



# **AMERICAN INTERNATIONAL UNIVERSITY–BANGLADESH (AIUB)**

## **Faculty of Science and Technology**

### **DEPARTMENT OF COMPUTER SCIENCE**

#### **ADVANCE DATABASE MANAGEMENT SYSTEM**

**Fall 2025-2026**

**Section: A**

**Title**

**Final Term Project Documentation**

**Supervised By**

**JUENA AHMED NOSHIN**

**Submitted by:**

SERIAL NO.	NAME	ID	TASK
1	HALDER, DIAMOND	22- 48152-2	Website, Normalizatin(25%)
2	KOUSHIK SAHA	22-47995-2	Table Createion, Query, Exception Handling(25%)
3	USHMITA MALLICK	22-47703-2	Normalization, Schema Diagram(25%)
4	ANIKA TAHSIN AUNU	22-47848-2	Relational Algebra, Exception Handling(25%)

**Date of Submission:** 4 January, 2026

## Table of Contents

Introduction .....	3
Project Proposal.....	3
User Interface Planning.....	97
Scenario Description .....	6
ER Diagram .....	7
Normalization .....	8
Schema Diagram.....	20
Table Creation Using SQL.....	21
Data Insertion Using SQL .....	34
Basic PL/SQL .....	45
Advance PL/SQL.....	61
Conclusion.....	77
Added exception handling section to Advance PL/SQL Codes.....	77
Relational Algebra.....	96
Bonus.....	97

## Projects Updates

### Introduction

Urban transportation systems play a vital role in supporting the economic growth and daily mobility of large cities. With the rapid increase in population and traffic congestion, efficient mass transit solutions have become essential. Metro rail systems provide a fast, reliable, and environmentally friendly mode of transportation, making them a critical component of modern urban infrastructure. Managing such a complex system requires accurate data handling, secure operations, and efficient coordination between passengers and operational staff.

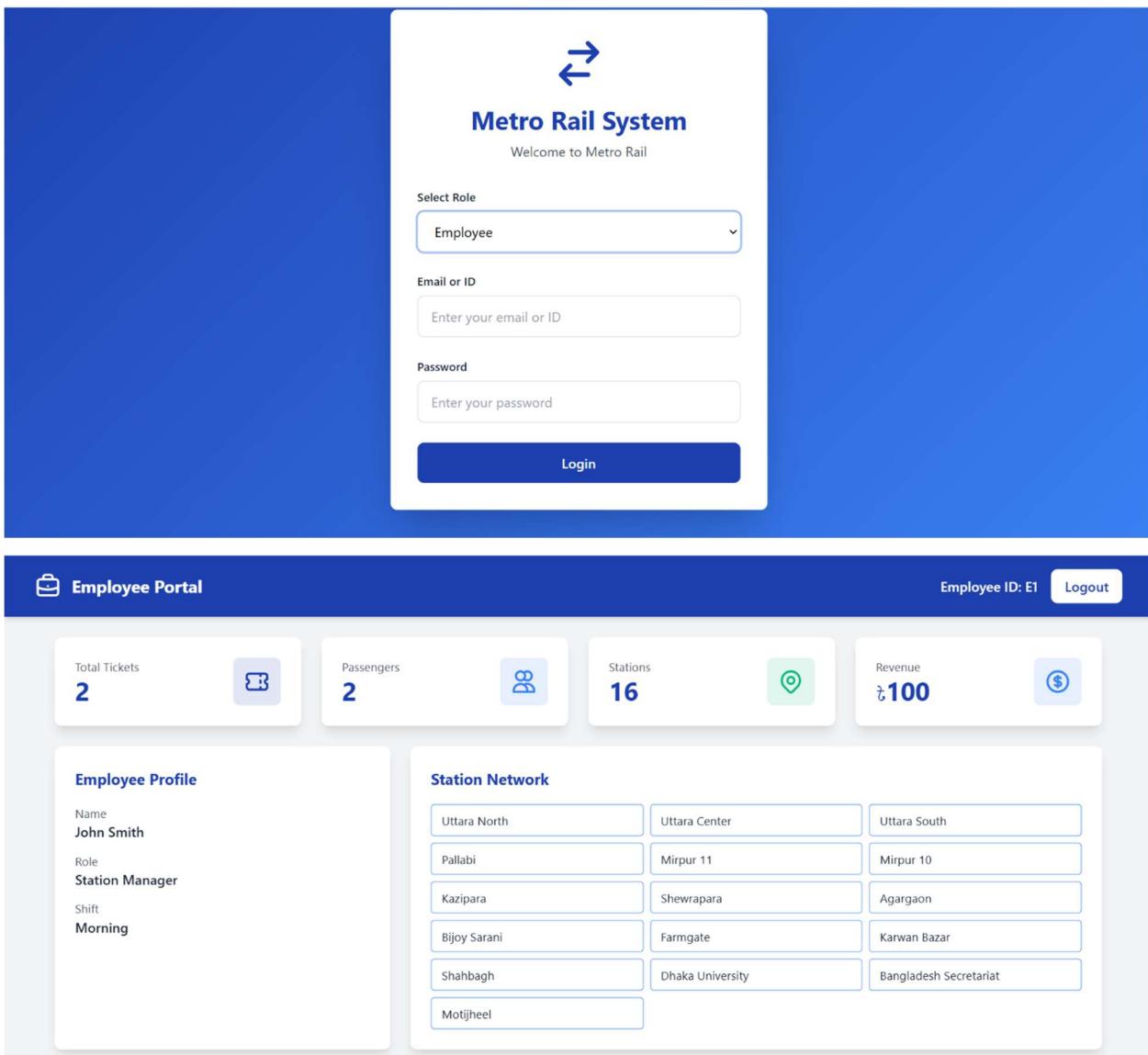
The Metro Rail Management System is a web-based application developed to digitalize and simplify metro rail operations and passenger services. The system is designed to manage passenger registration, ticket booking, fixed fare display, train routes, schedules, and maintenance activities in a structured manner. By using a centralized database and a well-defined Entity Relationship (ER) model, the system ensures data integrity, minimizes redundancy, and supports smooth information flow across different modules.

This project focuses on improving operational efficiency, enhancing passenger convenience, and reducing dependency on manual processes. The system is modeled based on real-world metro rail operations, with special relevance to the Dhaka Metro Rail system, making it practical, scalable, and suitable for real-life implementation.

### Project Proposal

This project proposes the design and development of a comprehensive Metro Rail Management System that integrates metro operations, workforce management, passenger services, ticketing, payments, fares, and maintenance into a single, well-structured database system. The system focuses on efficiently managing metro lines, trains, stations, routes, and schedules while ensuring accurate coordination between employees and operational activities. It enables passengers to seamlessly register, book tickets, and complete secure payments based on defined fares and travel routes. At the operational level, the project supports train scheduling, employee assignment, and maintenance logging to ensure service reliability and safety. By maintaining strong relationships among entities such as Line, Station, Train, Employee, Passenger, Ticket, and Payment, the proposed system aims to improve data consistency, reduce manual errors, and enhance overall service quality. Overall, the proposed Metro Rail Management System aims to digitalize metro rail management processes, improve operational efficiency, enhance passenger experience, and provide a secure and scalable solution suitable for real-world metro transportation systems such as the Dhaka Metro Rail.

# User Interface Planning



The image displays two screenshots of a Metro Rail System interface. The top screenshot shows the login page with a blue header containing a double-headed arrow icon and the text "Metro Rail System". Below the header is a welcome message "Welcome to Metro Rail". A dropdown menu labeled "Select Role" shows "Employee" selected. There are input fields for "Email or ID" and "Password", both with placeholder text "Enter your email or ID" and "Enter your password". A blue "Login" button is at the bottom. The bottom screenshot shows the Employee Portal home screen with a dark blue header featuring a briefcase icon, "Employee Portal", "Employee ID: E1", and a "Logout" button. Below the header are four summary cards: "Total Tickets 2" (with a ticket icon), "Passengers 2" (with a person icon), "Stations 16" (with a location pin icon), and "Revenue ₢100" (with a dollar sign icon). The main content area contains two sections: "Employee Profile" (listing Name: John Smith, Role: Station Manager, Shift: Morning) and "Station Network" (listing stations in a grid: Uttara North, Pallabi, Kazipara, Bijoy Sarani, Shahbagh, Motijheel; Uttara Center, Mirpur 11, Shewrapara, Farmgate, Dhaka University; Uttara South, Mirpur 10, Agargaon, Karwan Bazar, Bangladesh Secretariat).

**Metro Rail System**

Welcome to Metro Rail

Select Role

Passenger

Email or ID

Enter your email or ID

Password

Enter your password

**Login**

Don't have an account? [Register as Passenger](#)

**Passenger Registration**

Full Name

Enter full name

Gender

Select Gender

Date of Birth

dd----yyyy

National ID

Enter NID (This will be your password)

Your NID will be used as your password for login

Phone Number

Enter phone number

Email Address

Enter email

Address

Enter complete address

**Register**

### Book Your Ticket

Select Current Station

Choose your station

Select Destination Station

Choose destination

**Buy Ticket**

### My Tickets

Ticket #TKT1765694752715 Uttara North → Shahbagh 14/12/2025	₹50	Active
Ticket #TKT1765696451973 Uttara North → Shahbagh 14/12/2025	₹50	Active

### Fare Information

**Fixed Fare**

**₹50**

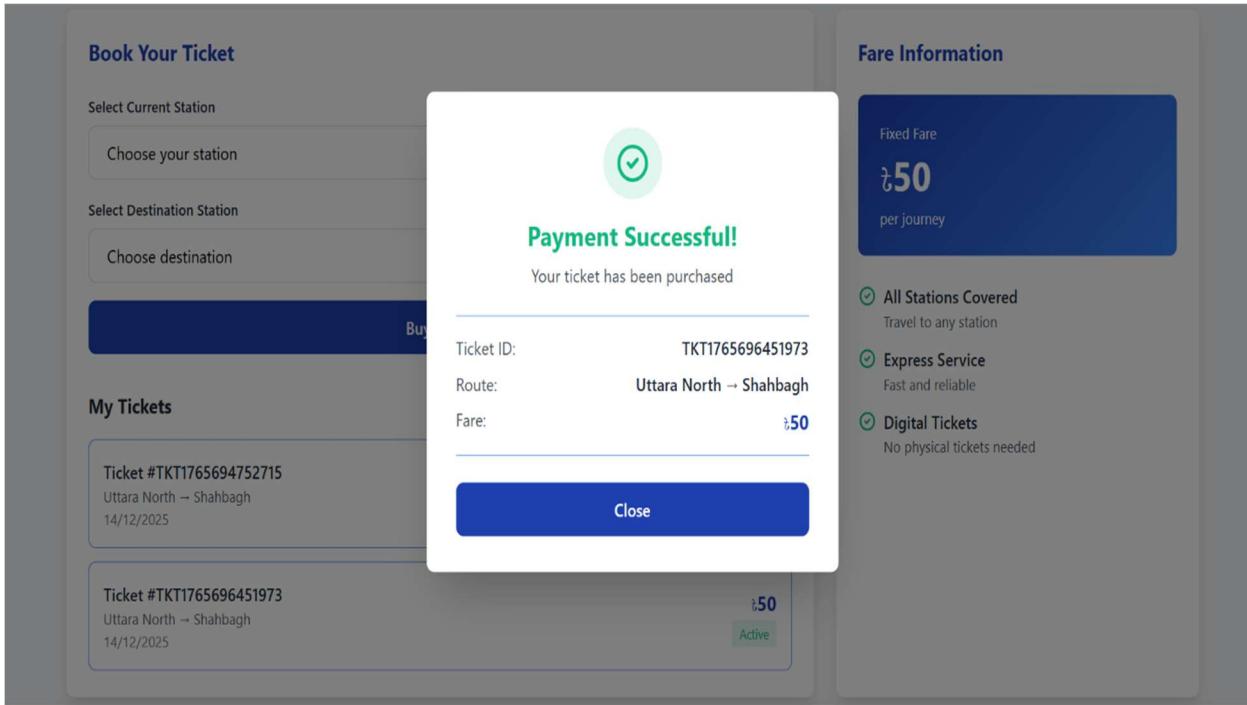
per journey

- All Stations Covered
- Express Service
- Digital Tickets

Travel to any station

Fast and reliable

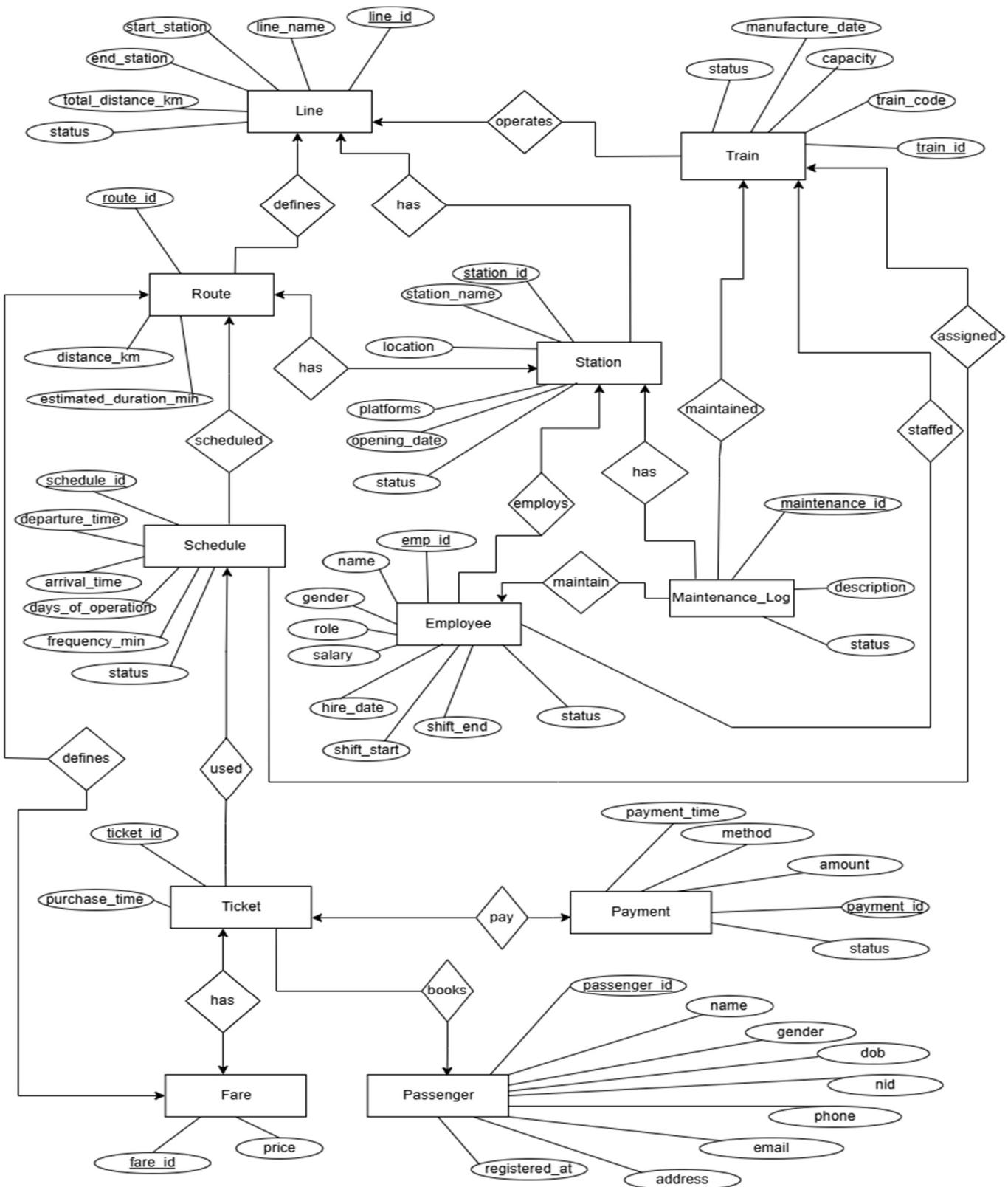
No physical tickets needed



## Scenario Description

The Metro Rail Management System is designed to efficiently manage metro operations, passengers, employees, ticketing, payments, fares, and maintenance activities. In this system, each Line is uniquely identified by a line\_id and is responsible for operating multiple Trains, managing several Stations, and defining various Routes. A Line also maintains records of its line\_name, start\_station, end\_station, total\_distance\_km, and operational status. Each Station and Train is dedicated to exactly one Line. Stations track information such as station\_name, location, number of platforms, opening\_date, and current status. Every Route has route\_id as primary key, distance\_km and estimated\_duration\_min. These routes are organized into multiple Schedules. Each Route defines a specific Fare that determines the price based on the travel distance. Each Schedule is assigned to a specific Train, where Employees are also assigned to manage the coordination between trains and their timings. A schedule includes departure\_time, arrival\_time, days\_of\_operation, frequency\_min, and status, which is then used to issue Tickets. The Train fleet is managed by storing train\_code, capacity, manufacture\_date, and status. To ensure smooth service, trains are staffed by multiple Employees and undergo regular maintenance. All maintenance activities are recorded in a Maintenance\_Log, detailing the description and status of the work. Each maintenance task is performed by an Employee at a specific Station. Stations employ a diverse workforce. Each Station can employ multiple Employees. Employees are registered with their name, gender, role, salary, hire\_date, shift\_start, shift\_end, and employment status. They are responsible for both daily operations and participating in maintenance activities. The system also manages Passenger services. Each passenger is identified by a passenger\_id and provides details such as name, gender, dob, nid, phone, email, address, and their account registered\_at time. A Passenger can book multiple Tickets, but each Ticket is linked to only one passenger and a specific Schedule. Each ticket also records the purchase\_time and is associated with a Fare that determines the price based on the from\_station and to\_station. To complete the process, every Ticket must have a corresponding Payment, maintaining a strict one to one relationship. The Payment record includes a payment\_id, amount, method, payment\_time, and transaction status. This integrated approach ensures seamless coordination of metro operations, staff management, and passenger satisfaction.

# ER Diagram



# Normalization

## 1. Line has Station

### UNF

has (line\_id , line\_name, start\_station, end\_station, total\_distance\_km, status, station\_id, station\_name, location, platforms, opening\_date, status)

### 1NF

There is no multi valued attribute.

1. line\_id , line\_name, start\_station, end\_station, total\_distance\_km, status, station\_id, station\_name, location, platforms, opening\_date, status

### 2NF

1. line\_id , line\_name, start\_station, end\_station, total\_distance\_km, status
2. station\_id , station\_name, location, platforms, opening\_date, status

### 3NF

There is no transitive dependency.

1. line\_id, line\_name, start\_station, end\_station, total\_distance\_km, status
2. station\_id, station\_name, location, platforms, opening\_date, status

## Table creation

1. line\_id , line\_name, start\_station, end\_station, total\_distance\_km, status
2. station\_id, station\_name, location, opening\_date, status, line\_id

## 2 . Line operates Train

### UNF

operates ( line\_id, line\_name, start\_station, end\_station, total\_distance\_km, status, train\_id, train\_code, capacity, manufacture\_date, status)

### 1NF

There is no multi valued attribute

- 1 line\_id, line\_name, start\_station, end\_station, total\_distance\_km, status, train\_id, train\_code, capacity, manufacture\_date, status

### 2NF

1. line\_id, line\_name, start\_station, end\_station, total\_distance\_km, status
2. train\_id, train\_code, capacity, manufacture\_date, status

### **3NF**

There is no transitive dependency.

1. line\_id, line\_name, start\_station, end\_station, total\_distance\_km, status
2. train\_id, train\_code, capacity, manufacture\_date, status

### **Table creation**

1. line\_id, line\_name, start\_station, end\_station, total\_distance\_km, status
2. train\_id, train\_code, capacity, manufacture\_date, status, **line\_id**

### **3.Line defines Route**

#### **UNF**

defines (line\_id, line\_name, start\_station, end\_station, total\_distance\_km, status, route\_id, distance\_km, estimated\_duration\_min)

#### **1NF**

There is no multi valued attribute

- 1 line\_id , line\_name, start\_station, end\_station, total\_distance\_km, status, route\_id , distance\_km, estimated\_duration\_min

#### **2NF**

1. line\_id, line\_name, start\_station, end\_station, total\_distance\_km, status  
route\_id,distance\_km, estimated\_duration\_min

### **3NF**

1. line\_id, line\_name, start\_station, end\_station, total\_distance\_km, status
2. route\_id ,distance\_km, estimated\_duration\_min

### **Table creation**

1. line\_id, line\_name, start\_station, end\_station, total\_distance\_km, status
2. route\_id ,distance\_km, estimated\_duration\_min, **line\_id**

### **4.Route scheduled Schedule**

#### **UNF**

schedule (route\_id, distance\_km, estimated\_duration\_min, schedule\_id departure\_time, arrival\_time, days\_of\_operation, frequency\_min, status)

## **1NF**

There is no multi-valued attribute.

1. route\_id, distance\_km, estimated\_duration\_min, schedule\_id, departure\_time, arrival\_time, days\_of\_operation, frequency\_min, status

## **2NF**

1. route\_id, distance\_km, estimated\_duration\_min
2. schedule\_id, departure\_time, arrival\_time, days\_of\_operation, frequency\_min, status

## **3NF**

There is no transitive dependency.

1. route\_id, distance\_km, estimated\_duration\_min
2. schedule\_id, departure\_time, arrival\_time, days\_of\_operation, frequency\_min, status

## **Table creation**

1. route\_id, distance\_km, estimated\_duration\_min
2. schedule\_id, departure\_time, arrival\_time, days\_of\_operation, frequency\_min, status, route\_id

## **5. Route has Station**

### **UNF**

has(route\_id, distance\_km, estimated\_duration\_min, station\_id, station\_name, location, platforms, opening\_date, status)

## **1NF**

There is no multi-valued attribute.

1. route\_id, distance\_km, estimated\_duration\_min, station\_id, station\_name, location, platforms, opening\_date, status

## **2NF**

1. route\_id, distance\_km, estimated\_duration\_min
2. station\_id, station\_name, location, platforms, opening\_date, status

## **3NF**

There is no transitive dependency.

1. route\_id, distance\_km, estimated\_duration\_min
2. station\_id, station\_name, location, platforms, opening\_date, status

## **Table creation**

1. route\_id, from\_station\_id, to\_station\_id, distance\_km, estimated\_duration\_min
2. station\_id, station\_name, location, platforms, opening\_date, status

## **6. Route defines Fare**

### **UNF**

defines(route\_id, distance\_km, estimated\_duration\_min, fare\_id, price)

### **1NF**

There is no multi-valued attribute.

1. route\_id, distance\_km, estimated\_duration\_min, fare\_id, price

### **2NF**

1. route\_id, distance\_km, estimated\_duration\_min
2. fare\_id, price

### **3NF**

There is no transitive dependency.

1. route\_id, distance\_km, estimated\_duration\_min
2. fare\_id, price

### **Table creation**

1. route\_id, distance\_km, estimated\_duration\_min, fare\_id
2. fare\_id, price

## **7. Train assigned Schedule**

### **UNF**

assigned (train\_id, train\_code, capacity, manufacture\_date, status, schedule\_id, departure\_time, arrival\_time, days\_of\_operation, frequency\_min, status)

### **1NF**

There is no multi valued attribute.

1. train\_id, train\_code, capacity, manufacture\_date, status, schedule\_id, departure\_time, arrival\_time, days\_of\_operation, frequency\_min, status

### **2NF**

1. train\_id, train\_code, capacity, manufacture\_date, status
2. schedule\_id, departure\_time, arrival\_time, days\_of\_operation, frequency\_min, status

### **3NF**

There is no transitive dependency.

1. train\_id, train\_code, capacity, manufacture\_date, status
2. schedule\_id, departure\_time, arrival\_time, days\_of\_operation, frequency\_min, status

### **Table creation**

1. train\_id, train\_code, capacity, manufacture\_date, status
2. schedule\_id, departure\_time, arrival\_time, days\_of\_operation, frequency\_min, status, **train\_id**

## **8. Schedule used Ticket**

### **UNF**

used (schedule\_id, departure\_time, arrival\_time, days\_of\_operation, frequency\_min, status, ticket\_id, purchase\_time)

### **1NF**

There is no multi valued attribute.

1. schedule\_id, departure\_time, arrival\_time, days\_of\_operation, frequency\_min, status, ticket\_id, purchase\_time

### **2NF**

1. schedule\_id, departure\_time, arrival\_time, days\_of\_operation, frequency\_min, status
2. ticket\_id, purchase\_time

### **3NF**

There is no transitive dependency.

1. schedule\_id, departure\_time, arrival\_time, days\_of\_operation, frequency\_min, status
2. ticket\_id, purchase\_time

### **Table creation**

1. schedule\_id, departure\_time, arrival\_time, days\_of\_operation, frequency\_min, status
2. ticket\_id, purchase\_time, **schedule\_id**

## **9. Passenger books Ticket**

### **UNF**

books ( passenger\_id, name, gender, dob, nid, phone, email, address, registered\_at, ticket\_id, purchase\_time)

## 1NF

There is no multi valued attribute.

1. passenger\_id, name, gender, dob, nid, phone, email, address, registered\_at, ticket\_id, purchase\_time

## 2NF

1. passenger\_id, name, gender, dob, nid, phone, email, address, registered\_at
2. ticket\_id, purchase\_time

## 3NF

There is no transitive dependency.

1. passenger\_id, name, gender, dob, nid, phone, email, address, registered\_at
2. ticket\_id, purchase\_time

## Table creation

1. passenger\_id, name, gender, dob, nid, phone, email, address, registered\_at
2. ticket\_id, purchase\_time, **passenger\_id**

## 10. Ticket pay Payment

### UNF

pay (payment\_id, method, amount, payment\_time, status, ticket\_id, purchase\_time)

## 1NF

There is no multi valued attribute.

1. payment\_id, method, amount, payment\_time, status, ticket\_id, purchase\_time

## 2NF

1. payment\_id, method, amount, payment\_time, status
2. ticket\_id, purchase\_time

## 3NF

There is no transitive dependency.

1. payment\_id, method, amount, payment\_time, status
2. ticket\_id, purchase\_time

## Table creation

1. payment\_id, method, amount, payment\_time, status
2. ticket\_id, purchase\_time, **payment\_id**

## **11. Ticket has Fare**

### **UNF**

has(ticket\_id, purchase\_time, fare\_id, price)

### **1NF**

There is no multi valued attribute.

1. ticket\_id, purchase\_time, fare\_id, from\_station, to\_station, price

### **2NF**

1. ticket\_id, purchase\_time

2. fare\_id, price

### **3NF**

There is no transitive dependency.

1. ticket\_id, purchase\_time

2. fare\_id, price

### **Table creation**

1. ticket\_id, purchase\_time, fare\_id

2. fare\_id, price

## **12. Station employs Employee**

### **UNF**

employs ( station\_id, station\_name, location, platforms, opening\_date, status, emp\_id, name, gender, role, salary, hire\_date, shift\_start, shift\_end, status)

### **1NF**

There is no multi valued attribute.

1. station\_id, station\_name, location, platforms, opening\_date, status, emp\_id, name, gender, role, salary, hire\_date, shift\_start, shift\_end, status

### **2NF**

1. station\_id, station\_name, location, platforms, opening\_date, status

2. emp\_id, name, gender, role, salary, hire\_date, shift\_start, shift\_end, status

## **3NF**

There is no transitive dependency.

1. station\_id, station\_name, location, platforms, opening\_date, status
2. emp\_id, name, gender, role, salary, hire\_date, shift\_start, shift\_end, status

## **Table creation**

1. station\_id, station\_name, location, platforms, opening\_date, status
2. emp\_id, name, gender, role, salary, hire\_date, shift\_start, shift\_end, status, **station\_id**

## **13. Train staffed Employee**

### **UNF**

staffed (train\_id, train\_code, capacity, manufacture\_date, status, emp\_id, name, gender, role, salary, hire\_date, shift\_start, shift\_end, status)

### **1NF**

There is no multi valued attribute.

1 train\_id, train\_code, capacity, manufacture\_date, status, emp\_id, name, gender, role, salary, hire\_date, shift\_start, shift\_end, status

### **2NF**

1. train\_id, train\_code, capacity, manufacture\_date, status
2. emp\_id, name, gender, role, salary, hire\_date, shift\_start, shift\_end, status

## **3NF**

There is no transitive dependency.

1. train\_id, train\_code, capacity, manufacture\_date, status
2. emp\_id, name, gender, role, salary, hire\_date, shift\_start, shift\_end, status

## **Table creation**

1. train\_id, train\_code, capacity, manufacture\_date, status
2. emp\_id, name, gender, role, salary, hire\_date, shift\_start, shift\_end, status, **train\_id**

## **14. Train maintained Maintenance Log**

### **UNF**

maintained (train\_id, train\_code, capacity, manufacture\_date, status, maintenance\_id, description, status)

## **1NF**

There is no multi valued attribute.

1 train\_id , train\_code, capacity, manufacture\_date, status, maintenance\_id, description, status

## **2NF**

1. train\_id , train\_code, capacity, manufacture\_date, status
2. maintenance\_id, description, status

## **3NF**

There is no transitive dependency.

1. train\_id , train\_code, capacity, manufacture\_date, status
2. maintenance\_id, description, status

## **Table creation**

1. train\_id, train\_code, capacity, manufacture\_date, status,
2. maintenance\_id, description, status, **train\_id**

## **15. Employee maintain Maintenance Log**

### **UNF**

maintain (emp\_id, name, gender, role, salary, hire\_date, shift\_start, shift\_end, status, maintenance\_id,  
description, status)

## **1NF**

There is no multi valued attribute.

1 emp\_id, name, gender, role, salary, hire\_date, shift\_start, shift\_end, status, maintenance\_id,  
description, status

## **2NF**

1. emp\_id, name, gender, role, salary, hire\_date, shift\_start, shift\_end, status
2. maintenance\_id, description, status

## **3NF**

There is no transitive dependency.

1. emp\_id, name, gender, role, salary, hire\_date, shift\_start, shift\_end, status
2. maintenance\_id, description, status

## Table creation

1. emp\_id, name, gender, role, salary, hire\_date, shift\_start, shift\_end, status
2. maintenance\_id, description, status, **emp\_id**

## **16. Station has Maintenance Log**

### UNF

has (station\_id, station\_name, location, platforms, opening\_date, status, maintenance\_id, description, status)

### 1NF

There is no multi-valued attribute.

1 station\_id, station\_name, location, platforms, opening\_date, status, maintenance\_id, description, status

### 2NF

1. station\_id, station\_name, location, platforms, opening\_date, status
2. maintenance\_id, description, status

### 3NF

1. station\_id, station\_name, location, platforms, opening\_date, status
2. maintenance\_id, description, status

## Table creation

1. station\_id, station\_name, location, platforms, opening\_date, status
2. maintenance\_id, description, status, **station\_id**

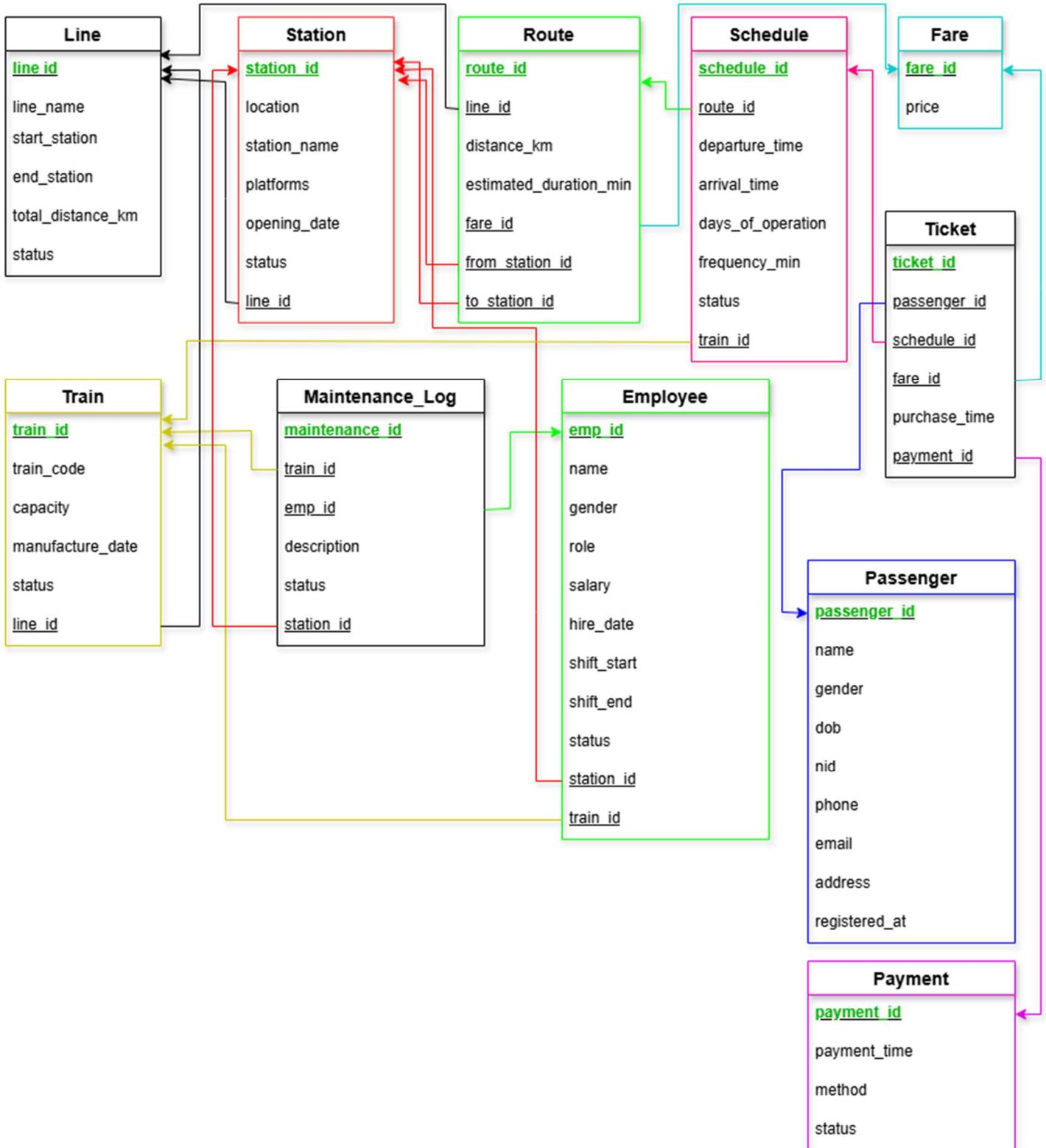
## Temporary Table

1. line\_id, line\_name, start\_station, end\_station, total\_distance\_km, status
2. station\_id, station\_name, location, opening\_date, status, **line\_id**
3. line\_id, line\_name, start\_station, end\_station, total\_distance\_km, status,
4. train\_id, train\_code, capacity, manufacture\_date, status, **line\_id**
5. line\_id, line\_name, start\_station, end\_station, total\_distance\_km, status
6. route\_id, distance\_km, estimated\_duration\_min, **line\_id**
7. route\_id, distance\_km, estimated\_duration\_min,
8. schedule\_id, departure\_time, arrival\_time, days\_of\_operation, frequency\_min, status, **route\_id**
9. route\_id, **from\_station\_id**, **to\_station\_id**, distance\_km, estimated\_duration\_min
10. station\_id, station\_name, location, platforms, opening\_date, status
11. route\_id, distance\_km, estimated\_duration\_min, **fare\_id**
12. fare\_id, price
13. train\_id, train\_code, capacity, manufacture\_date, status
14. schedule\_id, departure\_time, arrival\_time, days\_of\_operation, frequency\_min, status, **train\_id**
15. schedule\_id, departure\_time, arrival\_time, days\_of\_operation, frequency\_min, status
16. ticket\_id, purchase\_time, **schedule\_id**
17. passenger\_id, name, gender, dob, nid, phone, email, address, registered\_at
18. ticket\_id, purchase\_time, **passenger\_id**
19. payment\_id, method, amount, payment\_time, status
20. ticket\_id, purchase\_time, **payment\_id**
21. ticket\_id, purchase\_time, **fare\_id**
22. fare\_id, price
23. station\_id, station\_name, location, platforms, opening\_date, status
24. emp\_id, name, gender, role, salary, hire\_date, shift\_start, shift\_end, status, **station\_id**
25. train\_id, train\_code, capacity, manufacture\_date, status
26. emp\_id, name, gender, role, salary, hire\_date, shift\_start, shift\_end, status, **train\_id**
27. train\_id, train\_code, capacity, manufacture\_date, status,
28. maintenance\_id, description, status, **train\_id**
29. emp\_id, name, gender, role, salary, hire\_date, shift\_start, shift\_end, status
30. maintenance\_id, description, status, **emp\_id**
31. station\_id, station\_name, location, platforms, opening\_date, status
32. maintenance\_id, description, status, **station\_id**

## Final Table

1. line\_id, line\_name, start\_station, end\_station, total\_distance\_km, status
2. station\_id, station\_name, location, opening\_date, status, **line\_id**
3. route\_id, **from\_station\_id**, **to\_station\_id**, distance\_km, estimated\_duration\_min, **line\_id**, **fare\_id**
4. train\_id, train\_code, capacity, manufacture\_date, status, **line\_id**
5. fare\_id, price
6. schedule\_id, departure\_time, arrival\_time, days\_of\_operation, frequency\_min, status, **route\_id**, **train\_id**
7. ticket\_id, purchase\_time, **schedule\_id**, **passenger\_id**, **payment\_id**, **fare\_id**
8. payment\_id, method, amount, payment\_time, status
9. passenger\_id, name, gender, dob, nid, phone, email, address, registered\_at
10. emp\_id, name, gender, role, salary, hire\_date, shift\_start, shift\_end, status, **station\_id**, **train\_id**
11. maintenance\_id, description, status, **train\_id**, **station\_id**, **emp\_id**

# Schema Diagram



# Table Creation Using SQL

## Create users, assign roles and grant privileges

```
CREATE USER adms IDENTIFIED BY adms;
```

```
GRANT UNLIMITED TABLESPACE TO adms;
```

```
CREATE ROLE adms_info;
```

```
GRANT CREATE TABLE TO adms_info;
```

```
GRANT CREATE VIEW TO adms_info;
```

```
GRANT CREATE SEQUENCE TO adms_info;
```

```
GRANT CREATE TRIGGER TO adms_info;
```

```
GRANT CREATE SESSION TO adms;
```

```
GRANT CONNECT, RESOURCE TO adms;
```

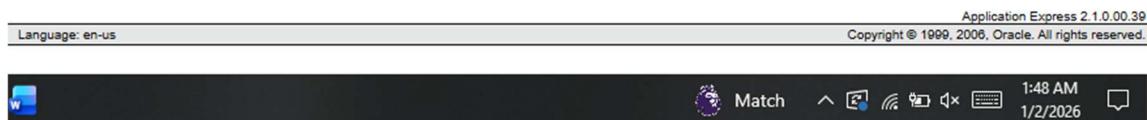
```
GRANT adms_info TO adms;
```

The screenshot shows the Oracle Application Express SQL Commands interface. The top navigation bar includes 'Home > SQL > SQL Commands'. Below the navigation is a toolbar with 'Autocommit' checked, 'Display' set to 10, and buttons for 'Save' and 'Run'. The main area contains the SQL script for creating a user and granting various privileges. The last line of the script, 'GRANT adms\_info TO adms;', is highlighted in blue. At the bottom of the interface, there are tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History', with 'Results' being the active tab.

```
CREATE USER adms IDENTIFIED BY adms;
GRANT UNLIMITED TABLESPACE TO adms;
CREATE ROLE adms_info;
GRANT CREATE TABLE TO adms_info;
GRANT CREATE VIEW TO adms_info;
GRANT CREATE SEQUENCE TO adms_info;
GRANT CREATE TRIGGER TO adms_info;
GRANT CREATE SESSION TO adms;
GRANT CONNECT, RESOURCE TO adms;
GRANT adms_info TO adms;
```

Statement processed.

0.00 seconds



## Line Table

```
CREATE TABLE Line (
    line_id NUMBER PRIMARY KEY,
    line_name VARCHAR2(100) UNIQUE NOT NULL,
    start_station VARCHAR2(100) NOT NULL,
    end_station VARCHAR2(100) NOT NULL,
    total_distance_km NUMBER CHECK (total_distance_km > 0),
    status VARCHAR2(20)
);
```

Autocommit Display 10 Save Run

```
CREATE TABLE Line (
    line_id NUMBER PRIMARY KEY,
    line_name VARCHAR2(100) UNIQUE NOT NULL,
    start_station VARCHAR2(100) NOT NULL,
    end_station VARCHAR2(100) NOT NULL,
    total_distance_km NUMBER CHECK (total_distance_km > 0),
    status VARCHAR2(20)
);

DESC Line;
```

Results Explain Describe Saved SQL History

Object Type TABLE Object LINE

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
LINE	LINE_ID	Number	-	-	-	1	-	-	-
	LINE_NAME	Varchar2	100	-	-	-	-	-	-
	START_STATION	Varchar2	100	-	-	-	-	-	-
	END_STATION	Varchar2	100	-	-	-	-	-	-
	TOTAL_DISTANCE_KM	Number	-	-	-	-	✓	-	-
	STATUS	Varchar2	20	-	-	-	✓	-	-
1 - 6									



## Station Table

```
CREATE TABLE Station (
    station_id NUMBER PRIMARY KEY,
    station_name VARCHAR2(100) NOT NULL,
    location VARCHAR2(100),
    opening_date DATE,
    status VARCHAR2(20),
    line_id NUMBER,
    CONSTRAINT fk_station_line FOREIGN KEY (line_id)
    REFERENCES Line(line_id)
);
```

Autocommit Display 10 Save Run

```
CREATE TABLE Station (
    station_id NUMBER PRIMARY KEY,
    station_name VARCHAR2(100) NOT NULL,
    location VARCHAR2(100),
    opening_date DATE,
    status VARCHAR2(20),
    line_id NUMBER,
    CONSTRAINT fk_station_line FOREIGN KEY (line_id)
    REFERENCES Line(line_id)
);

DESC Station;
```

Results Explain Describe Saved SQL History

Object Type TABLE Object STATION

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
STATION	STATION_ID	Number	-	-	-	1	-	-	-
	STATION_NAME	Varchar2	100	-	-	-	-	-	-
	LOCATION	Varchar2	100	-	-	-	✓	-	-
	OPENING_DATE	Date	7	-	-	-	✓	-	-
	STATUS	Varchar2	20	-	-	-	✓	-	-
	LINE_ID	Number	-	-	-	-	✓	-	-
1 - 6									

## Fare Table

```
CREATE TABLE Fare (
    fare_id NUMBER PRIMARY KEY,
    price NUMBER
);
```

User: SYSTEM

Home > SQL > SQL Commands

Autocommit Display 10

```
CREATE TABLE Fare (
    fare_id NUMBER PRIMARY KEY,
    price NUMBER
);

DESC Fare;
```

Results Explain Describe Saved SQL History

Object Type TABLE Object FARE

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
FARE	FARE_ID	Number	-	-	-	1	-	-	
	PRICE	Number	-	-	-	-	✓	-	1 - 2



## Route Table

```

CREATE TABLE Route (
    route_id NUMBER PRIMARY KEY,
    from_station_id NUMBER,
    to_station_id NUMBER,
    distance_km NUMBER CHECK (distance_km > 0),
    estimated_duration_min NUMBER CHECK (estimated_duration_min > 0),
    line_id NUMBER,
    fare_id NUMBER,
    CONSTRAINT fk_route_from_station FOREIGN KEY (from_station_id)
        REFERENCES Station(station_id),
    CONSTRAINT fk_route_to_station FOREIGN KEY (to_station_id)
        REFERENCES Station(station_id),
    CONSTRAINT fk_route_line FOREIGN KEY (line_id)
        REFERENCES Line(line_id),
    CONSTRAINT fk_route_fare FOREIGN KEY (fare_id)
        REFERENCES Fare(fare_id)
);

```

Autocommit Display 10 Save Run

```

CREATE TABLE Route (
    route_id NUMBER PRIMARY KEY,
    from_station_id NUMBER,
    to_station_id NUMBER,
    distance_km NUMBER CHECK (distance_km > 0),
    estimated_duration_min NUMBER CHECK (estimated_duration_min > 0),
    line_id NUMBER,
    fare_id NUMBER,
    CONSTRAINT fk_route_from_station FOREIGN KEY (from_station_id)
        REFERENCES Station(station_id),
    CONSTRAINT fk_route_to_station FOREIGN KEY (to_station_id)
        REFERENCES Station(station_id),
    CONSTRAINT fk_route_line FOREIGN KEY (line_id)
        REFERENCES Line(line_id),
    CONSTRAINT fk_route_fare FOREIGN KEY (fare_id)
        REFERENCES Fare(fare_id)
);

DESC Route;

```

Results Explain Describe Saved SQL History

Object Type TABLE Object ROUTE

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ROUTE	ROUTE_ID	Number	-	-	-	1	✓	-	-
	FROM_STATION_ID	Number	-	-	-	-	✓	-	-
	TO_STATION_ID	Number	-	-	-	-	✓	-	-
	DISTANCE_KM	Number	-	-	-	-	✓	-	-
	ESTIMATED_DURATION_MIN	Number	-	-	-	-	✓	-	-
	LINE_ID	Number	-	-	-	-	✓	-	-
	FARE_ID	Number	-	-	-	-	✓	-	-
1 - 7									

Application Express 2.1.0.00.31

Copyright © 1999, 2006, Oracle. All rights reserved

Language: en-us



1:59 AM  
1/2/2026

## Train Table

```
CREATE TABLE Train (
    train_id NUMBER PRIMARY KEY,
    train_code VARCHAR2(20) UNIQUE NOT NULL,
    capacity NUMBER CHECK (capacity > 0),
    manufacture_date DATE,
    status VARCHAR2(20),
    line_id NUMBER,
    CONSTRAINT fk_train_line FOREIGN KEY (line_id)
    REFERENCES Line(line_id)
);
```

Autocommit Display 10 Save Run

```
CREATE TABLE Train (
    train_id NUMBER PRIMARY KEY,
    train_code VARCHAR2(20) UNIQUE NOT NULL,
    capacity NUMBER CHECK (capacity > 0),
    manufacture_date DATE,
    status VARCHAR2(20),
    line_id NUMBER,
    CONSTRAINT fk_train_line FOREIGN KEY (line_id)
    REFERENCES Line(line_id)
);

DESC Train;
```

Results Explain Describe Saved SQL History

Object Type TABLE Object TRAIN

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
TRAIN	TRAIN_ID	Number	-	-	-	1	-	-	-
	TRAIN_CODE	Varchar2	20	-	-	-	-	-	-
	CAPACITY	Number	-	-	-	-	✓	-	-
	MANUFACTURE_DATE	Date	7	-	-	-	✓	-	-
	STATUS	Varchar2	20	-	-	-	✓	-	-
	LINE_ID	Number	-	-	-	-	✓	-	-

1 - 6

Application Express 2.1.0.00.39  
Language: en-us Copyright © 1999, 2006, Oracle. All rights reserved.  
Result 2:01 AM 1/2/2026

## Schedule Table

```
CREATE TABLE Schedule (
    schedule_id NUMBER PRIMARY KEY,
    departure_time VARCHAR2(20),
    arrival_time VARCHAR2(20),
    days_of_operation VARCHAR2(50),
    frequency_min NUMBER CHECK (frequency_min > 0),
    status VARCHAR2(20),
    route_id NUMBER,
    train_id NUMBER,
    CONSTRAINT fk_schedule_route FOREIGN KEY (route_id)
        REFERENCES Route(route_id),
    CONSTRAINT fk_schedule_train FOREIGN KEY (train_id)
        REFERENCES Train(train_id)
);
```

The screenshot shows the Oracle Application Express interface. At the top, there is a toolbar with 'Autocommit' checked, a 'Display' dropdown set to 10, and 'Save' and 'Run' buttons. The main area contains the SQL code for creating the 'Schedule' table. Below the code, a 'DESC Schedule;' command is shown. At the bottom, there are tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. A table titled 'Object Type TABLE Object SCHEDULE' displays the columns and their properties. The table has 8 rows corresponding to the columns defined in the SQL code. The 'Primary Key' column shows '1' for the first row ('SCHEDULE\_ID') and '0' for the others. The 'Nullable' column shows checkmarks for all rows except the primary key. The 'Default' and 'Comment' columns are empty. At the bottom right of the table, it says '1 - 8'. The footer of the application shows 'Language: en-us', 'Application Express 2.1.0.00.39', 'Copyright © 1999-2006, Oracle. All rights reserved.', and system status icons including weather (15°C Partly cloudy), time (2:03 AM), and date (1/2/2026).

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
SCHEDULE	SCHEDULE_ID	Number	-	-	-	1	-	-	-
	DEPARTURE_TIME	Varchar2	20	-	-	-	✓	-	-
	ARRIVAL_TIME	Varchar2	20	-	-	-	✓	-	-
	DAYS_OF_OPERATION	Varchar2	50	-	-	-	✓	-	-
	FREQUENCY_MIN	Number	-	-	-	-	✓	-	-
	STATUS	Varchar2	20	-	-	-	✓	-	-
	ROUTE_ID	Number	-	-	-	-	✓	-	-
	TRAIN_ID	Number	-	-	-	-	✓	-	-

## Payment Table

```
CREATE TABLE Payment (
    payment_id NUMBER PRIMARY KEY,
    method VARCHAR2(20),
    amount NUMBER,
    payment_time TIMESTAMP,
    status VARCHAR2(20)
);
```

Autocommit Display 10

```
CREATE TABLE Payment (
    payment_id NUMBER PRIMARY KEY,
    method VARCHAR2(20),
    amount NUMBER,
    payment_time TIMESTAMP,
    status VARCHAR2(20)
);
DESC Payment;
```

Results Explain Describe Saved SQL History

Object Type TABLE Object PAYMENT

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PAYMENT	PAYMENT_ID	Number	-	-	-	1	-	-	-
	METHOD	Varchar2	20	-	-	-	✓	-	-
	AMOUNT	Number	-	-	-	-	✓	-	-
	PAYMENT_TIME	Timestamp(6)	11	-	6	-	✓	-	-
	STATUS	Varchar2	20	-	-	-	✓	-	-

1 - 5



## Passenger Table

```
CREATE TABLE Passenger (
    passenger_id NUMBER PRIMARY KEY,
    name VARCHAR2(100),
    gender VARCHAR2(10),
    dob DATE,
    nid VARCHAR2(20),
    phone VARCHAR2(15),
    email VARCHAR2(100),
    address VARCHAR2(200),
    registered_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

The screenshot shows the Oracle Application Express SQL Workshop interface. At the top, there is a toolbar with 'Autocommit' checked, a 'Display' dropdown set to 10, and 'Save' and 'Run' buttons. The main area contains the SQL code for creating the Passenger table, which includes constraints for uniqueness and nullability. Below the code, the 'DESC Passenger;' command is run. The results section shows the table structure with columns: PASSENGER\_ID (Number, Primary Key), NAME (Varchar2(100)), GENDER (Varchar2(10)), DOB (Date), NID (Varchar2(20)), PHONE (Varchar2(15)), EMAIL (Varchar2(100)), ADDRESS (Varchar2(200)), and REGISTERED\_AT (Timestamp(6)). The table has 9 rows. At the bottom, the status bar shows 'Language: en-us', 'Application Express 2.1.0.00.39', 'Copyright © 1999-2006, Oracle. All rights reserved.', the date '1/2/2026', and the time '2:06 AM'. The system tray icons include a blue square, a red gear, a network signal, a battery, a speaker, a keyboard, and a speech bubble.

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PASSENGER	PASSENGER_ID	Number	-	-	-	1	-	-	-
	NAME	Varchar2	100	-	-	-	-	-	-
	GENDER	Varchar2	10	-	-	-	✓	-	-
	DOB	Date	7	-	-	-	✓	-	-
	NID	Varchar2	20	-	-	-	✓	-	-
	PHONE	Varchar2	15	-	-	-	✓	-	-
	EMAIL	Varchar2	100	-	-	-	✓	-	-
	ADDRESS	Varchar2	200	-	-	-	✓	-	-
	REGISTERED_AT	Timestamp(6)	11	-	6	-	✓	CURRENT_TIMESTAMP	-

## Ticket Table

```

CREATE TABLE Ticket (
    ticket_id NUMBER PRIMARY KEY,
    purchase_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    schedule_id NUMBER,
    passenger_id NUMBER,
    payment_id NUMBER,
    fare_id NUMBER,
    CONSTRAINT fk_ticket_schedule FOREIGN KEY (schedule_id)
        REFERENCES Schedule(schedule_id),
    CONSTRAINT fk_ticket_passenger FOREIGN KEY (passenger_id)
        REFERENCES Passenger(passenger_id),
    CONSTRAINT fk_ticket_payment FOREIGN KEY (payment_id)
        REFERENCES Payment(payment_id),
    CONSTRAINT fk_ticket_fare FOREIGN KEY (fare_id)
        REFERENCES Fare(fare_id)
);

```

Autocommit

```

CREATE TABLE Ticket (
    ticket_id NUMBER PRIMARY KEY,
    purchase_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    schedule_id NUMBER,
    passenger_id NUMBER,
    payment_id NUMBER,
    fare_id NUMBER,
    CONSTRAINT fk_ticket_schedule FOREIGN KEY (schedule_id)
        REFERENCES Schedule(schedule_id),
    CONSTRAINT fk_ticket_passenger FOREIGN KEY (passenger_id)
        REFERENCES Passenger(passenger_id),
    CONSTRAINT fk_ticket_payment FOREIGN KEY (payment_id)
        REFERENCES Payment(payment_id),
    CONSTRAINT fk_ticket_fare FOREIGN KEY (fare_id)
        REFERENCES Fare(fare_id)
);

```

**DESC Ticket:**

Results Explain Describe Saved SQL History									
Object Type TABLE Object TICKET									
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
TICKET	TICKET_ID	Number	-	-	-	1	-	-	-
	PURCHASE_TIME	Timestamp(6)	11	-	6	-	✓	CURRENT_TIMESTAMP	-
	SCHEDULE_ID	Number	-	-	-	-	✓	-	-
	PASSENGER_ID	Number	-	-	-	-	✓	-	-
	PAYMENT_ID	Number	-	-	-	-	✓	-	-
	FARE_ID	Number	-	-	-	-	✓	-	-

1 - 6



## Employee (Emp) Table

```
CREATE TABLE Emp (
    emp_id NUMBER PRIMARY KEY,
    name VARCHAR2(100),
    gender VARCHAR2(10),
    role VARCHAR2(50),
    salary NUMBER CHECK (salary > 0),
    hire_date DATE,
    shift_start VARCHAR2(20),
    shift_end VARCHAR2(20),
    status VARCHAR2(20),
    station_id NUMBER,
    train_id NUMBER,
    CONSTRAINT fk_emp_station FOREIGN KEY (station_id)
        REFERENCES Station(station_id),
    CONSTRAINT fk_emp_train FOREIGN KEY (train_id)
        REFERENCES Train(train_id)
);
```

Autocommit Display 10 ▾

Save Run

```
CREATE TABLE Emp (
    emp_id NUMBER PRIMARY KEY,
    name VARCHAR2(100),
    gender VARCHAR2(10),
    role VARCHAR2(50),
    salary NUMBER CHECK (salary > 0),
    hire_date DATE,
    shift_start VARCHAR2(20),
    shift_end VARCHAR2(20),
    status VARCHAR2(20),
    station_id NUMBER,
    train_id NUMBER,
    CONSTRAINT fk_emp_station FOREIGN KEY (station_id)
        REFERENCES Station(station_id),
    CONSTRAINT fk_emp_train FOREIGN KEY (train_id)
        REFERENCES Train(train_id)
);
```

Results Explain Describe Saved SQL History

Object Type TABLE Object EMP

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
EMP	EMP_ID	Number	-	-	-	1	-	-	-
	NAME	Varchar2	100	-	-	-	✓	-	-
	GENDER	Varchar2	10	-	-	-	✓	-	-
	ROLE	Varchar2	50	-	-	-	✓	-	-
	SALARY	Number	-	-	-	-	✓	-	-
	HIRE_DATE	Date	7	-	-	-	✓	-	-
	SHIFT_START	Varchar2	20	-	-	-	✓	-	-
	SHIFT_END	Varchar2	20	-	-	-	✓	-	-
	STATUS	Varchar2	20	-	-	-	✓	-	-
	STATION_ID	Number	-	-	-	-	✓	-	-
	TRAIN_ID	Number	-	-	-	-	✓	-	-

1 - 11



15°C Partly cloudy ⌂ 2:09 AM 1/2/2026

## Maintenance Table

```
CREATE TABLE Maintenance (
    maintenance_id NUMBER PRIMARY KEY,
    description VARCHAR2(255),
    status VARCHAR2(20),
    train_id NUMBER REFERENCES Train(train_id),
    station_id NUMBER REFERENCES Station(station_id),
    emp_id NUMBER REFERENCES Emp(emp_id)
);
```

Screenshot of Oracle Application Express SQL Worksheet interface:

Autocommit: checked | Display: 10 | Save | Run

```
CREATE TABLE Maintenance (
    maintenance_id NUMBER PRIMARY KEY,
    description VARCHAR2(255),
    status VARCHAR2(20),
    train_id NUMBER,
    station_id NUMBER,
    emp_id NUMBER,
    CONSTRAINT fk_maint_train FOREIGN KEY (train_id)
        REFERENCES Train(train_id),
    CONSTRAINT fk_maint_station FOREIGN KEY (station_id)
        REFERENCES Station(station_id),
    CONSTRAINT fk_maint_emp FOREIGN KEY (emp_id)
        REFERENCES Emp(emp_id)
);

DESC Maintenance;
```

Results Explain Describe Saved SQL History

Object Type TABLE Object MAINTENANCE

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
MAINTENANCE	MAINTENANCE_ID	Number	-	-	-	1	-	-	-
	DESCRIPTION	Varchar2	255	-	-	-	✓	-	-
	STATUS	Varchar2	20	-	-	-	✓	-	-
	TRAIN_ID	Number	-	-	-	-	✓	-	-
	STATION_ID	Number	-	-	-	-	✓	-	-
	EMP_ID	Number	-	-	-	-	✓	-	-
1 - 6									

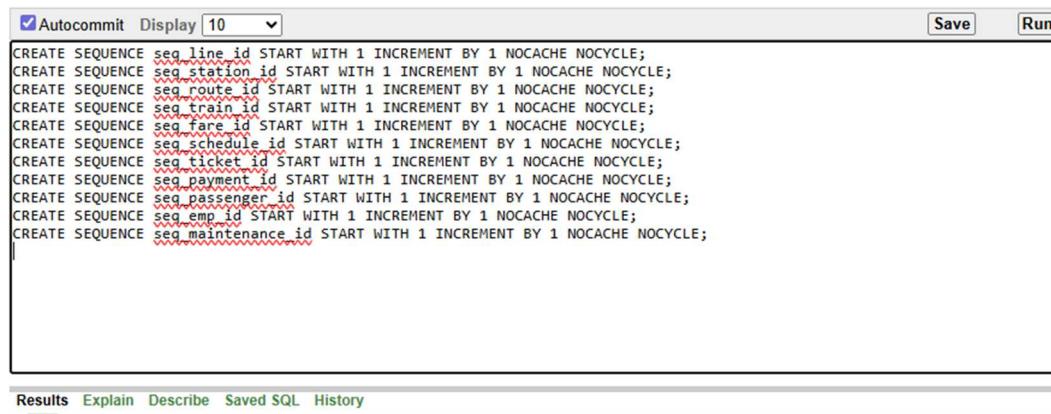
Application Express 2.1.0.00.3

Language: en-us Copyright © 1999, 2006, Oracle. All rights reserved.



## Creating Sequence

```
CREATE SEQUENCE seq_line_id START WITH 1 INCREMENT BY 1 NOCACHE NOCYCLE;  
  
CREATE SEQUENCE seq_station_id START WITH 1 INCREMENT BY 1 NOCACHE NOCYCLE;  
  
CREATE SEQUENCE seq_route_id START WITH 1 INCREMENT BY 1 NOCACHE NOCYCLE;  
  
CREATE SEQUENCE seq_train_id START WITH 1 INCREMENT BY 1 NOCACHE NOCYCLE;  
  
CREATE SEQUENCE seq_fare_id START WITH 1 INCREMENT BY 1 NOCACHE NOCYCLE;  
  
CREATE SEQUENCE seq_schedule_id START WITH 1 INCREMENT BY 1 NOCACHE NOCYCLE;  
  
CREATE SEQUENCE seq_ticket_id START WITH 1 INCREMENT BY 1 NOCACHE NOCYCLE;  
  
CREATE SEQUENCE seq_payment_id START WITH 1 INCREMENT BY 1 NOCACHE NOCYCLE;  
  
CREATE SEQUENCE seq_passenger_id START WITH 1 INCREMENT BY 1 NOCACHE NOCYCLE;  
  
CREATE SEQUENCE seq_emp_id START WITH 1 INCREMENT BY 1 NOCACHE NOCYCLE;  
  
CREATE SEQUENCE seq_maintenance_id START WITH 1 INCREMENT BY 1 NOCACHE NOCYCLE;
```



The screenshot shows the Oracle Application Express SQL Worksheet interface. The top bar includes checkboxes for 'Autocommit' and 'Display' (set to 10), and buttons for 'Save' and 'Run'. The main area contains the SQL code for creating ten sequences. The bottom navigation bar has tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History', with 'Results' being the active tab.

```
CREATE SEQUENCE seq_line_id START WITH 1 INCREMENT BY 1 NOCACHE NOCYCLE;  
CREATE SEQUENCE seq_station_id START WITH 1 INCREMENT BY 1 NOCACHE NOCYCLE;  
CREATE SEQUENCE seq_route_id START WITH 1 INCREMENT BY 1 NOCACHE NOCYCLE;  
CREATE SEQUENCE seq_train_id START WITH 1 INCREMENT BY 1 NOCACHE NOCYCLE;  
CREATE SEQUENCE seq_fare_id START WITH 1 INCREMENT BY 1 NOCACHE NOCYCLE;  
CREATE SEQUENCE seq_schedule_id START WITH 1 INCREMENT BY 1 NOCACHE NOCYCLE;  
CREATE SEQUENCE seq_ticket_id START WITH 1 INCREMENT BY 1 NOCACHE NOCYCLE;  
CREATE SEQUENCE seq_payment_id START WITH 1 INCREMENT BY 1 NOCACHE NOCYCLE;  
CREATE SEQUENCE seq_passenger_id START WITH 1 INCREMENT BY 1 NOCACHE NOCYCLE;  
CREATE SEQUENCE seq_emp_id START WITH 1 INCREMENT BY 1 NOCACHE NOCYCLE;  
CREATE SEQUENCE seq_maintenance_id START WITH 1 INCREMENT BY 1 NOCACHE NOCYCLE;
```

Sequence created.

0.00 seconds



# Data Insertion Using SQL

## 1. Line Table

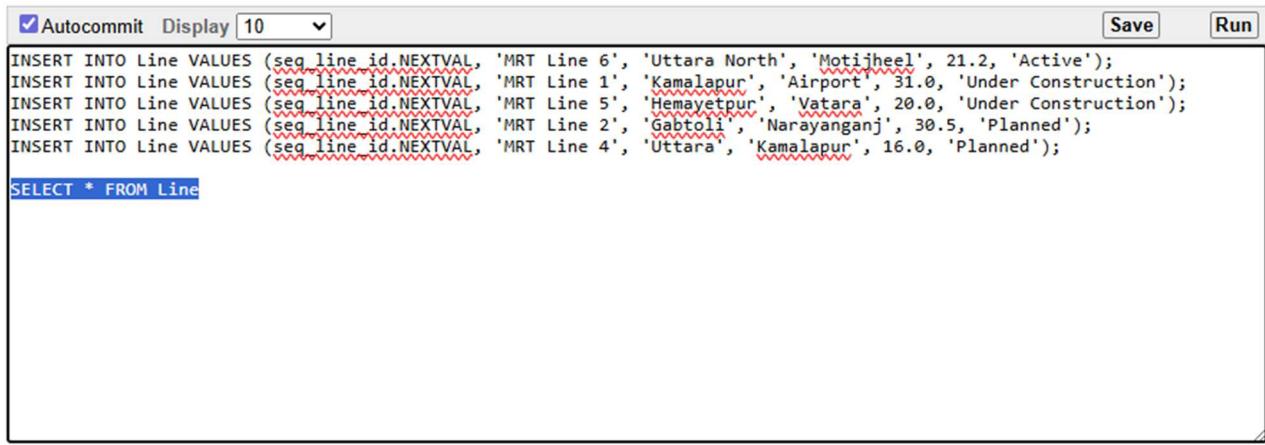
```
INSERT INTO Line VALUES (seq_line_id.NEXTVAL, 'MRT Line 6', 'Uttara North', 'Motijheel', 21.2, 'Active');
```

```
INSERT INTO Line VALUES (seq_line_id.NEXTVAL, 'MRT Line 1', 'Kamalapur', 'Airport', 31.0, 'Under Construction');
```

```
INSERT INTO Line VALUES (seq_line_id.NEXTVAL, 'MRT Line 5', 'Hemayetpur', 'Vatara', 20.0, 'Under Construction');
```

```
INSERT INTO Line VALUES (seq_line_id.NEXTVAL, 'MRT Line 2', 'Gabtoli', 'Narayanganj', 30.5, 'Planned');
```

```
INSERT INTO Line VALUES (seq_line_id.NEXTVAL, 'MRT Line 4', 'Uttara', 'Kamalapur', 16.0, 'Planned');
```



```
Autocommit Display 10 Save Run
INSERT INTO Line VALUES (seq_line_id.NEXTVAL, 'MRT Line 6', 'Uttara North', 'Motijheel', 21.2, 'Active');
INSERT INTO Line VALUES (seq_line_id.NEXTVAL, 'MRT Line 1', 'Kamalapur', 'Airport', 31.0, 'Under Construction');
INSERT INTO Line VALUES (seq_line_id.NEXTVAL, 'MRT Line 5', 'Hemayetpur', 'Vatara', 20.0, 'Under Construction');
INSERT INTO Line VALUES (seq_line_id.NEXTVAL, 'MRT Line 2', 'Gabtoli', 'Narayanganj', 30.5, 'Planned');
INSERT INTO Line VALUES (seq_line_id.NEXTVAL, 'MRT Line 4', 'Uttara', 'Kamalapur', 16.0, 'Planned');

SELECT * FROM Line
```

Results Explain Describe Saved SQL History

LINE_ID	LINE_NAME	START_STATION	END_STATION	TOTAL_DISTANCE_KM	STATUS
1	MRT Line 6	Uttara North	Motijheel	21.2	Active
2	MRT Line 1	Kamalapur	Airport	31	Under Construction
4	MRT Line 5	Hemayetpur	Vatara	20	Under Construction
5	MRT Line 2	Gabtoli	Narayanganj	30.5	Planned
6	MRT Line 4	Uttara	Kamalapur	16	Planned

5 rows returned in 0.02 seconds

[CSV Export](#)

Application Express 2.1.0.0.39

Language: en-us Copyright © 1999, 2006, Oracle. All rights reserved.



## Fare Table

```
INSERT INTO Fare VALUES (seq_fare_id.NEXTVAL, 20);
INSERT INTO Fare VALUES (seq_fare_id.NEXTVAL, 40);
INSERT INTO Fare VALUES (seq_fare_id.NEXTVAL, 60);
INSERT INTO Fare VALUES (seq_fare_id.NEXTVAL, 100);
INSERT INTO Fare VALUES (seq_fare_id.NEXTVAL, 150);
```

The screenshot shows the Oracle Application Express SQL Workshop interface. The top section contains the SQL code:

```
Autocommit Display 10 Save Run
INSERT INTO Fare VALUES (seq_fare_id.NEXTVAL, 20);
INSERT INTO Fare VALUES (seq_fare_id.NEXTVAL, 40);
INSERT INTO Fare VALUES (seq_fare_id.NEXTVAL, 60);
INSERT INTO Fare VALUES (seq_fare_id.NEXTVAL, 100);
INSERT INTO Fare VALUES (seq_fare_id.NEXTVAL, 150);

SELECT * FROM Fare
```

The bottom section shows the results of the executed query:

FARE_ID	PRICE
1	20
2	40
3	60
4	100
5	150

5 rows returned in 0.00 seconds [CSV Export](#)

Application Express 2.1.0.00.39

Live 2:45 AM 1/2/2026

## Station Table

```
INSERT INTO Station VALUES (seq_station_id.NEXTVAL, 'Uttara North', 'Uttara', TO_DATE('2022-12-28','YYYY-MM-DD'), 'Active', 1);
```

```
INSERT INTO Station VALUES (seq_station_id.NEXTVAL, 'Pallabi', 'Mirpur', TO_DATE('2023-01-25','YYYY-MM-DD'), 'Active', 1);
```

```
INSERT INTO Station VALUES (seq_station_id.NEXTVAL, 'Agargaon', 'Agargaon', TO_DATE('2022-12-28','YYYY-MM-DD'), 'Active', 1);
```

```
INSERT INTO Station VALUES (seq_station_id.NEXTVAL, 'Motijheel', 'Motijheel', TO_DATE('2023-11-04','YYYY-MM-DD'), 'Active', 1);
```

```
INSERT INTO Station VALUES (seq_station_id.NEXTVAL, 'Farmgate', 'Tejgaon', TO_DATE('2023-03-15','YYYY-MM-DD'), 'Active', 1);
```

The screenshot shows a PostgreSQL terminal window with the following content:

```
Autocommit Display 10
INSERT INTO Station VALUES (seq_station_id.NEXTVAL, 'Uttara North', 'Uttara', TO_DATE('2022-12-28','YYYY-MM-DD'), 'Active', 1);
INSERT INTO Station VALUES (seq_station_id.NEXTVAL, 'Pallabi', 'Mirpur', TO_DATE('2023-01-25','YYYY-MM-DD'), 'Active', 1);
INSERT INTO Station VALUES (seq_station_id.NEXTVAL, 'Agargaon', 'Agargaon', TO_DATE('2022-12-28','YYYY-MM-DD'), 'Active', 1);
INSERT INTO Station VALUES (seq_station_id.NEXTVAL, 'Motijheel', 'Motijheel', TO_DATE('2023-11-04','YYYY-MM-DD'), 'Active', 1);
INSERT INTO Station VALUES (seq_station_id.NEXTVAL, 'Farmgate', 'Tejgaon', TO_DATE('2023-03-15','YYYY-MM-DD'), 'Active', 1);
SELECT * FROM Station
```

Below the command input area, there is a navigation bar with tabs: Results, Explain, Describe, Saved SQL, and History. The Results tab is currently selected.

STATION_ID	STATION_NAME	LOCATION	OPENING_DATE	STATUS	LINE_ID
1	Uttara North	Uttara	28-DEC-22	Active	1
2	Pallabi	Mirpur	25-JAN-23	Active	1
3	Agargaon	Agargaon	28-DEC-22	Active	1
4	Motijheel	Motijheel	04-NOV-23	Active	1
5	Farmgate	Tejgaon	15-MAR-23	Active	1

5 rows returned in 0.00 seconds [CSV Export](#)

The system tray at the bottom right shows icons for battery, signal, and network, along with the date and time: 2:47 AM, 1/2/2026.

## Route Table

```
INSERT INTO Route VALUES (seq_route_id.NEXTVAL, 1, 2, 5, 10, 1, 1);
INSERT INTO Route VALUES (seq_route_id.NEXTVAL, 2, 3, 4.5, 8, 1, 1);
INSERT INTO Route VALUES (seq_route_id.NEXTVAL, 3, 4, 6, 12, 1, 2);
INSERT INTO Route VALUES (seq_route_id.NEXTVAL, 1, 4, 20, 35, 1, 4);
INSERT INTO Route VALUES (seq_route_id.NEXTVAL, 1, 5, 10, 18, 1, 3);
```

Autocommit Display 10

```
INSERT INTO Route VALUES (seq_route_id.NEXTVAL, 1, 2, 5, 10, 1, 1);
INSERT INTO Route VALUES (seq_route_id.NEXTVAL, 2, 3, 4.5, 8, 1, 1);
INSERT INTO Route VALUES (seq_route_id.NEXTVAL, 3, 4, 6, 12, 1, 2);
INSERT INTO Route VALUES (seq_route_id.NEXTVAL, 1, 4, 20, 35, 1, 4);
INSERT INTO Route VALUES (seq_route_id.NEXTVAL, 1, 5, 10, 18, 1, 3);
SELECT * FROM Route
```

Results Explain Describe Saved SQL History

ROUTE_ID	FROM_STATION_ID	TO_STATION_ID	DISTANCE_KM	ESTIMATED_DURATION_MIN	LINE_ID	FARE_ID
1	1	2	5	10	1	1
2	2	3	4.5	8	1	1
3	3	4	6	12	1	2
4	1	4	20	35	1	4
5	1	5	10	18	1	3

5 rows returned in 0.00 seconds [CSV Export](#)

Application Express 2.1.0.00.39  
Language: en-us Copyright © 1999, 2006, Oracle. All rights reserved.

 Result  2:51 AM 1/2/2026

## Train Table

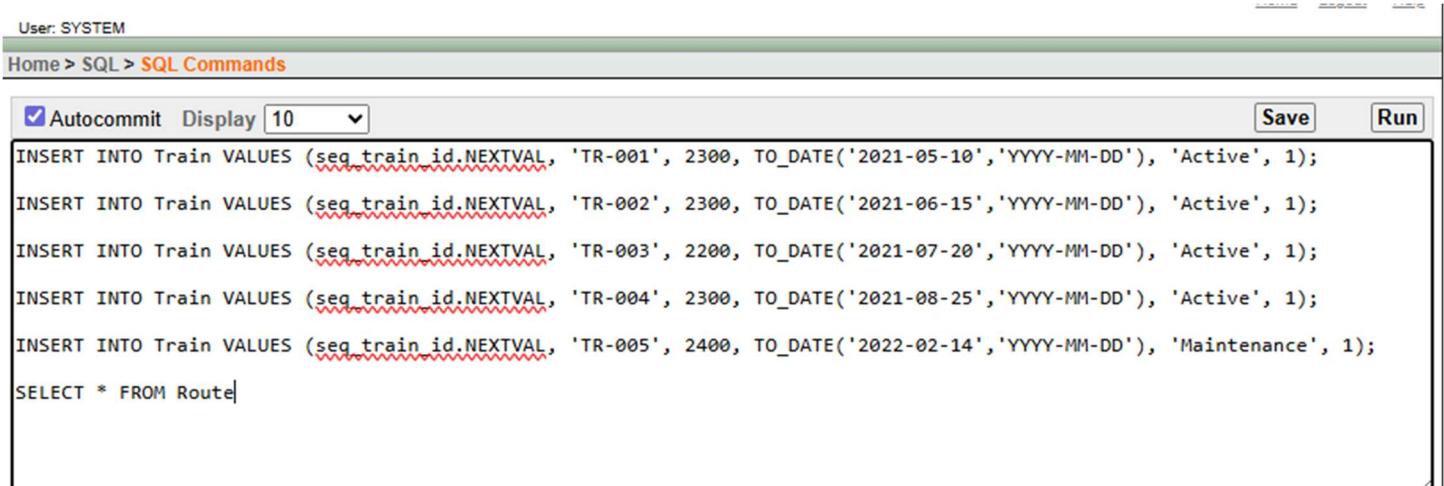
```
INSERT INTO Train VALUES (seq_train_id.NEXTVAL, 'TR-001', 2300, TO_DATE('2021-05-10','YYYY-MM-DD'),  
'Active', 1);
```

```
INSERT INTO Train VALUES (seq_train_id.NEXTVAL, 'TR-002', 2300, TO_DATE('2021-06-15','YYYY-MM-DD'),  
'Active', 1);
```

```
INSERT INTO Train VALUES (seq_train_id.NEXTVAL, 'TR-003', 2200, TO_DATE('2021-07-20','YYYY-MM-DD'),  
'Active', 1);
```

```
INSERT INTO Train VALUES (seq_train_id.NEXTVAL, 'TR-004', 2300, TO_DATE('2021-08-25','YYYY-MM-DD'),  
'Active', 1);
```

```
INSERT INTO Train VALUES (seq_train_id.NEXTVAL, 'TR-005', 2400, TO_DATE('2022-02-14','YYYY-MM-DD'),  
'Maintenance', 1);
```



```
User: SYSTEM  
Home > SQL > SQL Commands  
 Autocommit  10    
INSERT INTO Train VALUES (seq_train_id.NEXTVAL, 'TR-001', 2300, TO_DATE('2021-05-10','YYYY-MM-DD'), 'Active', 1);  
INSERT INTO Train VALUES (seq_train_id.NEXTVAL, 'TR-002', 2300, TO_DATE('2021-06-15','YYYY-MM-DD'), 'Active', 1);  
INSERT INTO Train VALUES (seq_train_id.NEXTVAL, 'TR-003', 2200, TO_DATE('2021-07-20','YYYY-MM-DD'), 'Active', 1);  
INSERT INTO Train VALUES (seq_train_id.NEXTVAL, 'TR-004', 2300, TO_DATE('2021-08-25','YYYY-MM-DD'), 'Active', 1);  
INSERT INTO Train VALUES (seq_train_id.NEXTVAL, 'TR-005', 2400, TO_DATE('2022-02-14','YYYY-MM-DD'), 'Maintenance', 1);  
SELECT * FROM Route
```

Results Explain Describe Saved SQL History

ROUTE_ID	FROM_STATION_ID	TO_STATION_ID	DISTANCE_KM	ESTIMATED_DURATION_MIN	LINE_ID	FARE_ID
1	1	2	5	10	1	1
2	2	3	4.5	8	1	1
3	3	4	6	12	1	2
4	1	4	20	35	1	4
5	1	5	10	18	1	3

5 rows returned in 0.02 seconds

[CSV Export](#)

Application Express 2.1.0.00.39

Language: en-us

Copyright © 1999, 2006, Oracle. All rights reserved.



## Schedule Table

```
INSERT INTO Schedule VALUES (seq_schedule_id.NEXTVAL, '08:00 AM', '08:40 AM', 'Sat-Thu', 10, 'Active', 4, 1);
```

```
INSERT INTO Schedule VALUES (seq_schedule_id.NEXTVAL, '09:00 AM', '09:40 AM', 'Sat-Thu', 10, 'Active', 1, 2);
```

```
INSERT INTO Schedule VALUES (seq_schedule_id.NEXTVAL, '10:00 AM', '10:30 AM', 'Sat-Thu', 10, 'Active', 2, 3);
```

```
INSERT INTO Schedule VALUES (seq_schedule_id.NEXTVAL, '11:00 AM', '11:50 AM', 'Sat-Thu', 15, 'Active', 3, 4);
```

```
INSERT INTO Schedule VALUES (seq_schedule_id.NEXTVAL, '12:00 PM', '12:45 PM', 'Sat-Thu', 15, 'Active', 5, 5);
```

```
Autocommit Display 10 Save Run
INSERT INTO Schedule VALUES (seq_schedule_id.NEXTVAL, '08:00 AM', '08:40 AM', 'Sat-Thu', 10, 'Active', 4, 1);
INSERT INTO Schedule VALUES (seq_schedule_id.NEXTVAL, '09:00 AM', '09:40 AM', 'Sat-Thu', 10, 'Active', 1, 2);
INSERT INTO Schedule VALUES (seq_schedule_id.NEXTVAL, '10:00 AM', '10:30 AM', 'Sat-Thu', 10, 'Active', 2, 3);
INSERT INTO Schedule VALUES (seq_schedule_id.NEXTVAL, '11:00 AM', '11:50 AM', 'Sat-Thu', 15, 'Active', 3, 4);
INSERT INTO Schedule VALUES (seq_schedule_id.NEXTVAL, '12:00 PM', '12:45 PM', 'Sat-Thu', 15, 'Active', 5, 5);
SELECT * FROM Schedule|
```

Results Explain Describe Saved SQL History

SCHEDULE_ID	DEPARTURE_TIME	ARRIVAL_TIME	DAY_OF_OPERATION	FREQUENCY_MIN	STATUS	ROUTE_ID	TRAIN_ID
1	08:00 AM	08:40 AM	Sat-Thu	10	Active	4	1
2	09:00 AM	09:40 AM	Sat-Thu	10	Active	1	2
3	10:00 AM	10:30 AM	Sat-Thu	10	Active	2	3
4	11:00 AM	11:50 AM	Sat-Thu	15	Active	3	4
5	12:00 PM	12:45 PM	Sat-Thu	15	Active	5	5

5 rows returned in 0.00 seconds

[CSV Export](#)

Application Express 2.1.0.00.39

Language: en-us

Copyright © 1999, 2008, Oracle. All rights reserved.



Supergirl official teas...



2:57 AM

1/2/2026

## Passenger Table

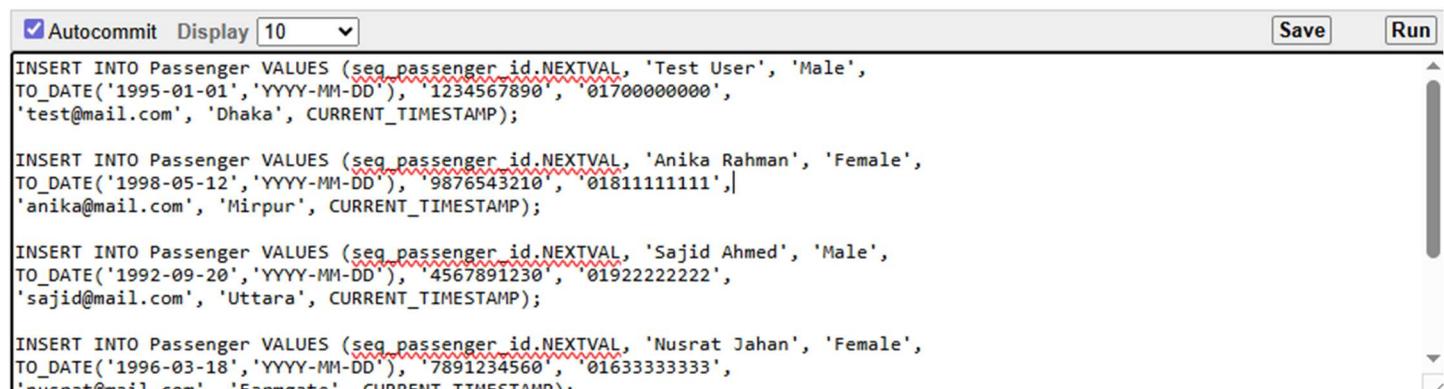
```
INSERT INTO Passenger VALUES (seq_passenger_id.NEXTVAL, 'Test User', 'Male', TO_DATE('1995-01-01','YYYY-MM-DD'), '1234567890', '01700000000', 'test@mail.com', 'Dhaka', CURRENT_TIMESTAMP);

INSERT INTO Passenger VALUES (seq_passenger_id.NEXTVAL, 'Anika Rahman', 'Female', TO_DATE('1998-05-12','YYYY-MM-DD'), '9876543210', '01811111111', 'anika@mail.com', 'Mirpur', CURRENT_TIMESTAMP);

INSERT INTO Passenger VALUES (seq_passenger_id.NEXTVAL, 'Sajid Ahmed', 'Male', TO_DATE('1992-09-20','YYYY-MM-DD'), '4567891230', '01922222222', 'sajid@mail.com', 'Uttara', CURRENT_TIMESTAMP);

INSERT INTO Passenger VALUES (seq_passenger_id.NEXTVAL, 'Nusrat Jahan', 'Female', TO_DATE('1996-03-18','YYYY-MM-DD'), '7891234560', '01633333333', 'nusrat@mail.com', 'Farmgate', CURRENT_TIMESTAMP);

INSERT INTO Passenger VALUES (seq_passenger_id.NEXTVAL, 'Imran Hossain', 'Male', TO_DATE('1990-11-02','YYYY-MM-DD'), '3216549870', '01544444444', 'imran@mail.com', 'Motijheel', CURRENT_TIMESTAMP);
```



The screenshot shows the Oracle SQL Developer interface with the following details:

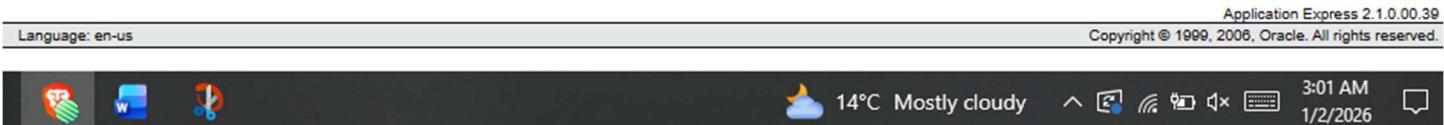
- Autocommit:** Checked.
- Display:** Set to 10 rows.
- Save:** Button.
- Run:** Button.
- SQL Editor Content:** Five INSERT statements for the Passenger table, each with a red underline under the column names and values. The statements are identical except for the passenger's name and address.

Results Explain Describe Saved SQL History

PASSENGER_ID	NAME	GENDER	DOB	NID	PHONE	EMAIL	ADDRESS	REGISTERED_AT
1	Test User	Male	01-JAN-95	1234567890	01700000000	test@mail.com	Dhaka	01-JAN-26 09.00.51.283000 PM
2	Anika Rahman	Female	12-MAY-98	9876543210	01811111111	anika@mail.com	Mirpur	01-JAN-26 09.00.59.150000 PM
3	Sajid Ahmed	Male	20-SEP-92	4567891230	01922222222	sajid@mail.com	Uttara	01-JAN-26 09.01.03.382000 PM
4	Nusrat Jahan	Female	18-MAR-96	7891234560	01633333333	nusrat@mail.com	Farmgate	01-JAN-26 09.01.10.069000 PM
5	Imran Hossain	Male	02-NOV-90	3216549870	01544444444	imran@mail.com	Motijheel	01-JAN-26 09.01.19.105000 PM

5 rows returned in 0.00 seconds

[CSV Export](#)



## Payment Table

```
INSERT INTO Payment VALUES (seq_payment_id.NEXTVAL, 'Bkash', 100, CURRENT_TIMESTAMP, 'Completed');
```

```
INSERT INTO Payment VALUES (seq_payment_id.NEXTVAL, 'Nagad', 20, CURRENT_TIMESTAMP, 'Completed');
```

```
INSERT INTO Payment VALUES (seq_payment_id.NEXTVAL, 'Card', 40, CURRENT_TIMESTAMP, 'Completed');
```

```
INSERT INTO Payment VALUES (seq_payment_id.NEXTVAL, 'Bkash', 60, CURRENT_TIMESTAMP, 'Completed');
```

```
INSERT INTO Payment VALUES (seq_payment_id.NEXTVAL, 'Cash', 150, CURRENT_TIMESTAMP, 'Completed');
```

Screenshot of a SQL query editor showing the execution of five INSERT statements and a SELECT statement.

Autocommit:  Display: 10

Save Run

```
INSERT INTO Payment VALUES (seq_payment_id.NEXTVAL, 'Bkash', 100, CURRENT_TIMESTAMP, 'Completed');
INSERT INTO Payment VALUES (seq_payment_id.NEXTVAL, 'Nagad', 20, CURRENT_TIMESTAMP, 'Completed');
INSERT INTO Payment VALUES (seq_payment_id.NEXTVAL, 'Card', 40, CURRENT_TIMESTAMP, 'Completed');
INSERT INTO Payment VALUES (seq_payment_id.NEXTVAL, 'Bkash', 60, CURRENT_TIMESTAMP, 'Completed');
INSERT INTO Payment VALUES (seq_payment_id.NEXTVAL, 'Cash', 150, CURRENT_TIMESTAMP, 'Completed');

Select * from Payment
```

Results Explain Describe Saved SQL History

PAYMENT_ID	METHOD	AMOUNT	PAYMENT_TIME	STATUS
1	Bkash	100	01-JAN-26 09.03.09.286000 PM	Completed
2	Nagad	20	01-JAN-26 09.03.13.937000 PM	Completed
3	Card	40	01-JAN-26 09.03.18.227000 PM	Completed
4	Bkash	60	01-JAN-26 09.03.22.495000 PM	Completed
5	Cash	150	01-JAN-26 09.03.26.555000 PM	Completed

5 rows returned in 0.00 seconds

[CSV Export](#)



## Ticket Table

```
INSERT INTO Ticket VALUES (seq_ticket_id.NEXTVAL, CURRENT_TIMESTAMP, 1, 1, 1, 4);
```

```
INSERT INTO Ticket VALUES (seq_ticket_id.NEXTVAL, CURRENT_TIMESTAMP, 2, 2, 2, 1);
```

```
INSERT INTO Ticket VALUES (seq_ticket_id.NEXTVAL, CURRENT_TIMESTAMP, 3, 3, 3, 2);
```

```
INSERT INTO Ticket VALUES (seq_ticket_id.NEXTVAL, CURRENT_TIMESTAMP, 4, 4, 4, 3);
```

```
INSERT INTO Ticket VALUES (seq_ticket_id.NEXTVAL, CURRENT_TIMESTAMP, 5, 5, 5, 5);
```

```
Select * from Ticket
```

The screenshot shows a MySQL Workbench interface. The top part is a query editor with the following content:

```
Autocommit Display 10 Save Run
INSERT INTO Ticket VALUES (seq_ticket_id.NEXTVAL, CURRENT_TIMESTAMP, 1, 1, 1, 4);
INSERT INTO Ticket VALUES (seq_ticket_id.NEXTVAL, CURRENT_TIMESTAMP, 2, 2, 2, 1);
INSERT INTO Ticket VALUES (seq_ticket_id.NEXTVAL, CURRENT_TIMESTAMP, 3, 3, 3, 2);
INSERT INTO Ticket VALUES (seq_ticket_id.NEXTVAL, CURRENT_TIMESTAMP, 4, 4, 4, 3);
INSERT INTO Ticket VALUES (seq_ticket_id.NEXTVAL, CURRENT_TIMESTAMP, 5, 5, 5, 5);
Select * from Payment;
```

Below the editor, there is a navigation bar with links: Results, Explain, Describe, Saved SQL, and History. The main area displays the results of the query:

PAYMENT_ID	METHOD	AMOUNT	PAYMENT_TIME	STATUS
1	Bkash	100	01-JAN-26 09.03.09.286000 PM	Completed
2	Nagad	20	01-JAN-26 09.03.13.937000 PM	Completed
3	Card	40	01-JAN-26 09.03.18.227000 PM	Completed
4	Bkash	60	01-JAN-26 09.03.22.495000 PM	Completed
5	Cash	150	01-JAN-26 09.03.26.555000 PM	Completed

Below the table, it says "5 rows returned in 0.00 seconds" and has a "CSV Export" link. The bottom of the screen shows a taskbar with various icons and system status information.

## Employee (Emp) Table

```
INSERT INTO Emp VALUES (seq_emp_id.NEXTVAL, 'Rahim Khan', 'Male', 'Driver', 45000, TO_DATE('2022-01-01','YYYY-MM-DD'), '08:00 AM', '04:00 PM', 'Active', 1, 1);
```

```
INSERT INTO Emp VALUES (seq_emp_id.NEXTVAL, 'Karim Ali', 'Male', 'Controller', 40000, TO_DATE('2021-06-10','YYYY-MM-DD'), '09:00 AM', '05:00 PM', 'Active', 2, 2);
```

```
INSERT INTO Emp VALUES (seq_emp_id.NEXTVAL, 'Salma Akter', 'Female', 'Station Master', 42000, TO_DATE('2020-03-01','YYYY-MM-DD'), '08:00 AM', '04:00 PM', 'Active', 3, NULL);
```

```
INSERT INTO Emp VALUES (seq_emp_id.NEXTVAL, 'Hasan Mahmud', 'Male', 'Technician', 38000, TO_DATE('2019-08-15','YYYY-MM-DD'), '10:00 AM', '06:00 PM', 'Active', 4, 3);
```

```
INSERT INTO Emp VALUES (seq_emp_id.NEXTVAL, 'Rina Sultana', 'Female', 'Ticket Officer', 35000, TO_DATE('2022-11-01','YYYY-MM-DD'), '07:00 AM', '03:00 PM', 'Active', 1, NULL);
```

Select \* from Emp

The screenshot shows the Oracle Application Express SQL Workshop interface. At the top, there are buttons for Autocommit (checked), Display (set to 10), Save, and Run. Below the input area, the five INSERT statements are listed:

```
INSERT INTO Emp VALUES (seq_emp_id.NEXTVAL, 'Rahim Khan', 'Male', 'Driver', 45000, TO_DATE('2022-01-01','YYYY-MM-DD'), '08:00 AM', '04:00 PM', 'Active', 1, 1);
INSERT INTO Emp VALUES (seq_emp_id.NEXTVAL, 'Karim Ali', 'Male', 'Controller', 40000, TO_DATE('2021-06-10','YYYY-MM-DD'), '09:00 AM', '05:00 PM', 'Active', 2, 2);
INSERT INTO Emp VALUES (seq_emp_id.NEXTVAL, 'Salma Akter', 'Female', 'Station Master', 42000, TO_DATE('2020-03-01','YYYY-MM-DD'), '08:00 AM', '04:00 PM', 'Active', 3, NULL);
INSERT INTO Emp VALUES (seq_emp_id.NEXTVAL, 'Hasan Mahmud', 'Male', 'Technician', 38000, TO_DATE('2019-08-15','YYYY-MM-DD'), '10:00 AM', '06:00 PM', 'Active', 4, 3);
INSERT INTO Emp VALUES (seq_emp_id.NEXTVAL, 'Rina Sultana', 'Female', 'Ticket Officer', 35000, TO_DATE('2022-11-01','YYYY-MM-DD'), '07:00 AM', '03:00 PM', 'Active', 1, NULL);
```

Below the statements is a Results tab, which displays the following table:

EMP_ID	NAME	GENDER	ROLE	SALARY	HIRE_DATE	SHIFT_START	SHIFT_END	STATUS	STATION_ID	TRAIN_ID
1	Rahim Khan	Male	Driver	45000	01-JAN-22	08:00 AM	04:00 PM	Active	1	1
2	Karim Ali	Male	Controller	40000	10-JUN-21	09:00 AM	05:00 PM	Active	2	2
3	Salma Akter	Female	Station Master	42000	01-MAR-20	08:00 AM	04:00 PM	Active	3	-
4	Hasan Mahmud	Male	Technician	38000	15-AUG-19	10:00 AM	06:00 PM	Active	4	3
5	Rina Sultana	Female	Ticket Officer	35000	01-NOV-22	07:00 AM	03:00 PM	Active	1	-

At the bottom left, it says "5 rows returned in 0.03 seconds". On the right, there are CSV Export and Application Express 2.1.0.0.39 links. The status bar at the bottom right shows "3:34 AM 1/2/2026".

## Maintenance Table

```
INSERT INTO Maintenance VALUES (seq_maintenance_id.NEXTVAL,'Monthly Engine Checkup',  
'Completed', 1, 1, 1);
```

```
INSERT INTO Maintenance VALUES (seq_maintenance_id.NEXTVAL,'Brake Inspection', 'Completed', 2, 2,  
2);
```

```
INSERT INTO Maintenance VALUES (seq_maintenance_id.NEXTVAL,'AC Repair', 'Completed', 3, 3, 4);
```

```
INSERT INTO Maintenance VALUES (seq_maintenance_id.NEXTVAL,'Door Sensor Fix', 'Pending', 4, 4, 4);
```

```
INSERT INTO Maintenance VALUES (seq_maintenance_id.NEXTVAL,'Electrical Check', 'Completed', 5, 1,  
1);
```

The screenshot shows a SQL query editor interface. At the top, there are checkboxes for 'Autocommit' and 'Display', a dropdown menu set to '10', and buttons for 'Save' and 'Run'. The main area contains the following SQL code:

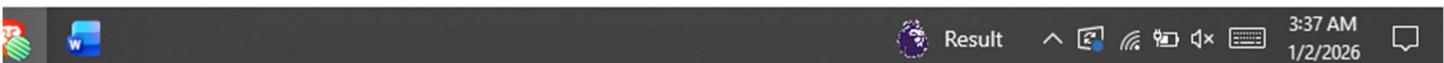
```
INSERT INTO Maintenance VALUES (seq_maintenance_id.NEXTVAL,'Monthly Engine Checkup', 'Completed', 1, 1, 1);  
INSERT INTO Maintenance VALUES (seq_maintenance_id.NEXTVAL,'Brake Inspection', 'Completed', 2, 2, 2);  
INSERT INTO Maintenance VALUES (seq_maintenance_id.NEXTVAL,'AC Repair', 'Completed', 3, 3, 4);  
INSERT INTO Maintenance VALUES (seq_maintenance_id.NEXTVAL,'Door Sensor Fix', 'Pending', 4, 4, 4);  
INSERT INTO Maintenance VALUES (seq_maintenance_id.NEXTVAL,'Electrical Check', 'Completed', 5, 1, 1);  
Select * from Maintenance
```

Results Explain Describe Saved SQL History

MAINTENANCE_ID	DESCRIPTION	STATUS	TRAIN_ID	STATION_ID	EMP_ID
1	Monthly Engine Checkup	Completed	1	1	1
2	Brake Inspection	Completed	2	2	2
3	AC Repair	Completed	3	3	4
4	Door Sensor Fix	Pending	4	4	4
5	Electrical Check	Completed	5	1	1

5 rows returned in 0.02 seconds

[CSV Export](#)



# Basic PL/SQL

## Q1.

Write a PL/SQL block that declares a variable and stores the **total number of passengers** from the PASSENGER table and displays it.

```
DECLARE
    v_total_passengers NUMBER;
BEGIN
    SELECT COUNT(*) INTO v_total_passengers
    FROM passenger;
    DBMS_OUTPUT.PUT_LINE('Total Passengers: ' || v_total_passengers);
END;
/
```

The screenshot shows a database query editor window. At the top, there are checkboxes for 'Autocommit' and 'Display' set to 10, and buttons for 'Save' and 'Run'. The SQL code area contains the provided PL/SQL block. Below the code, a results pane shows the output: 'Total Passengers: 5' followed by 'Statement processed.' and a timing of '0.00 seconds'. The bottom of the screen shows a dark taskbar with various icons and system status information, including the date '1/4/2026' and time '8:24 PM'.

```
DECLARE
    v_total_passengers NUMBER;
BEGIN
    SELECT COUNT(*) INTO v_total_passengers
    FROM passenger;
    DBMS_OUTPUT.PUT_LINE('Total Passengers: ' || v_total_passengers);
END;
/
```

Results Explain Describe Saved SQL History

Total Passengers: 5  
Statement processed.  
0.00 seconds

Air quality forecast 8:24 PM 1/4/2026

## Q2.

Write a PL/SQL program to store the **fare price of a specific fare** in a variable and print it.

```
DECLARE
    v_price NUMBER;
BEGIN
    SELECT price INTO v_price
    FROM fare
    WHERE fare_id = 1;
    DBMS_OUTPUT.PUT_LINE('Fare Price: ' || v_price);
END;
```

The screenshot shows the Oracle SQL Developer interface. The top bar has checkboxes for 'Autocommit' and 'Display' set to 10, and buttons for 'Save' and 'Run'. The main area contains the PL/SQL code provided above. Below the code, the 'Results' tab is selected, showing the output of the executed query: 'Fare Price: 20' and 'Statement processed.' The bottom status bar shows system icons and the text 'Air quality forecast', '8:25 PM', '1/4/2026', and a battery icon.

```
DECLARE
    v_price NUMBER;
BEGIN
    SELECT price INTO v_price
    FROM fare
    WHERE fare_id = 1;
    DBMS_OUTPUT.PUT_LINE('Fare Price: ' || v_price);
END;
```

Fare Price: 20  
Statement processed.  
0.00 seconds

Air quality forecast 8:25 PM 1/4/2026

### Q3.

Write a PL/SQL block using **arithmetic operators** to calculate total salary including bonus.

DECLARE

```
v_salary NUMBER := 30000;
```

```
v_bonus NUMBER := 5000;
```

BEGIN

```
DBMS_OUTPUT.PUT_LINE('Total Salary: ' || (v_salary + v_bonus));
```

END;

/

The screenshot shows a SQL developer interface. The top section contains the PL/SQL code:

```
Autocommit Display 10 Save Run
DECLARE
    v_salary NUMBER := 30000;
    v_bonus NUMBER := 5000;
BEGIN
    DBMS_OUTPUT.PUT_LINE('Total Salary: ' || (v_salary + v_bonus));
END;
/
```

The bottom section shows the results of the execution:

Results Explain Describe Saved SQL History

```
Total Salary: 35000
Statement processed.

0.00 seconds
```

The system tray at the bottom right shows the date and time: 8:27 PM 1/4/2026.

#### Q4.

Write a PL/SQL program using a **comparison operator** to check whether a train has **large capacity**.

DECLARE

```
v_capacity NUMBER;
```

BEGIN

```
    SELECT capacity INTO v_capacity
```

```
    FROM train
```

```
    WHERE train_id = 1;
```

```
    IF v_capacity > 500 THEN
```

```
        DBMS_OUTPUT.PUT_LINE('Large capacity train');
```

```
    END IF;
```

END;

/

The screenshot shows the Oracle SQL Developer interface. The code area contains a PL/SQL block. The output area shows the results of the execution, including the output message and the statement processed message. The system tray at the bottom right shows the date and time as 8:28 PM on 1/4/2026.

```
DECLARE
  v_capacity NUMBER;
BEGIN
    SELECT capacity INTO v_capacity
    FROM train
    WHERE train_id = 1;

    IF v_capacity > 500 THEN
        DBMS_OUTPUT.PUT_LINE('Large capacity train');
    END IF;
END;
/
```

Results Explain Describe Saved SQL History

Large capacity train  
Statement processed.  
0.00 seconds

8:28 PM  
1/4/2026

**Q5.** Write a PL/SQL program using the **UPPER()** function to display a passenger's name in uppercase.

DECLARE

```
v_name VARCHAR2(100);
```

BEGIN

```
    SELECT UPPER(name) INTO v_name
```

```
    FROM passenger
```

```
    WHERE passenger_id = 1;
```

```
    DBMS_OUTPUT.PUT_LINE(v_name);
```

END;

/

The screenshot shows the Oracle SQL Developer interface. The top section is the SQL editor window, which contains the following PL/SQL code:

```
DECLARE
  v_name VARCHAR2(100);
BEGIN
  SELECT UPPER(name) INTO v_name
  FROM passenger
  WHERE passenger_id = 1;
  DBMS_OUTPUT.PUT_LINE(v_name);
END;
/
```

The bottom section is the Results tab, showing the output of the executed statement:

```
TEST USER
Statement processed.

0.00 seconds
```

The status bar at the bottom right shows system information: 17°C Mostly clear, 8:28 PM, 1/4/2026, and a battery icon with a '1'.

**Q6.** Write a PL/SQL block using the **LENGTH()** function to find the length of an employee's name.

DECLARE

```
v_length NUMBER;
```

BEGIN

```
    SELECT LENGTH(name) INTO v_length
```

```
    FROM emp
```

```
    WHERE emp_id = 1;
```

```
    DBMS_OUTPUT.PUT_LINE('Name Length: ' || v_length);
```

END;

/

The screenshot shows the Oracle SQL Developer interface. In the top-left corner, there are checkboxes for 'Autocommit' and 'Display' (set to 10), and buttons for 'Save' and 'Run'. The main area contains the PL/SQL code:

```
DECLARE
    v_length NUMBER;
BEGIN
    SELECT LENGTH(name) INTO v_length
    FROM emp
    WHERE emp_id = 1;
    DBMS_OUTPUT.PUT_LINE('Name Length: ' || v_length);
END;
/
```

Below the code, there is a toolbar with buttons for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is selected. The output window displays the following:

```
Name Length: 10
Statement processed.

0.00 seconds
```

At the bottom of the screen, there is a system tray with icons for battery, signal, and volume, along with the current temperature (17°C), weather (Mostly clear), date (1/4/2026), and time (8:30 PM).

**Q7.** Write a PL/SQL program using a group function to count the total number of trains.

DECLARE

```
v_total_trains NUMBER;
```

BEGIN

```
    SELECT COUNT(*) INTO v_total_trains
```

```
    FROM train;
```

```
    DBMS_OUTPUT.PUT_LINE('Total Trains: ' || v_total_trains);
```

END;

/

The screenshot shows the Oracle SQL Developer interface. The top window displays a PL/SQL script:

```
DECLARE
  v_total_trains NUMBER;
BEGIN
    SELECT COUNT(*) INTO v_total_trains
    FROM train;

    DBMS_OUTPUT.PUT_LINE('Total Trains: ' || v_total_trains);
END;
/
```

The bottom window shows the execution results:

Total Trains: 5  
Statement processed.  
0.00 seconds

The system tray at the bottom right shows the date and time as 8:30 PM on 1/4/2026, along with other system icons.

**Q8.** Write a PL/SQL block to calculate the **average salary** of all employees.

## DECLARE

v\_avg\_salary NUMBER;

BEGIN

```
SELECT AVG(salary) INTO v_avg_salary
```

FROM emp;

```
DBMS_OUTPUT.PUT_LINE('Average Salary: ' || v_avg_salary);
```

END;

1

**Q9.** Write a PL/SQL program using a **FOR loop** to display all station names.

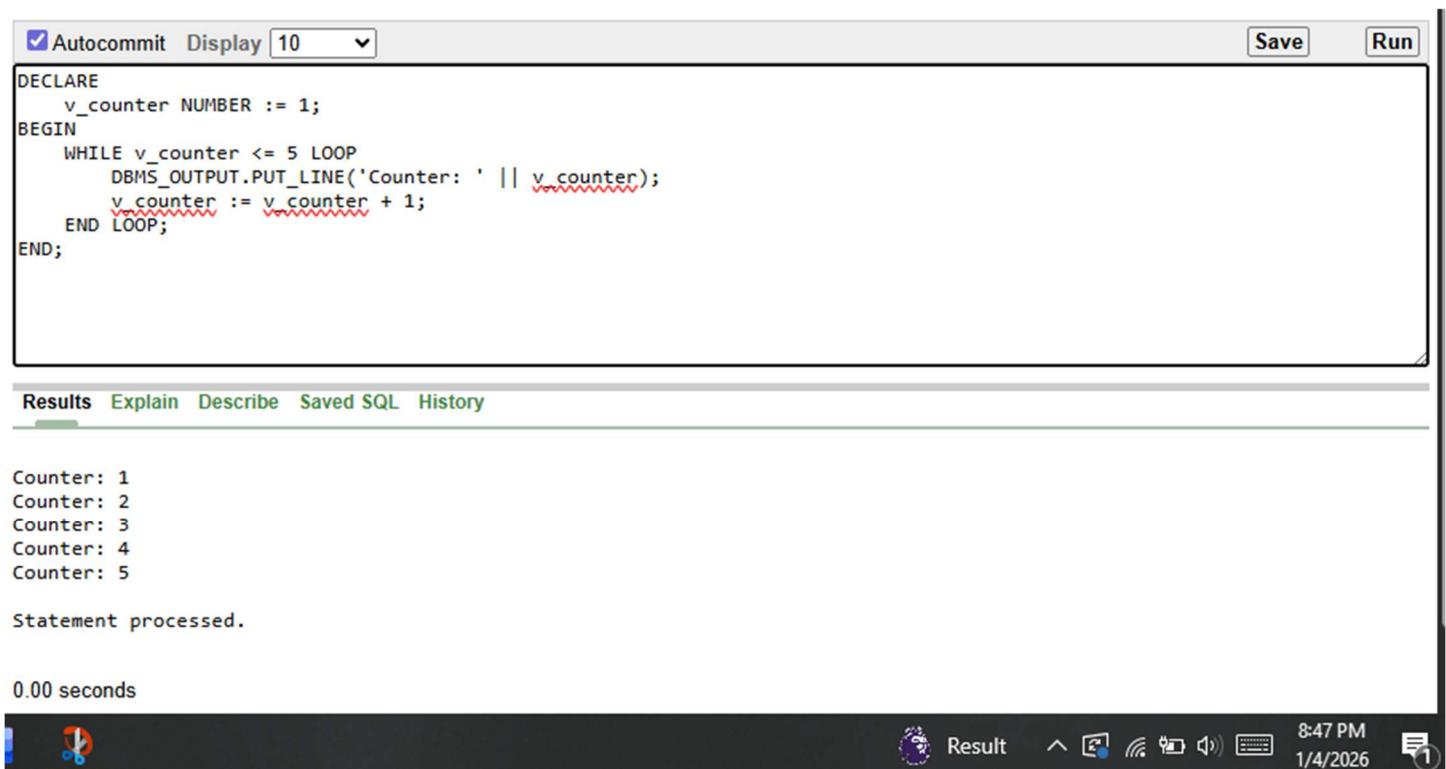
BEGIN

```
FOR r IN (
    SELECT station_name
    FROM Station
) LOOP
    DBMS_OUTPUT.PUT_LINE(r.station_name);
END LOOP;
END;
```

**Q10.** Write a PL/SQL program using a **WHILE loop** to display numbers from 1 to 5.

DECLARE

```
v_counter NUMBER := 1;  
  
BEGIN  
  
  WHILE v_counter <= 5 LOOP  
  
    DBMS_OUTPUT.PUT_LINE('Counter: ' || v_counter);  
  
    v_counter := v_counter + 1;  
  
  END LOOP;  
  
END;
```



The screenshot shows a database query editor window. At the top, there are checkboxes for 'Autocommit' and 'Display' (set to 10), and buttons for 'Save' and 'Run'. The main area contains the PL/SQL code. Below the code, a 'Results' tab is selected, showing the output of the script. The output displays the values of the counter variable from 1 to 5, followed by a message indicating the statement was processed. The bottom bar shows the time as 8:47 PM and the date as 1/4/2026.

```
DECLARE  
  v_counter NUMBER := 1;  
BEGIN  
  WHILE v_counter <= 5 LOOP  
    DBMS_OUTPUT.PUT_LINE('Counter: ' || v_counter);  
    v_counter := v_counter + 1;  
  END LOOP;  
END;
```

Results Explain Describe Saved SQL History

```
Counter: 1  
Counter: 2  
Counter: 3  
Counter: 4  
Counter: 5  
  
Statement processed.  
  
0.00 seconds
```

Result 8:47 PM 1/4/2026

**Q11.** Write a PL/SQL block using **IF-ELSE** to check whether an employee has a high salary.

DECLARE

v\_salary NUMBER;

BEGIN

SELECT salary INTO v\_salary

FROM emp

WHERE emp\_id = 1;

IF v\_salary > 30000 THEN

DBMS\_OUTPUT.PUT\_LINE('High Salary');

ELSE

DBMS\_OUTPUT.PUT\_LINE('Normal Salary');

END IF;

END;

/

The screenshot shows the Oracle SQL Developer interface. In the top-left corner, there's a toolbar with a checked 'Autocommit' checkbox, a 'Display' dropdown set to '10', and buttons for 'Save' and 'Run'. Below the toolbar is the SQL editor area containing the provided PL/SQL block. The code uses several reserved words like 'DECLARE', 'BEGIN', 'END', 'IF', 'ELSE', and 'END IF' in red, and variable names like 'v\_salary' in blue. The SQL editor has a scroll bar on the right. At the bottom of the editor, there are tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History', with 'Results' being the active tab. The results pane below shows the output of the executed query: 'High Salary' followed by 'Statement processed.' and '0.00 seconds'.

```
DECLARE
    v_salary NUMBER;
BEGIN
    SELECT salary INTO v_salary
    FROM emp
    WHERE emp_id = 1;

    IF v_salary > 30000 THEN
        DBMS_OUTPUT.PUT_LINE('High Salary');
    ELSE
        DBMS_OUTPUT.PUT_LINE('Normal Salary');
    END IF;
END;
/
```

Results Explain Describe Saved SQL History

High Salary

Statement processed.

0.00 seconds

Application Express 2.1.0.00.39  
Language: en-us Copyright © 1999, 2008, Oracle. All rights reserved.



**Q12.** Write a PL/SQL program using a **CASE statement** to display whether a schedule is on a weekday or holiday.

DECLARE

v\_day VARCHAR2(50);

BEGIN

```
SELECT days_of_operation INTO v_day  
FROM Schedule WHERE schedule_id = 14;
```

CASE

```
WHEN v_day LIKE '%Sat%' OR v_day LIKE '%Sun%' THEN
```

```
    DBMS_OUTPUT.PUT_LINE('Holiday');
```

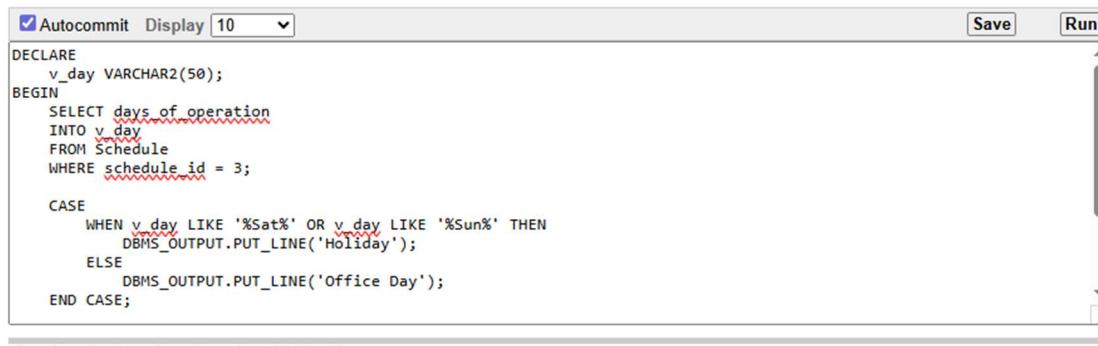
ELSE

```
    DBMS_OUTPUT.PUT_LINE('Office Day');
```

END CASE;

END;

/



```
Autocommit Display 10 Save Run  
DECLARE  
    v_day VARCHAR2(50);  
BEGIN  
    SELECT days_of_operation  
    INTO v_day  
    FROM Schedule  
    WHERE schedule_id = 3;  
  
    CASE  
        WHEN v_day LIKE '%Sat%' OR v_day LIKE '%Sun%' THEN  
            DBMS_OUTPUT.PUT_LINE('Holiday');  
        ELSE  
            DBMS_OUTPUT.PUT_LINE('Office Day');  
    END CASE;  
END;
```

Results Explain Describe Saved SQL History

Holiday

Statement processed.

0.00 seconds

Application Express 2.1.0.00.39  
Language: en-us Copyright © 1999, 2006, Oracle. All rights reserved.

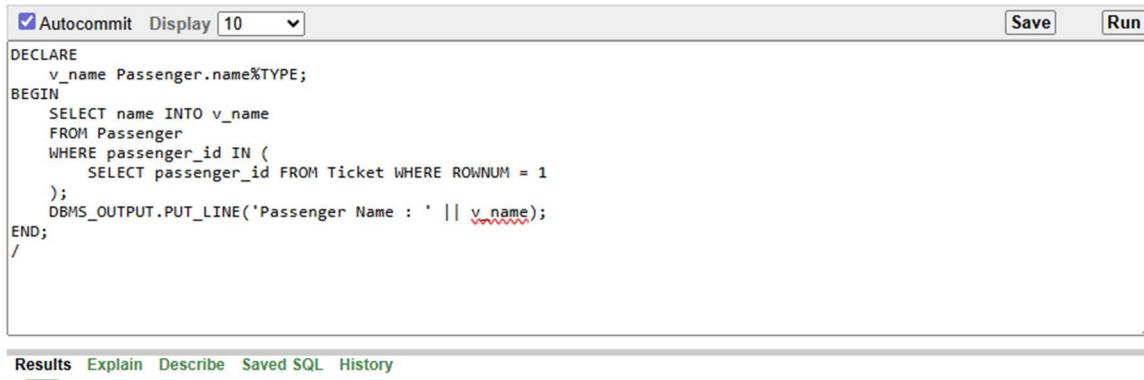


**Q13.** Write a PL/SQL block using a subquery to display passenger names who have booked tickets.

DECLARE

```
v_name Passenger.name%TYPE;  
  
BEGIN  
  
    SELECT name INTO v_name  
  
    FROM Passenger  
  
    WHERE passenger_id IN (  
  
        SELECT passenger_id FROM Ticket WHERE ROWNUM = 1  
  
    );  
  
    DBMS_OUTPUT.PUT_LINE('Passenger Name : ' || v_name);  
  
END;
```

/



The screenshot shows the Oracle Application Express SQL Workshop interface. The top bar has 'Autocommit' checked and 'Display' set to 10. There are 'Save' and 'Run' buttons. The main area contains the PL/SQL code for Question 13. Below the code is a toolbar with 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is selected.

```
DECLARE  
    v_name Passenger.name%TYPE;  
BEGIN  
    SELECT name INTO v_name  
    FROM Passenger  
    WHERE passenger_id IN (  
        SELECT passenger_id FROM Ticket WHERE ROWNUM = 1  
    );  
    DBMS_OUTPUT.PUT_LINE('Passenger Name : ' || v_name);  
END;  
/
```

Results Explain Describe Saved SQL History

Passenger Name : Test User

Statement processed.

0.01 seconds



The screenshot shows the results page of Oracle Application Express. The top status bar says 'Application Express 2.1.0.00.39' and 'Language: en-us'. The bottom status bar shows the date and time: 'Copyright © 1999, 2006, Oracle. All rights reserved.' and '7:56 PM 1/2/2026'. The main area displays the output from the previous screenshot.

**Q14.** Write a PL/SQL block using a subquery to find the train code assigned to a schedule.

DECLARE

```
v_code Train.train_code%TYPE;
```

BEGIN

```
    SELECT train_code INTO v_code
```

```
    FROM Train
```

```
    WHERE train_id = (
```

```
        SELECT train_id FROM Schedule WHERE ROWNUM = 1
```

```
    );
```

```
    DBMS_OUTPUT.PUT_LINE('Train Code : ' || v_code);
```

END;

/

The screenshot shows the Oracle SQL Developer interface. The code area contains the PL/SQL block provided above. The output area shows the results of the execution:

```
Autocommit Display 10 Save Run
DECLARE
  v_code Train.train_code%TYPE;
BEGIN
  SELECT train_code INTO v_code
  FROM Train
  WHERE train_id = (
    SELECT train_id FROM Schedule WHERE ROWNUM = 1
  );
  DBMS_OUTPUT.PUT_LINE('Train Code : ' || v_code);
END;
/

```

Results Explain Describe Saved SQL History

Train Code : TR-001

Statement processed.

0.00 seconds

Application Express 2.1.0.00.35

Copyright © 1999, 2006, Oracle. All rights reserved



**Q15.** Write a PL/SQL block using JOIN to display passenger name and ticket ID.

DECLARE

```
v_name Passenger.name%TYPE;
```

```
v_ticket Ticket.ticket_id%TYPE;
```

BEGIN

```
SELECT p.name, t.ticket_id
```

```
INTO v_name, v_ticket
```

```
FROM Passenger p
```

```
JOIN Ticket t
```

```
ON p.passenger_id = t.passenger_id
```

```
WHERE ROWNUM = 1;
```

```
DBMS_OUTPUT.PUT_LINE('Passenger : ' || v_name);
```

```
DBMS_OUTPUT.PUT_LINE('Ticket ID : ' || v_ticket);
```

END;

The screenshot shows the Oracle Application Express (APEX) interface. At the top, there is a toolbar with 'Autocommit' checked, 'Display' set to 10, and buttons for 'Save' and 'Run'. The main area contains the PL/SQL code. Below the code, the 'Results' tab is selected, showing the output of the executed query. The bottom status bar indicates the application version, language, and copyright information.

```
/ Autocommit Display 10 Save Run
v_name Passenger.name%TYPE;
v_ticket Ticket.ticket_id%TYPE;
BEGIN
  SELECT p.name, t.ticket_id
  INTO v_name, v_ticket
  FROM Passenger p
  JOIN Ticket t
  ON p.passenger_id = t.passenger_id
  WHERE ROWNUM = 1;

  DBMS_OUTPUT.PUT_LINE('Passenger : ' || v_name);
  DBMS_OUTPUT.PUT_LINE('Ticket ID : ' || v_ticket);
END;
/

```

Results Explain Describe Saved SQL History

Passenger : Test User  
Ticket ID : 1  
Statement processed.  
0.00 seconds

Application Express 2.1.0.00.39  
Language: en-us Copyright © 1999, 2006, Oracle. All rights reserved.

**Q16. Write a PL/SQL block using JOIN to display employee name and station name.**

DECLARE

```
v_emp Emp.name%TYPE;
```

```
v_station Station.station_name%TYPE;
```

BEGIN

```
SELECT e.name, s.station_name
```

```
INTO v_emp, v_station
```

```
FROM Emp e
```

```
JOIN Station s
```

```
ON e.station_id = s.station_id
```

```
WHERE ROWNUM = 1;
```

```
DBMS_OUTPUT.PUT_LINE('Employee : ' || v_emp);
```

```
DBMS_OUTPUT.PUT_LINE('Station : ' || v_station);
```

END;

/

```
Autocommit Display 10 Save Run
v_station Station.station_name%TYPE;
BEGIN
    SELECT e.name, s.station_name
    INTO v_emp, v_station
    FROM Emp e
    JOIN Station s
    ON e.station_id = s.station_id
    WHERE ROWNUM = 1;

    DBMS_OUTPUT.PUT_LINE('Employee : ' || v_emp);
    DBMS_OUTPUT.PUT_LINE('Station : ' || v_station);
END;
/
```

Results Explain Describe Saved SQL History

Employee : Rina Sultana  
Station : Uttara North  
Statement processed.  
0.01 seconds

Application Express 2.1.0.0.39  
Language: en-us Copyright © 1999, 2006, Oracle. All rights reserved.  
8:01 PM 1/2/2026

# Advance PL/SQL

**Q1.** Create a stored function that returns the **total number of passengers** in the metro system.

**Answer:**

```
CREATE OR REPLACE FUNCTION total_passengers
RETURN NUMBER IS
    v_count NUMBER;
BEGIN
    SELECT COUNT(*) INTO v_count FROM passenger;
    RETURN v_count;
END;

BEGIN
    DBMS_OUTPUT.PUT_LINE('Total passengers: ' || total_passengers);
END;
/
```

The screenshot shows the Oracle SQL Developer interface. In the top toolbar, 'Autocommit' is checked, 'Display' is set to 10, and there are 'Save' and 'Run' buttons. The main area contains the PL/SQL code for the 'total\_passengers' function. The code includes a function body with a local variable 'v\_count', a SELECT statement to count passengers, and a DBMS\_OUTPUT.PUT\_LINE call to print the result. Below the code, the 'Results' tab is selected, showing the output: 'Total passengers: 5' and 'Statement processed.'. At the bottom, the status bar indicates 'Application Express 2.1.0.00.39', 'Language: en-us', 'Copyright © 1999, 2006, Oracle. All rights reserved.', and a timestamp '8:51 PM 1/4/2026'.

```
CREATE OR REPLACE FUNCTION total_passengers
RETURN NUMBER IS
    v_count NUMBER;
BEGIN
    SELECT COUNT(*) INTO v_count FROM passenger;
    RETURN v_count;
END;

BEGIN
    DBMS_OUTPUT.PUT_LINE('Total passengers: ' || total_passengers);
END;
/
```

Total passengers: 5  
Statement processed.  
0.00 seconds

Application Express 2.1.0.00.39  
Language: en-us Copyright © 1999, 2006, Oracle. All rights reserved.  
Air quality forecast 8:51 PM 1/4/2026

**Q2.** Create a stored function that returns the **salary of an employee** based on employee ID.

**Answer:**

```
CREATE OR REPLACE FUNCTION get_employee_salary(p_emp_id NUMBER)
```

```
RETURN NUMBER IS
```

```
    v_salary NUMBER;
```

```
BEGIN
```

```
    SELECT salary INTO v_salary
```

```
    FROM emp
```

```
    WHERE emp_id = p_emp_id;
```

```
    RETURN v_salary;
```

```
END;
```

```
BEGIN
```

```
    DBMS_OUTPUT.PUT_LINE(
```

```
        'Employee Salary: ' || get_employee_salary(1)
```

```
);
```

```
END;
```

The screenshot shows the Oracle Application Express interface. In the top-left corner, there is a dropdown menu set to 'Display 10'. On the right side, there are 'Save' and 'Run' buttons. The main workspace contains the PL/SQL code for the function. The code is as follows:

```
CREATE OR REPLACE FUNCTION get_employee_salary(p_emp_id NUMBER)
RETURN NUMBER IS
    v_salary NUMBER;
BEGIN
    SELECT salary INTO v_salary
    FROM emp
    WHERE emp_id = p_emp_id;
    RETURN v_salary;
END;
/
BEGIN
    DBMS_OUTPUT.PUT_LINE(
        'Employee Salary: ' || get_employee_salary(1)
    );
END;
```

Below the code, the results of the execution are displayed:

```
Employee Salary: 45000
Statement processed.

0.00 seconds
```

In the bottom status bar, it shows 'Language: en-us', 'Application Express 2.1.0.0.39', 'Copyright © 1999, 2006, Oracle. All rights reserved.', the date '1/2/2026', and the time '8:05 PM'. There are also icons for file operations like 'New', 'Open', and 'Save'.

**Q3.** Create a stored procedure to **increase employee salary by 10%**.

**Answer:**

```
CREATE OR REPLACE PROCEDURE increase_salary(p_emp_id NUMBER) IS
```

```
BEGIN
```

```
    UPDATE emp
```

```
    SET salary = salary * 1.10
```

```
    WHERE emp_id = p_emp_id;
```

```
END;
```

```
/
```

The screenshot shows the Oracle Application Express SQL Workshop interface. The code area contains the following PL/SQL code:

```
CREATE OR REPLACE PROCEDURE increase_salary(p_emp_id NUMBER) IS
BEGIN
    UPDATE employee
    SET salary = salary * 1.10
    WHERE employee_id = p_emp_id;
END;
/
```

The code editor has syntax highlighting where 'employee' and 'employee\_id' are red, indicating they are identifiers. The 'Run' button is visible at the top right of the code editor.

**Results** Explain Describe Saved SQL History

Employee Salary: 45000

Statement processed.

0.00 seconds

Application Express 2.1.0.00.3

Copyright © 1999, 2006, Oracle. All rights reserved.

Language: en-us



**Q4. Create a stored procedure to insert a new passenger.**

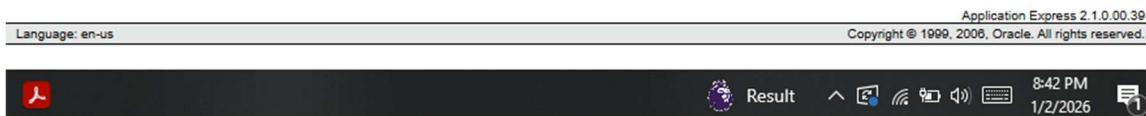
```
CREATE OR REPLACE PROCEDURE add_passenger (p_name IN Passenger.name%TYPE, p_gender IN Passenger.gender%TYPE, p_dob IN Passenger.dob%TYPE, p_nid IN Passenger.nid%TYPE, p_phone IN Passenger.phone%TYPE, p_email IN Passenger.email%TYPE, p_address IN Passenger.address%TYPE
) IS
BEGIN
INSERT INTO Passenger VALUES ( seq_passenger_id.NEXTVAL, p_name, p_gender, p_dob, p_nid,
p_phone, p_email, p_address,CURRENT_TIMESTAMP );
DBMS_OUTPUT.PUT_LINE('Passenger inserted successfully');
END;
/
```

```
CREATE OR REPLACE PROCEDURE add_passenger (p_name IN Passenger.name%TYPE, p_gender IN Passenger.gender%TYPE,
p_dob IN Passenger.dob%TYPE, p_nid IN Passenger.nid%TYPE, p_phone IN Passenger.phone%TYPE, p_email IN
Passenger.email%TYPE, p_address IN Passenger.address%TYPE
) IS
BEGIN
INSERT INTO Passenger VALUES ( seq_passenger_id.NEXTVAL, p_name, p_gender, p_dob, p_nid, p_phone, p_email,
p_address,CURRENT_TIMESTAMP );
DBMS_OUTPUT.PUT_LINE('Passenger inserted successfully');
END;
/
```

Results Explain Describe Saved SQL History

Procedure created.

0.03 seconds



**Q5.** Write a PL/SQL block using a **table-based record** to fetch an employee row.

**Answer:**

DECLARE

v\_emp emp%ROWTYPE;

BEGIN

SELECT \* INTO v\_emp

FROM emp

WHERE emp\_id = 1;

DBMS\_OUTPUT.PUT\_LINE(v\_emp.name || ' - ' || v\_emp.salary);

END;

/

The screenshot shows the Oracle Application Express SQL Workshop interface. The code area contains a PL/SQL block:

```
DECLARE
    v_emp emp%ROWTYPE;
BEGIN
    SELECT * INTO v_emp
    FROM emp
    WHERE emp_id = 1;

    DBMS_OUTPUT.PUT_LINE(v_emp.name || ' - ' || v_emp.salary);
END;
/
```

The results pane shows the output of the query:

```
Rahim Khan - 45000
Statement processed.
```

At the bottom, system status icons are visible, including a weather icon (16°C Mostly clear), signal strength, battery level, and network connectivity.

**Q6.** Write a PL/SQL block to display **train details** using a table-based record.

**Answer:**

DECLARE

v\_train train%ROWTYPE;

BEGIN

SELECT \* INTO v\_train

FROM train

WHERE train\_id = 1;

DBMS\_OUTPUT.PUT\_LINE(v\_train.train\_code || ' Capacity: ' || v\_train.capacity);

END;

/

The screenshot shows a PL/SQL editor window with the following code:

```
DECLARE
    v_train train%ROWTYPE;
BEGIN
    SELECT * INTO v_train
    FROM train
    WHERE train_id = 1;

    DBMS_OUTPUT.PUT_LINE(v_train.train_code || ' Capacity: ' || v_train.capacity);
END;
/
```

The code is run and produces the following output:

Results Explain Describe Saved SQL History

TR-001 Capacity: 2300  
Statement processed.  
0.00 seconds

Application Express 2.1.0.00.39  
Language: en-us Copyright © 1999, 2006, Oracle. All rights reserved.

16°C Mostly clear 8:46 PM 1/2/2026

**Q7.** Write a PL/SQL program using an **explicit cursor** to list all station names.

**Answer:**

```
DECLARE
  CURSOR c_station IS SELECT station_name FROM station;
  v_station station.station_name%TYPE;
BEGIN
  OPEN c_station;
  LOOP
    FETCH c_station INTO v_station;
    EXIT WHEN c_station%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE(v_station);
  END LOOP;
  CLOSE c_station;
END;
```

/

The screenshot shows the Oracle SQL Developer interface. The code editor window displays a PL/SQL block. The output window below shows the results of the query, listing station names: Uttara North, Pallabi, Agargaon, Motijheel, and Farmgate. The status bar at the bottom right shows the date and time as 8:53 PM 1/4/2026.

```
Autocommit Display 10 Save Run
DECLARE
  CURSOR c_station IS SELECT station_name FROM station;
  v_station station.station_name%TYPE;
BEGIN
  OPEN c_station;
  LOOP
    FETCH c_station INTO v_station;
    EXIT WHEN c_station%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE(v_station);
  END LOOP;
  CLOSE c_station;
END;
```

Results Explain Describe Saved SQL History

Uttara North  
Pallabi  
Agargaon  
Motijheel  
Farmgate

Statement processed.

0.02 seconds

Air quality forecast 8:53 PM 1/4/2026

**Q8.** Write a PL/SQL block using an **explicit cursor** to display employee names.

**Answer:**

DECLARE

CURSOR c\_emp IS SELECT name FROM emp;

v\_name emp.name%TYPE;

BEGIN

OPEN c\_emp;

LOOP

FETCH c\_emp INTO v\_name;

EXIT WHEN c\_emp%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE(v\_name);

END LOOP;

CLOSE c\_emp;

END;

/

The screenshot shows the Oracle Application Express interface. In the top window, a PL/SQL block is being written:

```
DECLARE
    CURSOR c_emp IS SELECT name FROM emp;
    v_name emp.name%TYPE;
BEGIN
    OPEN c_emp;
    LOOP
        FETCH c_emp INTO v_name;
        EXIT WHEN c_emp%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE(v_name);
    END LOOP;
    CLOSE c_emp;
END;
```

The bottom window shows the results of the execution:

Rahim Khan  
Karim Ali  
Salma Akter  
Hasan Mahmud  
Rinu Sultana

Statement processed.

0.00 seconds

Application Express 2.1.0.00.39      8:48 PM      1/2/2026

**Q9.** Write a PL/SQL block using a **cursor-based record** to display employee details.

**Answer:**

DECLARE

CURSOR c\_emp IS SELECT \* FROM emp;

v\_emp c\_emp%ROWTYPE;

BEGIN

OPEN c\_emp;

LOOP

FETCH c\_emp INTO v\_emp;

EXIT WHEN c\_emp%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE(v\_emp.full\_name || ' ' || v\_emp.role);

END LOOP;

CLOSE c\_emp;

END;

/

The screenshot shows the Oracle Application Express (APEX) interface. The top bar has 'Autocommit' checked, 'Display' set to 10, and buttons for 'Save' and 'Run'. The main area contains the PL/SQL code provided above. Below the code, the 'Results' tab is selected, showing the output of the query:

```
Rahim Khan Driver
Karim Ali Controller
Salma Akter Station Master
Hasan Mahmud Technician
Rina Sultana Ticket Officer
```

At the bottom, it says 'Statement processed.' and '0.02 seconds'. The footer shows 'Application Express 2.1.0.00.30', the time '8:50 PM', and the date '1/2/2026'.

**Q10.** Write a PL/SQL program using a **cursor-based record** to show train information.

**Answer:**

```
DECLARE
```

```
    CURSOR c_train IS SELECT * FROM train;
```

```
    v_train c_train%ROWTYPE;
```

```
BEGIN
```

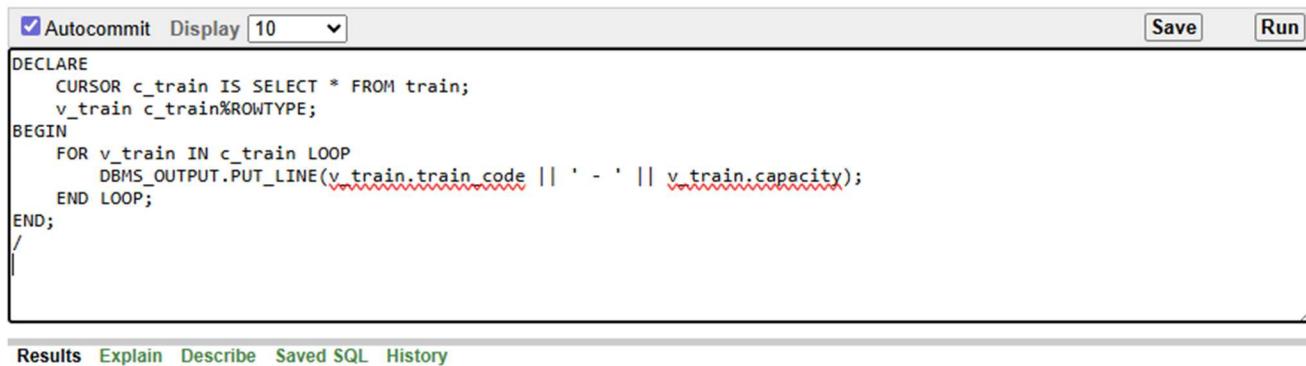
```
    FOR v_train IN c_train LOOP
```

```
        DBMS_OUTPUT.PUT_LINE(v_train.train_code || ' - ' || v_train.capacity);
```

```
    END LOOP;
```

```
END;
```

```
/
```



```
Autocommit Display 10 Save Run
DECLARE
    CURSOR c_train IS SELECT * FROM train;
    v_train c_train%ROWTYPE;
BEGIN
    FOR v_train IN c_train LOOP
        DBMS_OUTPUT.PUT_LINE(v_train.train_code || ' - ' || v_train.capacity);
    END LOOP;
END;
/
```

Results Explain Describe Saved SQL History

```
TR-001 - 2300
TR-002 - 2300
TR-003 - 2200
TR-004 - 2300
TR-005 - 2400
```

```
Statement processed.
```

```
0.02 seconds
```

Application Express 2.1.0.00.39



8:55 PM

1/2/2026

**Q11.** Create a **row-level trigger** that prevents inserting employees with salary less than 20,000.

**Answer:**

```
CREATE OR REPLACE TRIGGER trg_check_salary
```

```
BEFORE INSERT ON Emp
```

```
FOR EACH ROW
```

```
BEGIN
```

```
IF :NEW.salary < 20000 THEN
```

```
    :NEW.salary := 20000;
```

```
    DBMS_OUTPUT.PUT_LINE(
```

```
        'Salary was below 20000. Automatically set to 20000.'
```

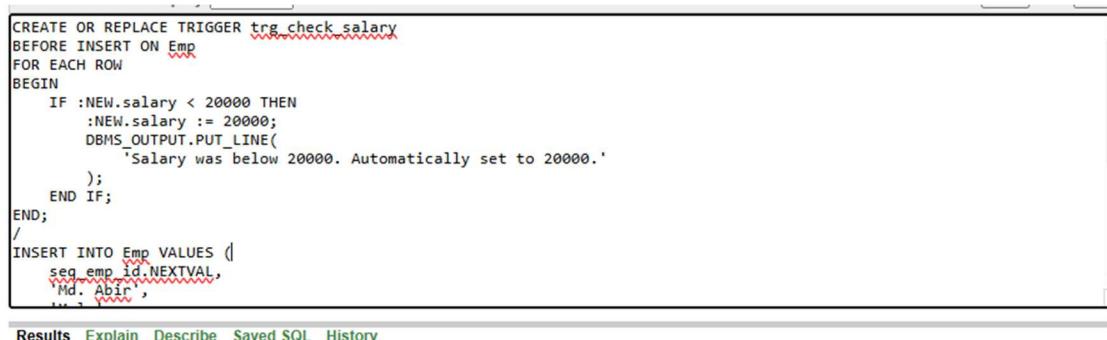
```
);
```

```
END IF;
```

```
END;
```

```
/
```

```
INSERT INTO Emp VALUES( seq_emp_id.NEXTVAL, 'Md. Abir', 'Male', 'Cleaner', 15000, SYSDATE, '08:00 AM', '04:00 PM', 'Active', 1, 1);
```



The screenshot shows the SQL code for creating a trigger. The code is as follows:

```
CREATE OR REPLACE TRIGGER trg_check_salary
BEFORE INSERT ON Emp
FOR EACH ROW
BEGIN
    IF :NEW.salary < 20000 THEN
        :NEW.salary := 20000;
        DBMS_OUTPUT.PUT_LINE(
            'Salary was below 20000. Automatically set to 20000.'
        );
    END IF;
END;
/
INSERT INTO Emp VALUES (
    seq_emp_id.NEXTVAL,
    'Md. Abir',
```

Results Explain Describe Saved SQL History

Salary was below 20000. Automatically set to 20000.

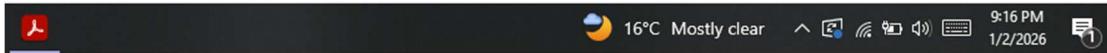
1 row(s) inserted.

0.00 seconds

Language: en-us

Application Express 2.1.0.00.3

Copyright © 1999, 2006, Oracle. All rights reserved.



**Q12.** Create a **row-level trigger** to automatically update maintenance status.

**Answer:**

```
CREATE OR REPLACE TRIGGER trg_maintenance_status
```

```
BEFORE INSERT ON Maintenance
```

```
FOR EACH ROW
```

```
BEGIN
```

```
    :NEW.status := 'Completed';
```

```
END;
```

```
/
```

```
INSERT INTO Maintenance VALUES ( seq_maintenance_id.NEXTVAL,'Routine brake check','Pending',2,2,2);
```

```
select * from maintenance where maintenance_id = 7
```

The screenshot shows the Oracle Application Express SQL Workshop interface. The code area contains the following SQL statements:

```
CREATE OR REPLACE TRIGGER trg_maintenance_status
BEFORE INSERT ON Maintenance
FOR EACH ROW
BEGIN
    :NEW.status := 'Completed';
END;
/
INSERT INTO Maintenance VALUES ( seq_maintenance_id.NEXTVAL,'Routine brake check','Pending',2,2,2);

select * from maintenance where maintenance_id = 7
```

Results Explain Describe Saved SQL History

Maintenance_ID	Description	Status	Train_ID	Station_ID	Emp_ID
7	Routine brake check	Completed	2	2	2

1 rows returned in 0.00 seconds

[CSV Export](#)

Application Express 2.1.0.00.39

Copyright © 1999, 2008, Oracle. All rights reserved.



**Q13.** Create a **statement-level trigger** that displays a message after inserting passengers.

**Answer:**

```
CREATE OR REPLACE TRIGGER trg_passenger_insert
```

```
AFTER INSERT ON Passenger
```

```
BEGIN
```

```
    DBMS_OUTPUT.PUT_LINE('Passenger record inserted successfully');
```

```
END;
```

```
/
```

```
INSERT INTO passenger (passenger_id, full_name, nid, phone, email, gender, date_of_birth) VALUES  
(seq_passenger_id.NEXTVAL,'Abir
```

```
Saha','1234567890132','01787070862','abir@mail.com','Male',TO_DATE('2000-07-15','YYYY-MM-DD'));
```

The screenshot shows the Oracle SQL Developer interface. The code area contains the trigger creation script and an insert statement. The results tab shows the output of the trigger message and the successful insertion of a new row into the passenger table. The bottom status bar indicates the language is en-us and the application version is 2.1.0.00.39.

```
CREATE OR REPLACE TRIGGER trg_passenger_insert
AFTER INSERT ON Passenger
BEGIN
    DBMS_OUTPUT.PUT_LINE('Passenger record inserted successfully');
END;
/
INSERT INTO passenger VALUES (seq_passenger_id.NEXTVAL,'Ashish Saha','Male',TO_DATE('2000-07-15','YYYY-MM-  
DD'),'1234567890132','01787070862','abir@mail.com','Dhaka',CURRENT_TIMESTAMP);

select * from passenger
```

Results Explain Describe Saved SQL History

Passenger record inserted successfully

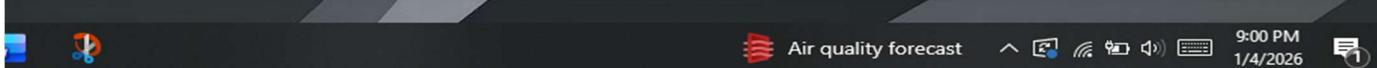
1 row(s) inserted.

0.01 seconds

Application Express 2.1.0.00.39

Copyright © 1999, 2006, Oracle. All rights reserved.

Language: en-us



**Q14.** Create a **statement-level trigger** that prevents deleting all trains.

**Answer:**

```
CREATE OR REPLACE TRIGGER trg_no_train_delete
BEFORE DELETE ON Train
BEGIN
    DBMS_OUTPUT.PUT_LINE('WARNING: Train records are being deleted');
END;
/
delete from train
```

The screenshot shows a SQL developer window with the following details:

- Autocommit:** Checked.
- Display:** Set to 10.
- Buttons:** Save and Run.
- SQL Editor Content:**

```
CREATE OR REPLACE TRIGGER trg_no_train_delete
BEFORE DELETE ON Train
BEGIN
    DBMS_OUTPUT.PUT_LINE('WARNING: Train records are being deleted');
END;
/
delete from train
```

**Results** Explain Describe Saved SQL History

Trigger created.

0.02 seconds



**Q15.** Create a package that contains a procedure to display all station names.

**Answer (Package Spec & Body):**

```
CREATE OR REPLACE PACKAGE station_pkg IS
    PROCEDURE show_stations;
END;
/
CREATE OR REPLACE PACKAGE BODY station_pkg IS
    PROCEDURE show_stations IS
        v_count NUMBER;
    BEGIN
        SELECT COUNT(*) INTO v_count FROM Station;
        IF v_count = 0 THEN
            DBMS_OUTPUT.PUT_LINE('No stations available');
        ELSE
            FOR r IN (SELECT station_name FROM Station) LOOP
                DBMS_OUTPUT.PUT_LINE(r.station_name);
            END LOOP;
        END IF;
    END show_stations;
END station_pkg;
/BEGIN
station_pkg.show_stations;
END;
/
```

The screenshot shows the Oracle Application Express SQL Worksheet interface. The code area displays the package specification and body. The results area shows the output of the package execution, listing station names: Uttara North, Pallabi, Agargaon, Motijheel, and Farmgate. The bottom status bar indicates the statement was processed in 0.02 seconds.

```
CREATE OR REPLACE PACKAGE station_pkg IS
    PROCEDURE show_stations;
END;
/
CREATE OR REPLACE PACKAGE BODY station_pkg IS
    PROCEDURE show_stations IS
        v_count NUMBER;
    BEGIN
        SELECT COUNT(*) INTO v_count FROM Station;
        IF v_count = 0 THEN
            DBMS_OUTPUT.PUT_LINE('No stations available');
        ELSE
            FOR r IN (SELECT station_name FROM Station) LOOP
                DBMS_OUTPUT.PUT_LINE(r.station_name);
            END LOOP;
        END IF;
    END show_stations;
END station_pkg;/BEGIN
station_pkg.show_stations;
END;
```

Results Explain Describe Saved SQL History

```
Uttara North
Pallabi
Agargaon
Motijheel
Farmgate
```

Statement processed.

0.02 seconds

**Q16.** Create a package that contains a function to return total trains.

**Answer:**

```
CREATE OR REPLACE PACKAGE train_pkg IS
    FUNCTION total_trains RETURN NUMBER;
END train_pkg;
/
CREATE OR REPLACE PACKAGE BODY train_pkg IS
    FUNCTION total_trains RETURN NUMBER IS
        v_count NUMBER;
    BEGIN
        SELECT COUNT(*) INTO v_count FROM Train;
        RETURN v_count;
    END total_trains;
END train_pkg;
/
DECLARE
    v_total NUMBER;
BEGIN
    v_total := train_pkg.total_trains;
    DBMS_OUTPUT.PUT_LINE('Total Trains = ' || v_total);
END;
/
```

```

CREATE OR REPLACE PACKAGE train_pkg IS
  FUNCTION total_trains RETURN NUMBER;
END train_pkg;
/
CREATE OR REPLACE PACKAGE BODY train_pkg IS
  FUNCTION total_trains RETURN NUMBER IS
    v_count NUMBER;
  BEGIN
    SELECT COUNT(*)
      INTO v_count
      FROM Train;
    RETURN v_count;
  END total_trains;
END train_pkg;

```

Results Explain Describe Saved SQL History

Total Trains = 5  
Statement processed.  
0.00 seconds

Application Express 2.1.0.00.3  
Language: en-us Copyright © 1999, 2006, Oracle. All rights reserved.

Result 9:04 PM 1/2/2026

## Conclusion

The Metro Rail Management System provides a comprehensive and well-structured framework for managing the complex operations of a modern metro service. By integrating key components such as lines, trains, stations, routes, schedules, employees, passengers, ticketing, payments, fares, and maintenance activities, the system ensures efficient coordination across all operational levels. Clearly defined relationships between entities help maintain data consistency, accountability, and transparency, from daily train operations to long-term infrastructure maintenance. The inclusion of detailed employee management supports smooth scheduling, staffing, and maintenance execution, while passenger-focused features such as ticket booking, fare calculation, and secure payment processing enhance user convenience and trust. The one-to-one relationship between tickets and payments ensures financial accuracy, and maintenance logs promote safety and reliability by tracking service activities. Overall, this system not only improves operational efficiency but also supports better decision-making through organized data management. By ensuring reliable services, optimized resource utilization, and improved passenger satisfaction, the Metro Rail Management System plays a vital role in delivering a safe, punctual, and customer-friendly urban transportation solution.

## Added exception handling section to Advance PL/SQL Codes

**Q1.** Create a stored function that returns the **total number of passengers** in the metro system.

**Answer:**

```
CREATE OR REPLACE FUNCTION total_passengers
RETURN NUMBER IS
  v_count NUMBER;
BEGIN
  SELECT COUNT(*) INTO v_count FROM passenger;
  RETURN v_count;

EXCEPTION
  WHEN NO_DATA_FOUND THEN
    RETURN 0;
  WHEN OTHERS THEN
    RETURN -1;
END;

BEGIN
  DBMS_OUTPUT.PUT_LINE('Total passengers: ' || total_passengers);
END;
/
```

The screenshot shows the Oracle Application Express interface. The top bar includes checkboxes for 'Autocommit' and 'Display 10', and buttons for 'Save' and 'Run'. The main area contains the PL/SQL code from the previous text. The code is executed, and the results are shown in the 'Results' tab. The output is:

```
Total passengers: 6
Statement processed.

0.00 seconds
```

At the bottom, the status bar displays 'Language: en-us', 'Application Express 2.1.0.00.39', 'Copyright © 1999, 2006, Oracle. All rights reserved.', and a series of icons including a Wi-Fi signal, battery level, and date/time (9:18 PM, 1/4/2026).

**Q2.** Create a stored function that returns the **salary of an employee** based on employee ID.

**Answer:**

```
CREATE OR REPLACE FUNCTION get_employee_salary(p_emp_id NUMBER)
RETURN NUMBER IS
v_salary NUMBER;
BEGIN
SELECT salary INTO v_salary
FROM emp
WHERE emp_id = p_emp_id;
RETURN v_salary;
EXCEPTION
WHEN NO_DATA_FOUND THEN
DBMS_OUTPUT.PUT_LINE('Employee not found');
RETURN NULL;
WHEN OTHERS THEN
DBMS_OUTPUT.PUT_LINE('Error retrieving salary');
RETURN NULL;
END;
BEGIN
DBMS_OUTPUT.PUT_LINE(
'Employee Salary: ' || get_employee_salary(1)
);
END;
```

```

Autocommit Display 10
v_salary NUMBER;
BEGIN
  SELECT salary INTO v_salary
  FROM emp
  WHERE emp_id = p_emp_id;
  RETURN v_salary;
EXCEPTION
  WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE('Employee not found');
    RETURN NULL;
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Error retrieving salary');
    RETURN NULL;
END;
BEGIN
  DBMS_OUTPUT.PUT_LINE(
    'Employee Salary: ' || get_employee_salary(1));
END;

```

Results Explain Describe Saved SQL History

Employee Salary: 45000  
Statement processed.  
0.02 seconds

Application Express 2.1.0.00.39  
Language: en-us Copyright © 1999, 2006, Oracle. All rights reserved.

Supergirl official teas... 9:21 PM 1/4/2026

**Q3.** Create a stored procedure to **increase employee salary by 10%**.

**Answer:**

```
CREATE OR REPLACE PROCEDURE increase_salary(p_emp_id NUMBER) IS
```

```
BEGIN
```

```
  UPDATE emp
```

```
    SET salary = salary * 1.10
```

```
    WHERE emp_id = p_emp_id;
```

```
EXCEPTION
```

```
  WHEN OTHERS THEN
```

```
    DBMS_OUTPUT.PUT_LINE('Error increasing salary');
```

```
END;
```

```
/
```

Autocommit

```
CREATE OR REPLACE PROCEDURE increase_salary(p_emp_id NUMBER) IS
BEGIN
    UPDATE emp
    SET salary = salary * 1.10
    WHERE emp_id = p_emp_id;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error increasing salary');
END;
/
```

Results Explain Describe Saved SQL History

Procedure created.

0.00 seconds

Application Express 2.1.0.00.39

Copyright © 1999, 2006, Oracle. All rights reserved.



**Q4. Create a stored procedure to insert a new passenger.**

```
CREATE OR REPLACE PROCEDURE add_passenger (p_name IN Passenger.name%TYPE, p_gender IN Passenger.gender%TYPE, p_dob IN Passenger.dob%TYPE, p_nid IN Passenger.nid%TYPE, p_phone IN Passenger.phone%TYPE, p_email IN Passenger.email%TYPE, p_address IN Passenger.address%TYPE ) IS
BEGIN
    INSERT INTO Passenger VALUES ( seq_passenger_id.NEXTVAL, p_name, p_gender, p_dob, p_nid, p_phone, p_email, p_address,CURRENT_TIMESTAMP );
    DBMS_OUTPUT.PUT_LINE('Passenger inserted successfully');
EXCEPTION
    WHEN DUP_VAL_ON_INDEX THEN
        DBMS_OUTPUT.PUT_LINE('Duplicate passenger entry');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error inserting passenger');
END;
```

/

The screenshot shows the Oracle Application Express interface. In the top navigation bar, 'Autocommit' is checked and 'Display' is set to 10. On the right, there are 'Save' and 'Run' buttons. The main area contains the PL/SQL code for the 'add\_passenger' procedure. Below the code, the 'Results' tab is selected, showing the message 'Procedure created.' and '0.00 seconds'. At the bottom, it shows 'Language: en-us' and 'Copyright © 1999, 2006, Oracle. All rights reserved.' The status bar at the bottom right includes icons for battery, signal, and network, along with the time '9:23 PM' and date '1/4/2026'.

```
CREATE OR REPLACE PROCEDURE add_passenger (p_name IN Passenger.name%TYPE, p_gender IN Passenger.gender%TYPE, p_dob IN Passenger.dob%TYPE, p_nid IN Passenger.nid%TYPE, p_phone IN Passenger.phone%TYPE, p_email IN Passenger.email%TYPE, p_address IN Passenger.address%TYPE ) IS
BEGIN
    INSERT INTO Passenger VALUES ( seq_passenger_id.NEXTVAL, p_name, p_gender, p_dob, p_nid, p_phone, p_email, p_address,CURRENT_TIMESTAMP );
    DBMS_OUTPUT.PUT_LINE('Passenger inserted successfully');
EXCEPTION
    WHEN DUP_VAL_ON_INDEX THEN
        DBMS_OUTPUT.PUT_LINE('Duplicate passenger entry');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error inserting passenger');
END;
```

**Q5.** Write a PL/SQL block using a **table-based record** to fetch an employee row.

**Answer:**

```
DECLARE
    v_emp emp%ROWTYPE;
BEGIN
    SELECT * INTO v_emp
    FROM emp
    WHERE emp_id = 1;
    DBMS_OUTPUT.PUT_LINE(v_emp.name || ' - ' || v_emp.salary);
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('Employee not found');
    WHEN TOO_MANY_ROWS THEN
        DBMS_OUTPUT.PUT_LINE('Multiple employees found');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error fetching employee');
END;
```

/

The screenshot shows the Oracle Application Express interface. The top bar includes checkboxes for 'Autocommit' and 'Display 10', and buttons for 'Save' and 'Run'. The main area contains the PL/SQL code. Below the code, the 'Results' tab is selected, showing the output of the query. The output displays 'Rahim Khan - 45000' followed by 'Statement processed.' and a timestamp '0.01 seconds'. At the bottom, a footer bar shows 'Language: en-us', the application version 'Application Express 2.1.0.00.39', and copyright information 'Copyright © 1999, 2008, Oracle. All rights reserved.'. The system tray at the bottom right shows the date '1/4/2026', time '9:24 PM', battery level, signal strength, and other system icons.

```
DECLARE
    v_emp emp%ROWTYPE;
BEGIN
    SELECT * INTO v_emp
    FROM emp
    WHERE emp_id = 1;
    DBMS_OUTPUT.PUT_LINE(v_emp.name || ' - ' || v_emp.salary);
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('Employee not found');
    WHEN TOO_MANY_ROWS THEN
        DBMS_OUTPUT.PUT_LINE('Multiple employees found');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error fetching employee');
END;
```

Rahim Khan - 45000  
Statement processed.  
0.01 seconds

Application Express 2.1.0.00.39  
Language: en-us Copyright © 1999, 2008, Oracle. All rights reserved.

16°C Partly cloudy 9:24 PM 1/4/2026

**Q6.** Write a PL/SQL block to display **train details** using a table-based record.

**Answer:**

DECLARE

```
v_train train%ROWTYPE;  
BEGIN  
    SELECT * INTO v_train  
    FROM train  
    WHERE train_id = 1;  
EXCEPTION  
    WHEN NO_DATA_FOUND THEN  
        DBMS_OUTPUT.PUT_LINE('Train not found');  
    WHEN TOO_MANY_ROWS THEN  
        DBMS_OUTPUT.PUT_LINE('Multiple trains found');  
    WHEN OTHERS THEN  
        DBMS_OUTPUT.PUT_LINE('Error fetching train');  
        DBMS_OUTPUT.PUT_LINE(v_train.train_code || ' Capacity: ' || v_train.capacity);  
END;
```

/

The screenshot shows the Oracle Application Express (APEX) interface. At the top, there is a toolbar with 'Autocommit' checked, a 'Display' dropdown set to 10, and 'Save' and 'Run' buttons. The main area contains a PL/SQL code editor with the following content:

```
BEGIN  
    SELECT * INTO v_train  
    FROM train  
    WHERE train_id = 1;  
EXCEPTION  
    WHEN NO_DATA_FOUND THEN  
        DBMS_OUTPUT.PUT_LINE('Train not found');  
    WHEN TOO_MANY_ROWS THEN  
        DBMS_OUTPUT.PUT_LINE('Multiple trains found');  
    WHEN OTHERS THEN  
        DBMS_OUTPUT.PUT_LINE('Error fetching train');  
        DBMS_OUTPUT.PUT_LINE(v_train.train_code || ' Capacity: ' || v_train.capacity);  
END;  
/
```

Below the code editor is a results panel with tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is selected, showing the output:

```
TR-001 Capacity: 2300  
Statement processed.  
0.00 seconds
```

At the bottom of the results panel, there is footer information: 'Application Express 2.1.0.00.39', 'Language: en-us', 'Copyright © 1999, 2006, Oracle. All rights reserved.', and a system status bar at the very bottom.

**Q7.** Write a PL/SQL program using an **explicit cursor** to list all station names.

**Answer:**

DECLARE

```
CURSOR c_station IS SELECT station_name FROM station;
v_station station.station_name%TYPE;
BEGIN
  OPEN c_station;
  LOOP
    FETCH c_station INTO v_station;
    EXIT WHEN c_station%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE(v_station);
  END LOOP;
  CLOSE c_station;
EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Error processing station cursor');
    IF c_station%ISOPEN THEN
      CLOSE c_station;
    END IF;
END;
```

```
Autocommit Display 10 Save Run
BEGIN
  OPEN c_station;
  LOOP
    FETCH c_station INTO v_station;
    EXIT WHEN c_station%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE(v_station);
  END LOOP;
  CLOSE c_station;
EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Error processing station cursor');
    IF c_station%ISOPEN THEN
      CLOSE c_station;
    END IF;
END;
```

Results Explain Describe Saved SQL History

Uttara North  
Pallabi  
Agargaon  
Motijheel  
Farmgate

Statement processed.

0.00 seconds

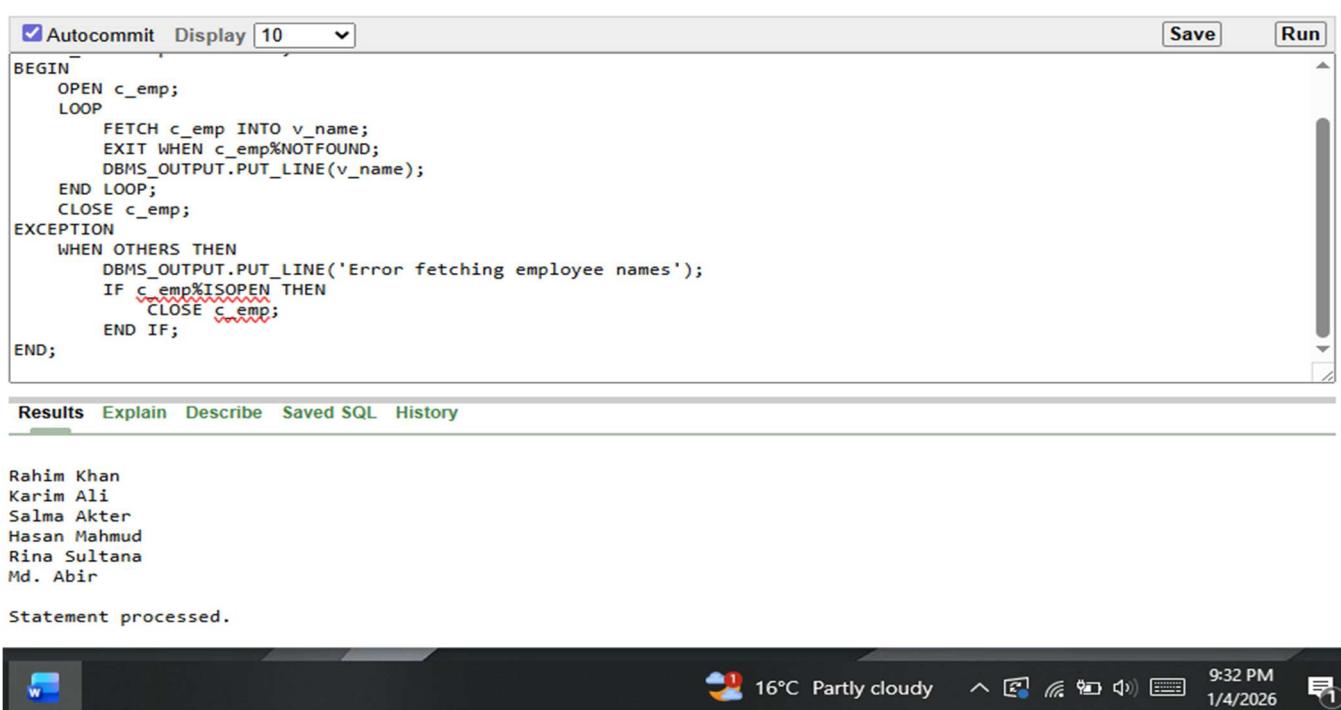
Supergirl official teas... 9:31 PM 1/4/2026

**Q8.** Write a PL/SQL block using an **explicit cursor** to display employee names.

**Answer:**

```
DECLARE
    CURSOR c_emp IS SELECT name FROM emp;
    v_name emp.name%TYPE;
BEGIN
    OPEN c_emp;
    LOOP
        FETCH c_emp INTO v_name;
        EXIT WHEN c_emp%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE(v_name);
    END LOOP;
    CLOSE c_emp;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error fetching employee names');
        IF c_emp%ISOPEN THEN
            CLOSE c_emp;
        END IF;
END;
```

/



The screenshot shows the Oracle SQL Developer interface. The top bar has checkboxes for 'Autocommit' and 'Display 10', and buttons for 'Save' and 'Run'. The main area contains the PL/SQL code from above. Below the code, there's a toolbar with 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The results pane at the bottom displays the output of the executed query, listing employee names: Rahim Khan, Karim Ali, Salma Akter, Hasan Mahmud, Rina Sultana, and Md. Abir. A message 'Statement processed.' is also shown. The bottom right corner shows the system status bar with icons for battery, signal, and time (9:32 PM, 1/4/2026).

```
Autocommit Display 10 Save Run
BEGIN
    OPEN c_emp;
    LOOP
        FETCH c_emp INTO v_name;
        EXIT WHEN c_emp%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE(v_name);
    END LOOP;
    CLOSE c_emp;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error fetching employee names');
        IF c_emp%ISOPEN THEN
            CLOSE c_emp;
        END IF;
END;
```

Rahim Khan  
Karim Ali  
Salma Akter  
Hasan Mahmud  
Rina Sultana  
Md. Abir

Statement processed.

**Q9.** Write a PL/SQL block using a **cursor-based record** to display employee details.

**Answer:**

```
DECLARE
    CURSOR c_emp IS SELECT * FROM emp;
    v_emp c_emp%ROWTYPE;
BEGIN
    OPEN c_emp;
    LOOP
        FETCH c_emp INTO v_emp;
        EXIT WHEN c_emp%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE(v_emp.full_name || ' ' || v_emp.role);
    END LOOP;
    CLOSE c_emp;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error processing employee cursor');
        IF c_emp%ISOPEN THEN
            CLOSE c_emp;
        END IF;
END;
/
```

```
Autocommit Display 10 Save Run
DECLARE
    CURSOR c_emp IS SELECT * FROM emp;
    v_emp c_emp%ROWTYPE;
BEGIN
    OPEN c_emp;
    LOOP
        FETCH c_emp INTO v_emp;
        EXIT WHEN c_emp%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE(v_emp.name || ' ' || v.emp.role);
    END LOOP;
    CLOSE c.emp;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error processing employee cursor');
        IF c.emp%ISOPEN THEN
            CLOSE c.emp;
    END IF;
END;
/
Rahim Khan Driver
Karim Ali Controller
Salma Akter Station Master
Hasan Mahmud Technician
Rina Sultana Ticket Officer
Md. Abir Cleaner
Statement processed.

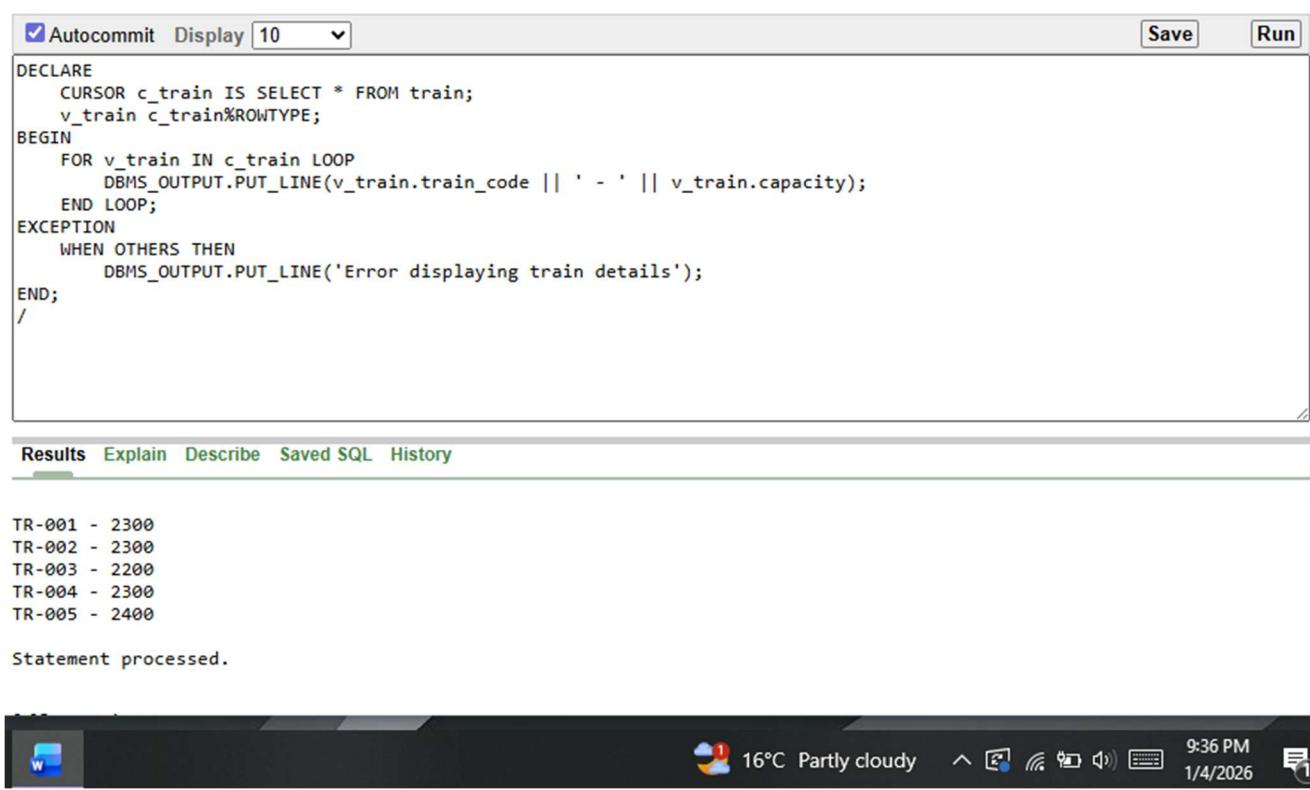
16°C Partly cloudy 9:34 PM 1/4/2026
```

**Q10.** Write a PL/SQL program using a **cursor-based record** to show train information.

**Answer:**

```
DECLARE
    CURSOR c_train IS SELECT * FROM train;
    v_train c_train%ROWTYPE;
BEGIN
    FOR v_train IN c_train LOOP
        DBMS_OUTPUT.PUT_LINE(v_train.train_code || ' - ' || v_train.capacity);
    END LOOP;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error displaying train details');
END;
```

/



The screenshot shows the Oracle SQL Developer interface. At the top, there's a toolbar with 'Autocommit' checked, a 'Display' dropdown set to 10, and buttons for 'Save' and 'Run'. Below the toolbar is the SQL editor window containing the provided PL/SQL code. The code uses a cursor to select all columns from the 'train' table, loops through each row, and prints the train code and capacity separated by a dash. It also includes an exception handling block that prints an error message if any other exception occurs. The code ends with an explicit end statement and a slash. Below the editor, a results tab is selected, showing the output of the executed code. The output consists of five lines of text, each representing a train record: 'TR-001 - 2300', 'TR-002 - 2300', 'TR-003 - 2200', 'TR-004 - 2300', and 'TR-005 - 2400'. At the bottom of the results window, it says 'Statement processed.' The system status bar at the bottom right shows the date as 1/4/2026, the time as 9:36 PM, and various system icons.

```
DECLARE
    CURSOR c_train IS SELECT * FROM train;
    v_train c_train%ROWTYPE;
BEGIN
    FOR v_train IN c_train LOOP
        DBMS_OUTPUT.PUT_LINE(v_train.train_code || ' - ' || v_train.capacity);
    END LOOP;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error displaying train details');
END;
```

Results Explain Describe Saved SQL History

```
TR-001 - 2300
TR-002 - 2300
TR-003 - 2200
TR-004 - 2300
TR-005 - 2400

Statement processed.
```

**Q11.** Create a **row-level trigger** that prevents inserting employees with salary less than 20,000.

**Answer:**

```
CREATE OR REPLACE TRIGGER trg_check_salary
BEFORE INSERT ON Emp
FOR EACH ROW
BEGIN
    IF :NEW.salary < 20000 THEN
        :NEW.salary := 20000;
        DBMS_OUTPUT.PUT_LINE(
            'Salary was below 20000. Automatically set to 20000.'
        );
    END IF;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Salary trigger error');
END;
/
INSERT INTO Emp VALUES( seq_emp_id.NEXTVAL, 'Md. Abir', 'Male', 'Cleaner', 15000, SYSDATE, '08:00 AM', '04:00 PM', 'Active', 1, 1);
```

The screenshot shows the Oracle Application Express SQL Workshop interface. The top bar includes checkboxes for Autocommit and Save, and a dropdown for Display (set to 10). The main area contains the PL/SQL code for the trigger. The bottom section shows the execution results, including the output of the DBMS\_OUTPUT.PUT\_LINE statement and the successful insertion of a new employee record. The status bar at the bottom right indicates the application version (2.1.0.00.39), copyright information (Copyright © 1999, 2006, Oracle. All rights reserved.), and system details like weather (16°C Partly cloudy), time (9:40 PM), date (1/4/2026), and notifications (1).

```
CREATE OR REPLACE TRIGGER trg_check_salary
BEFORE INSERT ON Emp
FOR EACH ROW
BEGIN
    IF :NEW.salary < 20000 THEN
        :NEW.salary := 20000;
        DBMS_OUTPUT.PUT_LINE(
            'Salary was below 20000. Automatically set to 20000.'
        );
    END IF;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Salary trigger error');
END;
/
INSERT INTO Emp VALUES( seq_emp_id.NEXTVAL, 'Abir Saha', 'Male', 'Cleaner', 15000, SYSDATE, '08:00 AM', '04:00 PM', 'Active', 1, 1);

Results Explain Describe Saved SQL History

Salary was below 20000. Automatically set to 20000.
1 row(s) inserted.

0.00 seconds

Application Express 2.1.0.00.39
Language: en-us
Copyright © 1999, 2006, Oracle. All rights reserved.

16°C Partly cloudy 9:40 PM 1/4/2026 1
```

**Q12.** Create a **row-level trigger** to automatically update maintenance status.

**Answer:**

```
CREATE OR REPLACE TRIGGER trg_maintenance_status
```

```
BEFORE INSERT ON Maintenance
```

```
FOR EACH ROW
```

```
BEGIN
```

```
:NEW.status := 'Completed';
```

```
EXCEPTION
```

```
WHEN OTHERS THEN
```

```
    DBMS_OUTPUT.PUT_LINE('Maintenance trigger error');
```

```
END;
```

```
/
```

```
INSERT INTO Maintenance VALUES ( seq_maintenance_id.NEXTVAL,'Routine brake check','Pending',2,2,2);
```

```
select * from maintenance where maintenance_id = 7
```

The screenshot shows the Oracle Application Express interface. The top half of the window displays the SQL code for creating a trigger named 'trg\_maintenance\_status'. The trigger is set to fire before an insert operation on the 'Maintenance' table. It contains logic to update the 'status' column of the inserted row to 'Completed'. If an error occurs during the insert, it will output an error message to the DBMS\_OUTPUT. The code concludes with an 'END;' statement, a slash to indicate the end of the trigger body, and an 'INSERT' statement to add a new record to the 'Maintenance' table with a specific description and status. Below the code, the 'Results' tab is selected, showing the output of the 'select \* from maintenance where maintenance\_id = 7' query. The result set contains one row with maintenance\_id 7, description 'Routine brake check', status 'Completed', and train\_id, station\_id, and emp\_id all set to 2. At the bottom of the results, it says '1 rows returned in 0.00 seconds' and provides a 'CSV Export' link. The bottom right corner of the interface shows the system status bar with icons for battery, signal, and network, along with the time '9:43 PM' and date '1/4/2026'.

MAINTENANCE_ID	DESCRIPTION	STATUS	TRAIN_ID	STATION_ID	EMP_ID
7	Routine brake check	Completed	2	2	2

**Q13.** Create a **statement-level trigger** that displays a message after inserting passengers.

**Answer:**

```
CREATE OR REPLACE TRIGGER trg_passenger_insert
AFTER INSERT ON Passenger
BEGIN
    DBMS_OUTPUT.PUT_LINE('Passenger record inserted successfully');
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error occurred in passenger insert trigger');
END;
/
INSERT INTO Passenger VALUES (seq_passenger_id.NEXTVAL, 'KOuhisk shaaha', 'male', TO_DATE('1998-05-12','YYYY-MM-DD'), '9876543211', '01821111111', 'anika@outlook.com', 'Mirpur', CURRENT_TIMESTAMP);
```

The screenshot shows the Oracle Application Express interface. In the top panel, there are checkboxes for 'Autocommit' and 'Display 10', and buttons for 'Save' and 'Run'. The main area contains the PL/SQL code for the trigger and the insert statement. Below the code, the 'Results' tab is selected, showing the output of the executed code. The output includes the message 'Passenger record inserted successfully', '1 row(s) inserted.', and a timestamp '0.01 seconds'. At the bottom, there is a footer with language information ('Language: en-us'), copyright details ('Copyright © 1999, 2006, Oracle. All rights reserved.'), and system status icons.

```
CREATE OR REPLACE TRIGGER trg_passenger_insert
AFTER INSERT ON Passenger
BEGIN
    DBMS_OUTPUT.PUT_LINE('Passenger record inserted successfully');
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error occurred in passenger insert trigger');
END;
/
INSERT INTO Passenger VALUES (seq_passenger_id.NEXTVAL, 'KOuhisk shaaha', 'male', TO_DATE('1998-05-12','YYYY-MM-DD'), '9876543211', '01821111111', 'anika@outlook.com', 'Mirpur', CURRENT_TIMESTAMP);
```

Results Explain Describe Saved SQL History

Passenger record inserted successfully  
1 row(s) inserted.  
0.01 seconds

Application Express 2.1.0.00.39  
Language: en-us Copyright © 1999, 2006, Oracle. All rights reserved.  
16°C Partly cloudy 9:57 PM 1/4/2026

**Q14.** Create a **statement-level trigger** that prevents deleting all trains.

**Answer:**

```
CREATE OR REPLACE TRIGGER trg_no_train_delete
BEFORE DELETE ON Train
BEGIN
    DBMS_OUTPUT.PUT_LINE('WARNING: Train records are being deleted');
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error occurred in delete trigger');
END;
/
delete from train
```

The screenshot shows the Oracle Application Express interface. The top bar includes checkboxes for 'Autocommit' and 'Display' set to 10, and buttons for 'Save' and 'Run'. The main area contains the SQL code for creating a trigger. The line 'CREATE OR REPLACE TRIGGER trg\_no\_train\_delete' has a red wavy underline, indicating a syntax error. The code is as follows:

```
CREATE OR REPLACE TRIGGER trg_no_train_delete
BEFORE DELETE ON Train
BEGIN
    DBMS_OUTPUT.PUT_LINE('WARNING: Train records are being deleted');
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error occurred in delete trigger');
END;
/
delete from train
```

Below the code, there are tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History', with 'Results' selected. The results pane displays the message 'Trigger created.' and '0.00 seconds'. At the bottom, the status bar shows 'Language: en-us', 'Application Express 2.1.0.00.39', 'Copyright © 1999, 2006, Oracle. All rights reserved.', and system icons for weather (15°C Clear), time (2:10 AM), date (1/3/2026), and notifications.

**Q15.** Create a package that contains a procedure to display all station names.

**Answer (Package Spec & Body):**

```
CREATE OR REPLACE PACKAGE station_pkg IS
    PROCEDURE show_stations;
END;
/
CREATE OR REPLACE PACKAGE BODY station_pkg IS
    PROCEDURE show_stations IS
        v_count NUMBER;
    BEGIN
        SELECT COUNT(*) INTO v_count FROM Station;
        IF v_count = 0 THEN
            DBMS_OUTPUT.PUT_LINE('No stations available');
        ELSE
            FOR r IN (SELECT station_name FROM Station) LOOP
                DBMS_OUTPUT.PUT_LINE(r.station_name);
            END LOOP;
        END IF;
    EXCEPTION
        WHEN OTHERS THEN
            DBMS_OUTPUT.PUT_LINE('Error while fetching stations');
    END show_stations;
END station_pkg;
/BEGIN
station_pkg.show_stations;
END;
/
```

Autocommit Display 10 Save Run

```

CREATE OR REPLACE PACKAGE station_pkg IS
    PROCEDURE show_stations;
END;
/
CREATE OR REPLACE PACKAGE BODY station_pkg IS
    PROCEDURE show_stations IS
        v_count NUMBER;
    BEGIN
        SELECT COUNT(*) INTO v_count FROM Station;
        IF v_count = 0 THEN
            DBMS_OUTPUT.PUT_LINE('No stations available');
        ELSE
            FOR r IN (SELECT station_name FROM Station) LOOP
                DBMS_OUTPUT.PUT_LINE(r.station_name);
            END LOOP;
        END IF;
    END;

```

Results Explain Describe Saved SQL History

```

Uttara North
Pallabi
Agargaon
Motijheel
Farmgate

Statement processed.

```

**Q16.** Create a package that contains a function to return total trains.

**Answer:**

```

CREATE OR REPLACE PACKAGE train_pkg IS
    FUNCTION total_trains RETURN NUMBER;
END train_pkg;
/
CREATE OR REPLACE PACKAGE BODY train_pkg IS
    FUNCTION total_trains RETURN NUMBER IS
        v_count NUMBER;
    BEGIN
        SELECT COUNT(*) INTO v_count FROM Train;
        RETURN v_count;
    EXCEPTION
        WHEN NO_DATA_FOUND THEN
            RETURN 0;
        WHEN OTHERS THEN
            DBMS_OUTPUT.PUT_LINE('Error occurred while counting trains');
            RETURN NULL;
    END;

```

```

END total_trains;
END train_pkg;
/
DECLARE
v_total NUMBER;
BEGIN
v_total := train_pkg.total_trains;
DBMS_OUTPUT.PUT_LINE('Total Trains = ' || v_total);
END;
/

```

Autocommit Display 10 Save Run

```

CREATE OR REPLACE PACKAGE train_pkg IS
    FUNCTION total_trains RETURN NUMBER;
END train_pkg;
/
CREATE OR REPLACE PACKAGE BODY train_pkg IS
    FUNCTION total_trains RETURN NUMBER IS
        v_count NUMBER;
    BEGIN
        SELECT COUNT(*) INTO v_count FROM Train;
        RETURN v_count;
    EXCEPTION
        WHEN NO_DATA_FOUND THEN
            RETURN 0;
        WHEN OTHERS THEN
            DBMS_OUTPUT.PUT_LINE('Error occurred while counting trains');
            RETURN NULL;
    END;

```

Results Explain Describe Saved SQL History

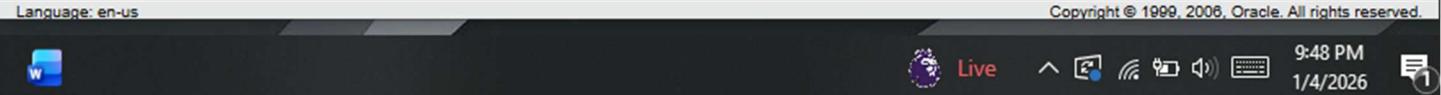
Total Trains = 5

Statement processed.

0.00 seconds

Application Express 2.1.0.00.39

Copyright © 1999, 2006, Oracle. All rights reserved.



## RELATIONAL ALGEBRA

### **Question 1**

Find all passengers whose gender is Male.

$$\prod_{all\_columns}(\sigma_{gender = "Male"}(Passenger))$$

### **Question 2**

Display only the station names from the Station table.

$$\prod_{station\_name}(Station)$$

### **Question 3**

Find the employee name whose employee ID is 'E101'.

$$\prod_{emp\_name}(\sigma_{emp\_id = '101'}(Emp))$$

### **Question 4**

Find the station names that are currently 'Active'.

$$\prod_{Station\_name}(\sigma_{status = "Active"}(Status))$$

### **Question 5**

Find the passenger ID who has booked a ticket with Ticket ID '5001'.

$$\prod_{passenger\_id}(\sigma_{ticket\_id = "5001"}(Ticket))$$

## Bonus

### User Interface Planning And Code

The image shows a user interface for the "Metro Rail System" and its corresponding code in a PHP editor.

**User Interface (Top):**

A modal window titled "Metro Rail System" with a subtitle "Welcome to Metro Rail". It features a "Select Role" dropdown set to "Employee", an "Email or ID" input field with placeholder "Enter your email or ID", a "Password" input field with placeholder "Enter your password", and a blue "Login" button at the bottom.

**Code Editor (Bottom):**

The code editor displays the PHP script for the "login.php" file. The code handles a POST request, checks the selected role (Passenger or Employee), and performs an Oracle database query using OCI API to fetch user information based on the provided email and password. The session variables are then updated with user details.

```
<?php
ob_start();
session_start();
include('config/db.php');

$error = "";

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $role = $_POST['role'];
    $id_email = $_POST['id_email'];
    $pass = $_POST['pass'];

    if (isset($conn) && $conn) {
        if ($role == 'Passenger') {
            $sql = "SELECT * FROM Passenger WHERE (email = :id OR TO_CHAR(passenger_id) = :id) AND nid = :nid";
            $stid = oci_parse($conn, $sql);
            oci_bind_by_name($stid, ":id", $id_email);
            oci_bind_by_name($stid, "nid", $pass);
        } else if ($role == 'Employee') {
            $sql = "SELECT * FROM Emp WHERE TO_CHAR(emp_id) = :id AND TO_CHAR(salary) = :pass";
            $stid = oci_parse($conn, $sql);
            oci_bind_by_name($stid, ":id", $id_email);
            oci_bind_by_name($stid, "pass", $pass);
        }

        if (isset($stid)) {
            $res = oci_execute($stid);
            if ($res) {
                $row = oci_fetch_array($stid, OCI_ASSOC);
                if ($row) {
                    $_SESSION['user_id'] = ($role == 'Passenger') ? $row['PASSENGER_ID'] : $row['EMP_ID'];
                    $_SESSION['role'] = $role;
                    $_SESSION['name'] = $row['NAME'];
                }
            }
        }
    }
}
```

File Edit Selection View ... ← → Q metro\_rail

EXPLORER ⌂ METRO\_RAIL

- assets
- css # style.css
- images
- js main.js
- auth
- logout.php
- register.php
- config
- db.php
- employee
- dashboard.php
- station\_map.php
- view\_employees.php
- view\_tickets.php
- passenger
- book\_ticket.php
- dashboard.php
- get\_fare.php
- travel\_history.php
- index.php
- login.php

login.php ...

```

37
38
39         if ($role == 'Passenger') {
40             header("Location: passenger/dashboard.php");
41         } else {
42             header("Location: employee/dashboard.php");
43         }
44         exit;
45     } else {
46         $error = "Invalid Credentials!";
47     }
48 }
oci_free_statement($stid);

}
>>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Login - Metro Rail Management</title>
    <script src="https://cdn.tailwindcss.com"></script>
</head>
<body class="bg-blue-900 flex items-center justify-center h-screen">
    <div class="bg-white p-10 rounded-[2.5rem] shadow-2xl w-full max-w-sm">
        <h2 class="text-2xl font-black text-center text-gray-800 mb-8 tracking-tighter italic uppercase">Metro Login</h2>
        <form method="POST" class="space-y-5" autocomplete="off">
            <div>
                <label class="block text-[10px] font-black text-gray-400 uppercase tracking-widest mb-1">Role</label>
                <select name="role" id="roleSelect" onchange="updateUI()" class="w-full border-b-2 py-2 outline-none cursor-pointer font-black">
                    <option value="" disabled selected>Enter your Role</option>
                    <option value="Passenger">Passenger</option>
                    <option value="Employee">Employee</option>
                </select>
            </div>
            <div>
                <input type="text" name="id_email" id="idInput" placeholder="Email or ID" class="w-full border-b-2 py-2 outline-none" required value="" autocomplete="off">
            </div>
            <div>
                <input type="password" name="pass" id="passInput" placeholder="Password" class="w-full border-b-2 py-2 outline-none" required value="" autocomplete="new-password">
            </div>
            <?php if ($error) echo "<p class='text-red-500 text-xs text-center font-bold uppercase'>$error</p>"; ?>
            <button type="submit" class="w-full bg-blue-800 text-white font-black py-4 rounded-2xl hover:bg-blue-700 transition duration-200">Login</button>
        </form>
        <div id="regContainer" class="mt-8 text-center text-xs">
            <span class="text-gray-400 font-medium">New traveler?</span>
            <a href="auth/register.php" class="text-blue-800 font-black ml-1 hover:underline uppercase">Register Now</a>
        </div>
    </div>
</body>
</html>
```

Ln 16, Col 31 Spaces: 4 UTF-8 CRLF () PHP

Search 10:49 AM 1/4/2026

File Edit Selection View ... ← → Q metro\_rail

EXPLORER ⌂ METRO\_RAIL

- assets
- css # style.css
- images
- js main.js
- auth
- logout.php
- register.php
- config
- db.php
- employee
- dashboard.php
- station\_map.php
- view\_employees.php
- view\_tickets.php
- passenger
- book\_ticket.php
- dashboard.php
- get\_fare.php
- travel\_history.php
- index.php
- login.php

login.php ...

```

56 <html lang="en">
57 <body class="bg-blue-900 flex items-center justify-center h-screen">
58     <div class="bg-white p-10 rounded-[2.5rem] shadow-2xl w-full max-w-sm">
59         <form method="POST" class="space-y-5" autocomplete="off">
60             <div>
61                 <option value="Employee">Employee</option>
62             </select>
63         </div>
64         <div>
65             <input type="text" name="id_email" id="idInput" placeholder="Email or ID" class="w-full border-b-2 py-2 outline-none" required value="" autocomplete="off">
66         </div>
67         <div>
68             <input type="password" name="pass" id="passInput" placeholder="Password" class="w-full border-b-2 py-2 outline-none" required value="" autocomplete="new-password">
69         </div>
70         <?php if ($error) echo "<p class='text-red-500 text-xs text-center font-bold uppercase'>$error</p>"; ?>
71         <button type="submit" class="w-full bg-blue-800 text-white font-black py-4 rounded-2xl hover:bg-blue-700 transition duration-200">Login</button>
72     </form>
73     <div id="regContainer" class="mt-8 text-center text-xs">
74         <span class="text-gray-400 font-medium">New traveler?</span>
75         <a href="auth/register.php" class="text-blue-800 font-black ml-1 hover:underline uppercase">Register Now</a>
76     </div>
77 </div>
78 </body>
79 </html>
```

Ln 16, Col 31 Spaces: 4 UTF-8 CRLF () PHP

Search 10:50 AM 1/4/2026

```

File Edit Selection View ... ← → Q metro_rail
EXPLORER employees.php book_ticket.php get_fare.php JS main.js logout.php travel_history.php dashboard.php employee login.php
METRO_RAIL assets # style.css images js main.js auth register.php config db.php employee dashboard.php station_map.php view_employees.php view_tickets.php passenger book_ticket.php dashboard.php get_fare.php travel_history.php index.php login.php
login.php > ...
<html lang="en">
  <body class="bg-blue-900 flex items-center justify-center h-screen">
    <script>
      function updateUI() {
        const role = document.getElementById('roleSelect').value;
        const regContainer = document.getElementById('regContainer');
        const idInput = document.getElementById('idInput');
        const passInput = document.getElementById('passInput');

        idInput.value = "";
        passInput.value = "";

        if (role === 'Employee') {
          regContainer.style.display = 'none';
          idInput.placeholder = "Employee Email/ID";
          passInput.placeholder = "Salary (Pass)";
        } else {
          regContainer.style.display = 'block';
          idInput.placeholder = "Email or ID";
          passInput.placeholder = "NID (Password)";
        }
      }

      window.onload = function() {
        document.getElementById('idInput').value = "";
        document.getElementById('passInput').value = "";
      };
    </script>
  </body>
</html>

```

Ln 16, Col 31 Spaces: 4 UTF-8 CRLF PHP 10:51 AM 1/4/2026

## Employee Portal

Employee ID: E1 Logout

Total Tickets **2**

Passengers **2**

Stations **16**

Revenue **₹100**

### Employee Profile

Name **John Smith**  
Role **Station Manager**  
Shift **Morning**

### Station Network

Uttara North	Uttara Center	Uttara South
Pallabi	Mirpur 11	Mirpur 10
Kazipara	Shewrapara	Agargaon
Bijoy Sarani	Farmgate	Karwan Bazar
Shahbagh	Dhaka University	Bangladesh Secretariat
Motijheel		

The screenshot shows a code editor interface with the following details:

- File Explorer:** On the left, it lists files and folders under the "METRO\_RAIL" project. The "dashboard.php" file is currently selected.
- Code Editor:** The main area displays the PHP code for "dashboard.php". The code includes session handling, database queries using OCI, and variable assignments for ticket and passenger counts.
- Search Bar:** At the top center, it says "metro\_rail".
- Bottom Bar:** It includes icons for search, refresh, and navigation, along with status information: "Ln 36, Col 1", "Spaces: 4", "UTF-8", "CRLF", "PHP", and "8".

```
<?php
session_start();
include('../config/db.php');

if (!isset($_SESSION['user_id']) || $_SESSION['role'] != 'Employee') {
    header("Location: ../login.php");
    exit();
}

$emp_id = $_SESSION['user_id'];

$q_tickets = oci_parse($conn, "SELECT COUNT(*) AS C FROM Ticket");
oci_execute($q_tickets);
$total_tickets = oci_fetch_array($q_tickets, OCI_ASSOC)[['C']];

$q_passengers = oci_parse($conn, "SELECT COUNT(*) AS C FROM Passenger");
oci_execute($q_passengers);
$total_passengers = oci_fetch_array($q_passengers, OCI_ASSOC)[['C']];

$q_rev = oci_parse($conn, "SELECT SUM(amount) AS S FROM Payment WHERE status = 'Completed'");
oci_execute($q_rev);
$revenue = oci_fetch_array($q_rev, OCI_ASSOC)[['S']] ?? 0;

$q_stations_count = oci_parse($conn, "SELECT COUNT(*) AS C FROM Station");
oci_execute($q_stations_count);
$station_count = oci_fetch_array($q_stations_count, OCI_ASSOC)[['C']];

$q_profile = oci_parse($conn, "SELECT name, role, shift_start, shift_end FROM Emp WHERE emp_id = :id");
oci_bind_by_name($q_profile, ":id", $emp_id);
oci_execute($q_profile);
$emp_profile = oci_fetch_array($q_profile, OCI_ASSOC);
```

The screenshot shows a code editor interface with the following details:

- File Menu:** File, Edit, Selection, View, ...
- Search Bar:** Q metro\_rail
- Explorer:** Shows project structure under METRO\_RAIL:
  - assets
  - css (# style.css)
  - images
  - js
    - main.js
  - auth
    - logout.php
    - register.php
  - config
    - db.php
  - employee
    - dashboard.php
    - station\_map.php
    - view\_employees.php
    - view\_tickets.php
  - passenger
    - book\_ticket.php
    - dashboard.php
    - get\_fare.php
    - travel\_history.php
  - index.php
  - login.php
- Code Editor:** Displays dashboard.php content.

```
37 $q_st_list = oci_parse($conn, "SELECT station_name FROM Station ORDER BY station_id ASC");
38 oci_execute($q_st_list);
39 ?>
40 <!DOCTYPE html>
41 <html lang="en">
42   <head>
43     <meta charset="UTF-8">
44     <title>Employee Dashboard - Metro Rail</title>
45     <script src="https://cdn.tailwindcss.com"></script>
46     <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css">
47   </head>
48 <body class="bg-gray-100 font-sans flex overflow-hidden">
49
50   <aside class="w-72 bg-slate-900 h-screen text-white flex flex-col shadow-2xl">
51     <div class="p-8 border-b border-slate-800">
52       <h1 class="text-[10px] font-black italic tracking-tighter text-blue-400">METRO<span class="text-white">RAIL</span></h1>
53       <p class="text-[10px] text-slate-500 uppercase font-bold tracking-widest mt-1">Employee Management System</p>
54     </div>
55
56     <nav class="flex-1 p-6 space-y-2">
57       <a href="dashboard.php" class="flex items-center space-x-4 p-4 rounded-xl bg-blue-600 text-white transition shadow-lg shadow-slate-800 py-2 px-3 w-full text-decoration-none text-slate-100 hover:bg-blue-700 hover:text-slate-100">
58         <i class="fas fa-th-large w-5"></i>
59         <span class="font-bold">Dashboard</span>
60       </a>
61
62       <a href="view_employees.php" class="flex items-center space-x-4 p-4 rounded-xl hover:bg-slate-800 text-slate-400 hover:text-white py-2 px-3 w-full text-decoration-none text-slate-100">
63         <i class="fas fa-users w-5"></i>
64         <span class="font-bold">View Employees</span>
65       </a>
66
67       <a href="view_tickets.php" class="flex items-center space-x-4 p-4 rounded-xl hover:bg-slate-800 text-slate-400 hover:text-white py-2 px-3 w-full text-decoration-none text-slate-100">
68         <i class="fas fa-ticket-alt w-5"></i>
69         <span class="font-bold">View Sold Tickets</span>
70       </a>
71
72     </nav>
73   </aside>
74   <div class="flex-grow flex flex-col justify-between">
75     <div>
76       <h2>Dashboard</h2>
77       <p>Welcome to the Employee Management System!</p>
78       <table border="1">
79         <thead>
80           <tr>
81             <th>Station Name</th>
82             <th>Address</th>
83             <th>Phone Number</th>
84           </tr>
85         </thead>
86         <tbody>
87           <tr>
88             <td>Station A</td>
89             <td>123 Main St</td>
90             <td>(555) 123-4567</td>
91           </tr>
92           <tr>
93             <td>Station B</td>
94             <td>456 Elm St</td>
95             <td>(555) 234-5678</td>
96           </tr>
97           <tr>
98             <td>Station C</td>
99             <td>789 Oak St</td>
100            <td>(555) 345-6789</td>
101          </tr>
102        </tbody>
103      </table>
104    </div>
105    <div>
106      <h2>Employee Management</h2>
107      <p>Manage employees and their details here.</p>
108      <table border="1">
109        <thead>
110          <tr>
111            <th>Employee ID</th>
112            <th>Name</th>
113            <th>Role</th>
114            <th>Actions</th>
115          </tr>
116        </thead>
117        <tbody>
118          <tr>
119            <td>E001</td>
120            <td>John Doe</td>
121            <td>Manager</td>
122            <td><a href="#">Edit</a> <a href="#">Delete</a></td>
123          </tr>
124          <tr>
125            <td>E002</td>
126            <td>Jane Smith</td>
127            <td>Supervisor</td>
128            <td><a href="#">Edit</a> <a href="#">Delete</a></td>
129          </tr>
130          <tr>
131            <td>E003</td>
132            <td>Mike Johnson</td>
133            <td>Employee</td>
134            <td><a href="#">Edit</a> <a href="#">Delete</a></td>
135          </tr>
136        </tbody>
137      </table>
138    </div>
139  </div>
140</body>
```
- Bottom Bar:** Search, Ln 36, Col 1, Spaces: 4, UTF-8, CRLF, PHP, 10:52 AM, 1/4/2026

The screenshot shows a code editor interface with the following details:

- File Explorer:** On the left, it lists files under the "METRO\_RAIL" project, including "employees.php", "book\_ticket.php", "get\_fare.php", "main.js", "logout.php", "travel\_history.php", "dashboard.php", "employee.php", "login.php", "index.php", and "db.php".
- Code Editor:** The main area displays the content of the "main.js" file. The code includes HTML, CSS, and JavaScript. It features a navigation bar with links for "Account" and "Logout", and a main content area displaying a welcome message and a "System Online" status indicator.
- Search Bar:** At the top center, it says "metro\_rail".
- Status Bar:** At the bottom, it shows "Ln 36, Col 1" and other standard status bar information.
- Icons:** A toolbar at the bottom includes icons for search, file operations, and other common functions.

The screenshot shows a code editor interface with the following details:

- File Menu:** File, Edit, Selection, View, ...
- Search Bar:** Q metro\_rail
- Explorer:** Shows a tree view of files and folders:
  - METRO\_RAIL folder contains:
    - assets
    - css (# style.css)
    - images
    - js (main.js)
  - auth folder contains:
    - logout.php
    - register.php
  - config folder contains:
    - db.php
  - employee folder contains:
    - dashboard.php (selected)
    - station\_map.php
    - view\_employees.php
    - view\_tickets.php
  - passenger folder contains:
    - book\_ticket.php
    - dashboard.php
    - get\_fare.php
    - travel\_history.php
  - index.php
  - login.php
- Code Editor:** The dashboard.php file is open, displaying PHP and CSS code. The code includes sections for ticket sales, total passengers, station count, and total revenue, each with a corresponding database query (e.g., \$total\_tickets, \$total\_passengers, \$station\_count, \$revenue). It also includes a "My Profile" section with a placeholder for the user's name (\$emp\_profile['NAME']).
- Bottom Bar:** Search, Outline, Timeline, and various system icons.
- Status Bar:** Ln 36, Col 1, Spaces: 4, UTF-8, CRLF, PHP, 10:53 AM, 1/4/2026

The screenshot shows a code editor interface with the following details:

- File Explorer:** Shows a tree view of files and folders. The `dashboard.php` file is currently selected.
- Code Editor:** Displays the contents of `dashboard.php`. The code includes:
  - User profile information (name, role) displayed in a grid format.
  - Shift details (start and end times) displayed in a grid format.
  - Station statistics (Active Stations) displayed in a grid format.
- Status Bar:** Shows the current line (Ln 36, Col 1), spaces used (Spaces: 4), and file encoding (UTF-8 CRLF).
- Bottom Icons:** Includes icons for search, refresh, file operations, and system status.

[← Passenger Registration](#)

Full Name

Gender

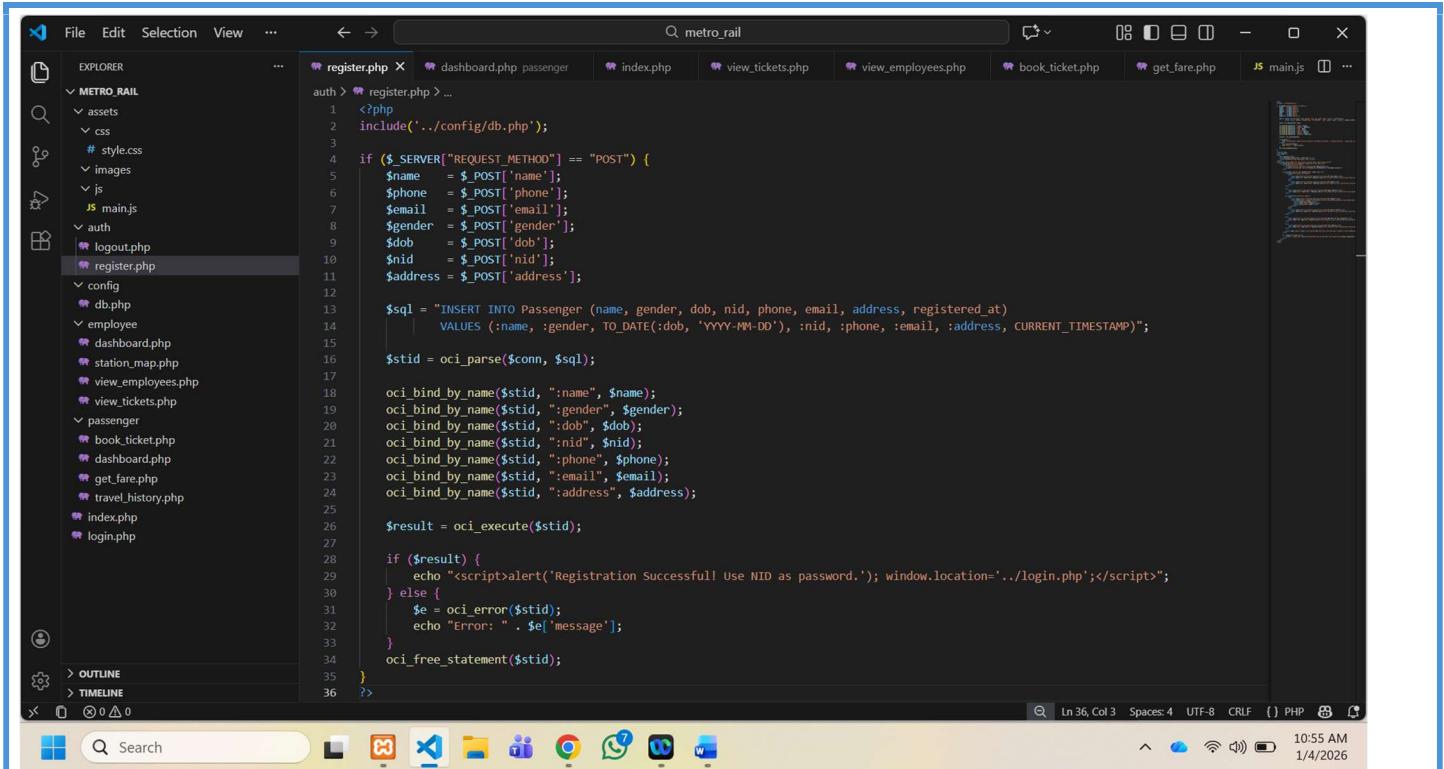
Date of Birth

National ID   
Your NID will be used as your password for login

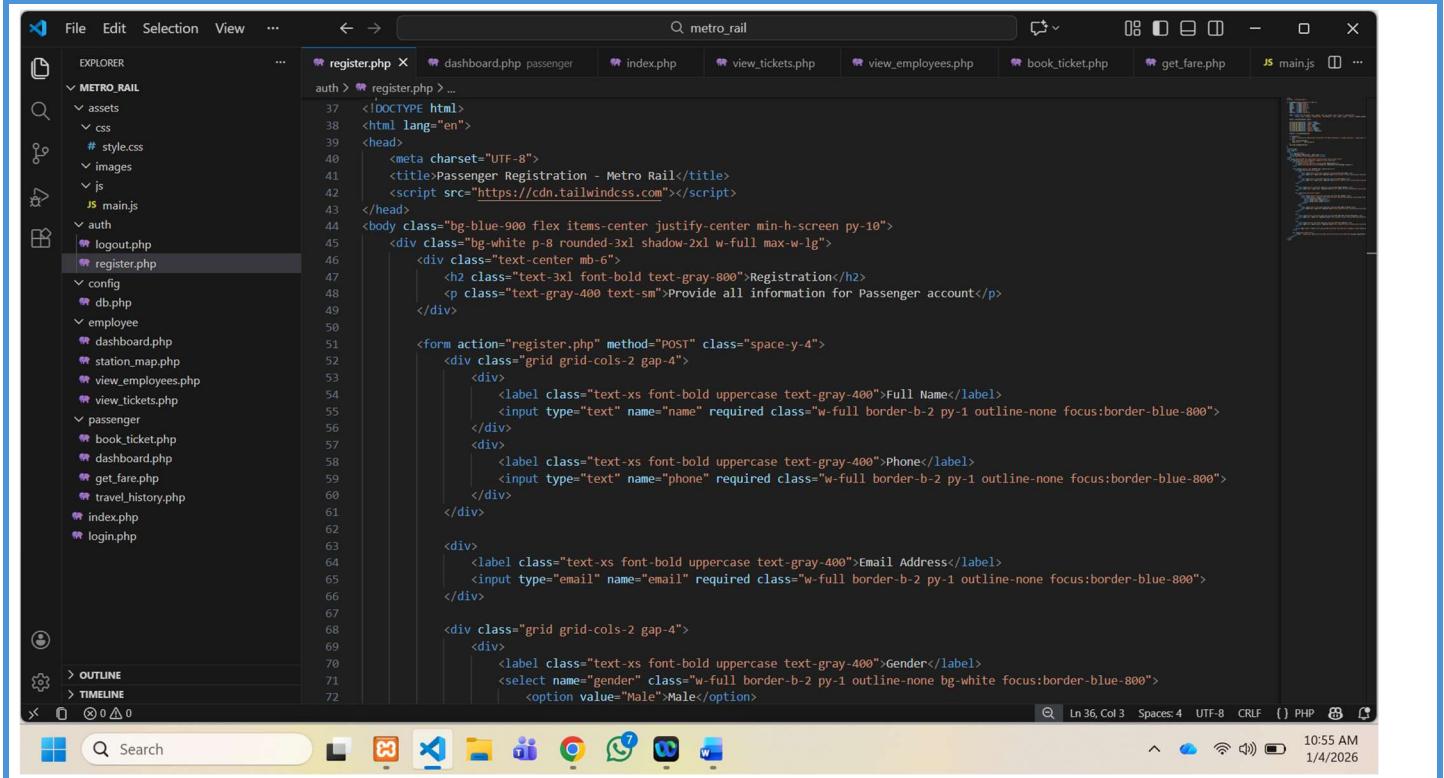
Phone Number

Email Address

Address



```
File Edit Selection View ... register.php dashboard.php passenger index.php view_tickets.php view_employees.php book_ticket.php get_fare.php main.js ...  
EXPLORER METRO_RAIL  
assets css style.css images js main.js auth logout.php register.php config db.php employee dashboard.php station_map.php view_employees.php view_tickets.php passenger book_ticket.php dashboard.php get_fare.php travel_history.php index.php login.php  
auth > register.php > ...  
auth > register.php > ...  
1 <?php  
2 include('../config/db.php');  
3  
4 if ($_SERVER["REQUEST_METHOD"] == "POST") {  
5     $name = $_POST['name'];  
6     $phone = $_POST['phone'];  
7     $email = $_POST['email'];  
8     $gender = $_POST['gender'];  
9     $dob = $_POST['dob'];  
10    $nid = $_POST['nid'];  
11    $address = $_POST['address'];  
12  
13    $sql = "INSERT INTO Passenger (name, gender, dob, nid, phone, email, address, registered_at)  
14        VALUES (:name, :gender, TO_DATE(:dob, 'YYYY-MM-DD'), :nid, :phone, :email, :address, CURRENT_TIMESTAMP)";  
15  
16    $stid = oci_parse($conn, $sql);  
17  
18    oci_bind_by_name($stid, ":name", $name);  
19    oci_bind_by_name($stid, ":gender", $gender);  
20    oci_bind_by_name($stid, ":dob", $dob);  
21    oci_bind_by_name($stid, ":nid", $nid);  
22    oci_bind_by_name($stid, ":phone", $phone);  
23    oci_bind_by_name($stid, ":email", $email);  
24    oci_bind_by_name($stid, ":address", $address);  
25  
26    $result = oci_execute($stid);  
27  
28    if ($result) {  
29        echo "<script>alert('Registration Successful! Use NID as password.');" .<script>";  
30    } else {  
31        $e = oci_error($stid);  
32        echo "Error: " . $e['message'];  
33    }  
34    oci_free_statement($stid);  
35 }  
?>  
Ln 36, Col 3 Spaces: 4 UTF-8 CRLF PHP 10:55 AM 1/4/2026
```



```
File Edit Selection View ... register.php dashboard.php passenger index.php view_tickets.php view_employees.php book_ticket.php get_fare.php main.js ...  
EXPLORER METRO_RAIL  
assets css style.css images js main.js auth logout.php register.php config db.php employee dashboard.php station_map.php view_employees.php view_tickets.php passenger book_ticket.php dashboard.php get_fare.php travel_history.php index.php login.php  
auth > register.php > ...  
auth > register.php > ...  
37 <!DOCTYPE html>  
38 <html lang="en">  
39 <head>  
40     <meta charset="UTF-8">  
41     <title>Passenger Registration - Metro Rail</title>  
42     <script src="https://cdn.tailwindcss.com"></script>  
43 </head>  
44 <body class="bg-blue-900 flex items-center justify-center min-h-screen py-10">  
45     <div class="bg-white p-8 rounded-3xl shadow-2xl w-full max-w-lg">  
46         <div class="text-center mb-6">  
47             <h2 class="text-3xl font-bold text-gray-800">Registration</h2>  
48             <p class="text-gray-400 text-sm">Provide all information for Passenger account</p>  
49         </div>  
50  
51         <form action="register.php" method="POST" class="space-y-4">  
52             <div class="grid grid-cols-2 gap-4">  
53                 <div>  
54                     <label class="text-xs font-bold uppercase text-gray-400">Full Name</label>  
55                     <input type="text" name="name" required class="w-full border-b-2 py-1 outline-none focus:border-blue-800">  
56                 </div>  
57                 <div>  
58                     <label class="text-xs font-bold uppercase text-gray-400">Phone</label>  
59                     <input type="text" name="phone" required class="w-full border-b-2 py-1 outline-none focus:border-blue-800">  
60                 </div>  
61             </div>  
62  
63             <div>  
64                 <label class="text-xs font-bold uppercase text-gray-400">Email Address</label>  
65                 <input type="email" name="email" required class="w-full border-b-2 py-1 outline-none focus:border-blue-800">  
66             </div>  
67  
68             <div class="grid grid-cols-2 gap-4">  
69                 <div>  
70                     <label class="text-xs font-bold uppercase text-gray-400">Gender</label>  
71                     <select name="gender" class="w-full border-b-2 py-1 outline-none bg-white focus:border-blue-800">  
72                         <option value="Male">Male</option>
```

The screenshot shows a code editor interface with the following details:

- File Menu:** File, Edit, Selection, View, ...
- Search Bar:** metro\_rail
- Left Sidebar (EXPLORER):** METRO\_RAIL folder containing assets (css, images), js (main.js), auth (register.php, logout.php), config (db.php), employee (dashboard.php, station\_map.php, view\_employees.php, view\_tickets.php), passenger (book\_ticket.php, dashboard.php, get\_fare.php, travel\_history.php), index.php, login.php.
- Right Sidebar:** Shows a preview of the code and a detailed code viewer.
- Code Editor:** Displays the contents of register.php. The code is a PHP file with HTML and CSS inline. It includes fields for gender (Female, Other), Date of Birth, NID (Password), Full Address, and a "Create Account" button. It also links to login.php for existing users.
- Bottom Status Bar:** Ln 36, Col 3, Spaces: 4, UTF-8, CRLF, PHP, 10:56 AM, 1/4/2026.

```
<?php require 'header.php'>
<?php if (!isset($_SESSION['user'])) { <?php include 'auth/register.php'> } else { header('Location: dashboard.php'); } ?>
<?php require 'footer.php'>
```

```

<html lang="en">
  <body class="bg-blue-900 flex items-center justify-center min-h-screen py-10">
    <div class="bg-white p-8 rounded-3xl shadow-2xl w-full max-w-lg">
      <form action="register.php" method="POST" class="space-y-4">
        <div class="grid grid-cols-2 gap-4">
          <div>
            <label class="text-xs font-bold uppercase text-gray-400">Gender:</label>
            <select required class="w-full border-b-2 py-1 outline-none focus:border-blue-800">
              <option value="Female">Female</option>
              <option value="Other">Other</option>
            </select>
          </div>
          <div>
            <label class="text-xs font-bold uppercase text-gray-400">Date of Birth:</label>
            <input type="date" name="dob" required class="w-full border-b-2 py-1 outline-none focus:border-blue-800">
          </div>
        </div>
        <div>
          <label class="text-xs font-bold uppercase text-gray-400">NID (Will be your Password)</label>
          <input type="text" name="nid" required class="w-full border-b-2 py-1 outline-none focus:border-blue-800">
        </div>
        <div>
          <label class="text-xs font-bold uppercase text-gray-400">Full Address:</label>
          <input type="text" name="address" required class="w-full border-b-2 py-1 outline-none focus:border-blue-800">
        </div>
        <button type="submit" class="w-full bg-blue-800 text-white font-bold py-3 rounded-xl hover:shadow-lg transition mt-6">Create Account</button>
      </form>
      <div class="text-center mt-6">
        <a href="../login.php" class="text-blue-800 text-sm font-bold hover:underline">Already registered? Login here</a>
      </div>
    </div>
  </body>
</html>
```

## Book Your Ticket

Select Current Station

Select Destination Station

[Buy Ticket](#)

## My Tickets

<b>Ticket #TKT1765694752715</b> Uttara North → Shahbagh 14/12/2025	<b>₹50</b> <span style="border: 1px solid green; padding: 2px 5px;">Active</span>
<b>Ticket #TKT1765696451973</b> Uttara North → Shahbagh 14/12/2025	<b>₹50</b> <span style="border: 1px solid green; padding: 2px 5px;">Active</span>

## Fare Information

Fixed Fare

**₹50**

per journey

- All Stations Covered  
Travel to any station
- Express Service  
Fast and reliable
- Digital Tickets  
No physical tickets needed

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface with a dark theme. The left sidebar contains a tree view of project files under 'METRO\_RAIL' (register.php, dashboard.php, passenger, index.php, view\_tickets.php, view\_employees.php, book\_ticket.php, get\_fare.php, main.js). The 'dashboard.php' file is currently selected and open in the main editor area. The code is a PHP script that starts a session, checks if a user is logged in, and then queries an Oracle database to get station names and IDs. It then generates an HTML response with Tailwind CSS and jQuery. The status bar at the bottom shows 'Ln 188, Col 21' and other standard status indicators.

```
<?php
session_start();
include('../config/db.php');

if (!isset($_SESSION['user_id'])) {
    header("location: ../login.php");
    exit();
}

$user_name = $_SESSION['name'];

$station_sql = "SELECT station_id, station_name FROM Station ORDER BY station_id ASC";
$stmt = oci_parse($conn, $station_sql);
oci_execute($stmt);

$stations = [];
while ($row = oci_fetch_array($stmt, OCI_ASSOC)) {
    $stations[] = ['ID' => $row['STATION_ID'], 'NAME' => $row['STATION_NAME']];
}
?>

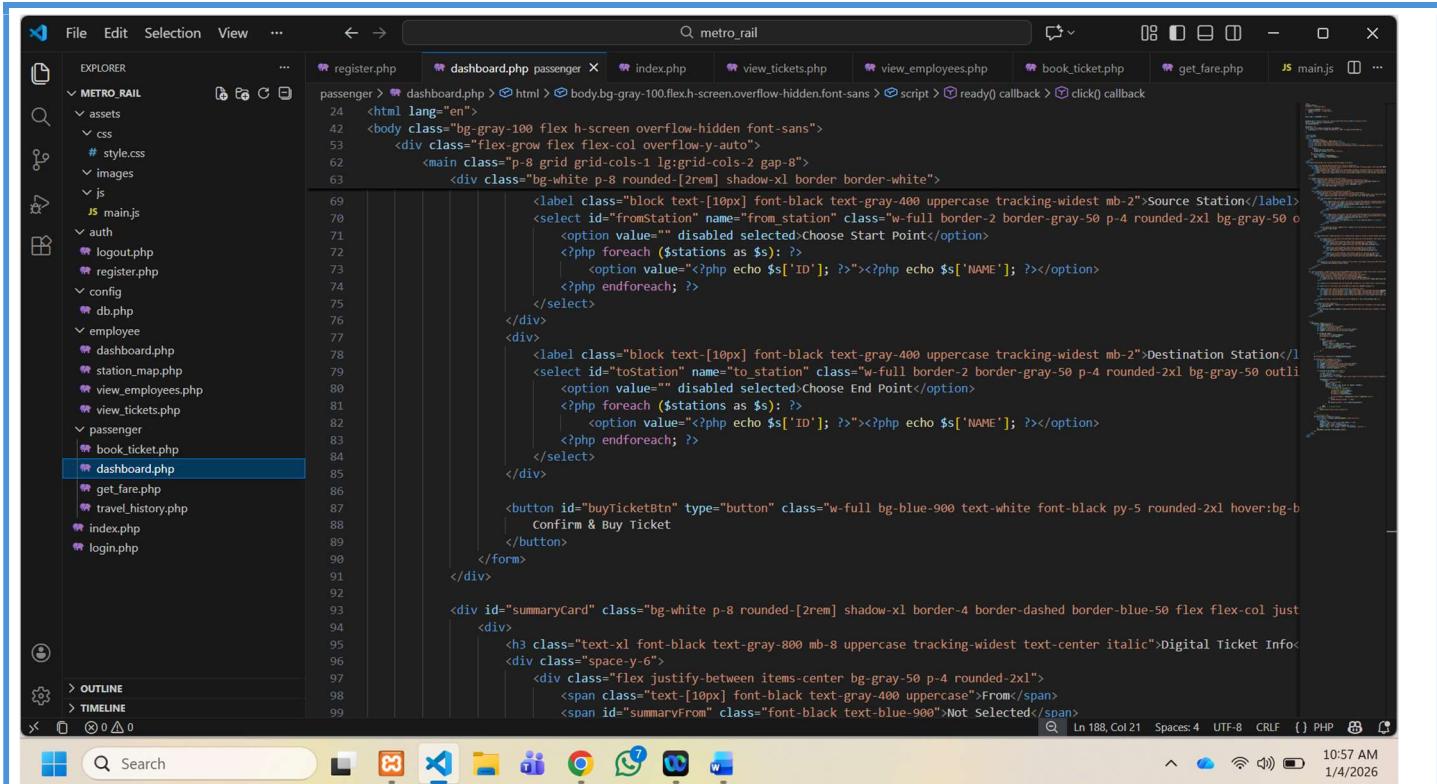
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Passenger Dashboard - Metro Rail</title>
    <script src="https://cdn.tailwindcss.com"></script>
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/html2pdf.js/0.10.1/html2pdf.bundle.min.js"></script>
    <style>
        .loader {
            border-top-color: #1e3a8a;
            animation: spinner 1.5s linear infinite;
        }
        @keyframes spinner {
```

The screenshot shows a code editor interface with a dark theme. On the left is a sidebar with a tree view of project files under 'METRO\_RAIL'. The main area displays the contents of 'dashboard.php'. The code is a PHP script with some CSS inline. It includes a CSS keyframe for a spinner, navigation links for Book Ticket, Travel History, and Logout, and a welcome message with a user name placeholder. The code editor has tabs for 'register.php', 'index.php', 'view\_tickets.php', 'view\_employees.php', 'book\_ticket.php', 'get\_fare.php', and 'JS main.js'. The status bar at the bottom shows 'Ln 188, Col 21' and '10:57 AM 1/4/2026'.

```
register.php dashboard.php passenger index.php view_tickets.php view_employees.php book_ticket.php get_fare.php JS main.js

register.php dashboard.php passenger index.php view_tickets.php view_employees.php book_ticket.php get_fare.php JS main.js

24 <html lang="en">
25   <head>
26     <style>
27       @keyframes spinner {
28         0% { transform: rotate(0deg); }
29         100% { transform: rotate(360deg); }
30       }
31     </style>
32   </head>
33   <body class="bg-gray-100 flex h-screen overflow-hidden font-sans">
34
35     <aside class="w-64 bg-blue-900 text-white flex flex-col shadow-2xl">
36       <div class="p-6 text-2xl font-black italic border-b border-blue-800 tracking-tighter text-blue-300">METRO<span class="text-white font-black italic border-b border-blue-800 tracking-tighter text-blue-300">Book Ticket</span></div>
37       <nav class="flex-grow p-4 space-y-2 mt-4">
38         <a href="dashboard.php" class="block p-3 bg-blue-800 rounded-xl font-bold border-l-4 border-white shadow-lg">Book Ticket</a>
39         <a href="travel_history.php" class="block p-3 hover:bg-blue-800 rounded-xl opacity-70 transition">Travel History</a>
40         <a href="../login.php" class="block p-3 hover:bg-red-800 rounded-xl mt-10 text-red-200 transition">Logout</a>
41       </nav>
42     </aside>
43
44     <div class="flex-grow flex flex-col overflow-y-auto">
45       <header class="bg-white shadow-sm p-6 flex justify-between items-center px-10">
46         <h2 class="text-2xl font-black text-slate-800 uppercase tracking-tight">Passenger Portal</h2>
47         <div class="font-black text-blue-900 italic flex items-center space-x-2">
48           <span class="text-xs text-gray-400 not-italic">WELCOME,</span>
49           <span><?php echo $user_name; ?></span>
50         </div>
51       </header>
52
53       <main class="p-8 grid grid-cols-1 lg:grid-cols-2 gap-8">
54         <div class="bg-white p-8 rounded-[2rem] shadow-xl border border-white">
55           <h3 class="text-2xl font-black text-blue-900 mb-8 uppercase tracking-widest flex items-center gap-2">
56             <span class="bg-blue-900 text-white p-1 rounded">Plan Your Journey</span>
57           </h3>
58           <form id="bookingForm" class="space-y-6">
59             <div>
```

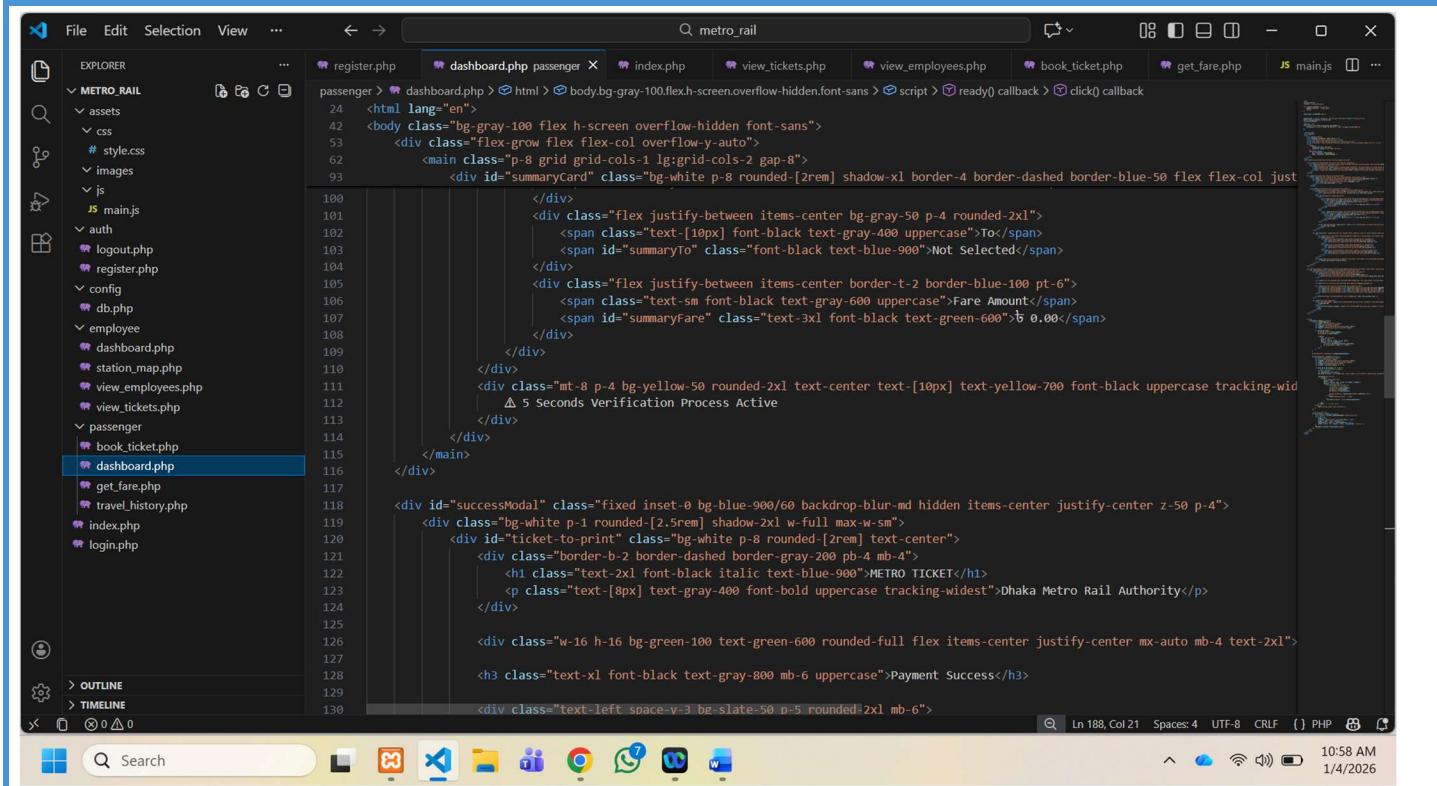


```
File Edit Selection View ... < > Q metro_rail
EXPLORER register.php dashboard.php passenger X index.php view_tickets.php view_employees.php book_ticket.php get_fare.php main.js ...
METRO_RAIL assets css # style.css images js main.js auth logout.php register.php config db.php employee dashboard.php station_map.php view_employees.php view_tickets.php passenger book_ticket.php dashboard.php get_fare.php travel_history.php index.php login.php
passenger > dashboard.php > html > body.bg-gray-100.flex.h-screen.overflow-hidden.font-sans > script > ready() callback > click() callback
24 <html lang="en">
42 <body class="bg-gray-100 flex h-screen overflow-hidden font-sans">
53   <div class="flex-grow flex flex-col overflow-y-auto">
62     <main class="p-8 grid grid-cols-1 lg:grid-cols-2 gap-8">
63       <div class="bg-white p-8 rounded-[2rem] shadow-xl border border-white">
69         <label class="block text-[10px] font-black text-gray-400 uppercase tracking-widest mb-2">Source Station</label>
70         <select id="fromstation" name="from station" class="w-full border-2 border-gray-50 p-4 rounded-2xl bg-gray-50 o
71           <option value="" disabled selected>Choose Start Point</option>
72           <?php foreach ($stations as $s); ?>
73             | <option value="<?php echo $s['ID']; ?>"><?php echo $s['NAME']; ?></option>
74           <?php endforeach; ?>
75         </select>
76       </div>
77       <div>
78         <label class="block text-[10px] font-black text-gray-400 uppercase tracking-widest mb-2">Destination Station</label>
79         <select id="tostation" name="to_station" class="w-full border-2 border-gray-50 p-4 rounded-2xl bg-gray-50 outline
80           <option value="" disabled selected>Choose End Point</option>
81           <?php foreach ($stations as $s); ?>
82             | <option value="<?php echo $s['ID']; ?>"><?php echo $s['NAME']; ?></option>
83           <?php endforeach; ?>
84         </select>
85       </div>
86       <button id="buyTicketBtn" type="button" class="w-full bg-blue-900 text-white font-black py-5 rounded-2xl hover:bg-b
87         Confirm & Buy Ticket
88       </button>
89     </form>
90   </div>
91
92
93   <div id="summaryCard" class="bg-white p-8 rounded-[2rem] shadow-xl border-4 border-dashed border-blue-50 flex flex-col just
94     <div>
95       <h3 class="text-xl font-black text-gray-800 mb-8 uppercase tracking-widest text-center italic">Digital Ticket Info</h3>
96       <div class="space-y-6">
97         <div class="flex justify-between items-center bg-gray-50 p-4 rounded-2xl">
98           <span class="text-[10px] font-black text-gray-400 uppercase">From:</span>
99           <span id="summaryFrom" class="font-black text-blue-900">Not Selected</span>
100      </div>
101      <div class="flex justify-between items-center bg-gray-50 p-4 rounded-2xl">
102        <span class="text-[10px] font-black text-gray-400 uppercase">To:</span>
103        <span id="summaryTo" class="font-black text-blue-900">Not Selected</span>
104      </div>
105      <div class="flex justify-between items-center border-t-2 border-blue-100 pt-6">
106        <span class="text-sm font-black text-gray-600 uppercase">Fare Amount</span>
107        <span id="summaryFare" class="text-3xl font-black text-green-600">5.00</span>
108      </div>
109      <div class="mt-8 p-4 bg-yellow-50 rounded-2xl text-center text-[10px] text-yellow-700 font-black uppercase tracking-wid
110        & 5 Seconds Verification Process Active
111      </div>
112    </div>
113  </div>
114 </main>
115 </div>
116 </div>
117 <div id="successModal" class="fixed inset-0 bg-blue-900/60 backdrop-blur-md hidden items-center justify-center z-50 p-4">
118   <div class="bg-white p-1 rounded-[2.5rem] shadow-2xl w-full max-w-sm">
119     <div id="ticket-to-print" class="bg-white p-8 rounded-[2rem] text-center">
120       <div class="border-b-2 border-dashed border-gray-200 pb-4 mb-4">
121         <h1 class="text-2xl font-black italic text-blue-900">METRO TICKET</h1>
122         <p class="text-[8px] text-gray-400 font-bold uppercase tracking-widest">Dhaka Metro Rail Authority</p>
123       </div>
124       <div class="w-16 h-16 bg-green-100 text-green-600 rounded-full flex items-center justify-center mx-auto mb-4 text-2xl">
125       </div>
126       <h3 class="text-xl font-black text-gray-800 mb-6 uppercase">Payment Success</h3>
127       <div class="text-left space-y-3 bg-slate-50 p-5 rounded-2xl mb-6">
128       </div>
129     </div>
130   </div>

```

Ln 188, Col 21 Spaces: 4 UTF-8 CRLF { PHP

10:57 AM 1/4/2026



```
File Edit Selection View ... < > Q metro_rail
EXPLORER register.php dashboard.php passenger X index.php view_tickets.php view_employees.php book_ticket.php get_fare.php main.js ...
METRO_RAIL assets css # style.css images js main.js auth logout.php register.php config db.php employee dashboard.php station_map.php view_employees.php view_tickets.php passenger book_ticket.php dashboard.php get_fare.php travel_history.php index.php login.php
passenger > dashboard.php > html > body.bg-gray-100.flex.h-screen.overflow-hidden.font-sans > script > ready() callback > click() callback
24 <html lang="en">
42 <body class="bg-gray-100 flex h-screen overflow-hidden font-sans">
53   <div class="flex-grow flex flex-col overflow-y-auto">
62     <main class="p-8 grid grid-cols-1 lg:grid-cols-2 gap-8">
63       <div class="bg-white p-8 rounded-[2rem] shadow-xl border border-white">
69         <label class="block text-[10px] font-black text-gray-400 uppercase tracking-widest mb-2">Source Station</label>
70         <select id="fromstation" name="from station" class="w-full border-2 border-gray-50 p-4 rounded-2xl bg-gray-50 o
71           <option value="" disabled selected>Choose Start Point</option>
72           <?php foreach ($stations as $s); ?>
73             | <option value="<?php echo $s['ID']; ?>"><?php echo $s['NAME']; ?></option>
74           <?php endforeach; ?>
75         </select>
76       </div>
77       <div>
78         <label class="block text-[10px] font-black text-gray-400 uppercase tracking-widest mb-2">Destination Station</label>
79         <select id="tostation" name="to_station" class="w-full border-2 border-gray-50 p-4 rounded-2xl bg-gray-50 outline
80           <option value="" disabled selected>Choose End Point</option>
81           <?php foreach ($stations as $s); ?>
82             | <option value="<?php echo $s['ID']; ?>"><?php echo $s['NAME']; ?></option>
83           <?php endforeach; ?>
84         </select>
85       </div>
86       <button id="buyTicketBtn" type="button" class="w-full bg-blue-900 text-white font-black py-5 rounded-2xl hover:bg-b
87         Confirm & Buy Ticket
88       </button>
89     </form>
90   </div>
91
92
93   <div id="summaryCard" class="bg-white p-8 rounded-[2rem] shadow-xl border-4 border-dashed border-blue-50 flex flex-col just
94     <div>
95       <h3 class="text-xl font-black text-gray-800 mb-8 uppercase tracking-widest text-center italic">Digital Ticket Info</h3>
96       <div class="space-y-6">
97         <div class="flex justify-between items-center bg-gray-50 p-4 rounded-2xl">
98           <span class="text-[10px] font-black text-gray-400 uppercase">From:</span>
99           <span id="summaryFrom" class="font-black text-blue-900">Not Selected</span>
100      </div>
101      <div class="flex justify-between items-center bg-gray-50 p-4 rounded-2xl">
102        <span class="text-[10px] font-black text-gray-400 uppercase">To:</span>
103        <span id="summaryTo" class="font-black text-blue-900">Not Selected</span>
104      </div>
105      <div class="flex justify-between items-center border-t-2 border-blue-100 pt-6">
106        <span class="text-sm font-black text-gray-600 uppercase">Fare Amount</span>
107        <span id="summaryFare" class="text-3xl font-black text-green-600">5.00</span>
108      </div>
109      <div class="mt-8 p-4 bg-yellow-50 rounded-2xl text-center text-[10px] text-yellow-700 font-black uppercase tracking-wid
110        & 5 Seconds Verification Process Active
111      </div>
112    </div>
113  </div>
114 </main>
115 </div>
116 </div>
117 <div id="successModal" class="fixed inset-0 bg-blue-900/60 backdrop-blur-md hidden items-center justify-center z-50 p-4">
118   <div class="bg-white p-1 rounded-[2.5rem] shadow-2xl w-full max-w-sm">
119     <div id="ticket-to-print" class="bg-white p-8 rounded-[2rem] text-center">
120       <div class="border-b-2 border-dashed border-gray-200 pb-4 mb-4">
121         <h1 class="text-2xl font-black italic text-blue-900">METRO TICKET</h1>
122         <p class="text-[8px] text-gray-400 font-bold uppercase tracking-widest">Dhaka Metro Rail Authority</p>
123       </div>
124       <div class="w-16 h-16 bg-green-100 text-green-600 rounded-full flex items-center justify-center mx-auto mb-4 text-2xl">
125       </div>
126       <h3 class="text-xl font-black text-gray-800 mb-6 uppercase">Payment Success</h3>
127       <div class="text-left space-y-3 bg-slate-50 p-5 rounded-2xl mb-6">
128       </div>
129     </div>
130   </div>

```

Ln 188, Col 21 Spaces: 4 UTF-8 CRLF { PHP

10:58 AM 1/4/2026

The screenshot shows a code editor interface with the following details:

- File Menu:** File, Edit, Selection, View, ...
- Search Bar:** metro\_rail
- Left Sidebar (EXPLORER):** Shows the project structure with files like register.php, dashboard.php, passenger, index.php, view\_tickets.php, view\_employees.php, book\_ticket.php, get\_fare.php, main.js, assets, style.css, images, and login.php.
- Code Editor Area:** Displays the `dashboard.php` file content. The code includes PHP logic for handling form submissions and displaying travel details, along with a large block of JavaScript at the bottom for updating a summary based on selected station options.

```
register.php dashboard.php passenger index.php view_tickets.php view_employees.php book_ticket.php get_fare.php JS main.js

passenger > dashboard.php passenger > html > body.bg-gray-100.flex.h-screen.overflow-hidden.font-sans > div#successModal.fixed.inset-0.bg-blue-900/60.backdrop-blur-md.hidden.items-center.justify-center.z-50.p-4>
  24   <html lang="en">
  25     <body class="bg-gray-100 flex h-screen overflow-hidden font-sans">
  26       <div id="successModal" class="fixed inset-0 bg-blue-900/60 backdrop-blur-md hidden items-center justify-center z-50 p-4">
  27         <div class="bg-white p-1 rounded-[2.5rem] shadow-2xl w-full max-w-sm">
  28           <div id="ticket-to-print" class="bg-white p-8 rounded-[2rem] text-center">
  29             <div class="flex justify-between"><span class="text-[10px] text-gray-400 font-bold">PASSENGER:</span><span class="text-[10px] text-gray-400 font-bold">FROM:</span><span id="pdfFrom" class="flex justify-between"><span class="text-[10px] text-gray-400 font-bold">TO:</span><span id="pdfTo" class="flex justify-between border-t border-gray-200 pt-2"><span class="text-[10px] text-gray-400 font-bold">F
  30               </div>
  31             <p class="text-[8px] text-gray-400 mb-6 italic">Valid for 1 hour from purchase time.</p>
  32           </div>
  33         <div class="px-8 pb-8 space-y-3">
  34           <button id="downloadBtn" class="w-full bg-green-600 text-white py-4 rounded-xl font-black uppercase tracking-widest shadow-lg transition duration-200 ease-in-out">
  35             Download PDF
  36           </button>
  37           <button onclick="location.reload()" class="w-full bg-blue-900 text-white py-4 rounded-xl font-black uppercase tracking-widest shadow-lg transition duration-200 ease-in-out">
  38             Done
  39           </button>
  40         </div>
  41       </div>
  42     </div>
  43   </div>
  44 </script>
  45 $(document).ready(function() {
  46   function updateSummary() {
  47     let fromId = $('#fromStation').val();
  48     let toId = $('#toStation').val();
  49     let fromName = $("#fromStation option:selected").text();
  50     let toName = $("#toStation option:selected").text();
  51
  52     if (fromId && toId) {
  53       $('#summaryFrom').text(fromName);
  54       $('#summaryTo').text(toName);
  55     }
  56   }
  57
  58   $('#fromStation').change(updateSummary);
  59   $('#toStation').change(updateSummary);
  60
  61   updateSummary();
  62
  63   // Additional logic for ticket printing
  64   // ...
  65
  66   // Function to handle PDF generation
  67   // ...
  68
  69   // Function to handle location reload
  70   // ...
  71
  72   // Function to handle download button click
  73   // ...
  74
  75   // Function to handle done button click
  76   // ...
  77
  78   // Function to handle summary update
  79   // ...
  80
  81   // Function to handle summary display
  82   // ...
  83
  84   // Function to handle summary styling
  85   // ...
  86
  87   // Function to handle summary content
  88   // ...
  89
  90   // Function to handle summary position
  91   // ...
  92
  93   // Function to handle summary visibility
  94   // ...
  95
  96   // Function to handle summary width
  97   // ...
  98
  99   // Function to handle summary height
  100  // ...
  101
  102  // Function to handle summary padding
  103  // ...
  104
  105  // Function to handle summary border radius
  106  // ...
  107
  108  // Function to handle summary shadow
  109  // ...
  110
  111  // Function to handle summary border
  112  // ...
  113
  114  // Function to handle summary border top
  115  // ...
  116
  117  // Function to handle summary border right
  118  // ...
  119
  120  // Function to handle summary border bottom
  121  // ...
  122
  123  // Function to handle summary border left
  124  // ...
  125
  126  // Function to handle summary border top right
  127  // ...
  128
  129  // Function to handle summary border top left
  130  // ...
  131
  132  // Function to handle summary border bottom right
  133  // ...
  134
  135  // Function to handle summary border bottom left
  136  // ...
  137
  138  // Function to handle summary border radius top right
  139  // ...
  140
  141  // Function to handle summary border radius top left
  142  // ...
  143
  144  // Function to handle summary border radius bottom right
  145  // ...
  146
  147  // Function to handle summary border radius bottom left
  148  // ...
  149
  150  // Function to handle summary border radius top
  151  // ...
  152
  153  // Function to handle summary border radius bottom
  154  // ...
  155
  156  // Function to handle summary border radius right
  157  // ...
  158
  159  // Function to handle summary border radius left
  160  // ...
  161
  162
  163
  164
  165
  166
  167
  168
  169
  170
  171
  172
  173
  174
  175
  176
  177
  178
  179
  180
  181
  182
  183
  184
  185
  186
  187
  188
  189
  190
  191
  192
  193
  194
  195
  196
  197
  198
  199
  200
  201
  202
  203
  204
  205
  206
  207
  208
  209
  210
  211
  212
  213
  214
  215
  216
  217
  218
  219
  220
  221
  222
  223
  224
  225
  226
  227
  228
  229
  230
  231
  232
  233
  234
  235
  236
  237
  238
  239
  240
  241
  242
  243
  244
  245
  246
  247
  248
  249
  250
  251
  252
  253
  254
  255
  256
  257
  258
  259
  260
  261
  262
  263
  264
  265
  266
  267
  268
  269
  270
  271
  272
  273
  274
  275
  276
  277
  278
  279
  280
  281
  282
  283
  284
  285
  286
  287
  288
  289
  290
  291
  292
  293
  294
  295
  296
  297
  298
  299
  300
  301
  302
  303
  304
  305
  306
  307
  308
  309
  310
  311
  312
  313
  314
  315
  316
  317
  318
  319
  320
  321
  322
  323
  324
  325
  326
  327
  328
  329
  330
  331
  332
  333
  334
  335
  336
  337
  338
  339
  340
  341
  342
  343
  344
  345
  346
  347
  348
  349
  350
  351
  352
  353
  354
  355
  356
  357
  358
  359
  360
  361
  362
  363
  364
  365
  366
  367
  368
  369
  370
  371
  372
  373
  374
  375
  376
  377
  378
  379
  380
  381
  382
  383
  384
  385
  386
  387
  388
  389
  390
  391
  392
  393
  394
  395
  396
  397
  398
  399
  400
  401
  402
  403
  404
  405
  406
  407
  408
  409
  410
  411
  412
  413
  414
  415
  416
  417
  418
  419
  420
  421
  422
  423
  424
  425
  426
  427
  428
  429
  430
  431
  432
  433
  434
  435
  436
  437
  438
  439
  440
  441
  442
  443
  444
  445
  446
  447
  448
  449
  450
  451
  452
  453
  454
  455
  456
  457
  458
  459
  460
  461
  462
  463
  464
  465
  466
  467
  468
  469
  470
  471
  472
  473
  474
  475
  476
  477
  478
  479
  480
  481
  482
  483
  484
  485
  486
  487
  488
  489
  490
  491
  492
  493
  494
  495
  496
  497
  498
  499
  500
  501
  502
  503
  504
  505
  506
  507
  508
  509
  510
  511
  512
  513
  514
  515
  516
  517
  518
  519
  520
  521
  522
  523
  524
  525
  526
  527
  528
  529
  530
  531
  532
  533
  534
  535
  536
  537
  538
  539
  540
  541
  542
  543
  544
  545
  546
  547
  548
  549
  550
  551
  552
  553
  554
  555
  556
  557
  558
  559
  560
  561
  562
  563
  564
  565
  566
  567
  568
  569
  570
  571
  572
  573
  574
  575
  576
  577
  578
  579
  580
  581
  582
  583
  584
  585
  586
  587
  588
  589
  590
  591
  592
  593
  594
  595
  596
  597
  598
  599
  600
  601
  602
  603
  604
  605
  606
  607
  608
  609
  610
  611
  612
  613
  614
  615
  616
  617
  618
  619
  620
  621
  622
  623
  624
  625
  626
  627
  628
  629
  630
  631
  632
  633
  634
  635
  636
  637
  638
  639
  640
  641
  642
  643
  644
  645
  646
  647
  648
  649
  650
  651
  652
  653
  654
  655
  656
  657
  658
  659
  660
  661
  662
  663
  664
  665
  666
  667
  668
  669
  670
  671
  672
  673
  674
  675
  676
  677
  678
  679
  680
  681
  682
  683
  684
  685
  686
  687
  688
  689
  690
  691
  692
  693
  694
  695
  696
  697
  698
  699
  700
  701
  702
  703
  704
  705
  706
  707
  708
  709
  710
  711
  712
  713
  714
  715
  716
  717
  718
  719
  720
  721
  722
  723
  724
  725
  726
  727
  728
  729
  730
  731
  732
  733
  734
  735
  736
  737
  738
  739
  740
  741
  742
  743
  744
  745
  746
  747
  748
  749
  750
  751
  752
  753
  754
  755
  756
  757
  758
  759
  759
  760
  761
  762
  763
  764
  765
  766
  767
  768
  769
  770
  771
  772
  773
  774
  775
  776
  777
  778
  779
  779
  780
  781
  782
  783
  784
  785
  786
  787
  788
  789
  789
  790
  791
  792
  793
  794
  795
  796
  797
  798
  799
  799
  800
  801
  802
  803
  804
  805
  806
  807
  808
  809
  809
  810
  811
  812
  813
  814
  815
  816
  817
  818
  819
  819
  820
  821
  822
  823
  824
  825
  826
  827
  828
  829
  829
  830
  831
  832
  833
  834
  835
  836
  837
  838
  839
  839
  840
  841
  842
  843
  844
  845
  846
  847
  848
  849
  849
  850
  851
  852
  853
  854
  855
  856
  857
  858
  859
  859
  860
  861
  862
  863
  864
  865
  866
  867
  868
  869
  869
  870
  871
  872
  873
  874
  875
  876
  877
  878
  879
  879
  880
  881
  882
  883
  884
  885
  886
  887
  888
  889
  889
  890
  891
  892
  893
  894
  895
  896
  897
  898
  899
  899
  900
  901
  902
  903
  904
  905
  906
  907
  908
  909
  909
  910
  911
  912
  913
  914
  915
  916
  917
  918
  919
  919
  920
  921
  922
  923
  924
  925
  926
  927
  928
  929
  929
  930
  931
  932
  933
  934
  935
  936
  937
  938
  939
  939
  940
  941
  942
  943
  944
  945
  946
  947
  948
  949
  949
  950
  951
  952
  953
  954
  955
  956
  957
  958
  959
  959
  960
  961
  962
  963
  964
  965
  966
  967
  968
  969
  969
  970
  971
  972
  973
  974
  975
  976
  977
  978
  979
  979
  980
  981
  982
  983
  984
  985
  986
  987
  988
  989
  989
  990
  991
  992
  993
  994
  995
  996
  997
  998
  999
  999
  1000
  1001
  1002
  1003
  1004
  1005
  1006
  1007
  1008
  1009
  1009
  1010
  1011
  1012
  1013
  1014
  1015
  1016
  1017
  1018
  1019
  1019
  1020
  1021
  1022
  1023
  1024
  1025
  1026
  1027
  1028
  1029
  1029
  1030
  1031
  1032
  1033
  1034
  1035
  1036
  1037
  1038
  1039
  1039
  1040
  1041
  1042
  1043
  1044
  1045
  1046
  1047
  1048
  1049
  1049
  1050
  1051
  1052
  1053
  1054
  1055
  1056
  1057
  1058
  1059
  1059
  1060
  1061
  1062
  1063
  1064
  1065
  1066
  1067
  1068
  1069
  1069
  1070
  1071
  1072
  1073
  1074
  1075
  1076
  1077
  1078
  1078
  1079
  1080
  1081
  1082
  1083
  1084
  1085
  1086
  1087
  1088
  1088
  1089
  1090
  1091
  1092
  1093
  1094
  1095
  1096
  1097
  1097
  1098
  1099
  1099
  1100
  1101
  1102
  1103
  1104
  1105
  1106
  1107
  1108
  1109
  1109
  1110
  1111
  1112
  1113
  1114
  1115
  1116
  1117
  1118
  1119
  1119
  1120
  1121
  1122
  1123
  1124
  1125
  1126
  1127
  1128
  1129
  1129
  1130
  1131
  1132
  1133
  1134
  1135
  1136
  1137
  1138
  1138
  1139
  1140
  1141
  1142
  1143
  1144
  1145
  1146
  1147
  1148
  1149
  1149
  1150
  1151
  1152
  1153
  1154
  1155
  1156
  1157
  1158
  1159
  1159
  1160
  1161
  1162
  1163
  1164
  1165
  1166
  1167
  1168
  1169
  1169
  1170
  1171
  1172
  1173
  1174
  1175
  1176
  1177
  1178
  1178
  1179
  1180
  1181
  1182
  1183
  1184
  1185
  1186
  1187
  1188
  1188
  1189
  1190
  1191
  1192
  1193
  1194
  1195
  1196
  1197
  1197
  1198
  1199
  1199
  1200
  1201
  1202
  1203
  1204
  1205
  1206
  1207
  1208
  1209
  1209
  1210
  1211
  1212
  1213
  1214
  1215
  1216
  1217
  1218
  1219
  1219
  1220
  1221
  1222
  1223
  1224
  1225
  1226
  1227
  1228
  1229
  1229
  1230
  1231
  1232
  1233
  1234
  1235
  1236
  1237
  1238
  1238
  1239
  1240
  1241
  1242
  1243
  1244
  1245
  1246
  1247
  1248
  1249
  1249
  1250
  1251
  1252
  1253
  1254
  1255
  1256
  1257
  1258
  1259
  1259
  1260
  1261
  1262
  1263
  1264
  1265
  1266
  1267
  1268
  1268
  1269
  1270
  1271
  1272
  1273
  1274
  1275
  1276
  1277
  1278
  1278
  1279
  1280
  1281
  1282
  1283
  1284
  1285
  1286
  1287
  1288
  1288
  1289
  1290
  1291
  1292
  1293
  1294
  1295
  1296
  1297
  1297
  1298
  1299
  1299
  1300
  1301
  1302
  1303
  1304
  1305
  1306
  1307
  1308
  1309
  1309
  1310
  1311
  1312
  1313
  1314
  1315
  1316
  1317
  1318
  1319
  1319
  1320
  1321
  1322
  1323
  1324
  1325
  1326
  1327
  1328
  1328
  1329
  1330
  1331
  1332
  1333
  1334
  1335
  1336
  1337
  1338
  1338
  1339
  1340
  1341
  1342
  1343
  1344
  1345
  1346
  1347
  1348
  1349
  1349
  1350
  1351
  1352
  1353
  1354
  1355
  1356
  1357
  1358
  1359
  1359
  1360
  1361
  1362
  1363
  1364
  1365
  1366
  1367
  1368
  1368
  1369
  1370
  1371
  1372
  1373
  1374
  1375
  1376
  1377
  1378
  1378
  1379
  1380
  1381
  1382
  1383
  1384
  1385
  1386
  1387
  1388
  1388
  1389
  1390
  1391
  1392
  1393
  1394
  1395
  1396
  1397
  1398
  1398
  1399
  1400
  1401
  1402
  1403
  1404
  1405
  1406
  1407
  1408
  1409
  1409
  1410
  1411
  1412
  1413
  1414
  1415
  1416
  1417
  1418
  1419
  1419
  1420
  1421
  1422
  1423
  1424
  1425
  1426
  1427
  1428
  1428
  1429
  1430
  1431
  1432
  1433
  1434
  1435
  1436
  1437
  1438
  1438
  1439
  1440
  1441
  1442
  1443
  1444
  1445
  1446
  1447
  1448
  1449
  1449
  1450
  1451
  1452
  1453
  1454
  1455
  1456
  1457
  1458
  1459
  1459
  1460
  1461
  1462
  1463
  1464
  1465
  1466
  1467
  1468
  1468
  1469
  1470
  1471
  1472
  1473
  1474
  1475
  1476
  1477
  1478
  1478
  1479
  1480
  1481
  1482
  1483
  1484
  1485
  1486
  1487
  1488
  1488
  1489
  1490
  1491
  1492
  1493
  1494
  1495
  1496
  1497
  1498
  1498
  1499
  1500
  1501
  1502
  1503
  1504
  1505
  1506
  1507
  1508
  1509
  1509
  1510
  1511
  1512
  1513
  1514
  1515
  1516
  1517
  1518
  1519
  1519
  1520
  1521
  1522
  1523
  1524
  1525
  1526
  1527
  1528
  1529
  1529
  1530
  1531
  1532
  1533
  1534
  1535
  1536
  1537
  1538
  1539
  1539
  1540
  1541
  1542
  1543
  1544
  1545
  1546
  1547
  1548
  1549
  1549
  1550
  1551
  1552
  1553
  1554
  1555
  1556
  1557
  1558
  1559
  1559
  1560
  1561
  1562
  1563
  1564
  1565
  1566
  1567
  1568
  1568
  1569
  1570
  1571
  1572
  1573
  1574
  1575
  1576
  1577
  1578
  1578
  1579
  1580
  1581
  1582
  1583
  1584
  1585
  1586
  1587
  1588
  1589
  1589
  1590
  1591
  1592
  1593
  1594
  1595
  1596
  1597
  1598
  1599
  1599
  1600
  1601
  1602
  1603
  1604
  1605
  1606
  1607
  1608
  1609
  1609
  1610
  1611
  1612
  1613
  1614
  1615
  1616
  1617
  1618
  1619
  1619
  1620
  1621
  1622
  1623
  1624
  1625
  1626
  1627
  1628
  1629
  1629
  1630
  1631
  1632
  1633
  1634
  1635
  1636
  1637
  1638
  1639
  1639
  1640
  1641
  1642
  1643
  1644
  1645
  1646
  1647
  1648
  1649
  1649
  1650
  1651
  1652
  1653
  1654
  1655
  1656
  1657
  1658
  1659
  1659
  1660
  1661
  1662
  1663
  1664
  1665
  1666
  1667
  1668
  1669
  1669
  1670
  
```

File Edit Selection View ... Q metro\_rail

EXPLORER METRO\_RAIL

- assets
- css # style.css
- images
- js main.js
- auth
- register.php
- config
- db.php
- employee
- dashboard.php
- station\_map.php
- view\_employees.php
- view\_tickets.php
- passenger
- book\_ticket.php
- dashboard.php
- get\_fare.php
- travel\_history.php
- index.php
- login.php

```

24 <html lang="en">
42 <body class="bg-gray-100 flex h-screen overflow-hidden font-sans">
150 <script>
151 $(document).ready(function() {
176     $('#buyTicketBtn').click(function() {
193         method: 'POST',
194         data: { from_id: from, to_id: to, amount: fareNum },
195         success: function(res) {
196             if (res.trim() === "Success") {
197                 // Set data for PDF Modal
198                 $('#pdfFrom').text(frontName);
199                 $('#pdfTo').text(toName);
200                 $('#pdfFare').text(fareText);
201
202                 $('#successModal').removeClass('hidden').addClass('flex');
203             } else {
204                 alert("Booking Failed: " + res);
205             }
206             btn.prop('disabled', false).html(originalText);
207         }
208     });
209 }, 5000); // 5 Seconds Buffer
210 } else {
211     alert("Please select valid stations!");
212 }
213 });

// PDF Download Logic
$('#downloadBtn').click(function() {
    const element = document.getElementById('ticket-to-print');
    const opt = {
        margin: 0.5,
        filename: 'Metro_Ticket_<?php echo time(); ?>.pdf',
        image: { type: 'jpeg', quality: 1 },
        html2canvas: { scale: 3, letterRendering: true },
        jsPDF: { unit: 'in', format: 'letter', orientation: 'portrait' }
    };
    html2pdf().set(opt).from(element).save();
});

```

Ln 149, Col 11 Spaces: 4 UTF-8 CRLF PHP

Search 10:59 AM 1/4/2026

File Edit Selection View ... Q metro\_rail

EXPLORER METRO\_RAIL

- assets
- css # style.css
- images
- js main.js
- auth
- register.php
- config
- db.php
- employee
- dashboard.php
- station\_map.php
- view\_employees.php
- view\_tickets.php
- passenger
- book\_ticket.php
- dashboard.php
- get\_fare.php
- travel\_history.php
- index.php
- login.php

```

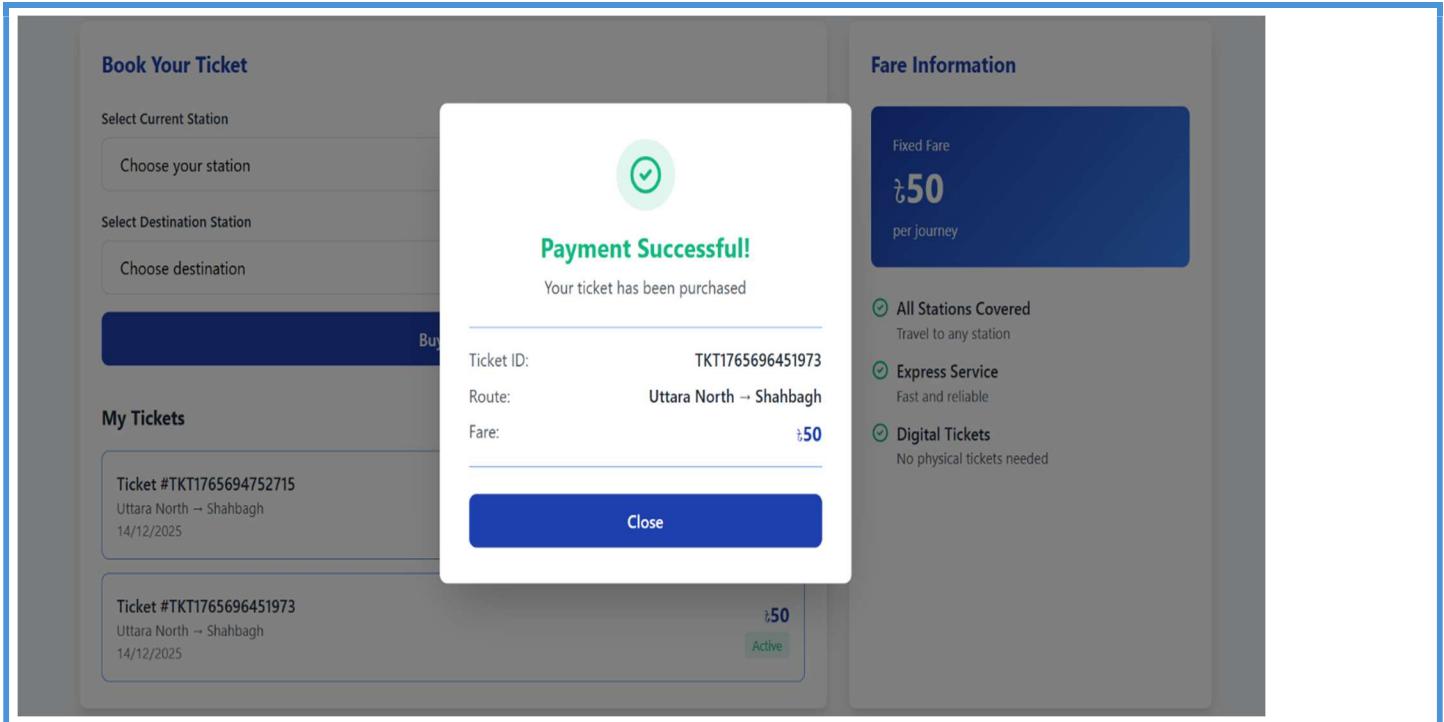
24 <html lang="en">
42 <body class="bg-gray-100 flex h-screen overflow-hidden font-sans">
150 <script>
151 $(document).ready(function() {
176     $('#buyTicketBtn').click(function() {
193         method: 'POST',
194         data: { from_id: from, to_id: to, amount: fareNum },
195         success: function(res) {
196             if (res.trim() === "Success") {
197                 // Set data for PDF Modal
198                 $('#pdfFrom').text(frontName);
199                 $('#pdfTo').text(toName);
200                 $('#pdfFare').text(fareText);
201
202                 $('#successModal').removeClass('hidden').addClass('flex');
203             } else {
204                 alert("Booking Failed: " + res);
205             }
206             btn.prop('disabled', false).html(originalText);
207         }
208     });
209 }, 5000); // 5 Seconds Buffer
210 } else {
211     alert("Please select valid stations!");
212 }
213 });

// PDF Download Logic
$('#downloadBtn').click(function() {
    const element = document.getElementById('ticket-to-print');
    const opt = {
        margin: 0.5,
        filename: 'Metro_Ticket_<?php echo time(); ?>.pdf',
        image: { type: 'jpeg', quality: 1 },
        html2canvas: { scale: 3, letterRendering: true },
        jsPDF: { unit: 'in', format: 'letter', orientation: 'portrait' }
    };
    html2pdf().set(opt).from(element).save();
});

```

Ln 149, Col 11 Spaces: 4 UTF-8 CRLF PHP

Search 10:59 AM 1/4/2026



File Edit Selection View ... ← → Q metro\_rail

EXPLORER

- METRO\_RAIL
  - assets
    - # style.css
    - images
    - js
      - main.js
  - auth
    - logout.php
    - register.php
  - config
    - db.php
  - employee
    - dashboard.php
    - station\_map.php
    - view\_employees.php
    - view\_tickets.php
  - passenger
    - book\_ticket.php
    - dashboard.php
    - get\_fare.php
    - travel\_history.php
- index.php
- login.php

register.php dashboard.php passenger index.php view\_tickets.php view\_employees.php book\_ticket.php get\_fare.php JS main.js

```

2rem].shadow-xl.border-4.border-dashed.border-blue-50.flex.flex-col.justify-between > div.mt-8.p-4.bg-yellow-50.rounded-2xl.text-center.text-[10px].text-yellow-700.font-black.uppercase.tracking-widest.italic
24 <html lang="en">
42 <body class="bg-gray-100 flex h-screen overflow-hidden font-sans">
117
118   <div id="successModal" class="fixed inset-0 bg-blue-900/60 backdrop-blur-md hidden items-center justify-center z-50 p-4">
119     <div class="bg-white p-1 rounded-[2.5rem] shadow-2xl w-full max-w-sm">
120       <div id="ticket-to-print" class="bg-white p-8 rounded-[2rem] text-center">
121         <div class="border-b-2 border-dashed border-gray-200 pb-4 mb-4">
122           <hi class="text-2xl font-black italic text-blue-900">METRO TICKET</hi>
123           <p class="text-[8px] text-gray-400 font-bold uppercase tracking-widest">Dhaka Metro Rail Authority</p>
124         </div>
125
126         <div class="w-16 h-16 bg-green-100 text-green-600 rounded-full flex items-center justify-center mx-auto mb-4 text-2xl">
127           <h3 class="text-xl font-black text-gray-800 mb-6 uppercase">Payment Success</h3>
128
129         <div class="text-left space-y-3 bg-slate-50 p-5 rounded-2xl mb-6">
130           <div class="flex justify-between"><span class="text-[10px] text-gray-400 font-bold">PASSENGER:</span><span class="text-[10px] text-gray-400 font-bold">FROM:</span><span id="pdfFrom" class="flex justify-between"><span class="text-[10px] text-gray-400 font-bold">TO:</span><span id="pdfTo" class="flex justify-between border-t border-gray-200 pt-2"><span class="text-[10px] text-gray-400 font-bold">F</span></div>
131           <div class="text-[8px] text-gray-400 mb-6 italic">Valid for 1 hour from purchase time.</p>
132         </div>
133
134         <div class="px-8 pb-8 space-y-3">
135           <button id="downloadBtn" class="w-full bg-green-600 text-white py-4 rounded-xl font-black uppercase tracking-widest shadow-lg">
136             <span>Download PDF</span>
137           </button>
138           <button onclick="location.reload()" class="w-full bg-blue-900 text-white py-4 rounded-xl font-black uppercase tracking-widest shadow-lg">
139             <span>Done</span>
140           </button>
141         </div>
142       </div>
143     </div>
144   </div>
145
146   </div>
147
148 </div>
149 </div>
150 <script>
```

Ln 112, Col 33 Spaces: 4 UTF-8 CRLF {} PHP ⚡ 11:06 AM 1/4/2026

