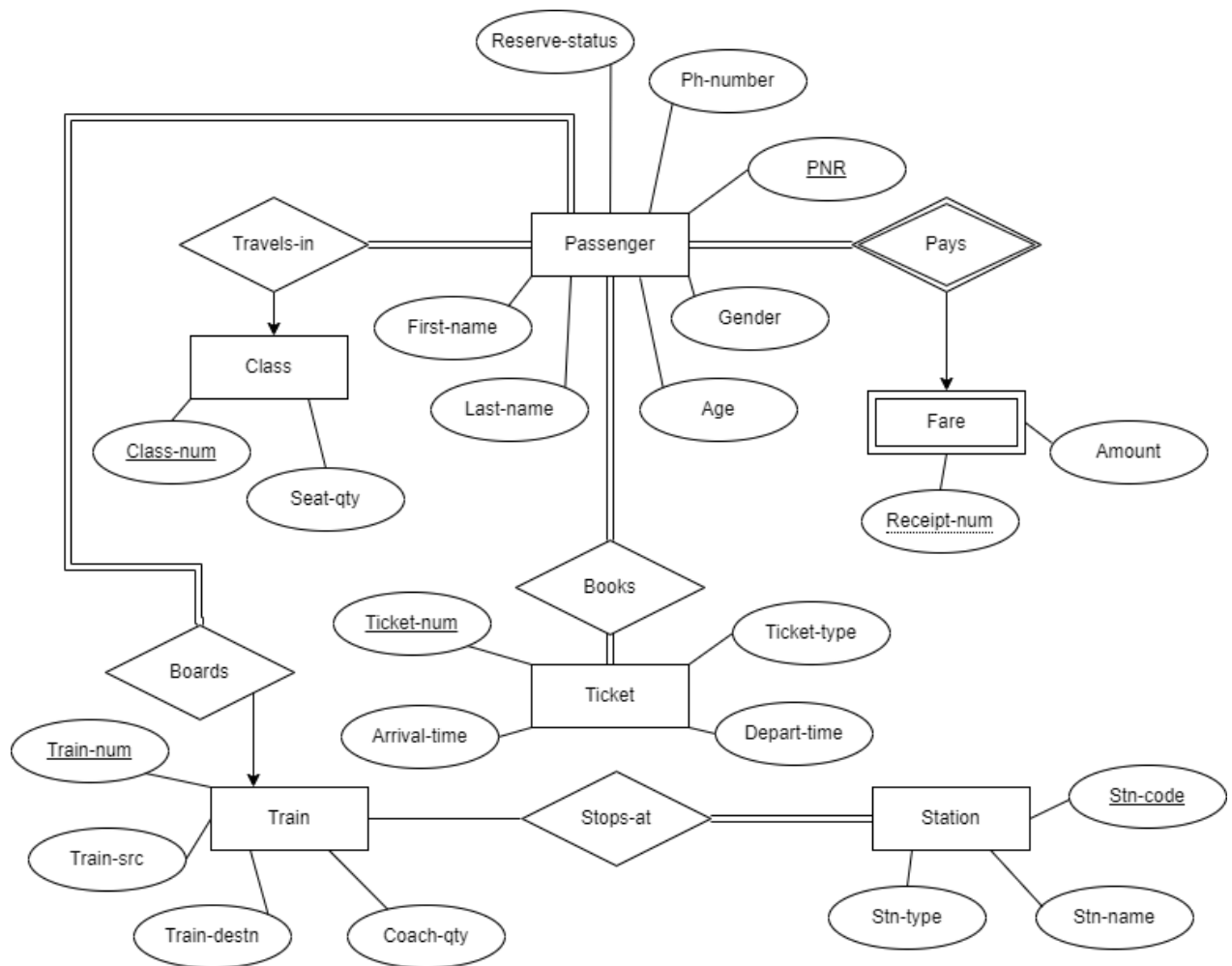
2.2 Relationships

The relationships in the database are as follows:

- Books: N:N Passenger to Ticket; Passenger total, Ticket total participation
- Stops-at: N:N Train to Station; Train partial, Station total participation
- Pays: N:1 Passenger to Fare; Passenger partial, Fare total participation
- Boards: N:1 Passenger to Train; Passenger partial, Train total participation
- Travels-in: N:1 Passenger to Class; Passenger partial, class total participation

2.3 Entity-Relationship Diagram

Following is the Entity-Relationship Diagram for the DB:

**Figure 2.1:** Entity-Relationship Diagram for the Database

Chapter 3

Relational Model

3.1 Rules for conversion from ERD to relational model

1. **Strong Entity Set:** Make a new table with same attributes.
2. **Composite Attribute:** Flatten the attribute.
3. **Multi-valued Attribute:** For an entity 'E' with a multi-valued attribute 'M', make a new table 'EM' with attributes as primary key of 'E' followed by that of 'M'.
4. **Weak Entity Set:** Make a new table for weak entity with all its attribute and the primary key of the strong entity it is dependent on.
5. **Many-to-Many Relationship:** Make a new table with the attributes as primary keys of both entities having N:N relationship.
6. **Many-to-One/One-to-Many Relationship:** Instead of making a new table, add the primary key of the one-side entity to the attributes of the many-side entity.
7. **One-to-One Relationship:** Instead of making a new table, consider any one entity to be on the many-side and apply Rule 6.

3.2 Relational Model

Following is the Relational Model of the Database after applying the above rules:

- Passenger(PNR, first_name, last_name, age, gender, phone_num, reserve_status, train_num*, class_num*)
- Station(Stn_code, Stn_name, Stn_type)
- Train(Train_num, train_src, train_dstn, coach_qty)
- Ticket(Ticket-num, Arrival-time, depart-time, ticket-type)
- Fare(Receipt-num, Amount, PNR*)
- Class (Class-num, Seat_qty)
- Books (PNR*, ticket_num)
- Stops_at (train_num*, Stn_code*)

Chapter 4

SQL Queries

This chapter includes the screenshots of:

1. Database Creation SQL Code
2. Data Insertion into the Database
3. SQL Queries run on the Database


```

Host: 127.0.0.1 Database: railway_reservation reservationdb.sql reservationdata.sql Queries.sql
1 CREATE TABLE CLASS
2 (
3 class_num INT NOT NULL PRIMARY KEY,
4 seat_qty INT NOT NULL
5 );
6
7 CREATE TABLE STATION
8 (
9 stn_code VARCHAR(6) NOT NULL PRIMARY KEY,
10 stn_name VARCHAR(50) NOT NULL,
11 stn_type VARCHAR(10) NOT NULL #Station type: Terminus, Junction, Central, Station etc.
12 );
13
14 CREATE TABLE TRAIN
15 (
16 train_num NUMERIC(5) NOT NULL PRIMARY KEY,
17 train_name VARCHAR(50) NOT NULL,
18 train_src VARCHAR(50) NOT NULL,
19 train_dstn VARCHAR(50) NOT NULL,
20 coach_qty INT NOT NULL
21 );
22
23 CREATE TABLE TICKET
24 (
25 ticket_num NUMERIC(6) NOT NULL PRIMARY KEY,
26 arrivalTime NUMERIC(4),
27 departureTime NUMERIC(4),
28 ticket_type VARCHAR(10)
29 );
30
31 CREATE TABLE PASSENGER
32 (
33 PNR VARCHAR(20) NOT NULL PRIMARY KEY,
34 first_name VARCHAR(15) NOT NULL,
35 last_name VARCHAR(15),
36 age INT NOT NULL,
37 gender VARCHAR(5),
38 phone_num VARCHAR(10) NOT NULL,
39 reserve_status VARCHAR(10) NOT NULL,
40 train_num NUMERIC(5),
41 class_num INT,
42 CONSTRAINT FK_PS_TRAIN FOREIGN KEY (train_num) REFERENCES TRAIN(train_num),
43 CONSTRAINT FK_PS_CLASS FOREIGN KEY (class_num) REFERENCES CLASS(class_num)
44 );
45
46 CREATE TABLE FARE
47 (
48 receipt_num NUMERIC(5) NOT NULL PRIMARY KEY,
49 amount NUMERIC,
50 PNR VARCHAR(20),
51 CONSTRAINT FK_FARE_PS FOREIGN KEY (PNR) REFERENCES PASSENGER(PNR)
52 );

```

Figure 4.1: Creating Tables

```

Host: 127.0.0.1 Database: railway_reservation reservationdb.sql reservationdata.sql Queries.sql
1 INSERT INTO CLASS VALUES (1, 178);
2 INSERT INTO CLASS VALUES (2, 280);
3
4 INSERT INTO STATION VALUES ('BVI', 'BORIVALI', 'STATION');
5 INSERT INTO STATION VALUES ('CSMT', 'MUMBAI CSMT', 'TERMINUS');
6 INSERT INTO STATION VALUES ('DR', 'DADAR (CENTRAL)', 'STATION');
7 INSERT INTO STATION VALUES ('DDR', 'DADAR (WESTERN)', 'STATION');
8 INSERT INTO STATION VALUES ('ADH', 'ANDHERI', 'STATION');
9 INSERT INTO STATION VALUES ('TNA', 'THANE', 'STATION');
10 INSERT INTO STATION VALUES ('RM', 'RATNAGIRI', 'STATION');
11 INSERT INTO STATION VALUES ('PUNE', 'PUNE JUNCTION', 'JUNCTION');
12 INSERT INTO STATION VALUES ('BSL', 'BHUSAVAL JUNCTION', 'JUNCTION');
13 INSERT INTO STATION VALUES ('MK', 'MASHIKI', 'STATION');
14 INSERT INTO STATION VALUES ('AGC', 'AGRA CANTT', 'STATION');
15 INSERT INTO STATION VALUES ('BOTS', 'BANDRA TERMINUS', 'TERMINUS');
16 INSERT INTO STATION VALUES ('LTT', 'LOKMANYA TILAK TERMINUS', 'TERMINUS');
17 INSERT INTO STATION VALUES ('MWC', 'MUMBAI CENTRAL', 'CENTRAL');
18 INSERT INTO STATION VALUES ('CNB', 'KANPUR CENTRAL', 'CENTRAL');
19 INSERT INTO STATION VALUES ('BNC', 'BANGLORE CANTT', 'STATION');
20 INSERT INTO STATION VALUES ('NDLS', 'NEW DELHI', 'STATION');
21 INSERT INTO STATION VALUES ('RAJP', 'RAJAPUR', 'STATION');
22 INSERT INTO STATION VALUES ('ST', 'SURAT', 'STATION');
23 INSERT INTO STATION VALUES ('KKW', 'KANKAVALI', 'STATION');
24 INSERT INTO STATION VALUES ('SUR', 'SOLAPUR', 'JUNCTION');
25 INSERT INTO STATION VALUES ('KOP', 'CSMT KOLHAPUR', 'TERMINUS');
26 INSERT INTO STATION VALUES ('SND', 'SINDHODURG', 'STATION');
27 INSERT INTO STATION VALUES ('SGR', 'SANGAMESHWAR ROAD', 'STATION');
28 INSERT INTO STATION VALUES ('DIVA', 'DIVA JUNCTION', 'JUNCTION');
29 INSERT INTO STATION VALUES ('SWV', 'SAWANTWADI ROAD', 'STATION');
30 INSERT INTO STATION VALUES ('NED', 'HAZUR SAHIB NANDED', 'STATION');
31 INSERT INTO STATION VALUES ('NZM', 'HAZRAT NIZAMUDDIN', 'TERMINUS');
32
33
34 INSERT INTO TRAIN VALUES (10111, 'KONKAN KANYA EXPRESS', 'MUMBAI CSMT', 'MADGAON', 22);
35 INSERT INTO TRAIN VALUES (10103, 'MANDOVI EXPRESS', 'MUMBAI CSMT', 'MADGAON', 22);
36 INSERT INTO TRAIN VALUES (01241, 'LTT GKP FEST SPECIAL', 'LTT', 'GKP', 18);
37 INSERT INTO TRAIN VALUES (12137, 'PUNJAB MAIL', 'MUMBAI CSMT', 'FIROZPUR CANTT', 19);
38 INSERT INTO TRAIN VALUES (17411, 'MAHALAXMI EXPRESS', 'MUMBAI CSMT', 'CSMT KOLHAPUR', 23);
39 INSERT INTO TRAIN VALUES (22119, 'TEJAS EXPRESS', 'MUMBAI CSMT', 'KARNALI', 15);
40 INSERT INTO TRAIN VALUES (22159, 'CHENNAI EXPRESS', 'MUMBAI CSMT', 'CHENNAI', 19);
41 INSERT INTO TRAIN VALUES (22177, 'MAHANAGRI EXPRESS', 'MUMBAI CSMT', 'VARANASI', 24);
42 INSERT INTO TRAIN VALUES (50103, 'DIVA RATNAGIRI PASS', 'DIVA JUNCTION', 'RATNAGIRI', 24);
43 INSERT INTO TRAIN VALUES (11007, 'DECCAN EXPRESS', 'MUMBAI CSMT', 'PUNE JUNCTION', 17);
44 INSERT INTO TRAIN VALUES (11010, 'SINHAGAD EXPRESS', 'PUNE JUNCTION', 'MUMBAI CSMT', 18);
45 INSERT INTO TRAIN VALUES (11003, 'TUTARI EXPRESS', 'DADAR', 'SAWANTWADI ROAD', 15);
46 INSERT INTO TRAIN VALUES (11020, 'KONARK EXPRESS', 'BHUBANESHWAR', 'MUMBAI CSMT', 23);
47 INSERT INTO TRAIN VALUES (12289, 'NGP DURGAM EXPRESS', 'MUMBAI CSMT', 'NAGPUR', 23);
48 INSERT INTO TRAIN VALUES (22221, 'NZM RAJDHANI EXPRESS', 'MUMBAI CSMT', 'HAZRAT NIZAMUDDIN', 24);
49 INSERT INTO TRAIN VALUES (22209, 'NDLS DURGAM EXPRESS', 'MUMBAI CENTRAL', 'NEW DELHI', 23);
50 INSERT INTO TRAIN VALUES (22927, 'LOKSHAKTI EXPRESS', 'BANDRA TERMINUS', 'AHMEDABAD JUNCTION', 23);
51
52 INSERT INTO TICKET VALUES (245940, 2317, 0634, 'ADULT');

```

Figure 4.2: Data Insertion

Host: 127.0.0.1 Database: railway_reservation Table: train Data reservationdb.sql reservationdata.sql Query #3*

```

1 #Q.1) Find the names of all trains originating from Mumbai CSMT
2
3 SELECT train_name FROM train
4 WHERE train_src='MUMBAI CSMT';

```

train (10x1c)

| train_name |
|----------------------|
| MANDOVI EXPRESS |
| KONKAN KANYA EXPRESS |
| DECCAN EXPRESS |
| PUNJAB MAIL |
| NGP DURONTO EXPRESS |
| MAHALAXMI EXPRESS |
| TEJAS EXPRESS |
| CHENNAI EXPRESS |
| MAHANAGRI EXPRESS |
| NZM RAJDHANI EXPRESS |

Figure 4.3: Query 1

Host: 127.0.0.1 Database: railway_reservation Table: passenger Data reservationdb.sql reservationdata.sql Query #3*

```

6
7 #Q.2) Display the details of stations having 'Road' in their name
8 SELECT * FROM station
9 WHERE stn_name LIKE '%ROAD';
10

```

station (3x3c)

| stn_code | stn_name | stn_type |
|----------|-------------------|----------|
| RAJP | RAJAPUR ROAD | STATION |
| SGR | SANGAMESHWAR ROAD | STATION |
| SWV | SAWANTWADI ROAD | STATION |

Figure 4.4: Query 2

Host: 127.0.0.1 Database: railway_reservation reservationdb.sql reservationdata.sql Query #3*

```

16 #Q.3) Find the names of unique trains by which the passengers are travelling
17 SELECT DISTINCT(t.train_name), t.train_num
18 FROM train AS t, passenger AS p
19 WHERE t.train_num=p.train_num;
20

```

train (10x2c)

| train_name | train_num |
|----------------------|-----------|
| LTT GKP FEST SPECIAL | 1,241 |
| MANDOVI EXPRESS | 10,103 |
| KONKAN KANYA EXPRESS | 10,111 |
| TUTARI EXPRESS | 11,003 |
| DECCAN EXPRESS | 11,007 |
| KONARK EXPRESS | 11,020 |
| MAHALAXMI EXPRESS | 17,411 |
| TEJAS EXPRESS | 22,119 |
| CHENNAI EXPRESS | 22,159 |
| DIVA RATNAGIRI PASS | 50,103 |

Figure 4.5: Query 3

Host: 127.0.0.1 Database: railway_reservation reservationdb.sql reservationdata.sql Query #3*

```

23 #Q.4) Display the details of all trains in the alphabetical order of their names
24 SELECT * FROM train
25 ORDER BY train.train_name
26
27
28
29
30

```

| train_num | train_name | train_src | train_dstn | coach_qty |
|-----------|----------------------|-----------------|--------------------|-----------|
| 22,159 | CHENNAI EXPRESS | MUMBAI CSMT | CHENNAI | 19 |
| 11,007 | DECCAN EXPRESS | MUMBAI CSMT | PUNE JUNCTION | 17 |
| 50,103 | DIVA RATNAGIRI PASS | DIVA JUNCTION | RATNAGIRI | 24 |
| 11,020 | KONARK EXPRESS | BHUBANESHWAR | MUMBAI CSMT | 23 |
| 10,111 | KONKAN KANYA EXPRESS | MUMBAI CSMT | MADGAON | 22 |
| 22,927 | LOKSHAKTI EXPRESS | BANDRA TERMINUS | AHMEDABAD JUNCTION | 23 |
| 1,241 | LTT GKP FEST SPECIAL | LTT | GKP | 18 |
| 17,411 | MAHALAXMI EXPRESS | MUMBAI CSMT | CSMT KOLHAPUR | 23 |
| 22,177 | MAHANAGRI EXPRESS | MUMBAI CSMT | VARANASI | 24 |
| 10,103 | MANDOWI EXPRESS | MUMBAI CSMT | MADGAON | 22 |
| 22,209 | NDLS DURONTO EXPRESS | MUMBAI CENTRAL | NEW DELHI | 23 |
| 12,289 | NGP DURONTO EXPRESS | MUMBAI CSMT | NAGPUR | 23 |
| 22,221 | NZM RAJDHANI EXPRESS | MUMBAI CSMT | HAZRAT NIZAMUDDIN | 24 |
| 12,137 | PUNJAB MAIL | MUMBAI CSMT | FIROZPUR CANTT | 19 |
| 11,010 | SINHAGAD EXPRESS | PUNE JUNCTION | MUMBAI CSMT | 18 |
| 22,119 | TEJAS EXPRESS | MUMBAI CSMT | KARMALI | 15 |
| 11,003 | TUTARI EXPRESS | DADAR | SAWANTWADI ROAD | 15 |

Figure 4.6: Query 4

Host: 127.0.0.1 Database: railway_reservation reservationdb.sql reservationdata.sql Query #3*

```

20 #Q.5) Find the maximum fare paid by any passenger
21 SELECT MAX(f.amount)
22 FROM fare AS f
23
24
25

```

| MAX(f.amount) |
|---------------|
| 2,300 |

Figure 4.7: Query 5

Host: 127.0.0.1 Database: railway_reservation reservationdb.sql reservationdata.sql Query #3*

```

24 #Q.6) Find the total number of stations which are junctions
25 SELECT COUNT(stn_code)
26 FROM station
27 WHERE stn_type='JUNCTION'
28
29

```

| COUNT(stn_code) |
|-----------------|
| 4 |

Figure 4.8: Query 6

Host: 127.0.0.1 Database: railway_reservation reservationdb.sql reservationdata.sql Query #3*

```

28 #Q.7) Find the number of trains originating from each station
29 SELECT t.train_src, COUNT(DISTINCT(t.train_num))
30 FROM train AS t
31 GROUP BY t.train_src
32
33

```

train (8r x 2c)

| train_src | COUNT(DISTINCT(t.train_num)) |
|-----------------|------------------------------|
| BANDRA TERMINUS | 1 |
| BHUBANESHWAR | 1 |
| DADAR | 1 |
| DIVA JUNCTION | 1 |
| LTT | 1 |
| MUMBAI CENTRAL | 1 |
| MUMBAI CSMT | 10 |
| PUNE JUNCTION | 1 |

Figure 4.9: Query 7

Host: 127.0.0.1 Database: railway_reservation reservationdb.sql reservationdata.sql Query #3*

```

33 #Q.8) Find the names of passengers and trains by which they are travelling
34 SELECT p.first_name, p.last_name, t.train_num, t.train_name
35 FROM passenger AS p, train AS t
36 WHERE t.train_num=p.train_num
37
38

```

Result #1 (15r x 4c)

| first_name | last_name | train_num | train_name |
|------------|-----------|-----------|----------------------|
| Maddy | Kunder | 17,411 | MAHALAXMI EXPRESS |
| Manav | Dhruve | 22,119 | TEJAS EXPRESS |
| Monish | DCosta | 11,003 | TUTARI EXPRESS |
| Manish | Ghosh | 22,119 | TEJAS EXPRESS |
| Tejas | Tawde | 10,111 | KONKAN KANYA EXPRESS |
| Sheru | Kunder | 11,007 | DECCAN EXPRESS |
| Ramani | Desai | 22,159 | CHENNAI EXPRESS |
| Yash | Kathe | 10,103 | MANDOVI EXPRESS |
| Sanika | Gawas | 10,111 | KONKAN KANYA EXPRESS |
| Trijal | Prjapati | 11,020 | KONARK EXPRESS |
| Anant | Shukla | 1,241 | LTT GKP FEST SPECIAL |
| Vinayak | Surve | 10,111 | KONKAN KANYA EXPRESS |
| Chirag | Kundre | 11,003 | TUTARI EXPRESS |
| Kaushik | Nalavde | 50,103 | DIVA RATNAGIRI PASS |
| Vaishnavi | Puthran | 10,111 | KONKAN KANYA EXPRESS |

Figure 4.10: Query 8

Host: 127.0.0.1 Database: railway_reservation reservationdb.sql reservationdata.sql Query #3*

```

39 #Q.9) Find the names of passengers travelling from 'Konkan Kanya Express'
40 SELECT p.first_name, p.last_name
41 FROM passenger AS p
42 WHERE p.train_num IN (SELECT t.train_num FROM train AS t
43 WHERE t.train_name='KONKAN KANYA EXPRESS');
44

```

passenger (4r x 2c)

| first_name | last_name |
|------------|-----------|
| Tejas | Tawde |
| Sanika | Gawas |
| Vinayak | Surve |
| Vaishnavi | Puthran |

Figure 4.11: Query 9

Host: 127.0.0.1 Database: railway_reservation reservationdb.sql reservationdata.sql Query #3*

```

45 #Q.10) Find the names of passengers who have paid a fare of more than Rs. 1500
46 SELECT p.first_name, p.last_name
47 FROM passenger AS p
48 WHERE p.PNR IN (SELECT f.PNR FROM fare AS f
49                 WHERE f.amount>1500);
50

```

passenger (4r x 2c)

| first_name | last_name |
|------------|-----------|
| Vaishnavi | Puthran |
| Manish | Ghosh |
| Vinayak | Surve |
| Maddy | Kunder |

Figure 4.12: Query 10

```

51 #Q.11) Update the reservation status of passengers having 'WAITING' to 'RAC'
52 SELECT first_name, reserve_status FROM passenger
53 UPDATE passenger
54 SET reserve_status='RAC'
55 WHERE reserve_status='WAITING'
56

```

passenger (15r x 2c)

| first_name | reserve_status |
|------------|----------------|
| Maddy | RAC |
| Manav | CONFIRM |
| Monish | WAITING |
| Manish | CONFIRM |
| Tejas | WAITING |
| Sheru | WAITING |
| Ramani | CONFIRM |
| Yash | CONFIRM |
| Sanika | CONFIRM |
| Trijal | WAITING |
| Anant | RAC |
| Vinayak | CONFIRM |
| Chirag | CONFIRM |
| Kaushik | CONFIRM |
| Vaishnavi | CONFIRM |

Figure 4.13: Query 11 - Before Execution

Host: 127.0.0.1 Database: railway_reservation reservationdb.sql reservationdata.sql Query #3*

```

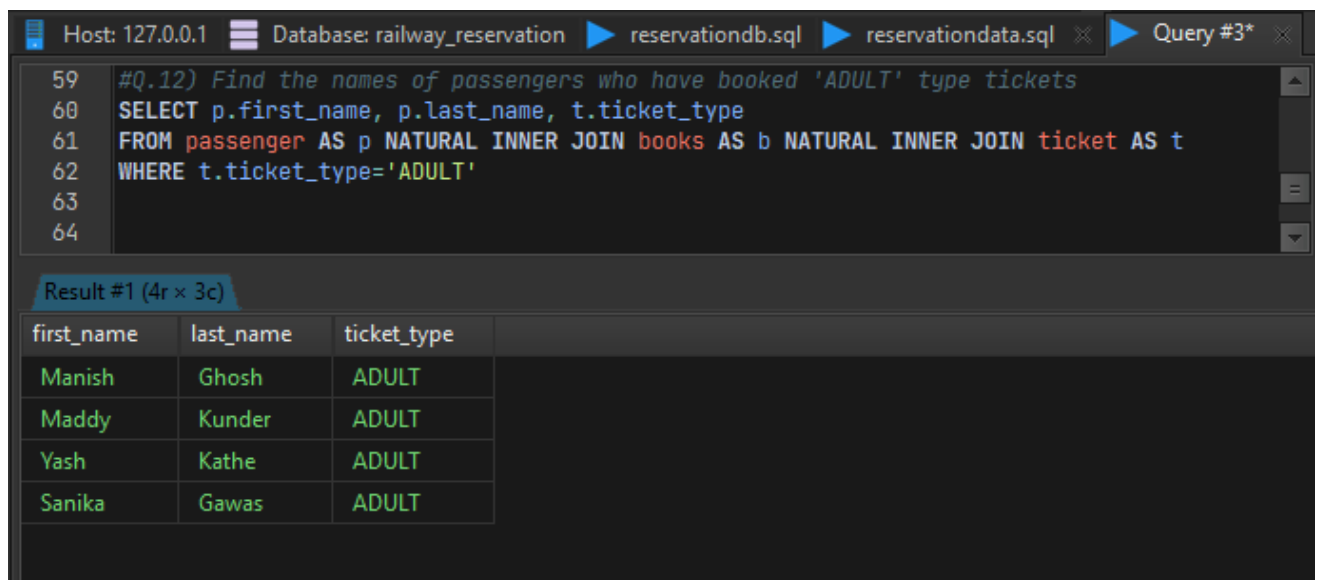
51 #Q.11) Update the reservation status of passengers having 'WAITING' to 'RAC'
52 SELECT first_name, reserve_status FROM passenger
53 UPDATE passenger
54 SET reserve_status='RAC'
55 WHERE reserve_status='WAITING'
56

```

passenger (15r x 2c)

| first_name | reserve_status |
|------------|----------------|
| Maddy | RAC |
| Manav | CONFIRM |
| Monish | RAC |
| Manish | CONFIRM |
| Tejas | RAC |
| Sheru | RAC |
| Ramani | CONFIRM |
| Yash | CONFIRM |
| Sanika | CONFIRM |
| Trijal | RAC |
| Anant | RAC |
| Vinayak | CONFIRM |
| Chirag | CONFIRM |
| Kaushik | CONFIRM |
| Vaishnavi | CONFIRM |

Figure 4.14: Query 11 - After Execution



The screenshot shows a SQL query execution window. The top bar indicates the host is 127.0.0.1, the database is railway_reservation, and the current query is Query #3*. The query text is as follows:

```
59 #Q.12) Find the names of passengers who have booked 'ADULT' type tickets
60 SELECT p.first_name, p.last_name, t.ticket_type
61 FROM passenger AS p NATURAL INNER JOIN books AS b NATURAL INNER JOIN ticket AS t
62 WHERE t.ticket_type='ADULT'
63
64
```

Below the query, the results are displayed as a table with the title "Result #1 (4r x 3c)". The table has three columns: first_name, last_name, and ticket_type. The data rows are as follows:

| first_name | last_name | ticket_type |
|------------|-----------|-------------|
| Manish | Ghosh | ADULT |
| Maddy | Kunder | ADULT |
| Yash | Kathe | ADULT |
| Sanika | Gawas | ADULT |

Figure 4.15: Query 12