AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

408/1, Kuratoli, Khilkhet, Dhaka 1229, Bangladesh

INTRODUCTION TO DATABASE

PROJECT REPORT

Project Title	:	Railway Management System
Date of submission	:	28th August, 2022
Course Teacher	:	Methila Farzana Woishe
Section	:	G
Semester	:	Summer [2022 - 2023]

Members:

No.	Name	ID	Contribution
01	MD. Istiak Ahamed	22-48998-3	Table Creation
02	Kayjer Islam	22-49005-3	Introduction, Scenario & Table Creation
03	Salman Sayeed	22-49006-3	ER DIagram, Data Insertion & Conclusion
04	MD. Rakib Hasan	22-49029-3	Normalization & Optimization

Table Of Contents

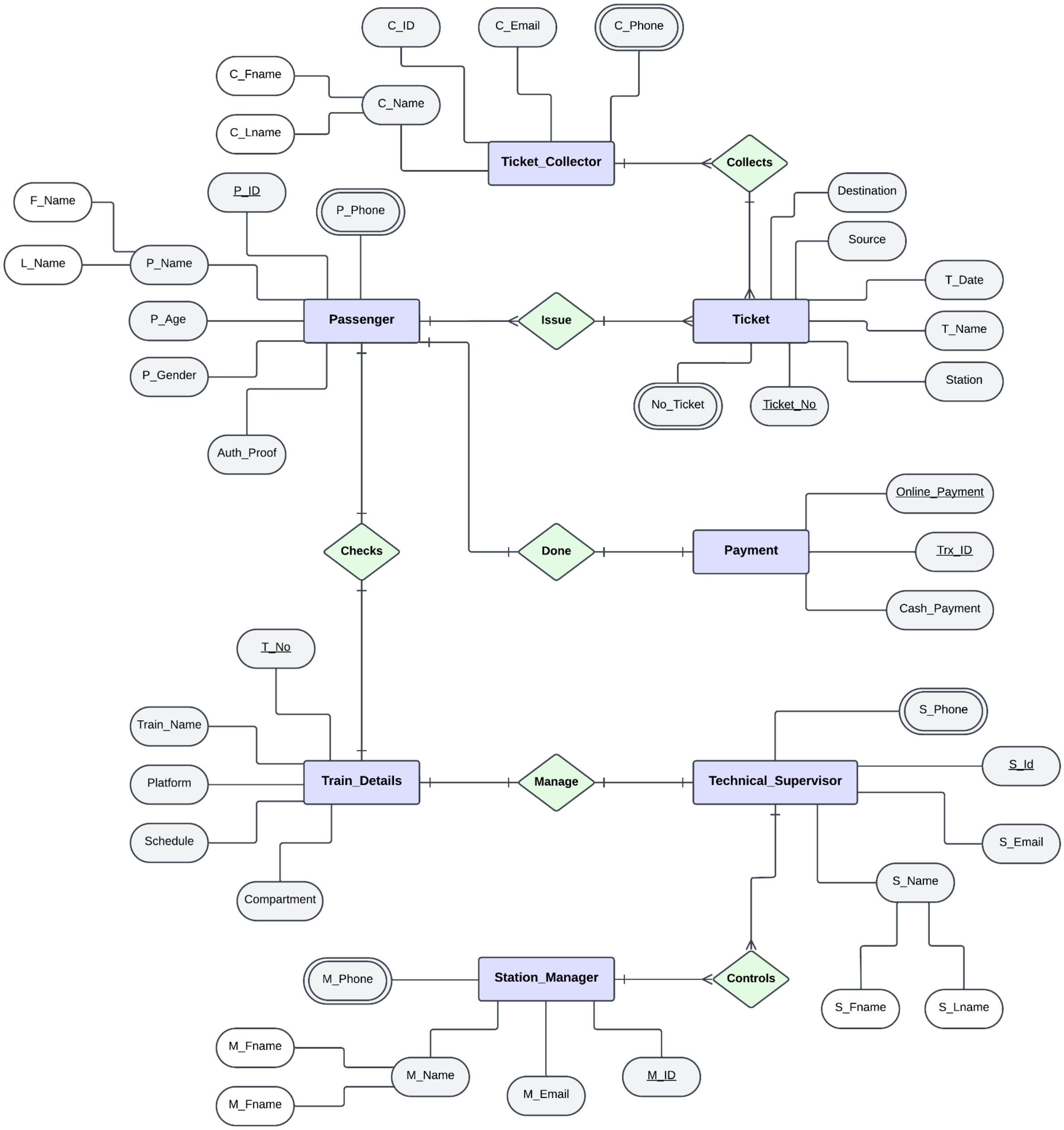
Introduction

Our team is in charge of guiding MetroControl, a state of the art railway management system, to implementation. A complete framework created to enable a variety of operations inside the metro rail domain is represented by the MetroControl Railway Management System. The seven distinct entities that make up this railway system are Passenger, Train_Details, Ticket, Payment, Technical_Supervisor, Ticket_Collector, and Station_Manager. Six relationships are used to construct the links between these entities, providing the basic framework for data management. The construction of data tables is predicated on both the attributes inherent to each entity and their relational dependencies.

Scenario

Within the **MetroControl Railway Management System**, a PASSENGER has the capability to access multiple TRAIN_DETAILS. PASSENGERS are discerned through a set of attributes, encompassing P_ID, P_Phone, P_Name, P_Age, P_Gender, and Auth_Proof. In parallel, TRAIN_DETAILS are uniquely identified by their attributes, namely Train_No, Train_Name, Schedule, Compartment, and Platform. These TRAIN_DETAILS are comprehensively stored within the system's database. Simultaneously, a PASSENGER can be associated with multiple TICKETS, each identified by distinct attributes including Ticket_No, No_Passenger, Source, Destination, Ticket_Name, T_Date, and Station. A PAYMENT is performed by a singular PASSENGER, with its identification hinging on attributes such as Trx_ID, Cash_Payment, or Online_Payment. Facilitating the system's operations, a TECHNICAL_SUPERVISOR assumes the responsibility of managing several TRAIN_DETAILS. The TECHNICAL_SUPERVISOR's identification is constituted by attributes such as S_ID, S_Name, S_Phone, and S_Email. Correspondingly, a TICKET_COLLECTOR undertakes the task of collecting multiple TICKETS, characterized by attributes like Collector_ID, C_Name, C_Phone, and C_Email.A MANAGER in the hierarchical structure is in charge of several TECHNICAL_SUPERVISORS.
The MANAGER is distinguished through attributes encompassing M_ID, M_Name, M_Phone, and M_Email. This intricate interplay of entities, their attributes, and relationships underscores the efficient and comprehensive functioning of the MetroControl Railway Management System.

ER Diagram


C_ID
C_Email
C_Phone
C_Fname
C_Name
C_Lname
Ticket_Collector
Collects
Destination
Source
T_Date
T_Name
Station
Ticket
No_Ticket
Ticket_No
F_Name
L_Name
P_ID
P_Phone
P_Name
P_Age
P_Gender
Auth_Proof
Passenger
Issue
Online_Payment
Trx_ID
Cash_Payment
Payment
Done
Checks
T_No
Train_Name
Platform
Schedule
Compartment
Train_Details
Manage
Technical_Supervisor
S_Phone
S_Id
S_Email
S_Name
S_Fname
S_Lname
Controls
Station_Manager
M_Phone
M_Fname
M_Name
M_Fname
M_Email
M_ID

Normalization

Indicators:

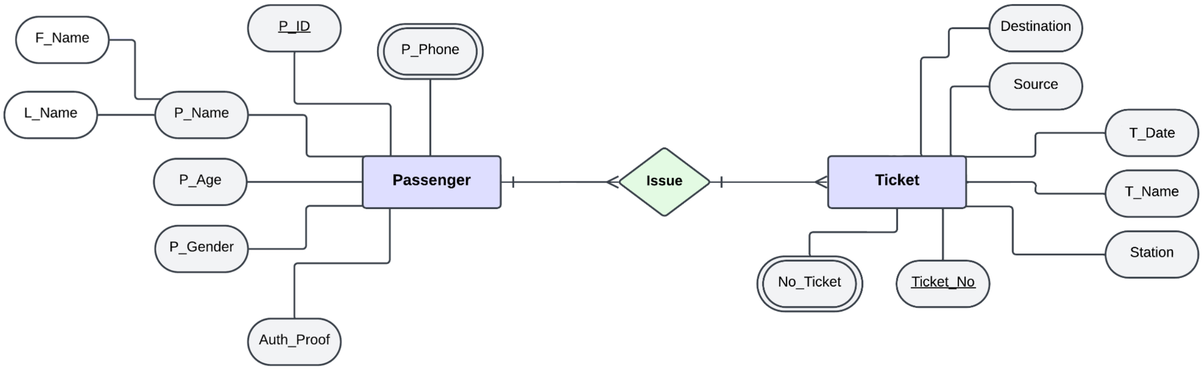
🟦 Primary Key

🟪 Secondary Key

—┼ One

—< Many

● Issue:



UNF: P_ID, F_Name, L_Name, P_Age, P_Gender, Auth_Proof, P_Phone
Ticket_No, T_Date, Station, T_Name, Destination, Source, No_Passenger

1NF: Multivalued Attributes: P_Phone, No_Passenger
P_ID, F_Name, L_Name,P_Age,P_Gender, Auth_Proof
Ticket_No, T_Date, Station, T_Name, Destination, Source

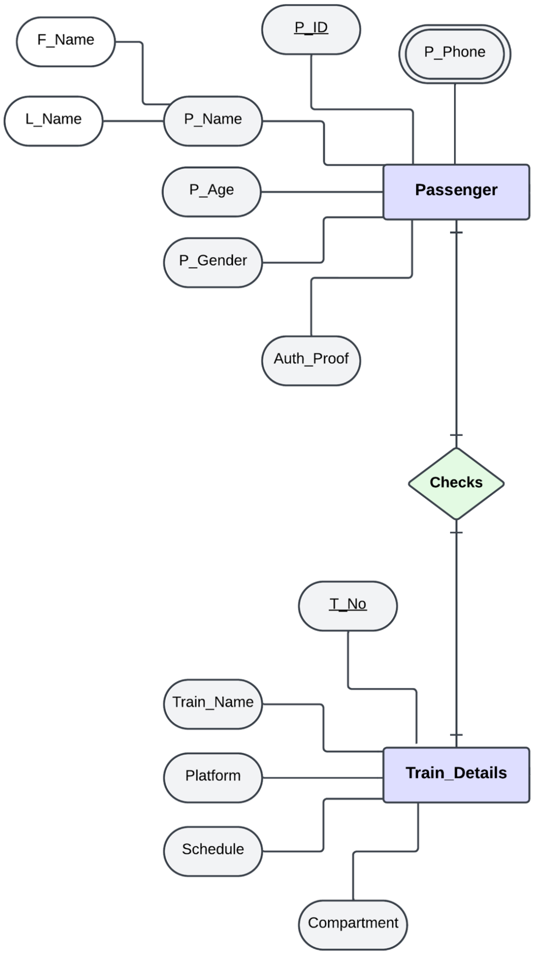
2NF:

1. P_ID, P_Phone
2. Ticket_No, No_Passenger
3. P_ID, F_Name, L_Name,P_Age,P_Gender, Auth_Proof
4. Ticket_No, T_Date, Station, T_Name, Destination, Source, P_ID

3NF:

1. P_ID, F_Name, L_Name
2. P_ID, P_Phone
3. Ticket_No, No_Passenger
4. P_ID, P_Age,P_Gender, Auth_Proof
5. Ticket_No, T_Date, Station, T_Name, Destination, Source, P_ID

● Checks:



UNF: P_ID, F_Name, L_Name, P_Age, P_Gender, Auth_Proof, P_Phone T_NO, Train_Name, Compartment, Platform, Schedule

1NF: Multivalued Attributes: P_Phone
P_ID, F_Name, L_Name,P_Age,P_Gender, Auth_Proof
T_NO , Train_Name, Compartment, Platform, Schedule

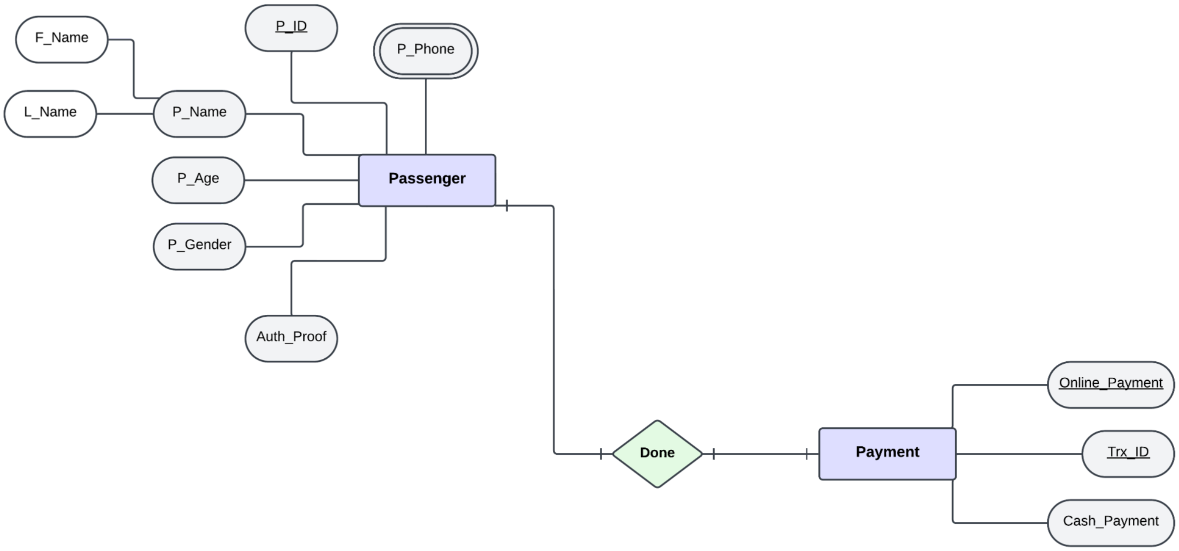
2NF:

1. P_ID, P_Phone
2. P_ID, F_Name, L_Name,P_Age,P_Gender, Auth_Proof
3. T_NO , Train_Name, Compartment, Platform, Schedule, P_ID

3NF:

1. P_ID, F_Name, L_Name
2. P_ID, P_Phone
3. P_ID, P_Age,P_Gender, Auth_Proof
4. T_NO , Train_Name, Compartment, Platform, Schedule, P_ID

- **Done:**



UNF: P_ID, F_Name, L_Name, P_Age, P_Gender, Auth_Proof, P_Phone
Trx_ID, Online_Payment, Cash_Payment

1NF: Multivalued Attributes: P_Phone
P_ID, F_Name, L_Name,P_Age,P_Gender, Auth_Proof
Trx_ID, Online_Payment, Cash_Payment

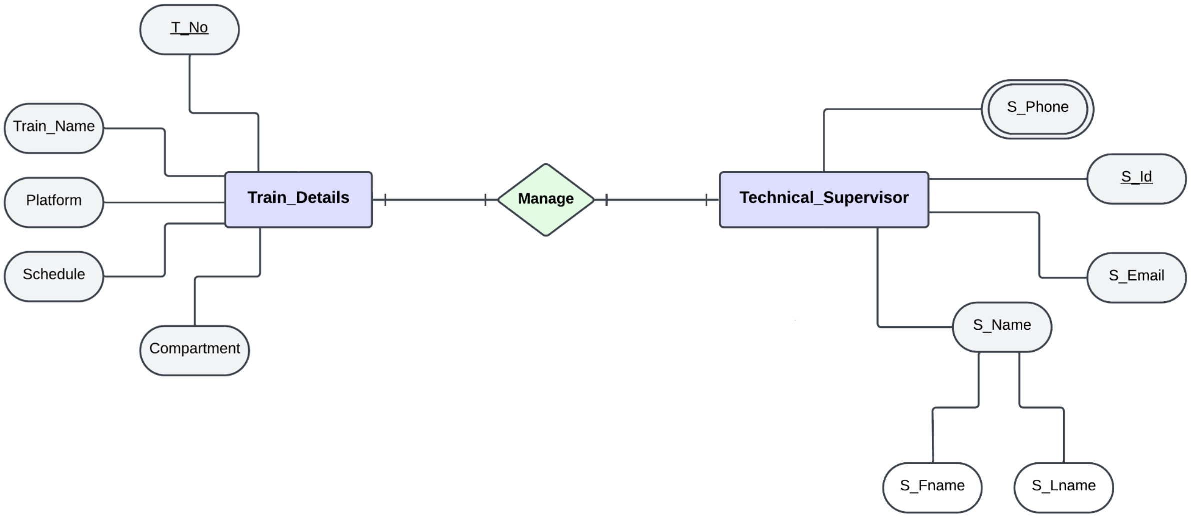
2NF:

1. P_ID, P_Phone
2. P_ID, F_Name, L_Name,P_Age,P_Gender, Auth_Proof
3. Trx_ID, Online_Payment, Cash_Payment, P_ID

3NF:

1. P_ID, F_Name, L_Name
2. P_ID, P_Phone
3. P_ID, P_Age,P_Gender, Auth_Proof
4. Trx_ID, Online_Payment, Cash_Payment, P_ID

- **Manage:**



UNF: T_NO , Train_Name, Compartment, Platform, Schedule
S_ID, S_Fname, S_Lname, S_Phone, S_Email

1NF: Multivalued Attributes: S_Phone

T_NO , Train_Name, Compartment, Platform, Schedule

S_ID, S_Fname, S_Lname, S_Email

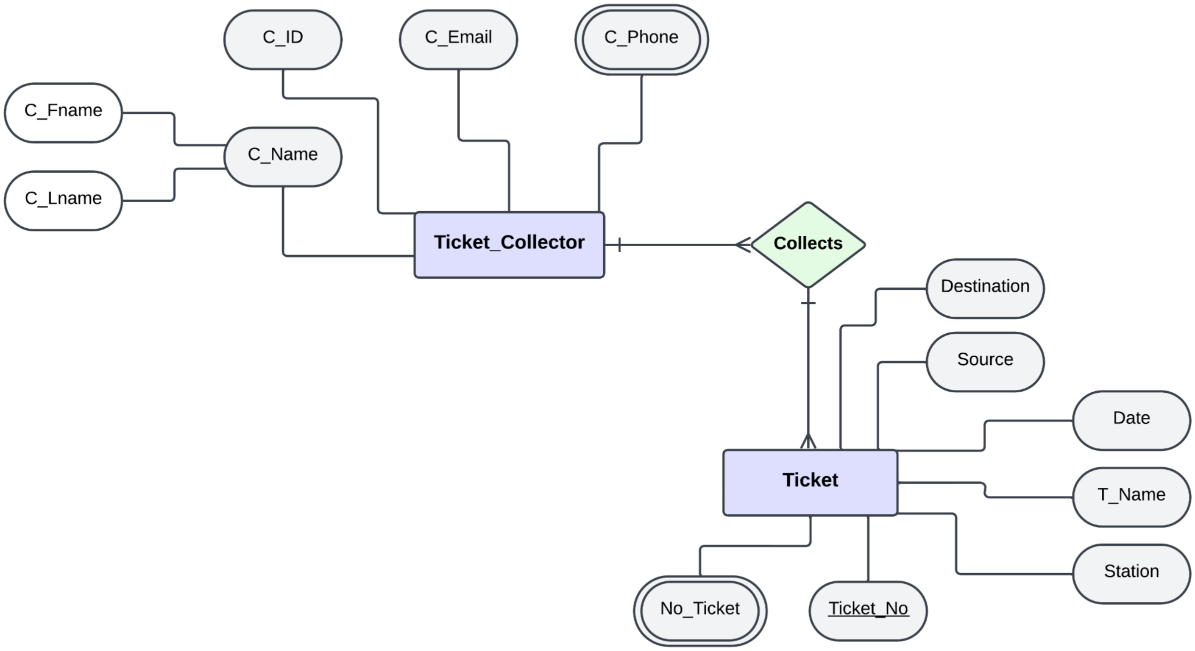
2NF:

1. S_ID, S_Phone
2. T_NO , Train_Name, Compartment, Platform, Schedule
3. S_ID, S_Fname, S_Lname, S_Email, T_No

3NF:

1. S_ID, S_Fname, S_Lname
2. S_ID, S_Phone
3. T_NO , Train_Name, Compartment, Platform, Schedule
4. S_ID, S_Email, T_No

- **Collects:**



UNF: C_ID, C_Fname, C_Lname, C_Phone, C_Email

Ticket_No, T_Date, Station, T_Name, Destination, Source, No_Passenger

1NF: Multivalued Attributes: C_Phone, No_Passenger
C_ID, C_Fname, C_Lname, C_Email
Ticket_No, T_Date, Station, T_Name, Destination, Source

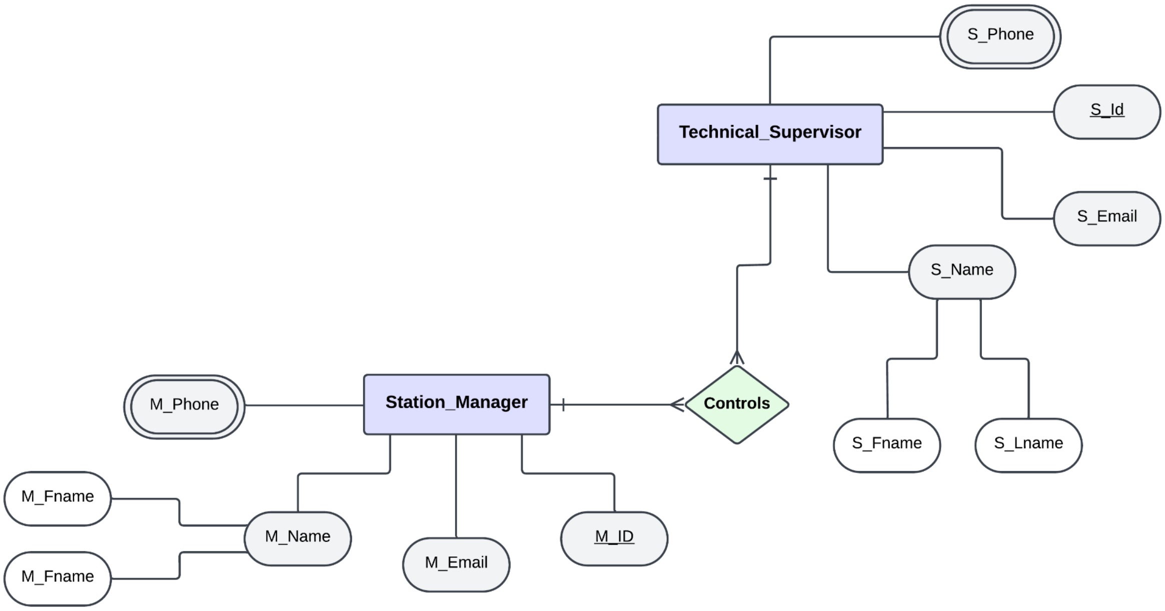
2NF:

1. C_ID, C_Phone
2. Ticket_No, No_Passenger
3. C_ID, C_Fname, C_Lname, C_Email
4. Ticket_No, T_Date, Station, T_Name, Destination, Source, C_ID

3NF:

1. Cr_ID, C_Fname, C_Lname
2. C_ID, C_Phone
3. Ticket_No, No_Passenger
4. C_ID, C_Email
5. Ticket_No, T_Date, Station, T_Name, Destination, Source, C_ID

● Controls:



UNF: M_ID, M_Fname, M_Lname, M_Phone, M_Email
S_ID, S_Fname, S_Lname, S_Phone, S_Email

1NF: Multivalued Attributes:S_Phone, M_Phone
M_ID, M_Fname, M_Lname, M_Email
S_ID, S_Fname, S_Lname S_Email

2NF:

1. M_ID, M_Phone
2. S_ID, S_Phone
3. M_ID, M_Fname, M_Lname_, M_Email
4. S_ID, S_Fname, S_Lname, S_Email, M_ID

3NF:

1. S_ID, S_Fname, S_Lname
2. M_ID, M_Fname, M_Lname
3. M_ID, M_Phone
4. S_ID, S_Phone
5. M_ID,M_Email
6. S_ID, S_Email, M_ID

FInalization

1. P_ID, F_Name, L_Name
2. P_ID, P_Phone
3. P_ID, P_Age,P_Gender, Auth_Proof
4. T_NO , Train_Name, Compartment, Platform, Schedule, P_ID
5. P_ID, F_Name, L_Name
6. P_ID, P_Phone
7. Ticket_No, No_Passenger
8. P_ID, P_Age,P_Gender, Auth_Proof
9. Ticket_No, T_Date, Station, T_Name, Destination, Source, P_ID
10. P_ID, F_Name, L_Name
11. P_ID, P_Phone
12. P_ID, P_Age,P_Gender, Auth_Proof
13. Trx_ID, Online_Payment, Cash_Payment, P_ID
14. S_ID, S_Fname, S_Lname
15. S_ID, S_Phone
16. T_NO , Train_Name, Compartment, Platform, Schedule
17. S_ID, S_Email, T_No
18. C_ID, C_Fname, C_Lname
19. C_ID, C_Phone
20. Ticket_No, No_Passenger
21. C_ID, C_Email
22. Ticket_No, T_Date, Station, T_Name, Destination, Source, C_ID
23. S_ID, S_Fname, S_Lname
24. M_ID, M_Fname, M_Lname
25. M_ID, M_Phone
26. S_ID, S_Phone
27. M_ID,M_Email
28. S_ID, S_Email, M_ID

Optimization

Final Table List:

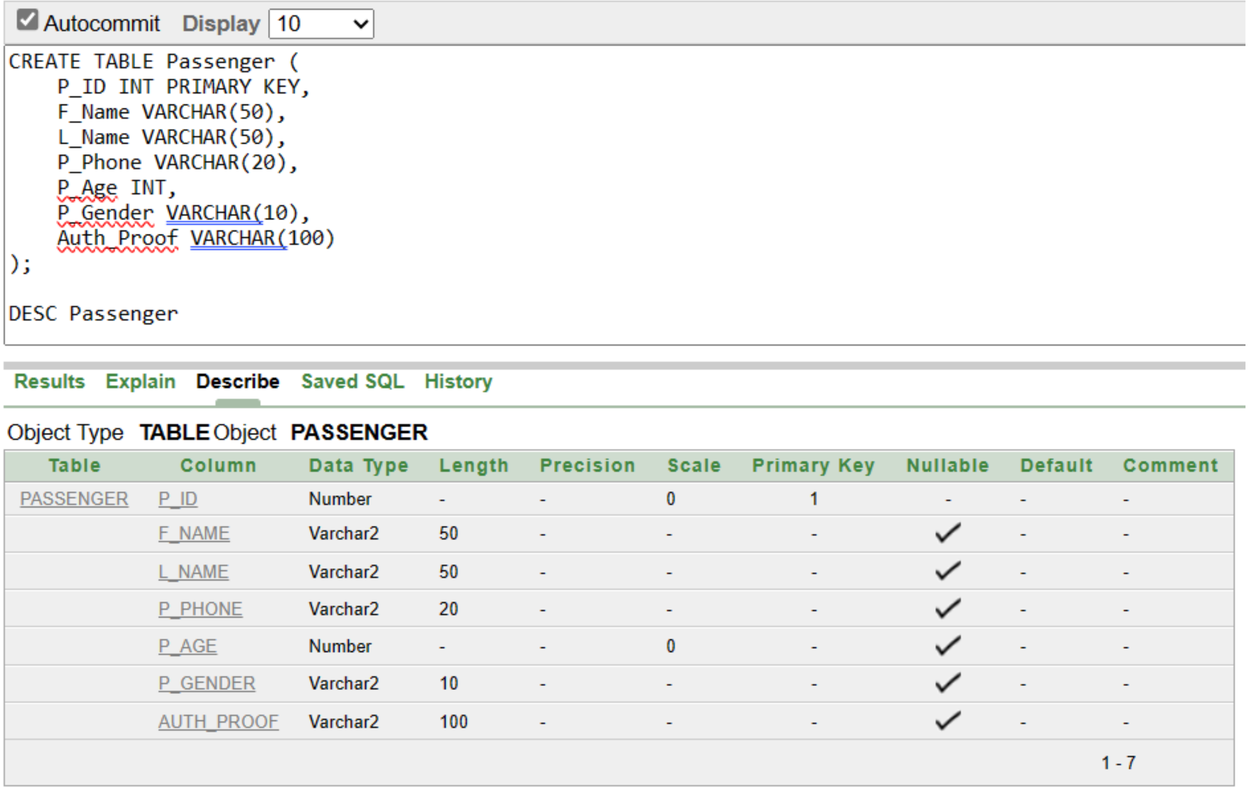
i	Passenger	:	P_ID, F_Name, L_Name, P_Phone, P_Age, P_Gender, Auth_Proof
ii	Train	:	T_No, Train_Name, Compartment, Platform, Schedule
iii	Collector	:	C_ID, C_Fname, C_Lname, C_Phone, C_Email
iv	Ticket	:	Ticket_No, No_Passenger, T_Date, Station, T_Name, Destination, Source, P_ID, C_ID
v	Payment	:	Trx_ID, Online_Payment, Cash_Payment, P_ID
vi	Technical Supervisor	:	S_ID, S_Fname, S_Lname, S_Phone, S_Email
vii	Station_Manager	:	M_ID, M_Fname, M_Lname, M_Phone, M_Email, S_ID

Table Creation

● Passenger Table:

Query:

```
CREATE TABLE Passenger (P_ID INT PRIMARY KEY, F_Name VARCHAR(50),
L_Name VARCHAR(50), P_Phone VARCHAR(20), P_Age INT, P_Gender
VARCHAR(10),  Auth_Proof  VARCHAR(100) );
```



```
CREATE TABLE Passenger (
    P_ID INT PRIMARY KEY,
    F_Name VARCHAR(50),
    L_Name VARCHAR(50),
    P_Phone VARCHAR(20),
    P_Age INT,
    P_Gender VARCHAR(10),
    Auth_Proof VARCHAR(100)
);

DESC Passenger
```

Object Type **TABLE** Object **PASSENGER**

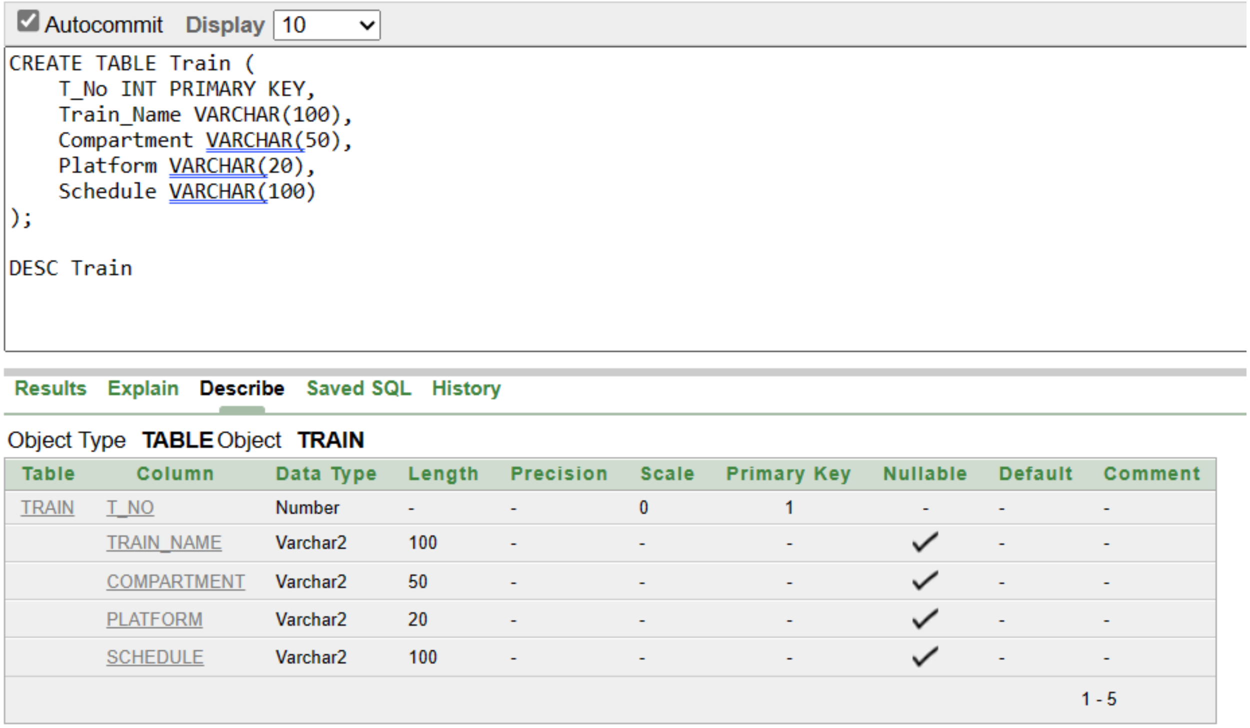
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PASSENGER	P_ID	Number	-	-	0	1	-	-	-
	F_NAME	Varchar2	50	-	-	-	✓	-	-
	L_NAME	Varchar2	50	-	-	-	✓	-	-
	P_PHONE	Varchar2	20	-	-	-	✓	-	-
	P_AGE	Number	-	-	0	-	✓	-	-
	P_GENDER	Varchar2	10	-	-	-	✓	-	-
	AUTH_PROOF	Varchar2	100	-	-	-	✓	-	-

1 - 7

● Train Table:

Query:

```
CREATE TABLE Train ( T_No INT PRIMARY KEY, Train_Name VARCHAR(100),
Compartment VARCHAR(50), Platform VARCHAR(20), Schedule VARCHAR(100) );
```



```
CREATE TABLE Train (
    T_No INT PRIMARY KEY,
    Train_Name VARCHAR(100),
    Compartment VARCHAR(50),
    Platform VARCHAR(20),
    Schedule VARCHAR(100)
);

DESC Train
```

Object Type **TABLE** Object **TRAIN**

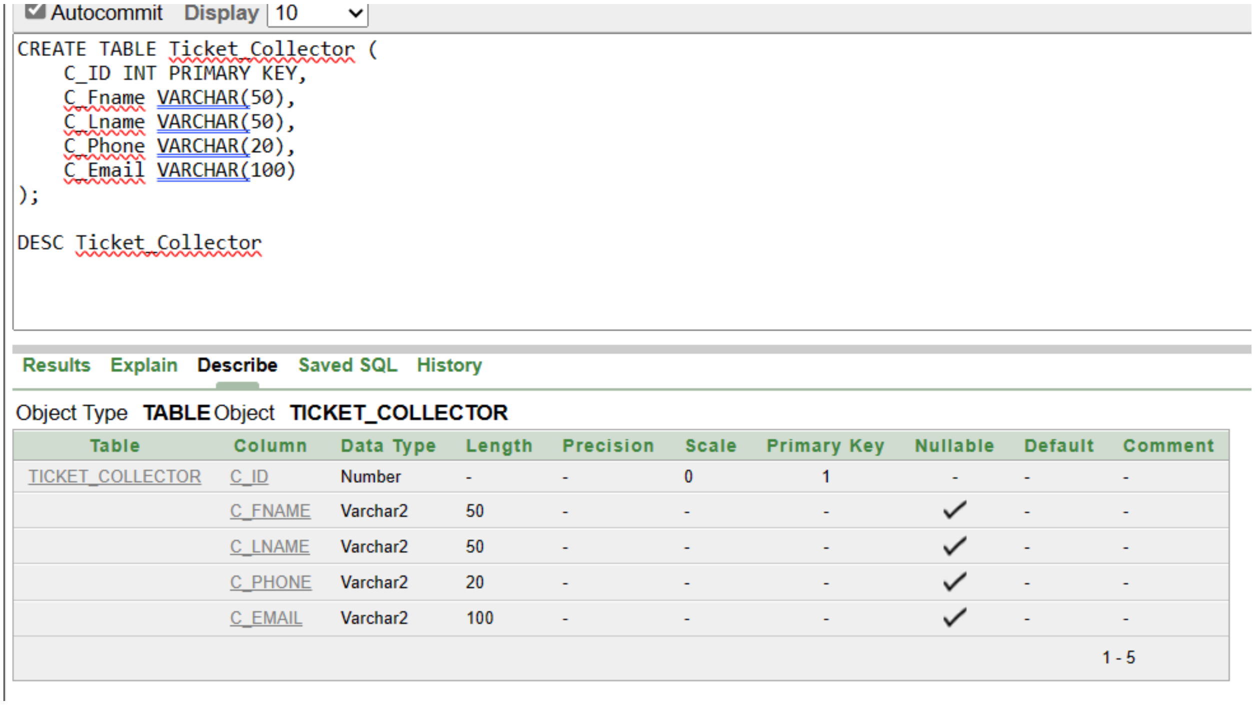
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
TRAIN	T_NO	Number	-	-	0	1	-	-	-
	TRAIN_NAME	Varchar2	100	-	-	-	✓	-	-
	COMPARTMENT	Varchar2	50	-	-	-	✓	-	-
	PLATFORM	Varchar2	20	-	-	-	✓	-	-
	SCHEDULE	Varchar2	100	-	-	-	✓	-	-

1 - 5

● Ticket_Collector Table:

Query:

```
CREATE TABLE Ticket_Collector ( C_ID INT PRIMARY KEY,C_Fname VARCHAR(50),
 C_Lname VARCHAR(50),C_Phone VARCHAR(20),C_Email VARCHAR(100));
```



```
CREATE TABLE Ticket_Collector (
    C_ID INT PRIMARY KEY,
    C_Fname VARCHAR(50),
    C_Lname VARCHAR(50),
    C_Phone VARCHAR(20),
    C_Email VARCHAR(100)
);

DESC Ticket_Collector
```

Object Type **TABLE** Object **TICKET_COLLECTOR**

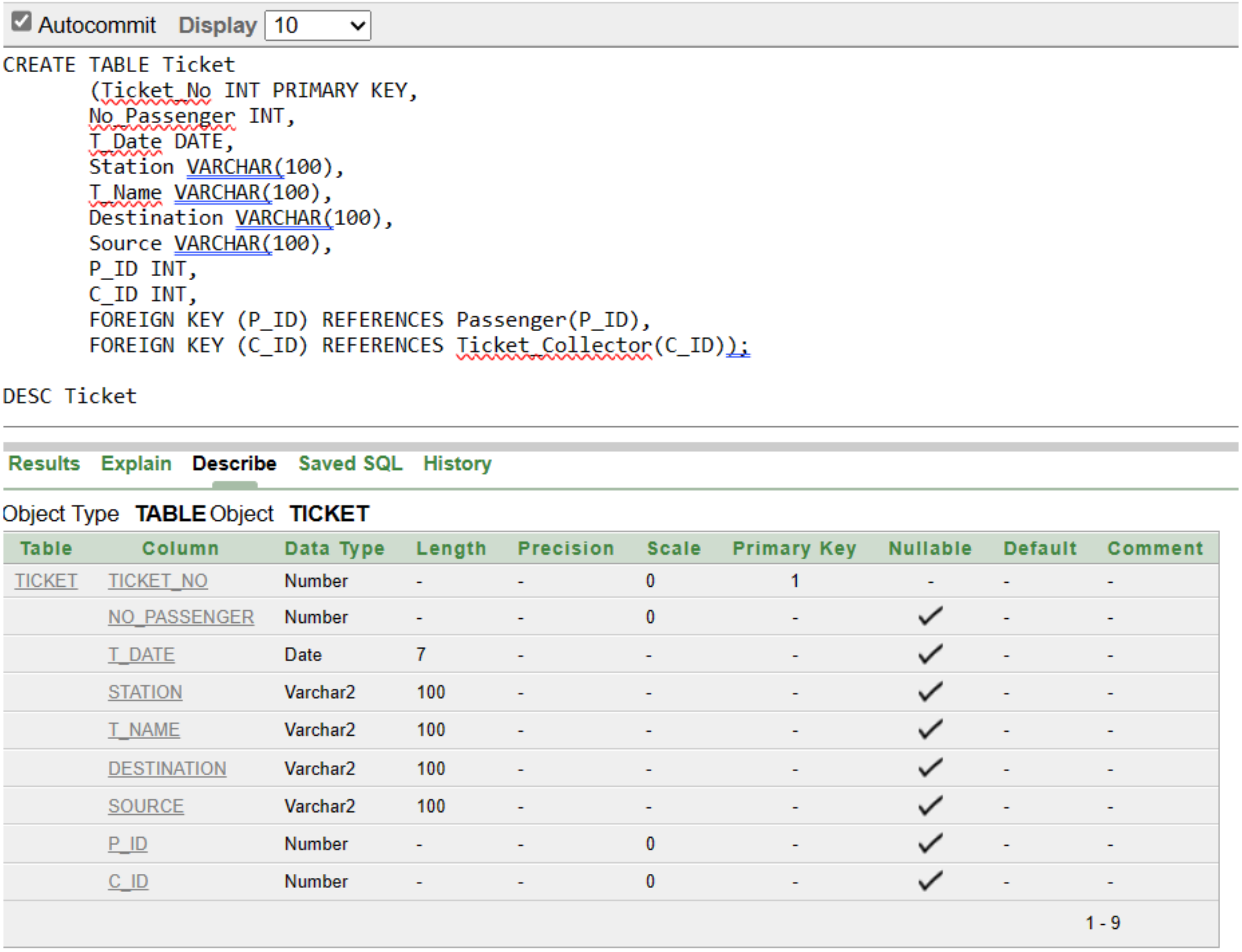
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
TICKET_COLLECTOR	C_ID	Number	-	-	0	1	-	-	-
	C_FNAME	Varchar2	50	-	-	-	✓	-	-
	C_LNAME	Varchar2	50	-	-	-	✓	-	-
	C_PHONE	Varchar2	20	-	-	-	✓	-	-
	C_EMAIL	Varchar2	100	-	-	-	✓	-	-

1 - 5

- ## Ticket Table:

Query:

```
CREATE TABLE Ticket (Ticket_No INT PRIMARY KEY, No_Passenger INT,
T_Date DATE,Station VARCHAR(100),T_Name VARCHAR(100),
Destination VARCHAR(100),Source VARCHAR(100),P_ID INT,C_ID INT,
FOREIGN KEY (P_ID) REFERENCES Passenger(P_ID),
FOREIGN KEY (C_ID) REFERENCES Collector(C_ID));
```



☑ Autocommit Display 10

```
CREATE TABLE Ticket
       (Ticket_No INT PRIMARY KEY,
       No_Passenger INT,
       T_Date DATE,
       Station VARCHAR(100),
       T_Name VARCHAR(100),
       Destination VARCHAR(100),
       Source VARCHAR(100),
       P_ID INT,
       C_ID INT,
       FOREIGN KEY (P_ID) REFERENCES Passenger(P_ID),
       FOREIGN KEY (C_ID) REFERENCES Ticket_Collector(C_ID));

DESC Ticket
```

Object Type **TABLE** Object **TICKET**

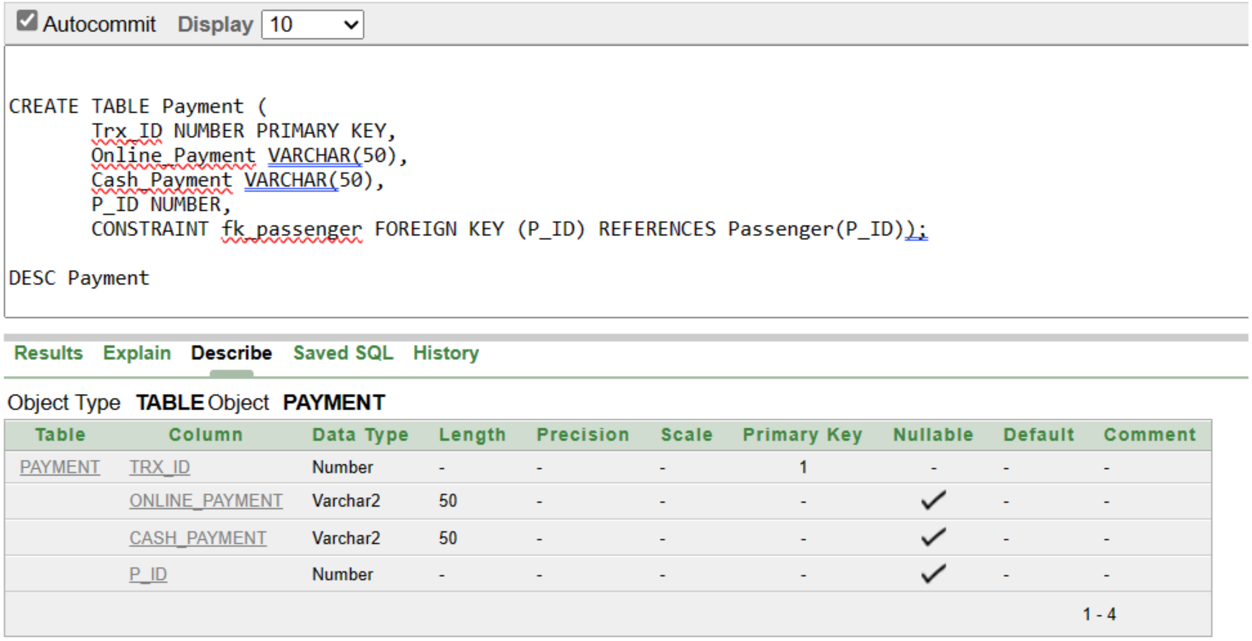
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
TICKET	TICKET_NO	Number	-	-	0	1	-	-	-
	NO_PASSENGER	Number	-	-	0	-	✓	-	-
	T_DATE	Date	7	-	-	-	✓	-	-
	STATION	Varchar2	100	-	-	-	✓	-	-
	T_NAME	Varchar2	100	-	-	-	✓	-	-
	DESTINATION	Varchar2	100	-	-	-	✓	-	-
	SOURCE	Varchar2	100	-	-	-	✓	-	-
	P_ID	Number	-	-	0	-	✓	-	-
	C_ID	Number	-	-	0	-	✓	-	-

1 - 9

● Payment Table:

Query:

```
CREATE TABLE Payment (Trx_ID NUMBER PRIMARY KEY,Online_Payment NUMBER,
Cash_Payment NUMBER,P_ID NUMBER,
CONSTRAINT fk_passenger FOREIGN KEY (P_ID) REFERENCES Passenger(P_ID));
```



```
CREATE TABLE Payment (
        Trx_ID NUMBER PRIMARY KEY,
        Online_Payment VARCHAR(50),
        Cash_Payment VARCHAR(50),
        P_ID NUMBER,
        CONSTRAINT fk_passenger FOREIGN KEY (P_ID) REFERENCES Passenger(P_ID));

DESC Payment
```

Object Type **TABLE** Object **PAYMENT**

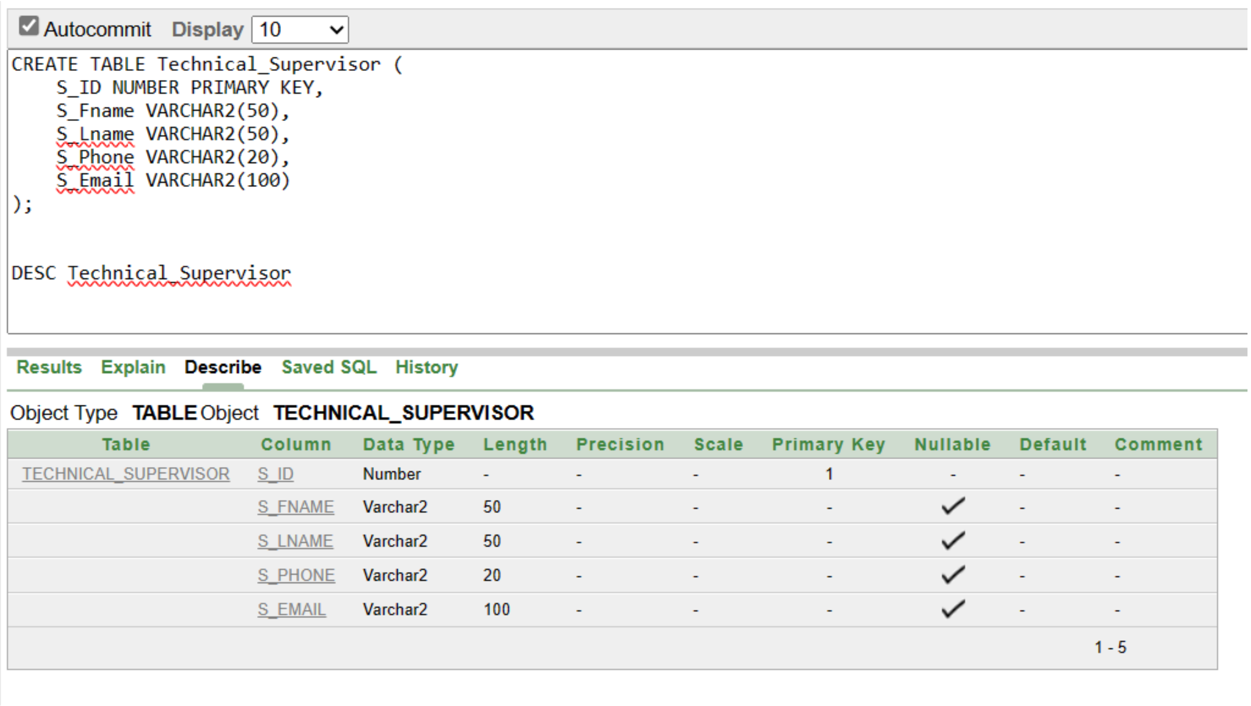
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PAYMENT	TRX_ID	Number	-	-	-	1	-	-	-
	ONLINE_PAYMENT	Varchar2	50	-	-	-	✓	-	-
	CASH_PAYMENT	Varchar2	50	-	-	-	✓	-	-
	P_ID	Number	-	-	-	-	✓	-	-

1 - 4

- **Technical_Supervisor Table:**

Query:

```
CREATE TABLE Technical_Supervisor (S_ID NUMBER PRIMARY KEY,
 S_Fname VARCHAR2(50),S_Lname VARCHAR2(50),S_Phone VARCHAR2(20),
 S_Email VARCHAR2(100));
```



```
CREATE TABLE Technical_Supervisor (
    S_ID NUMBER PRIMARY KEY,
    S_Fname VARCHAR2(50),
    S_Lname VARCHAR2(50),
    S_Phone VARCHAR2(20),
    S_Email VARCHAR2(100)
);


DESC Technical_Supervisor
```

Object Type **TABLE** Object **TECHNICAL_SUPERVISOR**

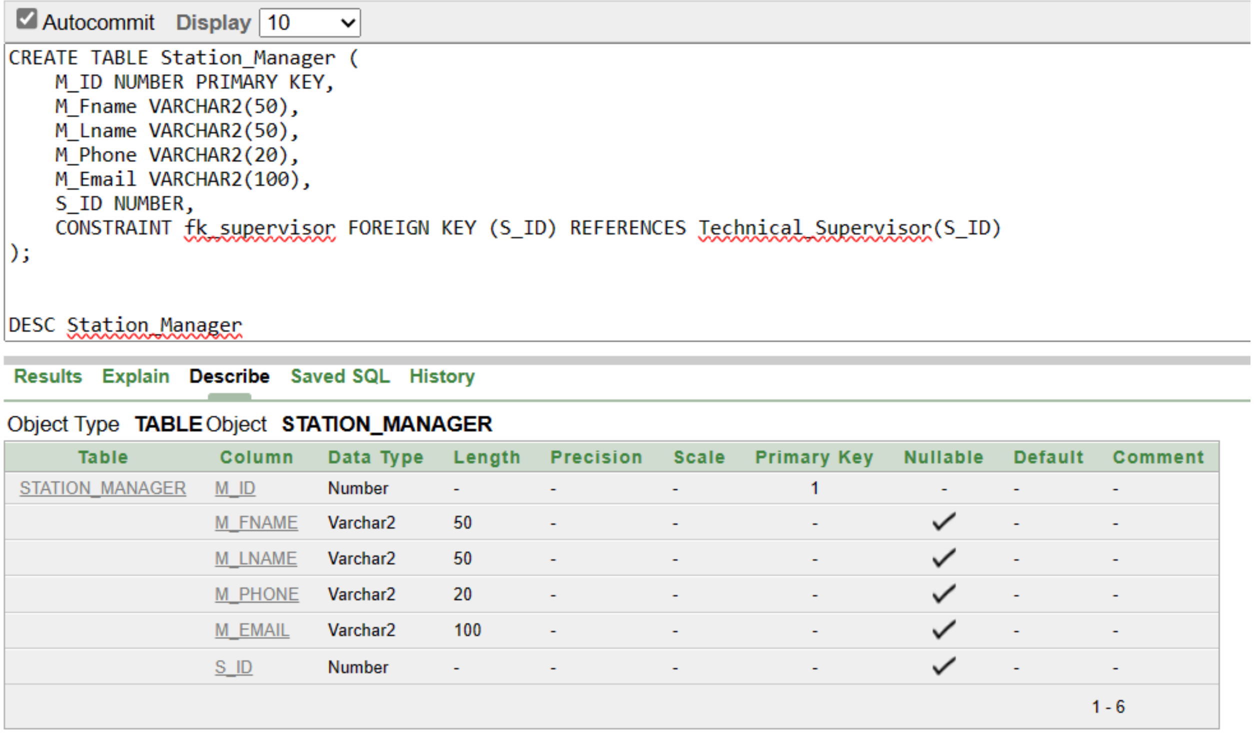
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
TECHNICAL_SUPERVISOR	S_ID	Number	-	-	-	1	-	-	-
	S_FNAME	Varchar2	50	-	-	-	✓	-	-
	S_LNAME	Varchar2	50	-	-	-	✓	-	-
	S_PHONE	Varchar2	20	-	-	-	✓	-	-
	S_EMAIL	Varchar2	100	-	-	-	✓	-	-

1 - 5

● Station_Manager Table:

Query:

```
CREATE TABLE Station_Manager ( M_ID NUMBER PRIMARY KEY,
M_Fname VARCHAR2(50),M_Lname VARCHAR2(50),M_Phone VARCHAR2(20),
M_Email VARCHAR2(100),S_ID NUMBER,
CONSTRAINT fk_supervisor FOREIGN KEY (S_ID) REFERENCES
Technical_Supervisor(S_ID));
```



```
CREATE TABLE Station_Manager (
    M_ID NUMBER PRIMARY KEY,
    M_Fname VARCHAR2(50),
    M_Lname VARCHAR2(50),
    M_Phone VARCHAR2(20),
    M_Email VARCHAR2(100),
    S_ID NUMBER,
    CONSTRAINT fk_supervisor FOREIGN KEY (S_ID) REFERENCES Technical_Supervisor(S_ID)
);


DESC Station_Manager
```

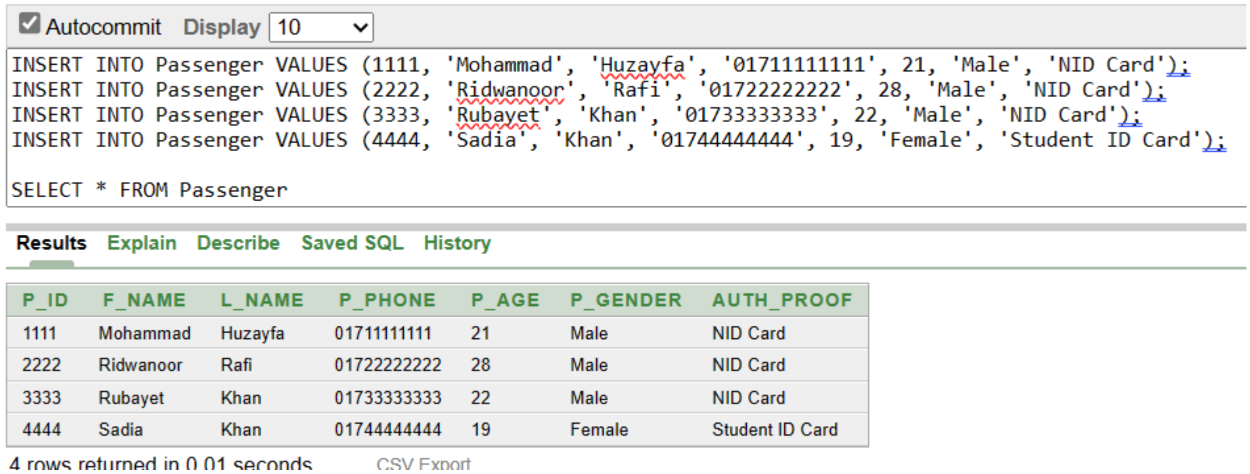
Object Type **TABLE** Object **STATION_MANAGER**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
STATION_MANAGER	M_ID	Number	-	-	-	1	-	-	-
	M_FNAME	Varchar2	50	-	-	-	✓	-	-
	M_LNAME	Varchar2	50	-	-	-	✓	-	-
	M_PHONE	Varchar2	20	-	-	-	✓	-	-
	M_EMAIL	Varchar2	100	-	-	-	✓	-	-
	S_ID	Number	-	-	-	-	✓	-	-

1 - 6

Data Insertion

- ## Passenger Table:

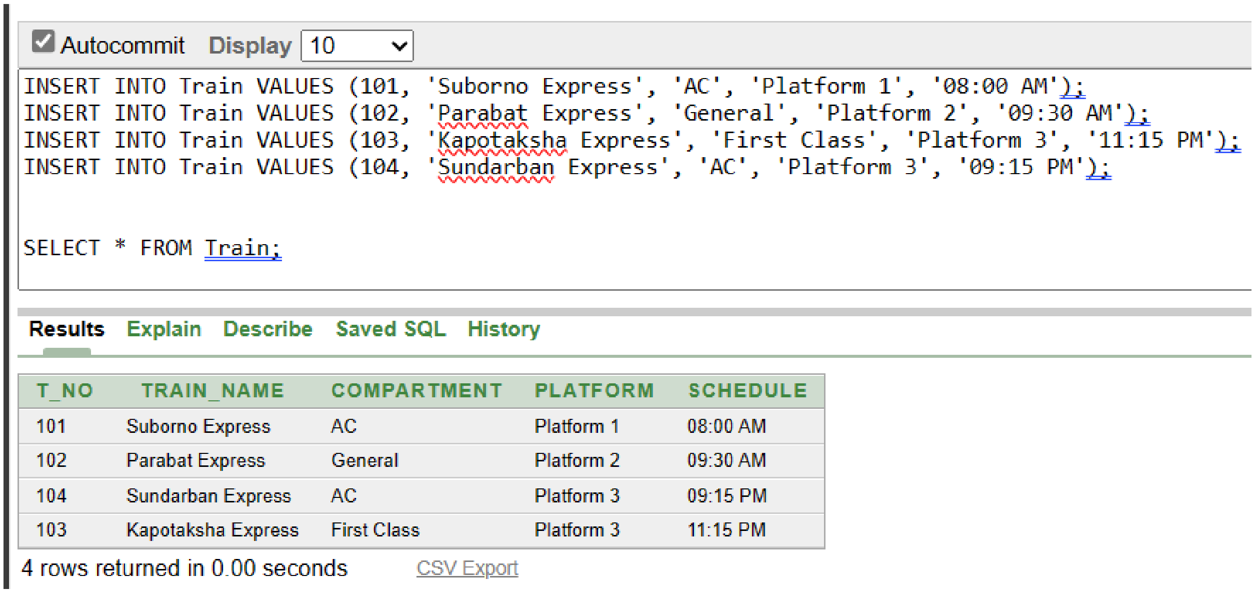


```
INSERT INTO Passenger VALUES (1111, 'Mohammad', 'Huzayfa', '01711111111', 21, 'Male', 'NID Card');
INSERT INTO Passenger VALUES (2222, 'Ridwanoor', 'Rafi', '01722222222', 28, 'Male', 'NID Card');
INSERT INTO Passenger VALUES (3333, 'Rubayet', 'Khan', '01733333333', 22, 'Male', 'NID Card');
INSERT INTO Passenger VALUES (4444, 'Sadia', 'Khan', '01744444444', 19, 'Female', 'Student ID Card');

SELECT * FROM Passenger
```

P_ID	F_NAME	L_NAME	P_PHONE	P_AGE	P_GENDER	AUTH_PROOF
1111	Mohammad	Huzayfa	01711111111	21	Male	NID Card
2222	Ridwanoor	Rafi	01722222222	28	Male	NID Card
3333	Rubayet	Khan	01733333333	22	Male	NID Card
4444	Sadia	Khan	01744444444	19	Female	Student ID Card

4 rows returned in 0.01 seconds

- ## Train Table:

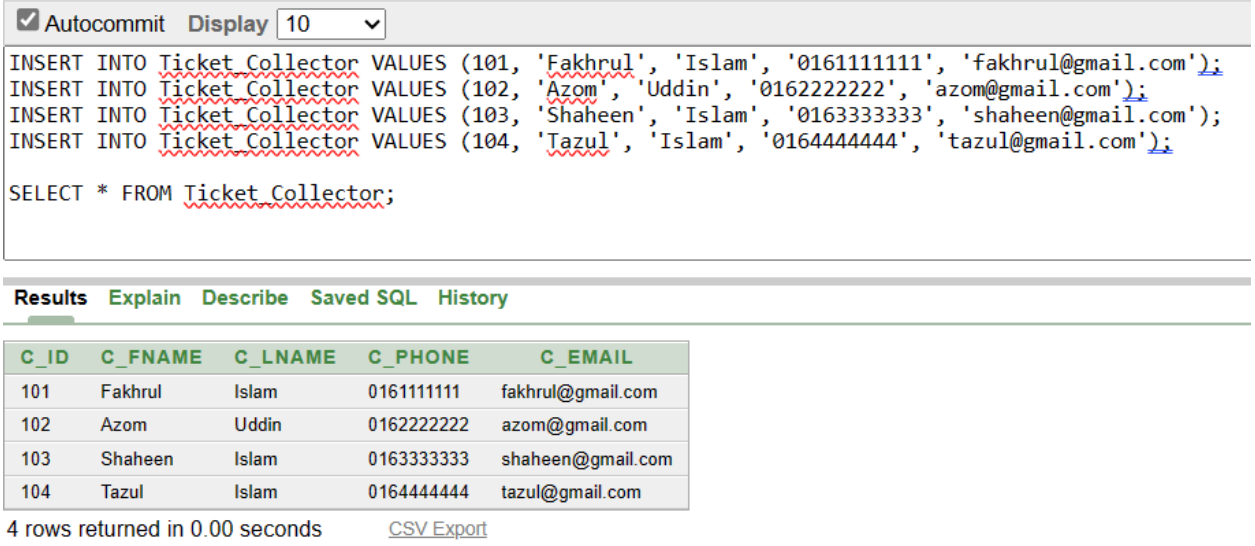


```
INSERT INTO Train VALUES (101, 'Suborno Express', 'AC', 'Platform 1', '08:00 AM');
INSERT INTO Train VALUES (102, 'Parabat Express', 'General', 'Platform 2', '09:30 AM');
INSERT INTO Train VALUES (103, 'Kapotaksha Express', 'First Class', 'Platform 3', '11:15 PM');
INSERT INTO Train VALUES (104, 'Sundarban Express', 'AC', 'Platform 3', '09:15 PM');


SELECT * FROM Train;
```

T_NO	TRAIN_NAME	COMPARTMENT	PLATFORM	SCHEDULE
101	Suborno Express	AC	Platform 1	08:00 AM
102	Parabat Express	General	Platform 2	09:30 AM
104	Sundarban Express	AC	Platform 3	09:15 PM
103	Kapotaksha Express	First Class	Platform 3	11:15 PM

4 rows returned in 0.00 seconds

- **Ticket_Collector Table:**

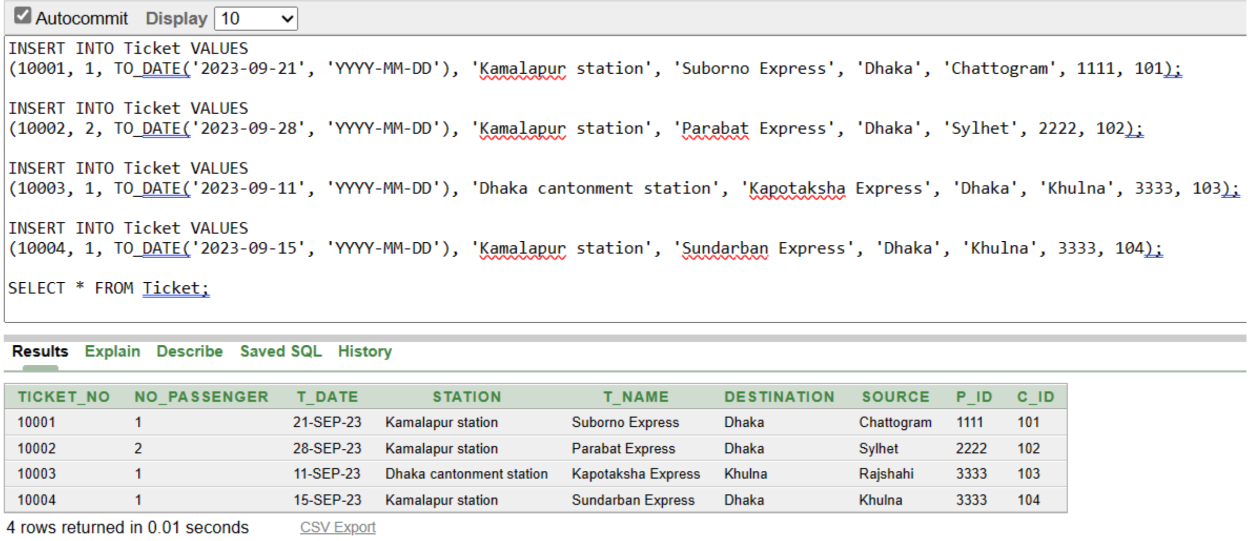


```
INSERT INTO Ticket_Collector VALUES (101, 'Fakhrul', 'Islam', '0161111111', 'fakhrul@gmail.com');
INSERT INTO Ticket_Collector VALUES (102, 'Azom', 'Uddin', '0162222222', 'azom@gmail.com');
INSERT INTO Ticket_Collector VALUES (103, 'Shaheen', 'Islam', '0163333333', 'shaheen@gmail.com');
INSERT INTO Ticket_Collector VALUES (104, 'Tazul', 'Islam', '0164444444', 'tazul@gmail.com');

SELECT * FROM Ticket_Collector;
```

C_ID	C_FNAME	C_LNAME	C_PHONE	C_EMAIL
101	Fakhrul	Islam	0161111111	fakhrul@gmail.com
102	Azom	Uddin	0162222222	azom@gmail.com
103	Shaheen	Islam	0163333333	shaheen@gmail.com
104	Tazul	Islam	0164444444	tazul@gmail.com

4 rows returned in 0.00 seconds

- **Ticket Table:**

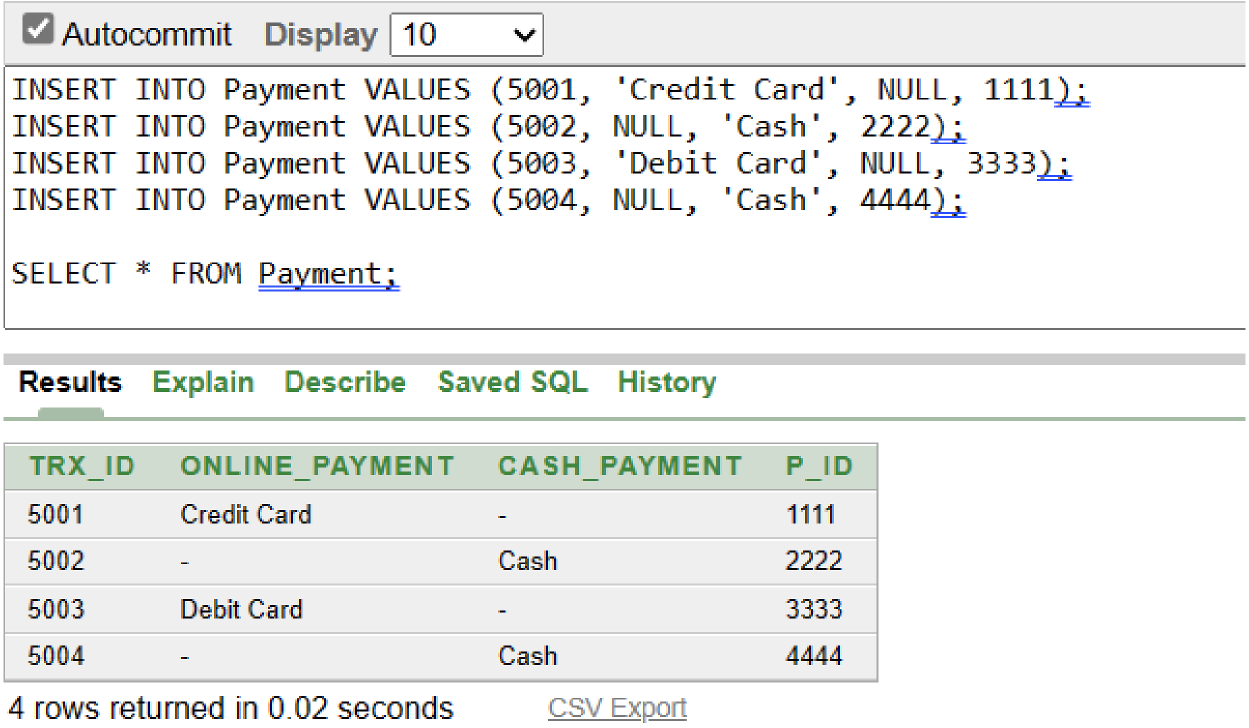


```
INSERT INTO Ticket VALUES
(10001, 1, TO_DATE('2023-09-21', 'YYYY-MM-DD'), 'Kamalapur station', 'Suborno Express', 'Dhaka', 'Chattogram', 1111, 101);

INSERT INTO Ticket VALUES
(10002, 2, TO_DATE('2023-09-28', 'YYYY-MM-DD'), 'Kamalapur station', 'Parabat Express', 'Dhaka', 'Sylhet', 2222, 102);

INSERT INTO Ticket VALUES
(10003, 1, TO_DATE('2023-09-11', 'YYYY-MM-DD'), 'Dhaka cantonment station', 'Kapotaksha Express', 'Dhaka', 'Khulna', 3333, 103);

INSERT INTO Ticket VALUES
(10004, 1, TO_DATE('2023-09-15', 'YYYY-MM-DD'), 'Kamalapur station', 'Sundarban Express', 'Dhaka', 'Khulna', 3333, 104);

SELECT * FROM Ticket;
```

TICKET_NO	NO_PASSENGER	T_DATE	STATION	T_NAME	DESTINATION	SOURCE	P_ID	C_ID
10001	1	21-SEP-23	Kamalapur station	Suborno Express	Dhaka	Chattogram	1111	101
10002	2	28-SEP-23	Kamalapur station	Parabat Express	Dhaka	Sylhet	2222	102
10003	1	11-SEP-23	Dhaka cantonment station	Kapotaksha Express	Khulna	Rajshahi	3333	103
10004	1	15-SEP-23	Kamalapur station	Sundarban Express	Dhaka	Khulna	3333	104

4 rows returned in 0.01 seconds

- **Payment Table:**

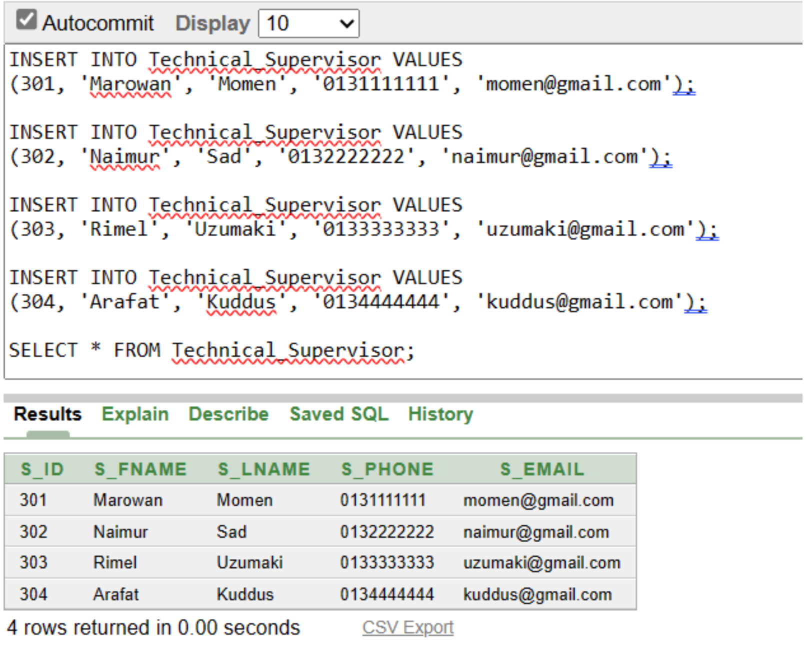


```
INSERT INTO Payment VALUES (5001, 'Credit Card', NULL, 1111);
INSERT INTO Payment VALUES (5002, NULL, 'Cash', 2222);
INSERT INTO Payment VALUES (5003, 'Debit Card', NULL, 3333);
INSERT INTO Payment VALUES (5004, NULL, 'Cash', 4444);

SELECT * FROM Payment;
```

TRX_ID	ONLINE_PAYMENT	CASH_PAYMENT	P_ID
5001	Credit Card	-	1111
5002	-	Cash	2222
5003	Debit Card	-	3333
5004	-	Cash	4444

4 rows returned in 0.02 seconds

- **Technical_Supervisor Table:**

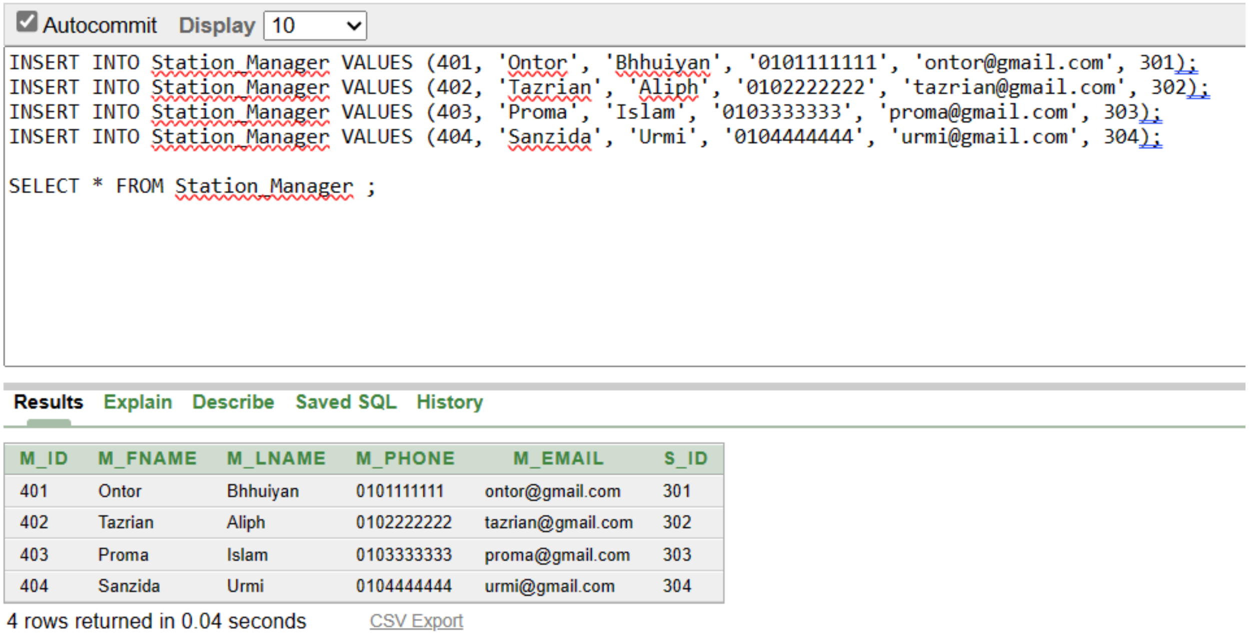


```
INSERT INTO Technical_Supervisor VALUES
(301, 'Marowan', 'Momen', '0131111111', 'momen@gmail.com');

INSERT INTO Technical_Supervisor VALUES
(302, 'Naimur', 'Sad', '0132222222', 'naimur@gmail.com');

INSERT INTO Technical_Supervisor VALUES
(303, 'Rimel', 'Uzumaki', '0133333333', 'uzumaki@gmail.com');

INSERT INTO Technical_Supervisor VALUES
(304, 'Arafat', 'Kuddus', '0134444444', 'kuddus@gmail.com');

SELECT * FROM Technical_Supervisor;
```

S_ID	S_FNAME	S_LNAME	S_PHONE	S_EMAIL
301	Marowan	Momen	0131111111	momen@gmail.com
302	Naimur	Sad	0132222222	naimur@gmail.com
303	Rimel	Uzumaki	0133333333	uzumaki@gmail.com
304	Arafat	Kuddus	0134444444	kuddus@gmail.com

4 rows returned in 0.00 seconds

- **Station_Manager Table:**



```
INSERT INTO Station_Manager VALUES (401, 'Ontor', 'Bhhuiyan', '0101111111', 'ontor@gmail.com', 301);
INSERT INTO Station_Manager VALUES (402, 'Tazrian', 'Aliph', '0102222222', 'tazrian@gmail.com', 302);
INSERT INTO Station_Manager VALUES (403, 'Proma', 'Islam', '0103333333', 'proma@gmail.com', 303);
INSERT INTO Station_Manager VALUES (404, 'Sanzida', 'Urmi', '0104444444', 'urmi@gmail.com', 304);

SELECT * FROM Station_Manager ;
```

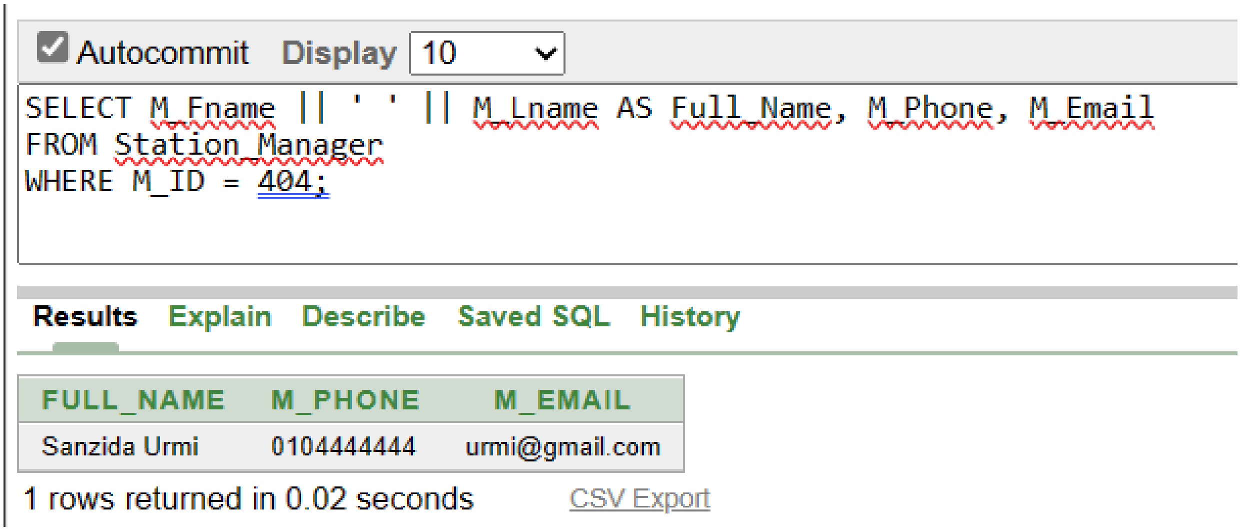
M_ID	M_FNAME	M_LNAME	M_PHONE	M_EMAIL	S_ID
401	Ontor	Bhhuiyan	0101111111	ontor@gmail.com	301
402	Tazrian	Aliph	0102222222	tazrian@gmail.com	302
403	Proma	Islam	0103333333	proma@gmail.com	303
404	Sanzida	Urmi	0104444444	urmi@gmail.com	304

4 rows returned in 0.04 seconds

Query Writing

• Single row function (1):

Question: Show Full Name (concat first name & last name), Phone, Email from Station_Manager table where the ID is 404.



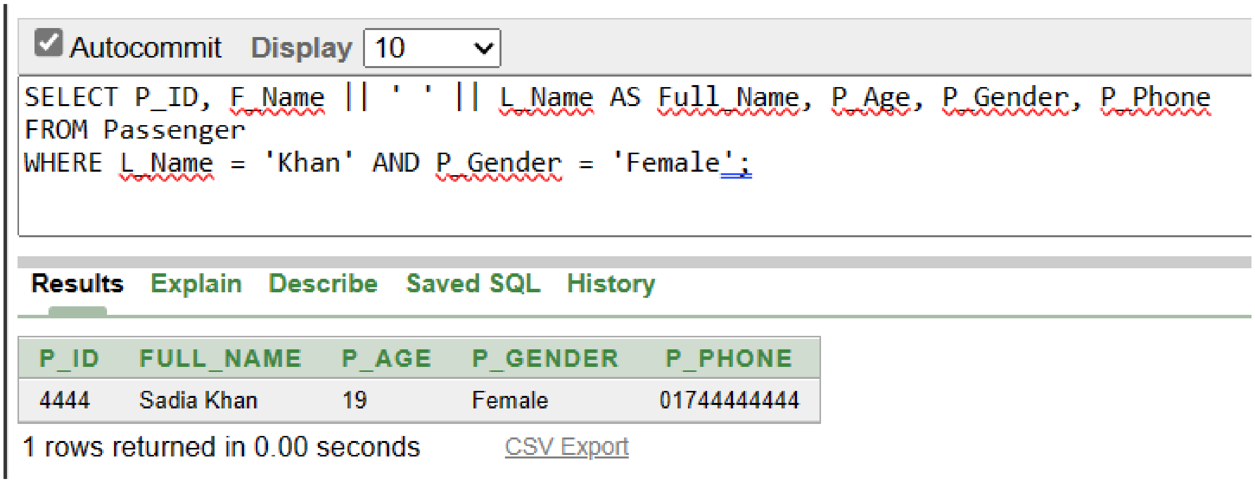
```
SELECT M_Fname || ' ' || M_Lname AS Full_Name, M_Phone, M_Email
FROM Station_Manager
WHERE M_ID = 404;
```

FULL_NAME	M_PHONE	M_EMAIL
Sanzida Urmi	0104444444	urmi@gmail.com

1 rows returned in 0.02 seconds

• Single row function (2):

Question: Show the ID, full name (concat first name & last name), age, gender & phone who has Khan as last name and is female.



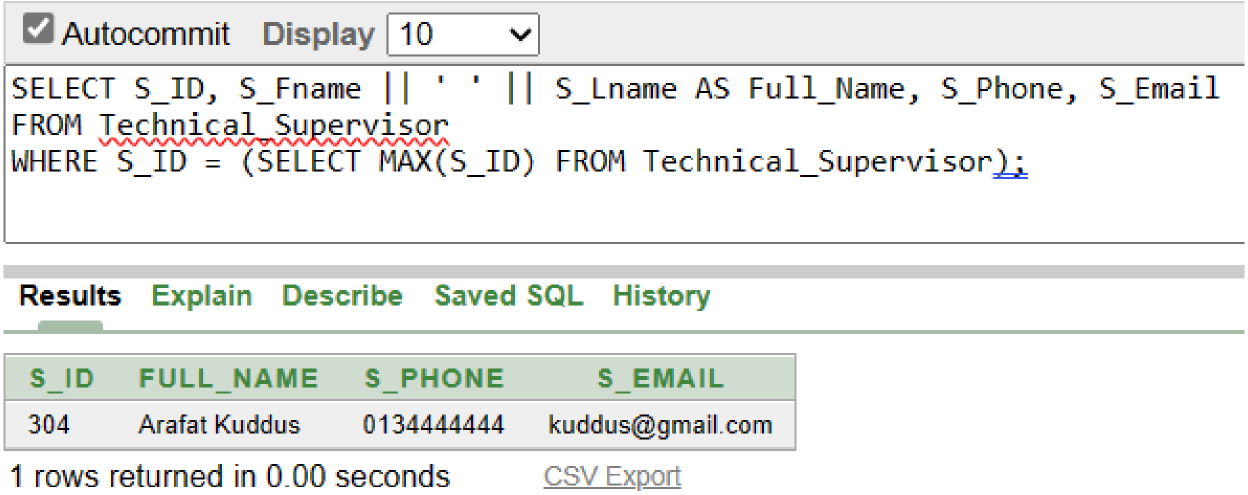
```
SELECT P_ID, F_Name || ' ' || L_Name AS Full_Name, P_Age, P_Gender, P_Phone
FROM Passenger
WHERE L_Name = 'Khan' AND P_Gender = 'Female';
```

P_ID	FULL_NAME	P_AGE	P_GENDER	P_PHONE
4444	Sadia Khan	19	Female	01744444444

1 rows returned in 0.00 seconds

● Group function (1):

Question: Show the ID, full name (concat first name & last name), phone & email from the Technical_Supervisor table where the id is max.



☑ Autocommit Display 10

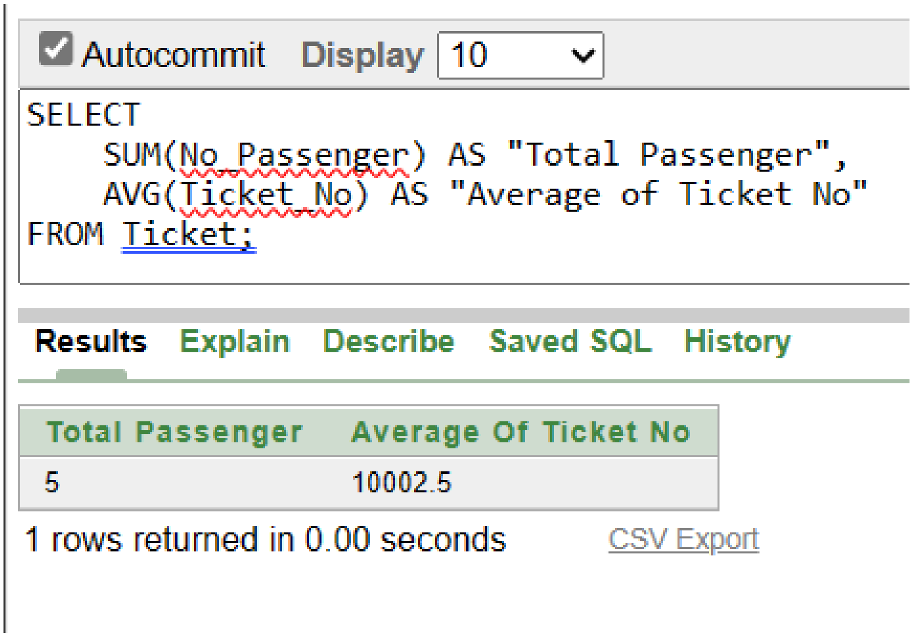
```
SELECT S_ID, S_Fname || ' ' || S_Lname AS Full_Name, S_Phone, S_Email
FROM Technical_Supervisor
WHERE S_ID = (SELECT MAX(S_ID) FROM Technical_Supervisor);
```

S_ID	FULL_NAME	S_PHONE	S_EMAIL
304	Arafat Kuddus	0134444444	kuddus@gmail.com

1 rows returned in 0.00 seconds

● Group function (2):

Question: Show sum of all the passenger numbers & average of ticket no also name the average of passenger number to exactly "Total Passenger".



☑ Autocommit Display 10

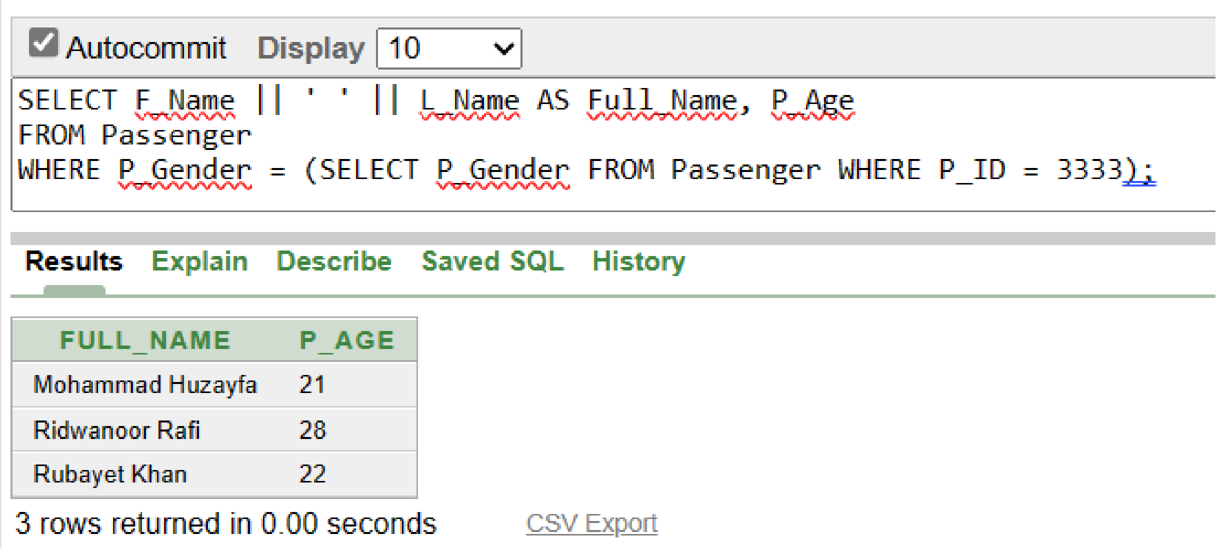
```
SELECT
    SUM(No_Passenger) AS "Total Passenger",
    AVG(Ticket_No) AS "Average of Ticket No"
FROM Ticket;
```

Total Passenger	Average Of Ticket No
5	10002.5

1 rows returned in 0.00 seconds

- **Sub query (1):**

Question: Show the the name, age and phone of passengers from the Passenger table who have the same gender as the passenger with ID 3333.



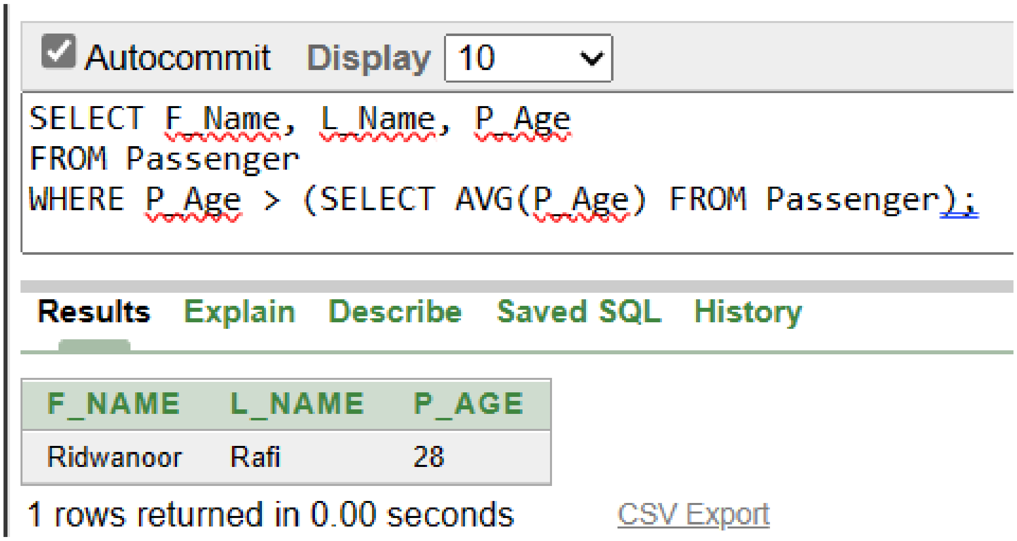
```
SELECT F_Name || ' ' || L_Name AS Full_Name, P_Age
FROM Passenger
WHERE P_Gender = (SELECT P_Gender FROM Passenger WHERE P_ID = 3333);
```

FULL_NAME	P_AGE
Mohammad Huzayfa	21
Ridwanoor Rafi	28
Rubayet Khan	22

3 rows returned in 0.00 seconds

- **Sub query (2):**

Question: Show the names and ages of passengers from the "Passenger" table who are older than the average age of all passengers.



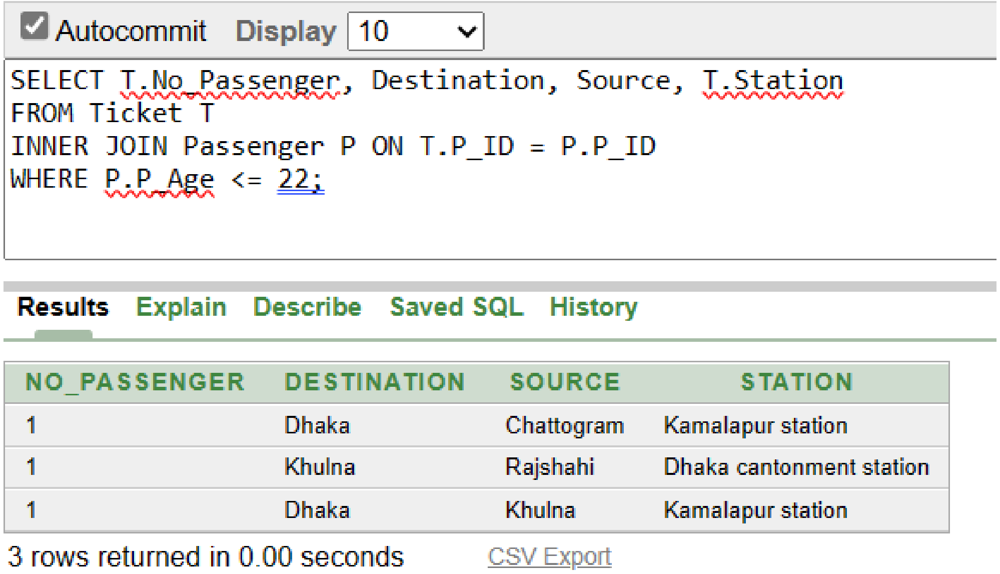
```
SELECT F_Name, L_Name, P_Age
FROM Passenger
WHERE P_Age > (SELECT AVG(P_Age) FROM Passenger);
```

F_NAME	L_NAME	P_AGE
Ridwanoor	Rafi	28

1 rows returned in 0.00 seconds

● Joining (1):

Question: Show the number of passengers, destination, source & station names for each ticket, where the passengers are aged 20 or younger.



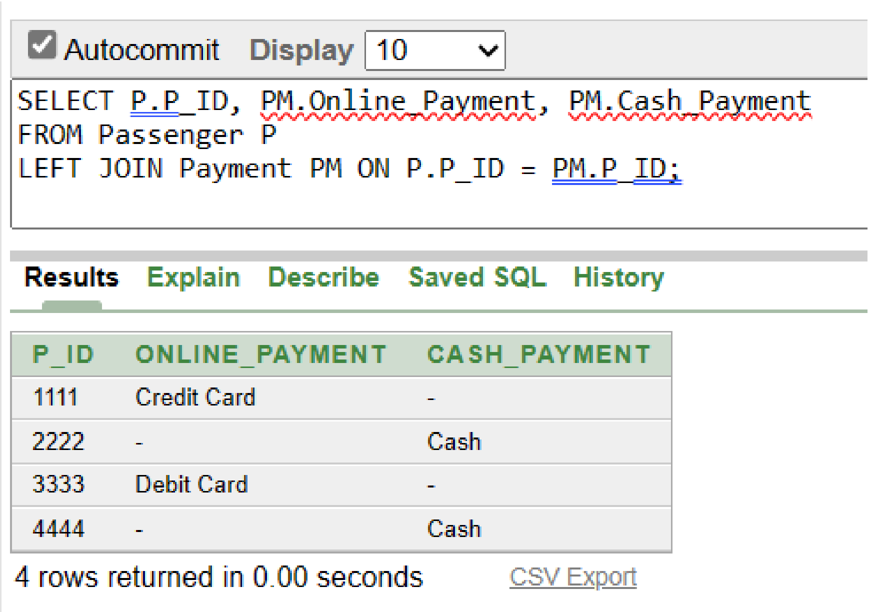
```
SELECT T.No_Passenger, Destination, Source, T.Station
FROM Ticket T
INNER JOIN Passenger P ON T.P_ID = P.P_ID
WHERE P.P_Age <= 22;
```

NO_PASSENGER	DESTINATION	SOURCE	STATION
1	Dhaka	Chattogram	Kamalapur station
1	Khulna	Rajshahi	Dhaka cantonment station
1	Dhaka	Khulna	Kamalapur station

3 rows returned in 0.00 seconds

● Joining (2):

Question: Show the passenger IDs and their respective payment methods, including passengers who haven't made any payments.



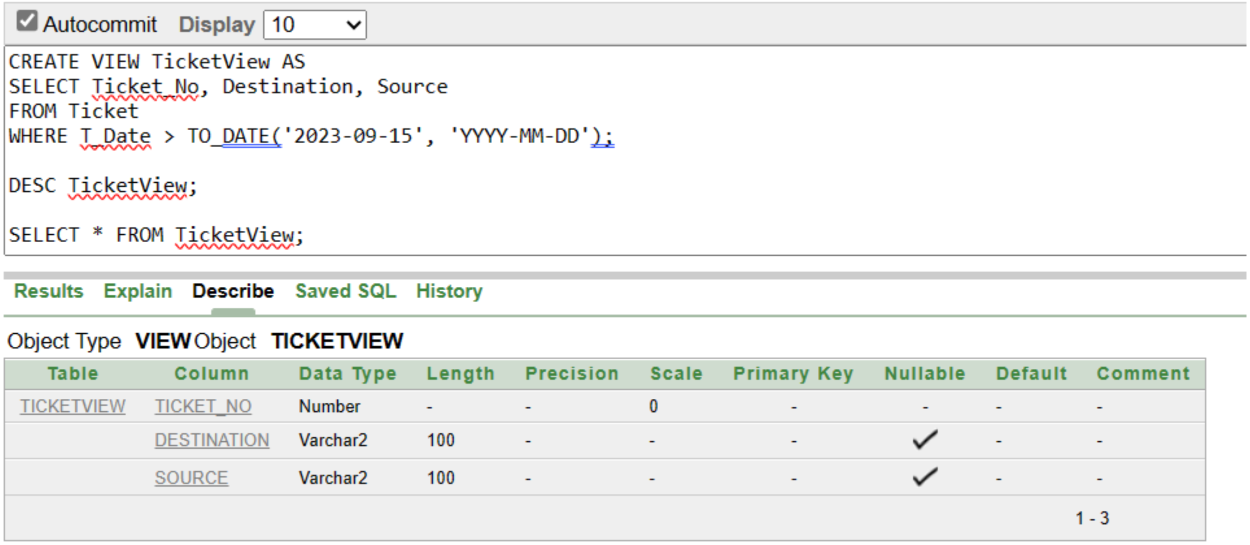
```
SELECT P.P_ID, PM.Online_Payment, PM.Cash_Payment
FROM Passenger P
LEFT JOIN Payment PM ON P.P_ID = PM.P_ID;
```

P_ID	ONLINE_PAYMENT	CASH_PAYMENT
1111	Credit Card	-
2222	-	Cash
3333	Debit Card	-
4444	-	Cash

4 rows returned in 0.00 seconds

● View (1):

Question: Create a view of the Ticket table showing Ticket no, Destination & source where the date is greater than 15 september.

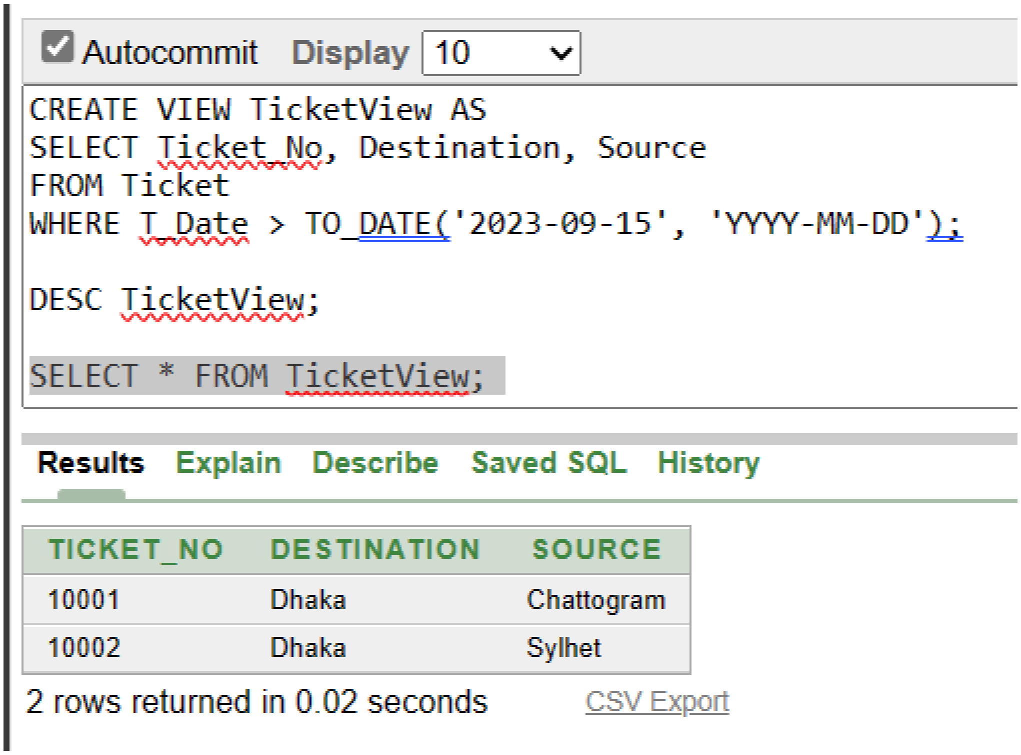


```
CREATE VIEW TicketView AS
SELECT Ticket_No, Destination, Source
FROM Ticket
WHERE T_Date > TO_DATE('2023-09-15', 'YYYY-MM-DD');

DESC TicketView;

SELECT * FROM TicketView;
```

Object Type **VIEW** Object **TICKETVIEW**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
TICKETVIEW	TICKET_NO	Number	-	-	0	-	-	-	-
	DESTINATION	Varchar2	100	-	-	-	✓	-	-
	SOURCE	Varchar2	100	-	-	-	✓	-	-

1 - 3



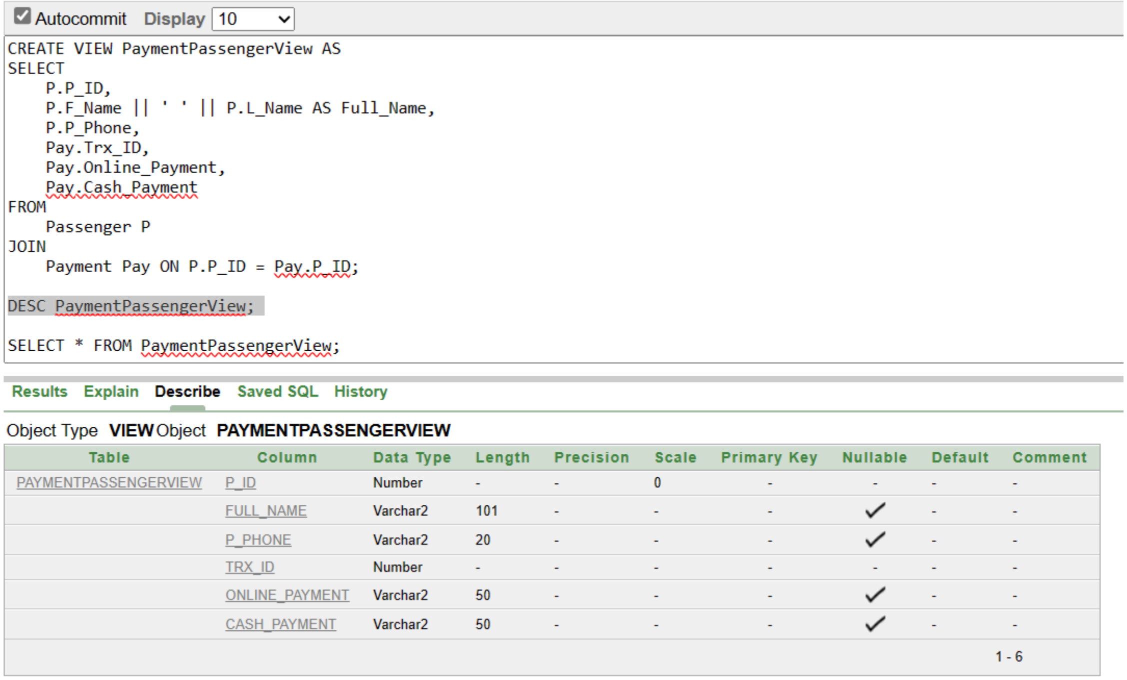
```
CREATE VIEW TicketView AS
SELECT Ticket_No, Destination, Source
FROM Ticket
WHERE T_Date > TO_DATE('2023-09-15', 'YYYY-MM-DD');

DESC TicketView;

SELECT * FROM TicketView;
```

TICKET_NO	DESTINATION	SOURCE
10001	Dhaka	Chattogram
10002	Dhaka	Sylhet

2 rows returned in 0.02 seconds

- **View (2):**

Question: Create a view to show the payment details along with the full name and contact information of the passengers who made the payment.



☑ Autocommit Display 10

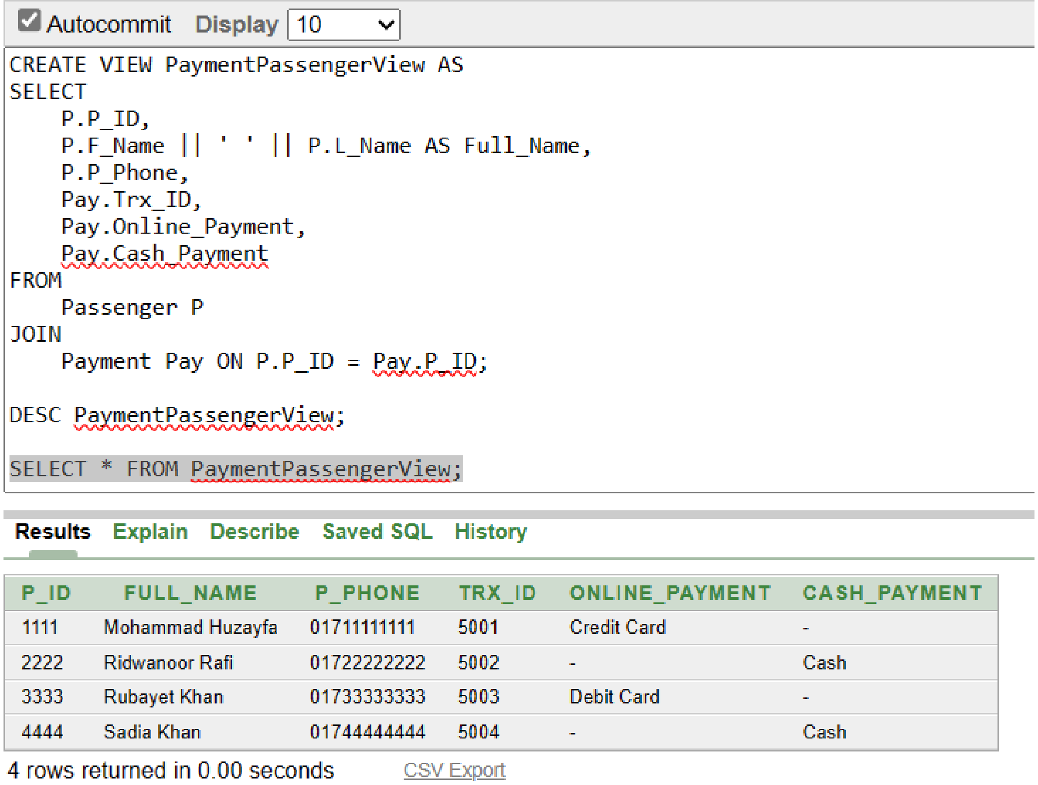
```
CREATE VIEW PaymentPassengerView AS
SELECT
    P.P_ID,
    P.F_Name || ' ' || P.L_Name AS Full_Name,
    P.P_Phone,
    Pay.Trx_ID,
    Pay.Online_Payment,
    Pay.Cash_Payment
FROM
    Passenger P
JOIN
    Payment Pay ON P.P_ID = Pay.P_ID;

DESC PaymentPassengerView;

SELECT * FROM PaymentPassengerView;
```

Results Explain **Describe** Saved SQL History

Object Type **VIEW** Object **PAYMENTPASSENGERVIEW**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PAYMENTPASSENGERVIEW	P_ID	Number	-	-	0	-	-	-	-
	FULL_NAME	Varchar2	101	-	-	-	✓	-	-
	P_PHONE	Varchar2	20	-	-	-	✓	-	-
	TRX_ID	Number	-	-	-	-	-	-	-
	ONLINE_PAYMENT	Varchar2	50	-	-	-	✓	-	-
	CASH_PAYMENT	Varchar2	50	-	-	-	✓	-	-

1 - 6



☑ Autocommit Display 10

```
CREATE VIEW PaymentPassengerView AS
SELECT
    P.P_ID,
    P.F_Name || ' ' || P.L_Name AS Full_Name,
    P.P_Phone,
    Pay.Trx_ID,
    Pay.Online_Payment,
    Pay.Cash_Payment
FROM
    Passenger P
JOIN
    Payment Pay ON P.P_ID = Pay.P_ID;

DESC PaymentPassengerView;

SELECT * FROM PaymentPassengerView;
```

Results Explain Describe Saved SQL History

P_ID	FULL_NAME	P_PHONE	TRX_ID	ONLINE_PAYMENT	CASH_PAYMENT
1111	Mohammad Huzayfa	01711111111	5001	Credit Card	-
2222	Ridwanoor Rafi	01722222222	5002	-	Cash
3333	Rubayet Khan	01733333333	5003	Debit Card	-
4444	Sadia Khan	01744444444	5004	-	Cash

4 rows returned in 0.00 seconds CSV Export

Conclusion

For our team, creating and implementing the MetroControl Railway Management System has proven to be a great accomplishment. In order to manage a contemporary metro rail system, we have developed a comprehensive framework. The project effectively illustrated the ability to manage multiple metro rail domain facets. The railway system's operations and data administration are now frictionless thanks to the seamless relationships between entities.

There are numerous opportunities for the project's expansion and improvement in the future. Enhanced User Experience, Advanced Security Features, and Real-time Data Management are a few possible topics to concentrate on.

The project may continue to develop and adapt to the changing environment of metro rail management by addressing user needs, adding cutting-edge technologies, and ensuring scalability. The entire railway ecosystem will profit from this ongoing work, which will help make travel for passengers more efficient and seamless.