



AMERICAN INTERNATIONAL UNIVERSITY–BANGLADESH (AIUB)

FACULTY OF SCIENCE & TECHNOLOGY

Advance Database Management System

Metro Rail Management System

Section: A

Supervised By

Juena Ahmed Noshin

Submitted By

1. Md. Amir Hossain Alif
2. Asir Foysal Al Mukit
3. Umme Jannatul Fariha
4. Afsana Jahan Onu

TABLE OF CONTENTS

I.	Title page	1
II.	Table of content	2
1.	Introduction	3
2.	User Interface	3
3.	Scenario Description	7
4.	Project Proposal	7
5.	ER Diagram	8
6.	Normalization	9
7.	Class Diagram	13
8.	Use Case Diagram	14
9.	Activity Diagram	15
10.	Schema Diagram	16
11.	Table Creation	16
12.	Describe Table	21
13.	Data Insertion	25
14.	Basic PL/SQL	32
15.	Advance PL/SQL	47
16.	Relational Algebra	68
17.	Conclusion	70

Introduction

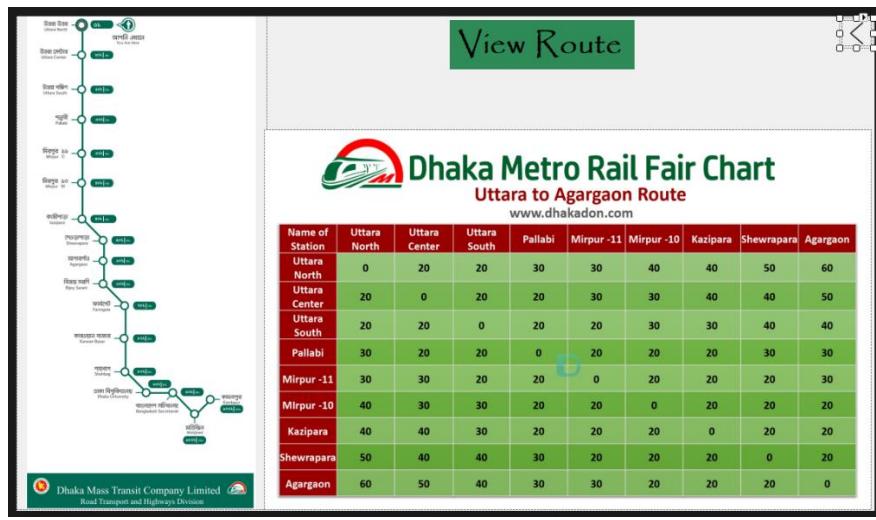
The project deals with designing and implementing a Metro Station Management System for managing all activities involving the metro. This would include ticketing, passenger information, train schedules, route management, and administration. The proposed system will be designed with improved operation efficiency and convenient and smooth travel for the passengers with various facilities such as rapid buying of tickets, up-to-date information, and route management on interactive pages.

The system will integrate core entities, such as passengers, tickets, stations, routes, transactions, and staff, for accurate data management and smooth coordination toward optimum service delivery. It caters to both passengers and administrators by allowing appropriate tracking of data in real time and optimizing the overall service.

User Interface



Two screenshots of the user interface. The left screenshot shows a 'SIGN UP HERE' form titled 'Welcome Back !'. It features a large blue background image of a metro train. The form includes fields for Name, Address, Phone Number, Email, Password, Age, Nid No, and Gender, along with 'Passenger' and 'Admin' radio button options and a 'Sign up' button. The right screenshot shows a 'Login' form with fields for 'Name' (with a user icon), 'ID' (with a card icon), and 'Password' (with a lock icon). It also features a 'Login' button and a back arrow icon in the top right corner.



Search _Plan Journey Section

Fields

Starting Station Ending Station

Date of Travel Tuesday , January 28.

Tickets Type Regular MRT Rapid

Passenger ID

Total Price

Payment Method

Book Tickets **View Route Details**

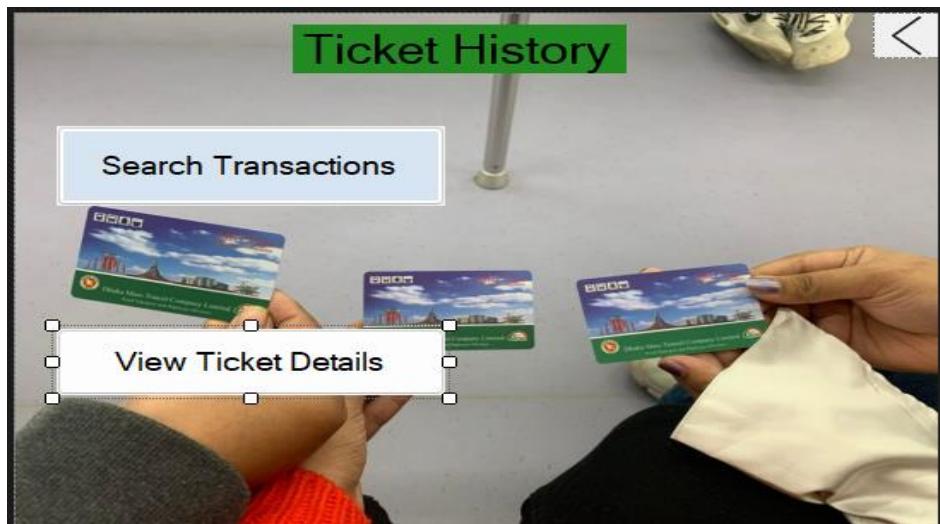
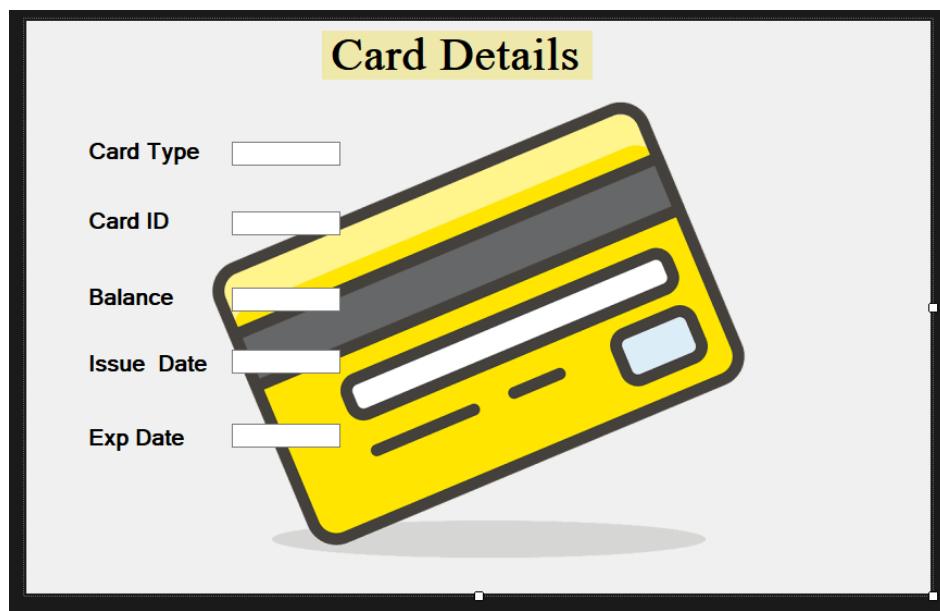
Ticket Purchase

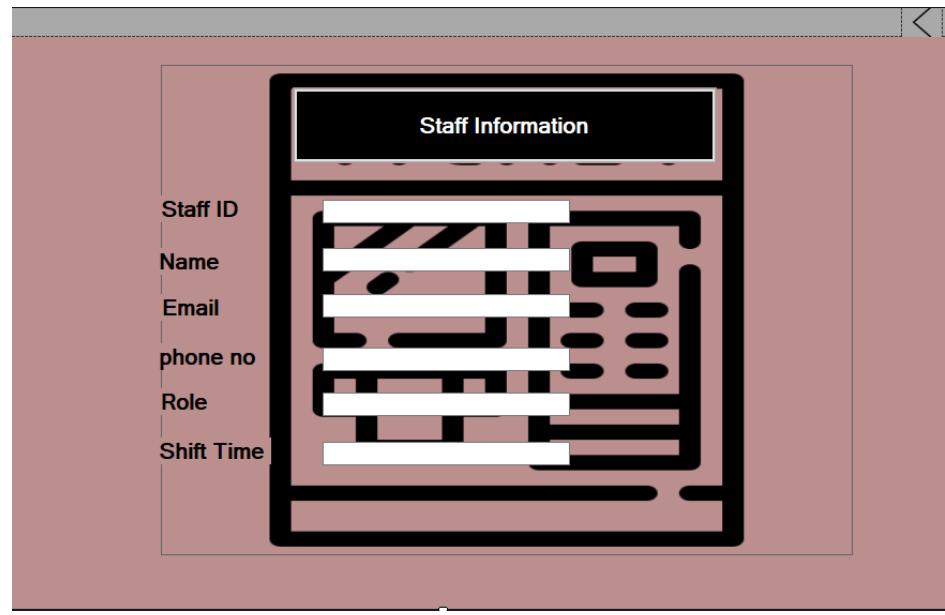
Tickets Type

Single Journey Day Pass MRT Rapid Regular

Payment Method

Back To Dashboard





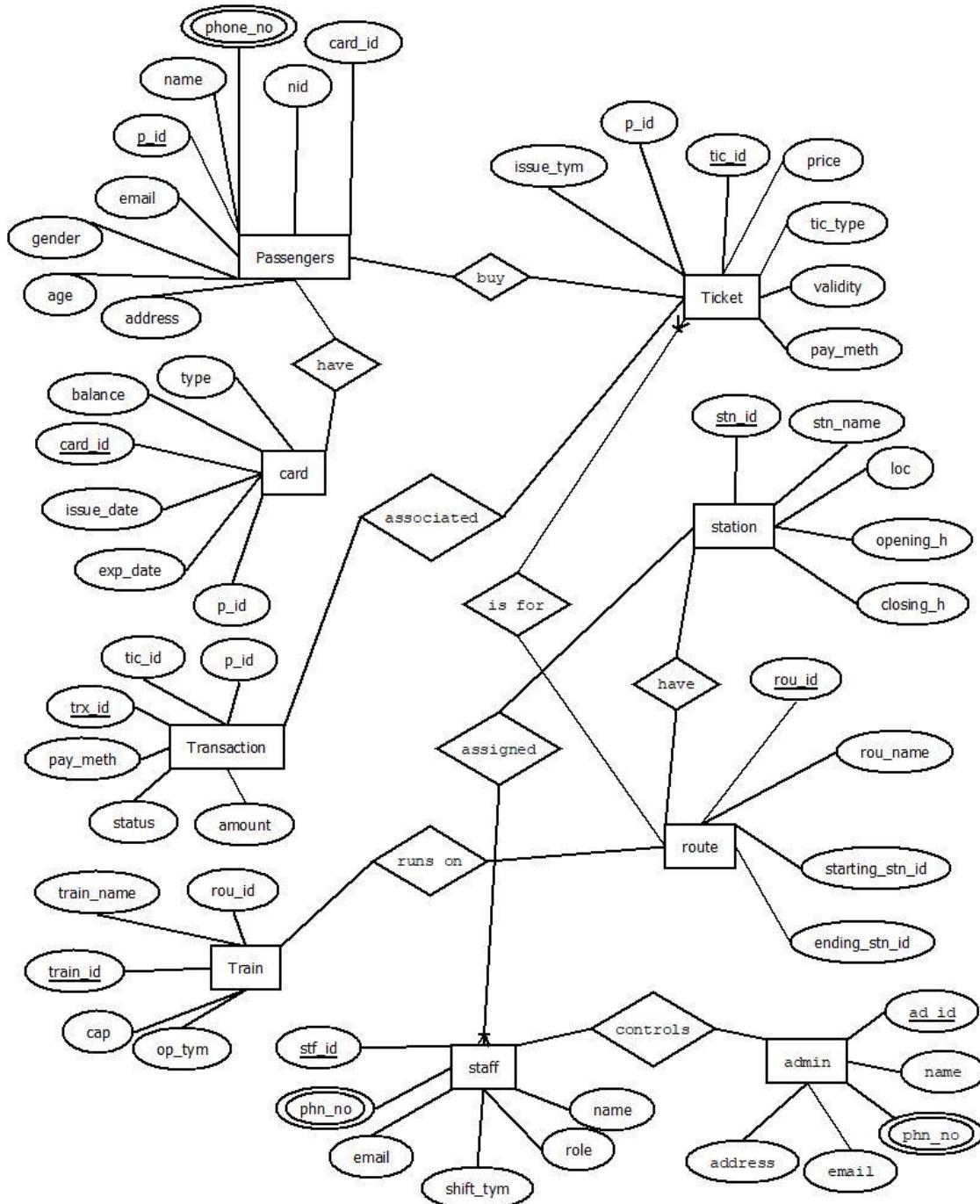
Scenario Description

The metro rail management system efficiently manages passenger ticketing, train operations, and station staffing. Passengers can register in the system with a unique passenger ID and provide details such as name, contact information, address, age, gender, and national identification. Each passenger is issued a single metro card containing a card ID, type, balance, issue date, and expiration date, which can be used for purchasing tickets. Tickets, identified by a unique ticket ID, include details such as type, issue time, validity, and payment method. Passengers can buy multiple tickets, with each ticket linked to a specific transaction. Transactions, identified by transaction ID, record the passenger ID, ticket ID, payment method, status, and amount. Tickets are associated with predefined routes, comprising a route ID, name, starting station, and ending station. Trains operate on these routes and are defined by train ID, name, route ID, capacity, and operating hours. Each route encompasses multiple stations, which are identified by station ID, name, location, and operational hours. The system is managed by admins who assign staff to stations, ensuring efficient operations. Admins are identified by admin ID and maintain their contact information and address. Stations are staffed by personnel with specific roles and shift times, with each staff member assigned to one station. This integrated system ensures seamless ticketing, route management, and operational efficiency for passengers and administrators alike.

Project Proposal

Metro Rail Management System is a digital solution for the betterment of railway operations by enabling passengers to purchase tickets online, maintain their travel cards, and manage transactions efficiently. The system features include passenger registration, secure ticket purchase, smart card integration, and automated financial transaction processing. It also includes train and route management, and administrative control for staff and operators. The system is focused on digitalizing railway ticketing, reducing manual workload, and enhancing passenger experience by using C# for both frontend and backend, and Oracle for the database. This project ensures a fast, seamless, and secure railway management system, hence guaranteeing efficiency, security, and user satisfaction with secure payment gateways and automated processes.

ER Diagram



Normalization

BUY:

UNF:

buy(p_id, name, nid, phn_no, email, age, gender, address, card_id, tic_id, tic_type, issue_tym, price, validity, pay_meth, p_id)

1NF:

Phn_no is a multivalued attribute.

1. p_id, name, nid, phn_no, email, age, gender, address, card_id, tic_id, tic_type, issue_tym, price, validity, pay_meth, p_id

2NF:

1. p_id, name, nid, phn_no, email, age, gender, address, card_id
2. tic_id, tic_type, issue_tym, price, validity, pay_meth, p_id

3NF:

There is no transitive dependency. Relation already in 3NF.

1. p_id, name, nid, phn_no, email, age, gender, address, card_id
2. tic_id, tic_type, issue_tym, price, validity, pay_meth, p_id

Table Creation:

1. p_id, name, nid, phn_no, email, age, gender, address, card_id
2. tic_id, tic_type, issue_tym, price, validity, pay_meth, p_id

ASSOCIATED:

UNF:

associated(trx_id, tic_id, p_id, pay_meth, status, amount, tic_id, tic_type, issue_tym, price, validity, pay_meth, p_id)

1NF:

There is no multivalued attribute. Relation already in 1nf.

1. trx_id, tic_id, p_id, pay_meth, status, amount, tic_id, tic_type, issue_tym, price, validity, pay_meth, p_id

2NF:

1. trx_id, tic_id, p_id, pay_meth, status, amount
2. tic_id, tic_type, issue_tym, price, validity, pay_meth, p_id

3NF:

There is no transitive dependency. Relation already in 3NF.

1. trx_id, tic_id, p_id, pay_meth, status, amount
2. tic_id, tic_type, issue_tym, price, validity, pay_meth, p_id

Table Creation:

1. trx_id, tic_id, p_id, pay_meth, status, amount
2. tic_id, tic_type, issue_tym, price, validity, pay_meth, p_id

HAVE:

UNF:

have(sttn_id,sttn_name,loc,opening_h,closing_h,rou_id,rou_name,strt_sttn_id,end_sttn_id)

1NF:

There is no multivalued attribute. Relation already in 1nf.

1.sttn_id,sttn_name,loc,opening_h,closing_h,rou_id,rou_name,strt_sttn_id,end_sttn_id

2NF:

1.sttn_id,sttn_name,loc,opening_h,closing_h

2.rou_id,rou_name,strt_sttn_id,end_sttn_id

3NF:

There is no transitive dependency. Relation already in 3NF.

1.sttn_id,sttn_name,loc,opening_h,closing_h

2.rou_id,rou_name,strt_sttn_id,end_sttn_id

Table Creation:

1.sttn_id,sttn_name,loc,opening_h,closing_h

2.rou_id,rou_name,strt_sttn_id,end_sttn_id

RUNS ON:

UNF:

runs_on(train_id,train_name,rou_id,cap,op_tym,rou_id,rou_name,strt_sttn_id,end_sttn_id)

1NF:

There is no multivalued attribute. Relation already in 1nf.

1.train_id,train_name,rou_id,cap,op_tym,rou_id,rou_name,strt_sttn_id,end_sttn_id

2NF:

1.train_id,train_name,rou_id,cap,op_tym

2.rou_id,rou_name,strt_sttn_id,end_sttn_id

3NF:

There is no transitive dependency. Relation already in 3NF.

1.train_id,train_name,rou_id,cap,op_tym

2.rou_id,rou_name,strt_sttn_id,end_sttn_id

Table Creation:

1.train_id,train_name,rou_id,cap,op_tym

2.rou_id,rou_name,strt_sttn_id,end_sttn_id

ASSIGNED:

UNF:

assigned(sttn_id,sttn_name,loc,opening_h,closing_h,staff_id,name,role,phn_no,email,shift_tym,assigner_id_sttn_id)

1NF:

Phn_no is a multivalued attribute.

1.sttn_id,sttn_name,loc,opening_h,closing_h,staff_id,name,role,phn_no,email,shift_tym,assigned_sttn_id

2NF:

1.sttn_id,sttn_name,loc,opening_h,closing_h

2.staff_id,name,role,phn_no,email,shift_tym,assigned_sttn_id

3NF:

There is no transitive dependency. Relation already in 3NF.

1.sttn_id,sttn_name,loc,opening_h,closing_h

2.staff_id,name,role,phn_no,email,shift_tym,assigned_sttn_id

Table Creation:

1.sttn_id,sttn_name,loc,opening_h,closing_h

2.staff_id,name,role,phn_no,email,shift_tym,assigned_sttn_id

HAVE_CARD:

UNF:

Have_card(p_id,name,nid,phn_no,email,age,gender,address,card_id,card_id,type,balance,issue_date,exp_date,p_id)

1NF:

Phn_no is a multivalued attribute.

1.p_id,name,nid,phn_no,email,age,gender,address,card_id,card_id,type,balance,issue_date,exp_date,p_id

2NF:

1. p_id,name,nid,phn_no,email,age,gender,address,card_id

2. card_id,type,balance,issue_date,exp_date,p_id

3NF:

There is no transitive dependency. Relation already in 3NF.

1. p_id,name,nid,phn_no,email,age,gender,address,card_id

2. card_id,type,balance,issue_date,exp_date,p_id

Table Creation:

1. p_id,name,nid,phn_no,email,age,gender,address,card_id

2. card_id,type,balance,issue_date,exp_date,p_id

CONTROL:

UNF:

control(ad_id,

name,phn_no,email,addrs,nid,staff_id,name,role,phn_no,email,shift_tym,assigned_sttn_id)

1NF:

Phn_no is a multivalued attribute.

1.ad_id, name,phn_no,email,addrs,nid,staff_id,name,role,phn_no,email,shift_tym,assigned_sttn_id

2NF:

1.ad_id, name,phn_no,email,addrs,nid

2. staff_id, name, role, phn_no, email, shift_tym, assigned_sttn_id

3NF:

There is no transitive dependency. Relation already in 3NF.

1. ad_id, name, phn_no, email, addrs, nid

2. staff_id, name, role, phn_no, email, shift_tym, assigned_sttn_id

Table Creation:

1. ad_id, name, phn_no, email, addrs, nid

2. staff_id, name, role, phn_no, email, shift_tym, assigned_sttn_id

Temporary Tables:

01. p_id, name, nid, phn_no, email, age, gender, address, card_id

02. tic_id, tic_type, issue_tym, price, validity, pay_meth, p_id

03. trx_id, tic_id, p_id, pay_meth, status, amount

04. tic_id, tic_type, issue_tym, price, validity, pay_meth, p_id

05. sttn_id, sttn_name, loc, opening_h, closing_h

06. rou_id, rou_name, strt_sttn_id, end_sttn_id

07. train_id, train_name, rou_id, cap, op_tym

08. rou_id, rou_name, strt_sttn_id, end_sttn_id

09. sttn_id, sttn_name, loc, opening_h, closing_h

10. staff_id, name, role, phn_no, email, shift_tym, assigned_sttn_id

11. p_id, name, nid, phn_no, email, age, gender, address, card_id

12. card_id, type, balance, issue_date, exp_date, p_id

13. ad_id, name, phn_no, email, addrs, nid

14. staff_id, name, role, phn_no, email, shift_tym, assigned_sttn_id

Final Tables:

01. p_id, name, nid, phn_no, email, age, gender, address, card_id

02. tic_id, tic_type, issue_tym, price, validity, pay_meth, p_id

03. trx_id, tic_id, p_id, pay_meth, status, amount

04. sttn_id, sttn_name, loc, opening_h, closing_h

05. rou_id, rou_name, strt_sttn_id, end_sttn_id

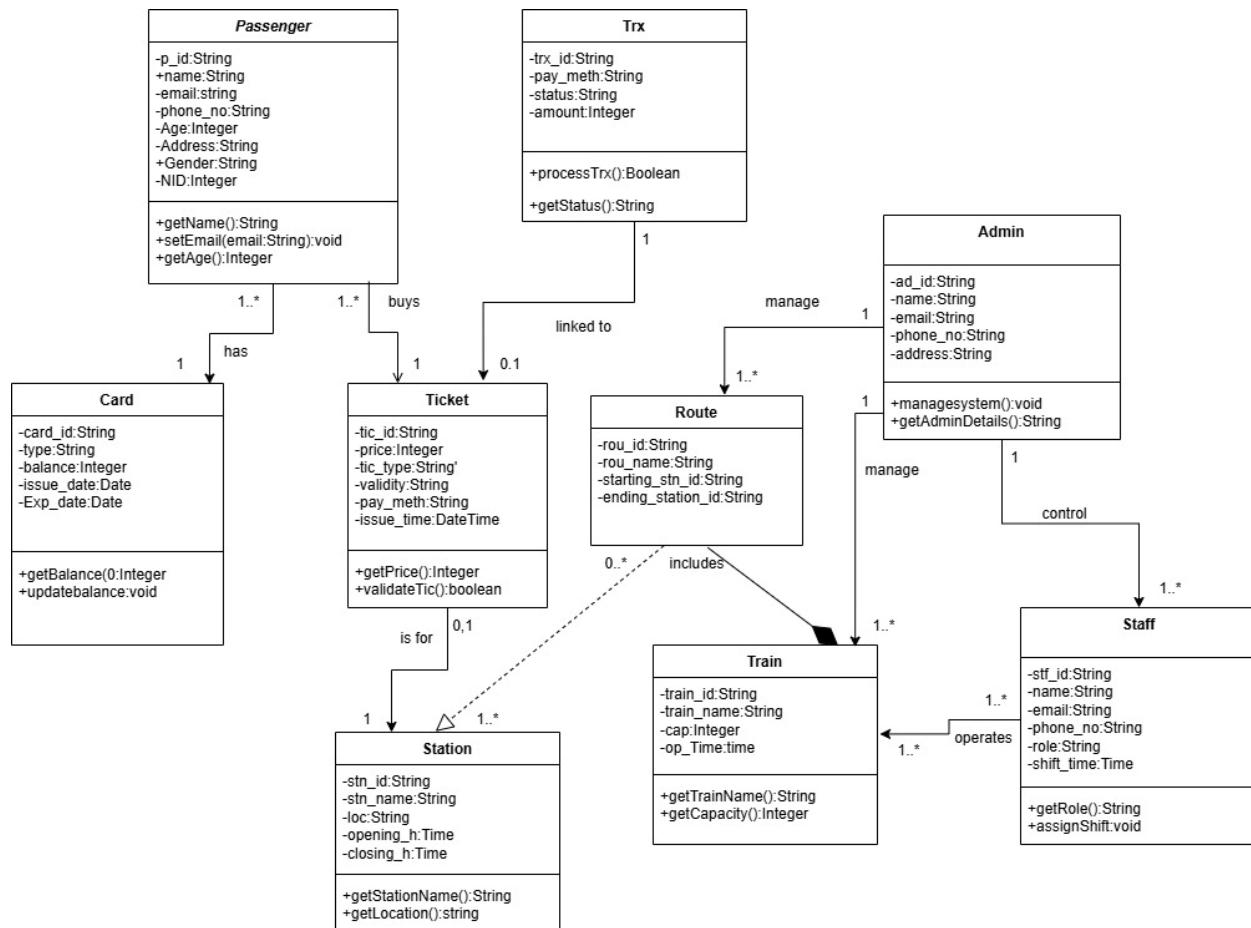
06. train_id, train_name, rou_id, cap, op_tym

07. staff_id, name, role, phn_no, email, shift_tym, assigned_sttn_id

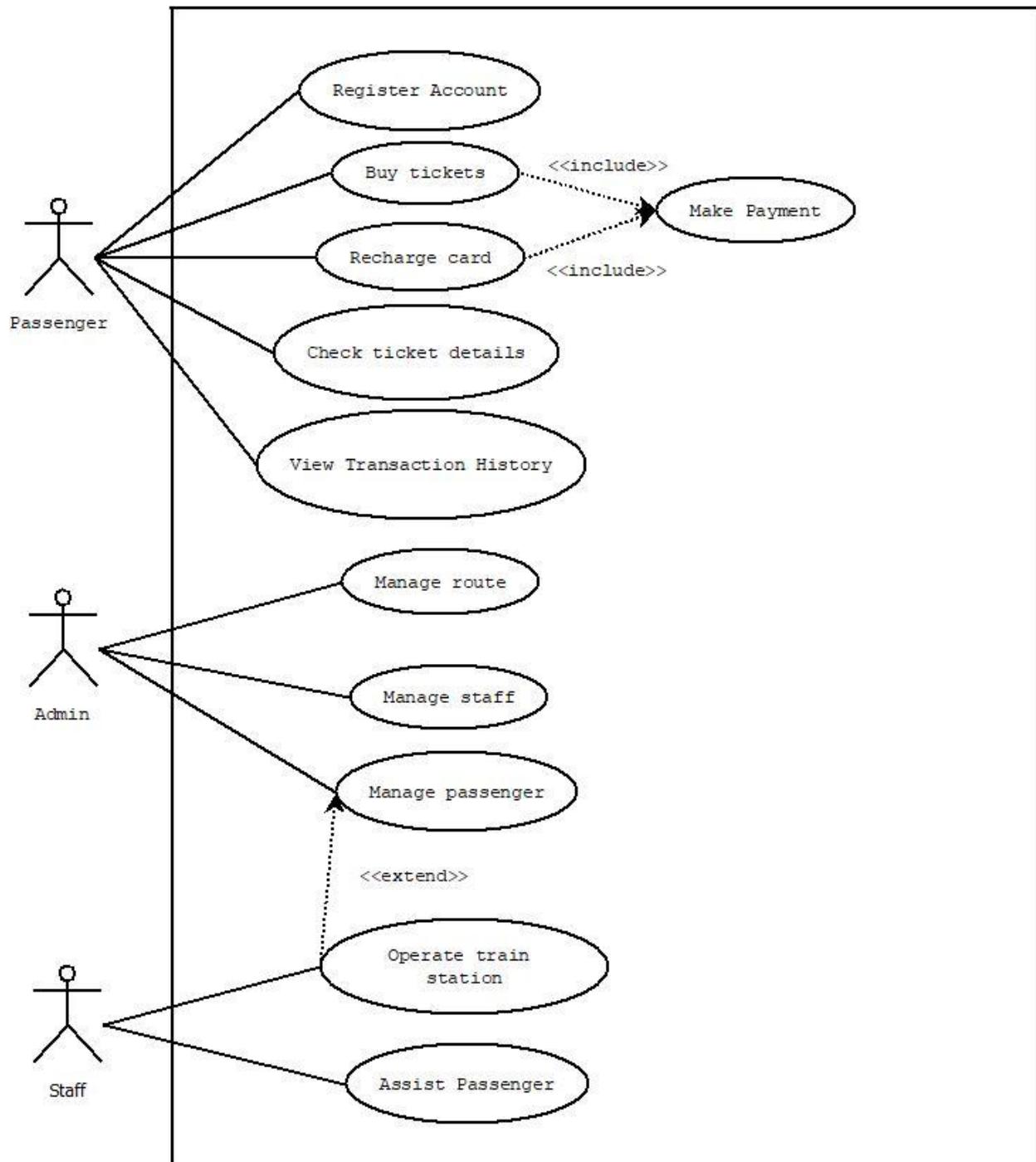
08. card_id, type, balance, issue_date, exp_date, p_id

09. ad_id, name, phn_no, email, addrs, nid

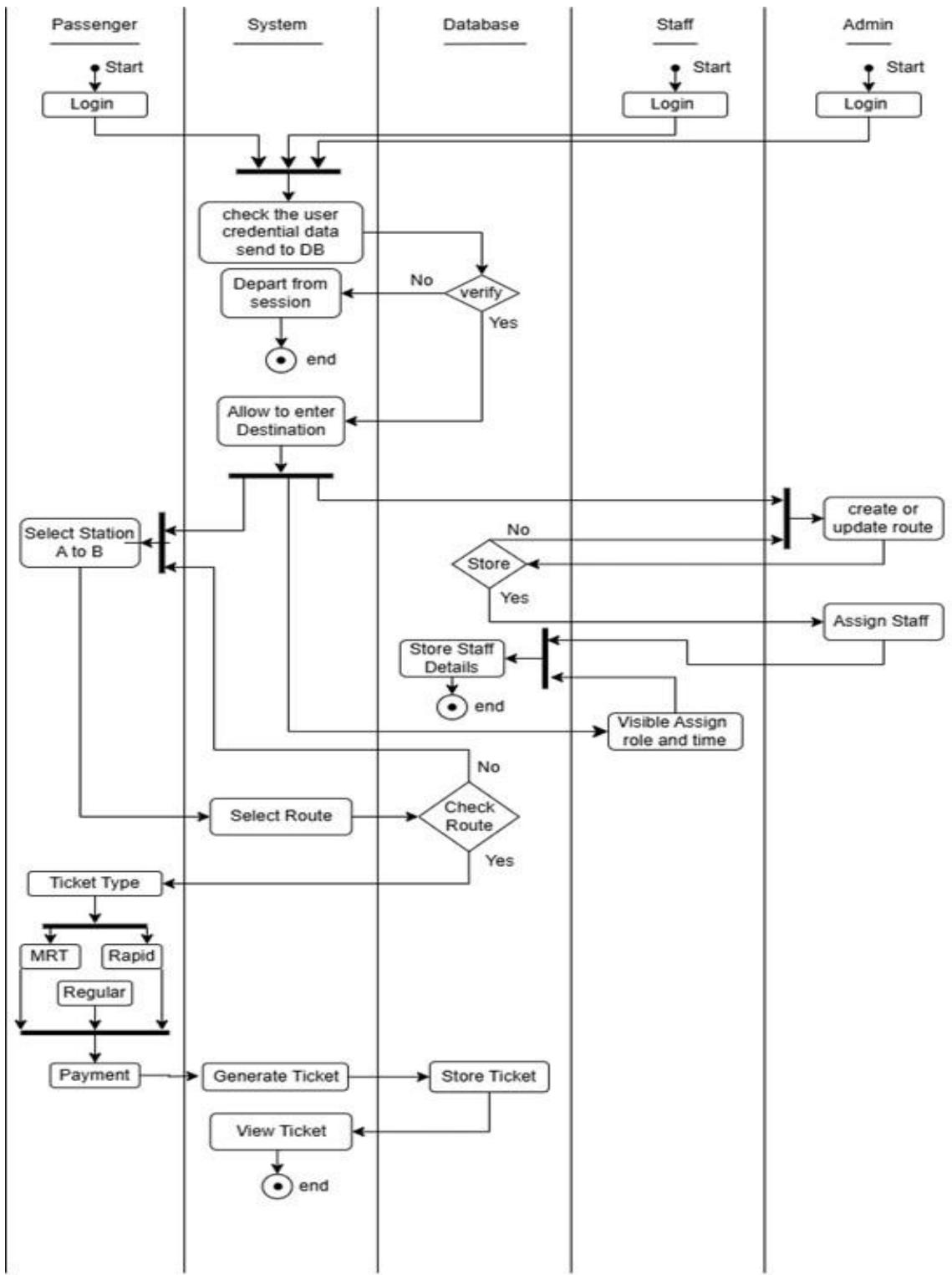
Class diagram:



Use Case Diagram:



Activity Diagram:



Schema Diagram:

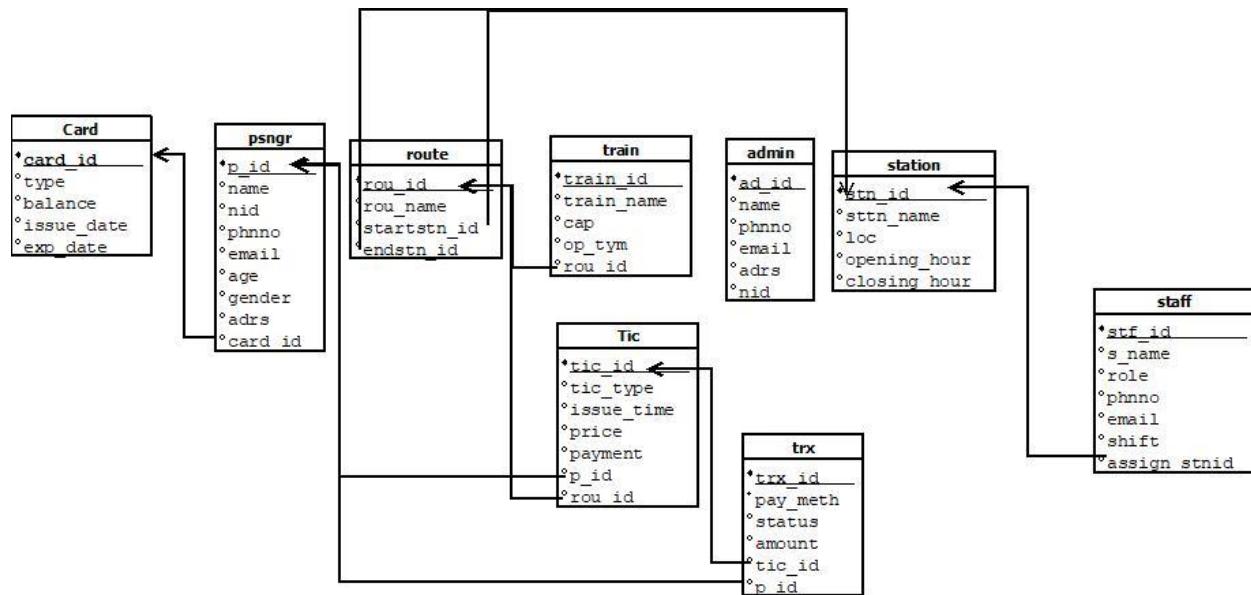


TABLE CREATION

1. Card:

create table Card(card_id varchar2(20) not null primary key, type varchar2(200), balance int, issue_date Date, exp_date date);

ORACLE Database Express Edition

User SCOTT

Home > SQL > SQL Commands

Autocommit Display 20 Save Run

```

create table Card(card_id varchar2(20) not null primary key, type varchar2(200), balance int, issue_date Date, exp_date date);
insert into Card values('001', 'MRT', 2000, to_date('2024-11-27', 'YYYY-MM-DD'), to_date('2025-11-27', 'YYYY-MM-DD'));

select * from card

describe passenger

create table psngr(pid varchar(20) not null primary key, name varchar2(50), nid int, phnno int, email Varchar2(50), age int, gender varchar2(50), adrs
varchar2(100), card_id varchar2(20), foreign key (card_id) references card(card_id));
  
```

Results Explain Describe Saved SQL History

Table created.

0.02 seconds

Language: en-us Application Express 2.1.0.0.39 Copyright © 1999, 2006, Oracle. All rights reserved.

AW01 -0.28% 11:34 PM 12/10/2024

2. Psngr:

create table psngr(p_id varchar(20) not null primary key, name varchar2(50), nid int, phnno int, email Varchar2(50) ,age int, gender varchar2(50), adrs varchar2(100),card_id varchar2(20), foreign key (card_id) references card(card_id));

The screenshot shows the Oracle Database Express Edition interface. The SQL Commands window contains the following SQL code:

```
select * from card
describe passenger

create table psngr(p_id varchar(20) not null primary key, name varchar2(50), nid int, phnno int, email Varchar2(50) ,age int, gender varchar2(50), adrs varchar2(100),card_id varchar2(20), foreign key (card_id) references card(card_id));

insert into passenger values('p001', 'asir', 200089087, 165243789, 'asir@gmail.com', 24, 'Male', 'dhaka', '001'
to_date('2024-11-27', 'YYYY-MM-DD'), to_date('2025-11-27', 'YYYY-MM-DD'))
```

The results pane shows the message "Table created." and a execution time of "0.11 seconds". The status bar at the bottom right indicates "Application Express 2.1.0.0.39" and the date "12/10/2024".

3. Ticket:

CREATE TABLE Tic (tic_id varchar2(20) not null PRIMARY KEY, tic_type VARCHAR2(50), issue_tym TIMESTAMP, price int, payment VARCHAR2(50),p_id varchar(20), foreign key (p_id) REFERENCES psngr(p_id));

The screenshot shows the Oracle Database Express Edition interface. The SQL Commands window contains the following SQL code:

```
CREATE TABLE Tic_id (
    tic_id varchar2(20) not null PRIMARY KEY,
    tic_type VARCHAR2(50),
    issue_tym TIMESTAMP,
    price int,
    payment VARCHAR2(50),
    p_id varchar(20),
    foreign key (p_id) REFERENCES psngr(p_id)
);

create table trx(trx_id varchar(20) not null primary key, pay_meth VARCHAR2(200),status VARCHAR2(200),amount int,tic, foreign key (card_id) references card(card_id));
```

The results pane shows the message "Table created." and a execution time of "0.12 seconds". The status bar at the bottom right indicates "Application Express 2.1.0.0.39" and the date "12/10/2024".

4. Transaction:

```
create table trx(trx_id varchar(20) not null primary key, pay_meth VARCHAR2(200),status
VARCHAR2(200),amount int, tic_id varchar2(20), p_id varchar2(20), foreign key (tic_id)
references tic(tic_id), foreign key (p_id) references psngr(p_id));
```

The screenshot shows the Oracle Database Express Edition interface. The SQL command window contains the following code:

```
create table trx(trx_id varchar(20) not null primary key, pay_meth VARCHAR2(200),status
VARCHAR2(200),amount int, tic_id varchar2(20), p_id varchar2(20), foreign key (tic_id)
references tic(tic_id), foreign key (p_id) references psngr(p_id));
create table station(stn_id varchar(20) not null primary key, sttn_name VARCHAR2(100), loc VARCHAR2(255), closing_hour TIMESTAMP, opening_h TIMESTAMP);
```

Below the code, the results show "Table created." and "0.03 seconds". The status bar at the bottom right indicates "Application Express 2.1.0.0.39" and "Copyright © 1999, 2006, Oracle. All rights reserved."

5. Station:

```
create table station(stn_id varchar(20) not null primary key, sttn_name VARCHAR2(100), loc
VARCHAR2(255), closing_hour TIMESTAMP, opening_h TIMESTAMP);
```

The screenshot shows the Oracle Database Express Edition interface. The SQL command window contains the following code:

```
create table trx(trx_id varchar(20) not null primary key, pay_meth VARCHAR2(200),status
VARCHAR2(200),amount int, tic_id varchar2(20), p_id NUMBER REFERENCES psngr(p_id));
create table station(stn_id varchar(20) not null primary key, sttn_name VARCHAR2(100), loc VARCHAR2(255), closing_hour TIMESTAMP, opening_h TIMESTAMP);
```

Below the code, the results show "Table created." and "0.05 seconds". The status bar at the bottom right indicates "Application Express 2.1.0.0.39" and "Copyright © 1999, 2006, Oracle. All rights reserved."

6. Route:

create table route(rou_id varchar(20) not null primary key, rou_name VARCHAR2(200), startstn_id varchar2(20), endstn_id varchar2(20), foreign key (startstn_id) references station(stn_id), foreign key (endstn_id) references station(stn_id));

The screenshot shows the Oracle Database Express Edition interface. The SQL Commands window contains the following code:

```
create table route(rou_id varchar(20) not null primary key, rou_name VARCHAR2(200), startstn_id varchar2(20), endstn_id varchar2(20), foreign key (startstn_id) references station(stn_id), foreign key (endstn_id) references station(stn_id));  
create table trx(trx_id varchar(20) not null primary key, pay_meth VARCHAR2(200), status VARCHAR2(200), amount int, tic_id varchar2(20), p_id varchar2(20), foreign key (tic_id) references tic(tic_id), foreign key (p_id) references pang(p_id));  
create table station(stn_id varchar(20) not null primary key, stn_name VARCHAR2(100), loc VARCHAR2(255), closing_hour TIMESTAMP, opening_h TIMESTAMP);  
  
create table route(rou_id varchar(20) not null primary key, rou_name VARCHAR2(200), startstn_id varchar2(20), endstn_id varchar2(20), foreign key (startstn_id) references station(stn_id), foreign key (endstn_id) references station(stn_id));  
train_id train_name, rou_id cap op_tym  
stf_id stf_name role phno email shift assign_stnid
```

The results section shows "Table created." and "0.04 seconds". The browser status bar indicates "Application Express 2.1.0.0.39 Copyright © 1999, 2006, Oracle. All rights reserved."

7. Train:

create table train(train_id varchar(20) not null primary key, train_name varchar(200), rou_id varchar(20), cap int, op_tym varchar2(200), foreign key (rou_id) references route(rou_id));

The screenshot shows the Oracle Database Express Edition interface. The SQL Commands window contains the following code:

```
create table route(rou_id varchar(20) not null primary key, rou_name VARCHAR2(200), startstn_id varchar2(20), endstn_id varchar2(20), foreign key (startstn_id) references station(stn_id), foreign key (endstn_id) references station(stn_id));  
create table train(train_id varchar(20) not null primary key, train_name varchar(200), rou_id varchar(20), cap int, op_tym timestamp, foreign key (rou_id) references route(rou_id));  
create table staff(stf_id varchar(20) not null primary key, stf_name role phno email shift assign_stnid  
create table admin(ad_id varchar(20) not null primary key name phno email add_nid
```

The results section shows "Table created." and "0.14 seconds". The browser status bar indicates "Application Express 2.1.0.0.39 Copyright © 1999, 2006, Oracle. All rights reserved."

8. Staff:

```
create table staff(stf_id varchar(20) not null primary key,s_name varchar(200),role varchar(200),
phnno int,email varchar(200),shift varchar(200), assign_stnid varchar(20),foreign key
(assign_stnid) references station(stn_id));
```

The screenshot shows the Oracle Database Express Edition interface. The SQL Commands window contains the following SQL code:

```
create table route(rou_id varchar(20) not null primary key, rou_name VARCHAR2(200), startstn_id varchar2(20), endstn_id varchar2(20),foreign key (startstn_id)
references station(stn_id), foreign key (endstn_id) references station(stn_id));

create table train(train_id varchar(20) not null primary key, train_name varchar(200) ,rou_id varchar(20) ,cap int, op_tym timestamp, foreign key (rou_id)
references route(rou_id));

create table staff(stf_id varchar(20) not null primary key,s_name varchar(200),role varchar(200), phnno int,email varchar(200),shift varchar(200), assign_stnid
varchar(20),foreign key (assign_stnid) references station(stn_id));

create table admin(ad_id varchar(20) not null primary key name varchar(200), phnno email add nid
```

The results section shows "Table created." and a execution time of "0.05 seconds". The status bar at the bottom right indicates "Application Express 2.1.0.0.39" and "Copyright © 1999, 2006, Oracle. All rights reserved."

9. Admin:

```
create table admin(ad_id varchar(20) not null primary key, name varchar2(200), phnno int, email
varchar2(200), adrs varchar2(200), nid int);
```

The screenshot shows the Oracle Database Express Edition interface. The SQL Commands window contains the same SQL code as the previous screenshot, but with a different execution time:

```
create table route(rou_id varchar(20) not null primary key, rou_name VARCHAR2(200), startstn_id varchar2(20), endstn_id varchar2(20),foreign key (startstn_id)
references station(stn_id), foreign key (endstn_id) references station(stn_id));

create table train(train_id varchar(20) not null primary key, train_name varchar(200) ,rou_id varchar(20) ,cap int, op_tym timestamp, foreign key (rou_id)
references route(rou_id));

create table staff(stf_id varchar(20) not null primary key,s_name varchar(200),role varchar(200), phnno int,email varchar(200),shift varchar(200), assign_stnid
varchar(20),foreign key (assign_stnid) references station(stn_id));

create table admin(ad_id varchar(20) not null primary key name varchar(200), phnno int, email varchar2(200), adrs varchar2(200),nid int);
```

The results section shows "Table created." and a execution time of "0.10 seconds". The status bar at the bottom right indicates "Application Express 2.1.0.0.39" and "Copyright © 1999, 2006, Oracle. All rights reserved."

Describe Table

1. card

The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands window, the following SQL code is run:

```
insert into Card values(001, 'MRT', 2000, to_date('2024-11-27', 'YYYY-MM-DD'), to_date('2025-11-27', 'YYYY-MM-DD'))  
insert into Card values(001, 'MRT', 2000, to_date('2024-11-27', 'YYYY-MM-DD'), to_date('2025-11-27', 'YYYY-MM-DD'))  
insert into Card values(001, 'MRT', 2000, to_date('2024-11-27', 'YYYY-MM-DD'), to_date('2025-11-27', 'YYYY-MM-DD'))  
insert into Card values(001, 'MRT', 2000, to_date('2024-11-27', 'YYYY-MM-DD'), to_date('2025-11-27', 'YYYY-MM-DD'))  
  
select * from card  
  
describe card  
  
create table psngr(p_id varchar(20) not null primary key, name varchar2(50), nid int, phnno int, email Varchar2(50) ,age int, gender varchar2(50), addrs  
varchar2(100) card_id varchar2(20) foreign key (card_id) references card(card_id));
```

The results show the description of the CARD table:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CARD	CARD_ID	Varchar2	20	-	-	1	-	-	-
	TYPE	Varchar2	200	-	-	-	✓	-	-
	BALANCE	Number	-	-	0	-	✓	-	-
	ISSUE_DATE	Date	7	-	-	-	✓	-	-
	EXP_DATE	Date	7	-	-	-	✓	-	-

Language: en-us Application Express 2.1.0.0.39 Copyright © 1999, 2006, Oracle. All rights reserved. AW01 -0.32% 12:33 AM 12/11/2024

2. Psngr

The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands window, the following SQL code is run:

```
describe psngr  
  
create table psngr(p_id varchar(20) not null primary key, name varchar2(50), nid int, phnno int, email Varchar2(50) ,age int, gender varchar2(50), addrs  
varchar2(100) card_id varchar2(20) foreign key (card_id) references card(card_id));
```

The results show the description of the PSNGR table:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PSNGR	P_ID	Varchar2	20	-	-	1	-	-	-
	NAME	Varchar2	50	-	-	-	✓	-	-
	NID	Number	-	-	0	-	✓	-	-
	PHNNO	Number	-	-	0	-	✓	-	-
	EMAIL	Varchar2	50	-	-	-	✓	-	-
	AGE	Number	-	-	0	-	✓	-	-
	GENDER	Varchar2	50	-	-	-	✓	-	-
	ADRS	Varchar2	100	-	-	-	✓	-	-
	CARD_ID	Varchar2	20	-	-	-	✓	-	-

Language: en-us Application Express 2.1.0.0.39 Copyright © 1999, 2006, Oracle. All rights reserved. AW01 -0.32% 12:34 AM 12/11/2024

3. tic

The screenshot shows the Oracle Application Express SQL Commands interface. The URL is 127.0.0.1:8080/apex/f?p=4500:1003:1694083273625485:NO:::1. The page displays SQL code for creating the 'tic' table and inserting five rows of data. Below the code is a table showing the columns and data types of the 'tic' table.

```
9.
create table admin(ad_id varchar(20) not null primary key, name varchar2(200), phnno int, email varchar2(200), addrs varchar2(200), nid int);

insert into admin values ('ad001', 'onu', 17139598545, 'onu01@gmail.com', 'dhaka', 200440050085)
insert into admin values ('ad002', 'alif', 17139545545, 'alif@gmail.com', 'pabna', 330400050085)
insert into admin values ('ad003', 'monu', 17139598845, 'monu01@gmail.com', 'dhaka', 444400050085)
insert into admin values ('ad004', 'sonu', 17139598565, 'sonu01@gmail.com', 'ctg', 200444500885)
insert into admin values ('ad005', 'amir', 17139598555, 'amir@gmail.com', 'dhaka', 900400050085)

desc tic
```

Object Type	TABLE	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
	TIC	TIC_ID	Varchar2	20	-	-	1	-	-	-
	TIC	TIC_TYPE	Varchar2	50	-	-	-	✓	-	-
	TIC	ISSUE_TYM	Timestamp(6)	11	-	6	-	✓	-	-
	TIC	PRICE	Number	-	-	0	-	✓	-	-
	TIC	PAYMENT	Varchar2	50	-	-	-	✓	-	-
	TIC	P_ID	Varchar2	20	-	-	-	✓	-	-

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

4. trx

The screenshot shows the Oracle Application Express SQL Commands interface. The URL is 127.0.0.1:8080/apex/f?p=4500:1003:1694083273625485:NO:::1. The page displays SQL code for creating the 'trx' table and inserting five rows of data. Below the code is a table showing the columns and data types of the 'trx' table.

```
9.
create table admin(ad_id varchar(20) not null primary key, name varchar2(200), phnno int, email varchar2(200), addrs varchar2(200), nid int);

insert into admin values ('ad001', 'onu', 17139598545, 'onu01@gmail.com', 'dhaka', 200440050085)
insert into admin values ('ad002', 'alif', 17139545545, 'alif@gmail.com', 'pabna', 330400050085)
insert into admin values ('ad003', 'monu', 17139598845, 'monu01@gmail.com', 'dhaka', 444400050085)
insert into admin values ('ad004', 'sonu', 17139598565, 'sonu01@gmail.com', 'ctg', 200444500885)
insert into admin values ('ad005', 'amir', 17139598555, 'amir@gmail.com', 'dhaka', 900400050085)

desc trx
```

Object Type	TABLE	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
	TRX	TRX_ID	Varchar2	20	-	-	1	-	-	-
	TRX	PAY_METH	Varchar2	200	-	-	-	✓	-	-
	TRX	STATUS	Varchar2	200	-	-	-	✓	-	-
	TRX	AMOUNT	Number	-	-	0	-	✓	-	-
	TRX	TIC_ID	Varchar2	20	-	-	-	✓	-	-
	TRX	P_ID	Varchar2	20	-	-	-	✓	-	-

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

5. staff

The screenshot shows the Oracle Application Express interface. In the top-left, the URL is 127.0.0.1:8080/apex/f?p=4500:1003:1694083273625485::NO. The top-right has Save and Run buttons. The main area contains SQL code for creating the STAFF table and inserting five rows of data. Below the code is a 'desc staff' command. A results grid shows the columns and data for the STAFF table. At the bottom, it says Language: en-us and Application Express 2.1.0.0.39 Copyright © 1999, 2006, Oracle. All rights reserved.

```

9.
create table admin(ad_id varchar(20) not null primary key, name varchar2(200), phnno int, email varchar2(200), addrs varchar2(200), nid int);

insert into admin values ('ad001', 'onu', 17139598545, 'onu01@gmail.com', 'dhaka', 200400050085);
insert into admin values ('ad002', 'alif', 17139545545, 'alif@gmail.com', 'pabna', 330400050085);
insert into admin values ('ad003', 'monu', 17139598845, 'monu01@gmail.com', 'dhaka', 444400050085);
insert into admin values ('ad004', 'sonu', 17139598565, 'sonu01@gmail.com', 'ctg', 200444500085);
insert into admin values ('ad005', 'amir', 17139598555, 'amir@gmail.com', 'dhaka', 900400050085);

desc staff

```

Object Type	Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
STAFF	STAFF_ID	VARCHAR2	20	-	-	-	1	-	-	-
	S_NAME	VARCHAR2	200	-	-	-	-	✓	-	-
	ROLE	VARCHAR2	200	-	-	-	-	✓	-	-
	PHNNO	NUMBER	-	-	0	-	-	✓	-	-
	EMAIL	VARCHAR2	200	-	-	-	-	✓	-	-
	SHIFT	VARCHAR2	200	-	-	-	-	✓	-	-
	ASSIGN_STAFFID	VARCHAR2	20	-	-	-	-	✓	-	-

1 - 7

Language: en-us Application Express 2.1.0.0.39 Copyright © 1999, 2006, Oracle. All rights reserved.

6. route

The screenshot shows the Oracle Application Express interface. In the top-left, the URL is 127.0.0.1:8080/apex/f?p=4500:1003:1694083273625485::NO. The top-right has Home, Logout, and Help buttons. The main area contains SQL code for creating the ROUTE table and inserting five rows of data. Below the code is a 'desc route' command. A results grid shows the columns and data for the ROUTE table. At the bottom, it says Language: en-us and Application Express 2.1.0.0.39 Copyright © 1999, 2006, Oracle. All rights reserved.

```

9.
create table admin(ad_id varchar(20) not null primary key, name varchar2(200), phnno int, email varchar2(200), addrs varchar2(200), nid int);

insert into admin values ('ad001', 'onu', 17139598545, 'onu01@gmail.com', 'dhaka', 200400050085);
insert into admin values ('ad002', 'alif', 17139545545, 'alif@gmail.com', 'pabna', 330400050085);
insert into admin values ('ad003', 'monu', 17139598845, 'monu01@gmail.com', 'dhaka', 444400050085);
insert into admin values ('ad004', 'sonu', 17139598565, 'sonu01@gmail.com', 'ctg', 200444500085);
insert into admin values ('ad005', 'amir', 17139598555, 'amir@gmail.com', 'dhaka', 900400050085);

desc route

```

Object Type	Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ROUTE	ROUT_ID	VARCHAR2	20	-	-	-	1	-	-	-
	ROUT_NAME	VARCHAR2	200	-	-	-	-	✓	-	-
	STARTSTN_ID	VARCHAR2	20	-	-	-	-	✓	-	-
	ENDSTN_ID	VARCHAR2	20	-	-	-	-	✓	-	-

1 - 4

Language: en-us Application Express 2.1.0.0.39 Copyright © 1999, 2006, Oracle. All rights reserved.

7. station

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The user is SCOTT. The code entered is:

```
9.
create table admin(ad_id varchar(20) not null primary key, name varchar2(200), phnno int, email varchar2(200), addrs varchar2(200), did int);

insert into admin values ('ad001', 'onu', 17139598545, 'onu01@gmail.com', 'dhaka', 20040050085)
insert into admin values ('ad002', 'alif', 17139545545, 'alif@gmail.com', 'pabna', 33040050085)
insert into admin values ('ad003', 'monu', 17139598845, 'monu01@gmail.com', 'dhaka', 44440050085)
insert into admin values ('ad004', 'sonu', 17139598565, 'sonu01@gmail.com', 'ctg', 20044450085)
insert into admin values ('ad005', 'amir', 17139598555, 'amir@gmail.com', 'dhaka', 90040050085)

desc station
```

The results show the creation of the STATION table with the following structure:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
STATION	STN_ID	Varchar2	20	-	-	1	-	-	-
STATION	STN_NAME	Varchar2	100	-	-	-	✓	-	-
STATION	LOC	Varchar2	255	-	-	-	✓	-	-
STATION	CLOSING_HOUR	Timestamp(6)	11	-	6	-	✓	-	-
STATION	OPENING_H	Timestamp(6)	11	-	6	-	✓	-	-

Language: en_us Application Express 2.1 00.39 Copyright © 1999, 2008, Oracle. All rights reserved.

8. train

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The user is SCOTT. The code entered is:

```
9.
create table admin(ad_id varchar(20) not null primary key, name varchar2(200), phnno int, email varchar2(200), addrs varchar2(200), did int);

insert into admin values ('ad001', 'onu', 17139598545, 'onu01@gmail.com', 'dhaka', 20040050085)
insert into admin values ('ad002', 'alif', 17139545545, 'alif@gmail.com', 'pabna', 33040050085)
insert into admin values ('ad003', 'monu', 17139598845, 'monu01@gmail.com', 'dhaka', 44440050085)
insert into admin values ('ad004', 'sonu', 17139598565, 'sonu01@gmail.com', 'ctg', 20044450085)
insert into admin values ('ad005', 'amir', 17139598555, 'amir@gmail.com', 'dhaka', 90040050085)

desc train
```

The results show the creation of the TRAIN table with the following structure:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
TRAIN	TRAIN_ID	Varchar2	20	-	-	1	-	-	-
TRAIN	TRAIN_NAME	Varchar2	200	-	-	-	✓	-	-
TRAIN	RQU_ID	Varchar2	20	-	-	-	✓	-	-
TRAIN	CAP	Number	-	-	0	-	✓	-	-
TRAIN	OP_TYM	Varchar2	200	-	-	-	✓	-	-

Language: en_us Application Express 2.1 00.39 Copyright © 1999, 2008, Oracle. All rights reserved.

9. admin

The screenshot shows the Oracle Database Express Edition Application Express interface. The SQL Commands page is open, showing the creation of a table 'admin' and its data insertion. The table has columns: AD_ID, NAME, PHNNO, EMAIL, ADRS, and NID. The data inserted includes rows for 'ad001' through 'ad005' with various names, phone numbers, emails, addresses, and IDs.

```
create table admin(ad_id varchar(20) not null primary key, name varchar2(200), phnno int, email varchar2(200), adrs varchar2(200), nid int);
insert into admin values ('ad001', 'enu', 17139598545, 'enu01@gmail.com', 'dhaka', 20040050085);
insert into admin values ('ad002', 'alif', 17139545545, 'alif@gmail.com', 'dhaka', 33040050085);
insert into admin values ('ad003', 'monu', 17139598845, 'monu01@gmail.com', 'dhaka', 44440050085);
insert into admin values ('ad004', 'sonu', 17139598565, 'sonu01@gmail.com', 'csg', 20044450085);
insert into admin values ('ad005', 'amir', 17139598555, 'amir@gmail.com', 'dhaka', 90040050085);

desc admin
```

Below the SQL window, the object list shows the 'ADMIN' table with its columns and data types:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ADMIN	AD_ID	VARCHAR2	20	-	-	1	-	-	-
ADMIN	NAME	VARCHAR2	200	-	-	-	✓	-	-
ADMIN	PHNNO	NUMBER	-	-	0	-	✓	-	-
ADMIN	EMAIL	VARCHAR2	200	-	-	-	✓	-	-
ADMIN	ADRS	VARCHAR2	200	-	-	-	✓	-	-
ADMIN	NID	NUMBER	-	-	0	-	✓	-	-

DATA INSERTION

1. CARD:

```
insert into Card values(001, 'MRT' , 2000, to_date('2024-11-27' , 'YYYY-MM-DD') ,to_date('2025-11-27' , 'YYYY-MM-DD'))
insert into Card values(002, 'RAPID' , 1000, to_date('2024-10-27' , 'YYYY-MM-DD')
,to_date('2025-10-27' , 'YYYY-MM-DD'))
insert into Card values(003, 'REGULAR' , 0, to_date('2024-12-27' , 'YYYY-MM-DD')
,to_date('2024-12-28' , 'YYYY-MM-DD'))
insert into Card values(004, 'REGULAR' , 0, to_date('2024-09-27' , 'YYYY-MM-DD')
,to_date('2024-09-28' , 'YYYY-MM-DD'))
insert into Card values(005, 'MRT' , 500, to_date('2024-10-01' , 'YYYY-MM-DD') ,to_date('2025-10-01' , 'YYYY-MM-DD'))
```

```

create table Card(card_id varchar2(20) not null primary key, type varchar2(200), balance int, issue_date Date, exp_date date);
insert into Card values(001, 'MRT', 2000, to_date('2024-11-27', 'YYYY-MM-DD'), to_date('2025-11-27', 'YYYY-MM-DD'));
insert into Card values(002, 'RAPID', 1000, to_date('2024-10-27', 'YYYY-MM-DD'), to_date('2025-10-27', 'YYYY-MM-DD'));
insert into Card values(003, 'REGULAR', 0, to_date('2024-12-27', 'YYYY-MM-DD'), to_date('2024-12-28', 'YYYY-MM-DD'));
insert into Card values(004, 'REGULAR', 0, to_date('2024-09-27', 'YYYY-MM-DD'), to_date('2024-09-28', 'YYYY-MM-DD'));
insert into Card values(005, 'MRT', 500, to_date('2024-10-01', 'YYYY-MM-DD'), to_date('2025-10-01', 'YYYY-MM-DD'));

select * from card

```

Results

CARD_ID	TYPE	BALANCE	ISSUE_DATE	EXP_DATE
1	MRT	2000	27-NOV-24	27-NOV-25
2	RAPID	1000	27-OCT-24	27-OCT-25
3	REGULAR	0	27-DEC-24	28-DEC-24
4	REGULAR	0	27-SEP-24	28-SEP-24
5	MRT	500	01-OCT-24	01-OCT-25

5 rows returned in 0.03 seconds CSV Export

Application Express 2.1.0.00.39
Copyright © 1999, 2006, Oracle. All rights reserved.

2. Psngr:

insert into psngr values('p001', 'Asir', 200089087, 165243789, 'asir@gmail.com', 24, 'Male', 'dhaka', '001)
 insert into psngr values('p002', 'Afsana', 3500089087, 178243789, 'afsana@gmail.com', 22, 'female', 'pabna', '002)
 insert into psngr values('p003', 'Suma', 450089087, 165993789, 'suma@gmail.com', 24, 'female', 'dhaka', '003)
 insert into psngr values('p004', 'Preo', 280089087, 165263789, 'preo@gmail.com', 21, 'female', 'kst', '004)
 insert into psngr values('p005', 'Isty', 500089087, 177773789, 'isty@gmail.com', 20, 'Male', 'ctg', '005)

```

create table psngr(p_id varchar(20) not null primary key, name varchar2(50), nid int, phnno int, email Varchar2(50), age int, gender varchar2(50), adrs varchar2(100), card_id varchar2(20), foreign key (card_id) references card(card_id));

insert into psngr values('p001', 'Asir', 200089087, 165243789, 'asir@gmail.com', 24, 'Male', 'dhaka', '001)
insert into psngr values('p002', 'Afsana', 3500089087, 178243789, 'afsana@gmail.com', 22, 'female', 'pabna', '002')
insert into psngr values('p003', 'Suma', 450089087, 165993789, 'suma@gmail.com', 24, 'female', 'dhaka', '003')
insert into psngr values('p004', 'Preo', 280089087, 165263789, 'preo@gmail.com', 21, 'female', 'kst', '004')
insert into psngr values('p005', 'Isty', 500089087, 177773789, 'isty@gmail.com', 20, 'Male', 'ctg', '005')

```

Results

P_ID	NAME	NID	PHNNO	EMAIL	AGE	GENDER	ADRS	CARD_ID
p001	Asir	200089087	165243789	asir@gmail.com	24	Male	dhaka	1
p002	Afsana	3500089087	178243789	afsana@gmail.com	22	female	pabna	2
p003	Suma	450089087	165993789	suma@gmail.com	24	female	dhaka	3
p004	Preo	280089087	165263789	preo@gmail.com	21	female	kst	4
p005	Isty	500089087	177773789	isty@gmail.com	20	Male	ctg	5

5 rows returned in 0.00 seconds CSV Export

Application Express 2.1.0.00.39
Copyright © 1999, 2006, Oracle. All rights reserved.

3. TIC:

```
insert into TIC values('t001', 'MRT' , TIMESTAMP '2024-08-09 14:30:00',100, 'BKASH' ,p001)
insert into tic values('t002', 'RAPID' ,TIMESTAMP '2024-09-09 15:30:00',100, 'BKASH' ,p002)
insert into tic values('t003', 'REGULAR' , TIMESTAMP '2024-10-09 12:35:00',100, 'CASH'
,'p003')
insert into tic values('t004', 'REGULAR' , TIMESTAMP '2024-12-09 10:40:00',100, 'CASH'
,'p004')
insert into tic values('t005', 'MRT' , TIMESTAMP '2024-11-09 16:43:00',100, 'BKASH' ,p005)
```

The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands window, the following SQL code is entered:

```
select * from tic;
drop table admin
SELECT table_name
FROM all_tables;
CREATE TABLE Tic (
tic_id varchar2(28) not null PRIMARY KEY,
tic_type VARCHAR2(58),
```

The Results window displays the data from the 'tic' table:

TIC_ID	TIC_TYPE	ISSUE_TYM	PRICE	PAYMENT	P_ID
t001	MRT	09-AUG-24 02.30.00.000000 PM	100	BKASH	p001
t002	RAPID	09-SEP-24 03.30.00.000000 PM	100	BKASH	p002
t003	REGULAR	09-OCT-24 12.35.00.000000 PM	100	CASH	p003
t004	REGULAR	09-DEC-24 10.40.00.000000 AM	100	CASH	p004
t005	MRT	09-NOV-24 04.43.00.000000 PM	100	BKASH	p005

5 rows returned in 0.02 seconds

CSV Export

Language: en-us Application Express 2.1.0.0.39 Copyright © 1999-2006, Oracle. All rights reserved.

4. Staff:

```
insert into staff values ('stf001' , 'fariha', 'manager' ,18520554441 , 'fariha@gmail.com' , 'morning'
,'s001')
insert into staff values ('stf002' , 'ariha', 'crew' ,18520554442 , 'ariha@gmail.com' , 'morning' ,
's002')
insert into staff values ('stf003' , 'riha', 'operator' ,18520554443 , 'riha@gmail.com' , 'morning' ,
's003')
insert into staff values ('stf004' , 'dia', 'crew' ,18520554541 , 'dia@gmail.com' , 'evening' , 's004')
insert into staff values ('stf005' , 'fari', 'crew' ,18520555541 , 'fari@gmail.com' , 'evening' , 's005')
```

User: SCOTT

Home > SQL > SQL Commands

Autocommit

```
9.
create table admin(ad_id varchar(20) not null primary key, name varchar(200), phnno int, email varchar(200), adrs varchar(200), did int);

insert into admin values ('ad001', 'oni', 17139598545, 'oni01@gmail.com', 'dhaka', 20040050085);
insert into admin values ('ad002', 'aliif', 17139545545, 'aliif@gmail.com', 'dhaka', 33040050085);
insert into admin values ('ad003', 'monu', 17139598845, 'monu01@gmail.com', 'dhaka', 44440050085);
insert into admin values ('ad004', 'sonu', 17139598565, 'sonu01@gmail.com', 'Ctg', 20044450085);
insert into admin values ('ad005', 'amir', 17139598555, 'amir@gmail.com', 'dhaka', 90040050085);

select * from staff
```

Results Explain Describe Saved SQL History

STF_ID	S_NAME	ROLE	PHNNO	EMAIL	SHIFT	ASSIGN_STNID
stf001	farha	manager	1852054441	farha@gmail.com	morning	s001
stf002	ariha	crew	1852054442	ariha@gmail.com	morning	s002
stf003	riha	operator	1852054443	rha@gmail.com	morning	s003
stf004	dia	crew	1852054541	dia@gmail.com	evening	s004
stf005	fan	crew	18520555541	fan@gmail.com	evening	s005

5 rows returned in 0.00 seconds

Application Express 2.1 00.39
Copyright © 1999, 2006, Oracle. All rights reserved.

Language: en-us

Windows Taskbar: Type here to search, Start button, File Explorer, Internet Explorer, Google, Firefox, Chrome, Microsoft Edge, File, Network, Task View, Taskbar settings, System tray: USD/EUR +0.20%, 7:22 PM, 12/11/2024

5. Trx:
- ```
insert into trx values('trx001', 'BKASH', 'paid', 100, 't001', 'p001')
insert into trx values('trx002', 'BKASH', 'paid', 100, 't002', 'p002')
insert into trx values('trx003', 'CASH', 'paid', 100, 't003', 'p003')
insert into trx values('trx004', 'CASH', 'unpaid', 100, 't004', 'p004')
insert into trx values('trx005', 'BKASH', 'paid', 100, 't005', 'p005')
```

User: SCOTT

Home > SQL > SQL Commands

Autocommit

```
9.
create table admin(ad_id varchar(20) not null primary key, name varchar(200), phnno int, email varchar(200), adrs varchar(200), did int);

insert into admin values ('ad001', 'oni', 17139598545, 'oni01@gmail.com', 'dhaka', 20040050085);
insert into admin values ('ad002', 'aliif', 17139545545, 'aliif@gmail.com', 'dhaka', 33040050085);
insert into admin values ('ad003', 'monu', 17139598845, 'monu01@gmail.com', 'dhaka', 44440050085);
insert into admin values ('ad004', 'sonu', 17139598565, 'sonu01@gmail.com', 'Ctg', 20044450085);
insert into admin values ('ad005', 'amir', 17139598555, 'amir@gmail.com', 'dhaka', 90040050085);

select * from trx
```

Results Explain Describe Saved SQL History

| TRX_ID | PAY_METH | STATUS | AMOUNT | TIC_ID | P_ID |
|--------|----------|--------|--------|--------|------|
| tx001  | BKASH    | paid   | 100    | t001   | p001 |
| tx002  | BKASH    | paid   | 100    | t002   | p002 |
| tx003  | CASH     | paid   | 100    | t003   | p003 |
| tx004  | CASH     | unpaid | 100    | t004   | p004 |
| tx005  | BKASH    | paid   | 100    | t005   | p005 |

5 rows returned in 0.01 seconds

Application Express 2.1 00.39  
Copyright © 1999, 2006, Oracle. All rights reserved.

Language: en-us

Windows Taskbar: Type here to search, Start button, File Explorer, Internet Explorer, Google, Firefox, Chrome, Microsoft Edge, File, Network, Task View, Taskbar settings, System tray: 20°C Haze, 7:19 PM, 12/11/2024

## 6. Route:

```
insert into route values ('rou001','uttara north-motijheel' , 's001' , 's003')
insert into route values ('rou002','uttara south-motijheel' , 's002' , 's003')
insert into route values ('rou003','motijheel-agargaon' , 's003' , 's004')
insert into route values ('rou004','uttara south-mirpur 10' , 's002' , 's005')
insert into route values ('rou005','uttara north-mirpur 10' , 's001' , 's005')
```

The screenshot shows the Oracle Database Express Edition SQL Command window. The code entered is:

```
9.
create table admin(ad_id varchar(20) not null primary key, name varchar2(200), phnno int, email varchar2(200), addrs varchar2(200), nid int);

insert into admin values ('ad001', 'onu', 17139598545, 'onu01@gmail.com', 'dhaka', 200400050085)
insert into admin values ('ad002', 'alif', 17139545545, 'alif@gmail.com', 'pabna', 330400050085)
insert into admin values ('ad003', 'monu', 17139598845, 'monu01@gmail.com', 'dhaka', 444400050085)
insert into admin values ('ad004', 'sonu', 17139598565, 'sonu01@gmail.com', 'ctg', 200444500085)
insert into admin values ('ad005', 'amir', 17139598555, 'amir@gmail.com', 'dhaka', 900400050085)

select * from route
```

The results section displays the data inserted into the route table:

| ROU_ID | ROU_NAME               | STARTSTN_ID | ENDSTN_ID |
|--------|------------------------|-------------|-----------|
| rou001 | uttara north-motijheel | s001        | s003      |
| rou002 | uttara south-motijheel | s002        | s003      |
| rou003 | motijheel-agargaon     | s003        | s004      |
| rou004 | uttara south-mirpur 10 | s002        | s005      |
| rou005 | uttara north-mirpur 10 | s001        | s005      |

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

Language: en-us Application Express 2.1 0.00.39 Copyright © 1999, 2006, Oracle. All rights reserved.

## 7. Station:

```
insert into station values ('s001', 'Uttara north', 'uttara', timestamp '2024-12-03 21:30:00'
,timestamp '2024-12-03 07:00:00')
insert into station values ('s002', 'Uttara south', 'uttara', timestamp '2024-12-03 21:30:00'
,timestamp '2024-12-03 07:00:00')
insert into station values ('s003', 'motijheel', 'motijheel', timestamp '2024-12-03 21:30:00'
,timestamp '2024-12-03 07:00:00')
insert into station values ('s004', 'agargaon', 'agargaon', timestamp '2024-12-03 21:30:00'
,timestamp '2024-12-03 07:00:00')
insert into station values ('s005', 'mirpur 10', 'mirpur 10', timestamp '2024-12-03 21:30:00'
,timestamp '2024-12-03 07:00:00')
```

```

User SCOTT
Home > SQL > SQL Commands
Autocommit Display 10 Save Run
9.
create table admin(ad_id varchar(20) not null primary key, name varchar2(200), phnno int, email varchar2(200), addrs varchar2(200), did int);

insert into admin values ('ad001', 'enu', 17139598545, 'enu0@gmail.com', 'dhaka', 208440050085);
insert into admin values ('ad002', 'ali', 17139545545, 'ali@gmail.com', 'Dhaka', 330400050085);
insert into admin values ('ad003', 'monu', 17139598845, 'monu0@gmail.com', 'dhaka', 444400050085);
insert into admin values ('ad004', 'sonu', 17139598565, 'sonu0@gmail.com', 'Ctg', 20844450085);
insert into admin values ('ad005', 'amir', 17139598555, 'amir@gmail.com', 'dhaka', 908400050085);

select * from station

```

Results Explain Describe Saved SQL History

| STN_ID | STTN_NAME    | LOC       | CLOSING_HOUR                 | OPENING_HOUR                 |
|--------|--------------|-----------|------------------------------|------------------------------|
| s001   | Uttara north | uttara    | 03-DEC-24 09.30.00.000000 PM | 03-DEC-24 07.00.00.000000 AM |
| s002   | Uttara south | uttara    | 03-DEC-24 09.30.00.000000 PM | 03-DEC-24 07.00.00.000000 AM |
| s003   | motijheel    | motjheel  | 03-DEC-24 09.30.00.000000 PM | 03-DEC-24 07.00.00.000000 AM |
| s004   | agargaon     | agargaon  | 03-DEC-24 09.30.00.000000 PM | 03-DEC-24 07.00.00.000000 AM |
| s005   | mirpur 10    | mirpur 10 | 03-DEC-24 09.30.00.000000 PM | 03-DEC-24 07.00.00.000000 AM |

5 rows returned in 0.00 seconds CSV Export

Application Express 2.1.0.0.39  
Copyright © 1999, 2006, Oracle. All rights reserved.

Language en-US

8. Train:
- ```

insert into train values ('trn001', 'modhumoti', 'rou001', 500, '4 hour')
insert into train values ('trn002', 'tista', 'rou001', 500, '4 hour')
insert into train values ('trn003', 'bosumoti', 'rou001', 500, '4 hour')
insert into train values ('trn004', 'kawla', 'rou001', 500, '4 hour')
insert into train values ('trn005', 'moinamoti', 'rou001', 500, '4 hour')

```

```

User SCOTT
Home > SQL > SQL Commands
Autocommit Display 10 Save Run
9.
create table admin(ad_id varchar(20) not null primary key, name varchar2(200), phnno int, email varchar2(200), addrs varchar2(200), did int);

insert into admin values ('ad001', 'enu', 17139598545, 'enu0@gmail.com', 'dhaka', 208440050085);
insert into admin values ('ad002', 'ali', 17139545545, 'ali@gmail.com', 'Dhaka', 330400050085);
insert into admin values ('ad003', 'monu', 17139598845, 'monu0@gmail.com', 'dhaka', 444400050085);
insert into admin values ('ad004', 'sonu', 17139598565, 'sonu0@gmail.com', 'Ctg', 20844450085);
insert into admin values ('ad005', 'amir', 17139598555, 'amir@gmail.com', 'dhaka', 908400050085);

select * from train

```

Results Explain Describe Saved SQL History

TRAIN_ID	TRAIN_NAME	ROU_ID	CAP	OP_TYM
tm001	modhumoti	rou001	500	4 hour
tm002	tista	rou001	500	4 hour
tm003	bosumoti	rou001	500	4 hour
tm004	kawla	rou001	500	4 hour
tm005	moinamoti	rou001	500	4 hour

5 rows returned in 0.00 seconds CSV Export

Application Express 2.1.0.0.39
Copyright © 1999, 2006, Oracle. All rights reserved.

Language en-US

9. Admin:

insert into admin values ('ad001', 'onu', 17139598545, 'onu01@gmail.com', 'dhaka', 20040050085)

insert into admin values ('ad002', 'alif', 17139545545, 'alif@gmail.com', 'pabna', 33040050085)

insert into admin values ('ad003', 'monu', 17139598845, 'monu01@gmail.com', 'dhaka', 44440050085)

insert into admin values ('ad004', 'sonu', 17139598565, 'sonu01@gmail.com', 'ctg', 20044450085)

insert into admin values ('ad005', 'amir', 17139598555, 'amir@gmail.com', 'dhaka', 90040050085)

The screenshot shows the Oracle Database Express Edition interface. The SQL Commands window contains the following code:

```
9.
create table admin(ad_id varchar(20) not null primary key, name varchar2(200), phnno int, email varchar2(200), adrs varchar2(200), nid int);

insert into admin values ('ad001', 'onu', 17139598545, 'onu01@gmail.com', 'dhaka', 20040050085)
insert into admin values ('ad002', 'alif', 17139545545, 'alif@gmail.com', 'pabna', 33040050085)
insert into admin values ('ad003', 'monu', 17139598845, 'monu01@gmail.com', 'dhaka', 44440050085)
insert into admin values ('ad004', 'sonu', 17139598565, 'sonu01@gmail.com', 'ctg', 20044450085)
insert into admin values ('ad005', 'amir', 17139598555, 'amir@gmail.com', 'dhaka', 90040050085)

select * from admin
```

The Results tab displays the data inserted into the 'admin' table:

AD_ID	NAME	PHNNO	EMAIL	ADR\$	NID
ad001	onu	17139598545	onu01@gmail.com	dhaka	20040050085
ad002	alif	17139545545	alif@gmail.com	pabna	33040050085
ad003	monu	17139598845	monu01@gmail.com	dhaka	44440050085
ad004	sonu	17139598565	sonu01@gmail.com	ctg	20044450085
ad005	amir	17139598555	amir@gmail.com	dhaka	90040050085

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

Application Express 2.1.0.00.39
Copyright © 1999, 2006, Oracle. All rights reserved.

Basic PL/SQL

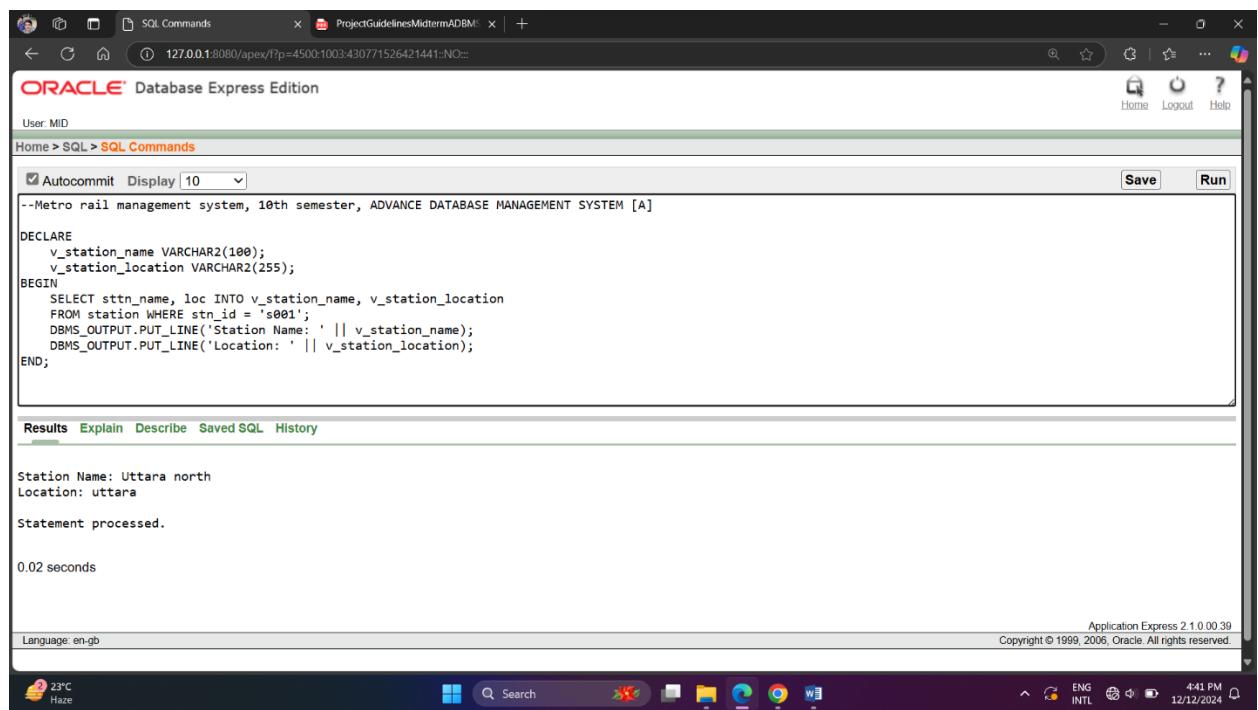
variables

- a) Write a query that can display Station_name and location using pl/sql

Ans:

--Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM [A]

```
DECLARE
    v_station_name VARCHAR2(100);
    v_station_location VARCHAR2(255);
BEGIN
    SELECT sttn_name, loc INTO v_station_name, v_station_location
    FROM station WHERE stn_id = 's001';
    DBMS_OUTPUT.PUT_LINE('Station Name: ' || v_station_name);
    DBMS_OUTPUT.PUT_LINE('Location: ' || v_station_location);
END;
```



The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands window, a PL/SQL block is entered:

```
--Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM [A]
DECLARE
    v_station_name VARCHAR2(100);
    v_station_location VARCHAR2(255);
BEGIN
    SELECT sttn_name, loc INTO v_station_name, v_station_location
    FROM station WHERE stn_id = 's001';
    DBMS_OUTPUT.PUT_LINE('Station Name: ' || v_station_name);
    DBMS_OUTPUT.PUT_LINE('Location: ' || v_station_location);
END;
```

After running the code, the results are displayed in the Results tab:

```
Station Name: Uttara north
Location: uttara

Statement processed.

0.02 seconds
```

The bottom status bar indicates the application version and copyright information.

- b) Write a query that can display passenger age from psngr table where p_id p001 using pl/sql

Ans:

--Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM
[A]

```
DECLARE
  v_age INT;
BEGIN
  SELECT age INTO v_age FROM psngr WHERE p_id = 'p001';
  DBMS_OUTPUT.PUT_LINE('Passenger Age: ' || v_age);
END
```

The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands window, a PL/SQL block is run. The code declares a variable v_age, selects the age for p_id 'p001' into it, and then prints the result. The output window shows the passenger age as 24 and a statement processed message. The bottom status bar indicates the application version is 2.1.0.00.39 and the copyright year is 1999-2006.

operators

- a) Write a query that display remaining balance after removing 500 from card where id = 1

Ans:

--Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM
[A]

```
DECLARE
  v_balance INT;
  v_amount INT;
BEGIN
  SELECT balance INTO v_balance FROM Card WHERE card_id = '1';
  v_amount := v_balance - 500;
  DBMS_OUTPUT.PUT_LINE('Remaining Balance: ' || v_amount);
END;
```

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL code entered is:

```
--Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM [A]

DECLARE
    v_balance INT;
    v_amount INT;
BEGIN
    SELECT balance INTO v_balance FROM Card WHERE card_id = '1';
    v_amount := v_balance - 500;
    DBMS_OUTPUT.PUT_LINE('Remaining Balance: ' || v_amount);
END;
```

The results pane shows the output of the query:

```
Remaining Balance: 1500
Statement processed.

0.00 seconds
```

At the bottom right, it says "Application Express 2.1.0.0.39 Copyright © 1999, 2006, Oracle. All rights reserved."

- b) Write a query that display ticket price including charges where ticket id t001

Ans:

--Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM
[A]

```
DECLARE
    v_price INT;
    v_final_price INT;
BEGIN
    SELECT price INTO v_price FROM Tic WHERE tic_id = 't001';
    v_final_price := v_price * 1.1;
    DBMS_OUTPUT.PUT_LINE('Final Ticket Price with service: ' || v_final_price);
END;
```

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL code entered is:

```
--Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM [A]

DECLARE
    v_price INT;
    v_final_price INT;
BEGIN
    SELECT price INTO v_price FROM Tic WHERE tic_id = 't001';
    v_final_price := v_price * 1.1;
    DBMS_OUTPUT.PUT_LINE('Final Ticket Price with service: ' || v_final_price);
END;
```

The results pane shows the output of the PL/SQL block:

```
Final Ticket Price with service: 110
Statement processed.

0.00 seconds
```

At the bottom right, it says "Application Express 2.1.0.0.39 Copyright © 1999, 2006, Oracle. All rights reserved."

single-row function

- a) Write a query that display name and email from psngr table where email show all in capital letter.

Ans:

--Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM [A]

```
DECLARE
    v_name VARCHAR2(50);
    v_email VARCHAR2(50);
BEGIN
    SELECT name, UPPER(email) INTO v_name, v_email FROM psngr WHERE p_id = 'p001';
    DBMS_OUTPUT.PUT_LINE('Name: ' || v_name || ', Email: ' || v_email);
END;
```

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL code entered is:

```
--Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM [A]
DECLARE
    v_name VARCHAR2(50);
    v_email VARCHAR2(50);
BEGIN
    SELECT name, UPPER(email) INTO v_name, v_email FROM psngr WHERE p_id = 'p001';
    DBMS_OUTPUT.PUT_LINE('Name: ' || v_name || ', Email: ' || v_email);
END;
```

The results pane displays the output:

```
Name: Asir, Email: ASIR@GMAIL.COM
Statement processed.

0.00 seconds
```

At the bottom right, it says "Application Express 2.1.0.0.39 Copyright © 1999, 2006, Oracle. All rights reserved."

- b) Write a query that display name and email from psngr table where name show all in small letter.

Ans: --Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM [A]

```
DECLARE
    v_name VARCHAR2(50);
    v_email VARCHAR2(50);
BEGIN
    SELECT LOWER(name), email INTO v_name, v_email FROM psngr WHERE p_id = 'p003';
    DBMS_OUTPUT.PUT_LINE('Name: ' || v_name || ', Email: ' || v_email);
END;
```

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL code entered is:

```
--Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM [A]
DECLARE
    v_name VARCHAR2(50);
    v_email VARCHAR2(50);
BEGIN
    SELECT LOWER(name), email INTO v_name, v_email FROM passenger WHERE p_id = 'p003';
    DBMS_OUTPUT.PUT_LINE('Name: ' || v_name || ', Email: ' || v_email);
END;
```

The results pane displays the output of the PL/SQL block:

```
Name: suma, Email: suma@gmail.com
Statement processed.
0.00 seconds
```

At the bottom right, it says "Application Express 2.1.0.0.39 Copyright © 1999, 2006, Oracle. All rights reserved."

Group function

- a) Write a query that display average and maximum balance from card table

Ans:

-Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM
[A]

```
DECLARE
    v_avg_balance INT;
    v_max_balance INT;
BEGIN
    SELECT AVG(balance), MAX(balance) INTO v_avg_balance, v_max_balance FROM card;
    DBMS_OUTPUT.PUT_LINE('Average Balance: ' || v_avg_balance || ', Maximum Balance: ' ||
    v_max_balance);
END;
```

```

-- Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM [A]

DECLARE
    v_avg_balance INT;
    v_max_balance INT;
BEGIN
    SELECT AVG(balance), MAX(balance) INTO v_avg_balance, v_max_balance FROM card;
    DBMS_OUTPUT.PUT_LINE('Average Balance: ' || v_avg_balance || ', Maximum Balance: ' || v_max_balance);
END;

```

Results Explain Describe Saved SQL History

Average Balance: 700, Maximum Balance: 2000
Statement processed.
0.00 seconds

Language: en-gb Application Express 2.1.0.0.39
Copyright © 1999, 2006, Oracle. All rights reserved.

- b) Write a query that display total balance and total card number from card table

Ans:

--Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM
[A]

DECLARE

```

v_sum_balance INT;
v_count_cards INT;
BEGIN
    SELECT SUM(balance), COUNT(card_id) INTO v_sum_balance, v_count_cards FROM card;
    DBMS_OUTPUT.PUT_LINE('Total Balance: ' || v_sum_balance || ', Number of Cards: ' || v_count_cards);
END;

```

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL code entered is:

```
--Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM [A]

DECLARE
    v_sum_balance INT;
    v_count_cards INT;
BEGIN
    SELECT SUM(balance), COUNT(card_id) INTO v_sum_balance, v_count_cards FROM card;
    DBMS_OUTPUT.PUT_LINE('Total Balance: ' || v_sum_balance || ', Number of Cards: ' || v_count_cards);
END;
```

The results pane displays the output of the query:

```
Total Balance: 3500, Number of Cards: 5
Statement processed.

0.00 seconds
```

At the bottom right, it says "Application Express 2.1.0.0.39 Copyright © 1999, 2006, Oracle. All rights reserved."

Loop

- a) Write a query that display card balance using for loop

Ans:

-Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM
[A]

```
DECLARE
    v_card_id VARCHAR2(20);
BEGIN
    FOR v_card IN (SELECT card_id, balance FROM card) LOOP
        DBMS_OUTPUT.PUT_LINE('Card ID: ' || v_card.card_id || ' Balance: ' || v_card.balance);
    END LOOP;
END;
```

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL editor contains the following PL/SQL code:

```
--Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM [A]

DECLARE
    v_card_id VARCHAR2(20);
BEGIN
    FOR v_card IN (SELECT card_id, balance FROM card) LOOP
        DBMS_OUTPUT.PUT_LINE('Card ID: ' || v_card.card_id || ' Balance: ' || v_card.balance);
    END LOOP;
END;
```

The results pane displays the output of the code execution:

```
Card ID: 1 Balance: 2000
Card ID: 2 Balance: 1000
Card ID: 3 Balance: 0
Card ID: 4 Balance: 0
Card ID: 5 Balance: 500

Statement processed.

0.02 seconds
```

The status bar at the bottom right indicates "Application Express 2.1.0.0.30" and the date/time "12/12/2024 5:49 PM".

- b) Write a query that display number as counter using loop

Ans:

-Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM
[A]

```
DECLARE
    v_counter INT := 1;
BEGIN
    FOR i IN 1..5 LOOP
        DBMS_OUTPUT.PUT_LINE('Count: ' || v_counter);
        v_counter := v_counter + 1;
    END LOOP;
END;
```

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL editor contains the following PL/SQL code:

```
--Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM [A]

DECLARE
    v_counter INT := 1;
BEGIN
    FOR i IN 1..5 LOOP
        DBMS_OUTPUT.PUT_LINE('Count: ' || v_counter);
        v_counter := v_counter + 1;
    END LOOP;
END;
```

The results pane shows the output of the code execution:

```
Count: 1
Count: 2
Count: 3
Count: 4
Count: 5

Statement processed.

0.01 seconds
```

The status bar at the bottom right indicates "Application Express 2.1.0.0.30" and the date/time "12/12/2024 6:06 PM".

Conditional statements

- a) Write a query that display transaction status from trx table

Ans:

```
--Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM
[A]
```

```
DECLARE
    v_status VARCHAR2(20);
BEGIN
    SELECT status INTO v_status FROM trx WHERE trx_id = 'trx001';
    IF v_status = 'paid' THEN
        DBMS_OUTPUT.PUT_LINE('Transaction Paid');
    ELSE
        DBMS_OUTPUT.PUT_LINE('Transaction Unpaid');
    END IF;
END;
```

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL editor contains the following PL/SQL code:

```
--Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM [A]

DECLARE
    v_status VARCHAR2(20);
BEGIN
    SELECT status INTO v_status FROM trx WHERE trx_id = 'trx001';
    IF v_status = 'paid' THEN
        DBMS_OUTPUT.PUT_LINE('Transaction Paid');
    ELSE
        DBMS_OUTPUT.PUT_LINE('Transaction Unpaid');
    END IF;
END;]
```

The results pane shows the output of the code execution:

```
Transaction Paid
Statement processed.

0.00 seconds
```

At the bottom right, it says "Application Express 2.1.0.0.0.39 Copyright © 1999, 2006, Oracle. All rights reserved."

- b) Write a query that display the ticket type using ticket id from tic table

Ans:

--Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM
[A]

```
DECLARE
    v_ticket_type VARCHAR2(50);
BEGIN
    SELECT tic_type INTO v_ticket_type
    FROM tic
    WHERE tic_id = 't001';

    CASE v_ticket_type
        WHEN 'MRT' THEN
            DBMS_OUTPUT.PUT_LINE('Ticket Type: MRT');
        WHEN 'RAPID' THEN
            DBMS_OUTPUT.PUT_LINE('Ticket Type: RAPID');
        ELSE
            DBMS_OUTPUT.PUT_LINE('Ticket Type: Regular');
    END CASE;
END;
```

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL code entered is:

```
--Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM [A]

DECLARE
    v_ticket_type VARCHAR2(50);
BEGIN
    SELECT tic_type INTO v_ticket_type
    FROM tic
    WHERE tic_id = 't001';

    CASE v_ticket_type
        WHEN 'MRT' THEN
            DBMS_OUTPUT.PUT_LINE('Ticket Type: MRT');
        WHEN 'RAPID' THEN
            DBMS_OUTPUT.PUT_LINE('Ticket Type: RAPID');
        ELSE
            DBMS_OUTPUT.PUT_LINE('Ticket Type: Regular');
    END CASE;
END;
```

The results pane shows the output of the DBMS_OUTPUT.PUT_LINE statements:

Ticket Type: MRT
Statement processed.
0.00 seconds

At the bottom, the status bar indicates "Application Express 2.1.0.0.39" and the system tray shows icons for battery, temperature (27°C), and network.

Subquery

- a) Write a query that display ticket price from tic where passenger id p001 and ticket type MRT

Ans:

--Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM
[A]

```
DECLARE
    v_ticket_price INT;
BEGIN
    SELECT price INTO v_ticket_price
    FROM tic
    WHERE tic_id = (SELECT tic_id FROM tic WHERE p_id = 'p001' AND tic_type = 'MRT');

    DBMS_OUTPUT.PUT_LINE('Ticket Price: ' || v_ticket_price);
END;
```

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL code entered is:

```
--Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM [A]

DECLARE
    v_ticket_price INT;
BEGIN
    SELECT price INTO v_ticket_price
    FROM tic
    WHERE tic_id = (SELECT tic_id FROM tic WHERE p_id = 'p001' AND tic_type = 'MRT');

    DBMS_OUTPUT.PUT_LINE('Ticket Price: ' || v_ticket_price);
END;
```

The results pane shows the output:

```
Ticket Price: 100
Statement processed.

0.02 seconds
```

At the bottom right, it says "Application Express 2.1.0.0.39 Copyright © 1999, 2006, Oracle. All rights reserved."

- b) Write a query that display card balance from card table where p_id = p001

Ans:

--Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM
[A]

```
DECLARE
    v_balance NUMBER;
BEGIN
    SELECT balance INTO v_balance FROM Card WHERE card_id = (SELECT card_id FROM psngr
    WHERE p_id = 'p001');
    DBMS_OUTPUT.PUT_LINE('Card Balance: ' || v_balance);
END;
```

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL code entered is:

```
--Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM [A]
DECLARE
    v_balance NUMBER;
BEGIN
    SELECT balance INTO v_balance FROM Card WHERE card_id = (SELECT card_id FROM psngr WHERE p_id = 'p001');
    DBMS_OUTPUT.PUT_LINE('Card Balance: ' || v_balance);
END;
```

The results pane shows the output:

```
Card Balance: 2000
Statement processed.

0.01 seconds
```

At the bottom right, it says Application Express 2.1.0.0.39 Copyright © 1999, 2006, Oracle. All rights reserved.

Joining

- a) Write a query that display card type and balance from card and psngr using joining.

Ans:

--Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM
[A]

```
DECLARE
    v_card_type VARCHAR2(200);
    v_balance INT;
BEGIN
    SELECT c.type, c.balance
    INTO v_card_type, v_balance
    FROM Card c
    JOIN psngr p ON c.card_id = p.card_id
    WHERE p.p_id = 'p001';
    DBMS_OUTPUT.PUT_LINE('Card Type: ' || v_card_type || ', Balance: ' || v_balance);
END;
```

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL code is as follows:

```
--Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM [A]

DECLARE
    v_card_type VARCHAR2(200);
    v_balance INT;
BEGIN
    SELECT c.type, c.balance
    INTO v_card_type, v_balance
    FROM Card c
    JOIN psngr p ON c.card_id = p.card_id
    WHERE p.p_id = 'p001';
    DBMS_OUTPUT.PUT_LINE('Card Type: ' || v_card_type || ', Balance: ' || v_balance);
END;
```

The results show the output of the query:

```
Card Type: MRT, Balance: 2000
Statement processed.

0.02 seconds
```

Language: en-gb Application Express 2.1.0.0.39
Copyright © 1999, 2006, Oracle. All rights reserved.

- b) Write a query that display train name and route name from train and route table using joining

Ans:

--Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM
[A]

```
DECLARE
    v_train_name VARCHAR2(200);
    v_route_name VARCHAR2(200);
BEGIN
    SELECT t.train_name, r.rou_name
    INTO v_train_name, v_route_name
    FROM train t
    JOIN route r ON t.rou_id = r.rou_id
    WHERE t.train_id = 'trn001';

    DBMS_OUTPUT.PUT_LINE('Train Name: ' || v_train_name);
    DBMS_OUTPUT.PUT_LINE('Route Name: ' || v_route_name);
END;
```

The screenshot shows the Oracle Database Express Edition SQL Commands interface. A PL/SQL block is being executed:

```
--Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM [A]

DECLARE
    v_train_name VARCHAR2(200);
    v_route_name VARCHAR2(200);
BEGIN
    SELECT t.train_name, r.rou_name
    INTO v_train_name, v_route_name
    FROM train t
    JOIN route r ON t.rou_id = r.rou_id
    WHERE t.train_id = 'trn001';

    DBMS_OUTPUT.PUT_LINE('Train Name: ' || v_train_name);
    DBMS_OUTPUT.PUT_LINE('Route Name: ' || v_route_name);
END;
```

The results show the output of the PL/SQL block:

```
Train Name: modhumoti
Route Name: uttara north-motijheel

Statement processed.

0.00 seconds
```

At the bottom, the status bar indicates "Application Express 2.1.0.00.39" and "Copyright © 1999, 2006, Oracle. All rights reserved."

Advance PL/SQL

stored function

- a) Write a query to create a function to check card balance

Ans:

--Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM [A]

```
CREATE OR REPLACE FUNCTION check_balance(card_id_input VARCHAR2) RETURN INT IS
    card_balance INT;
BEGIN
    SELECT balance INTO card_balance
    FROM Card
    WHERE card_id = card_id_input;

    RETURN card_balance;
END;
```

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL code entered is:

```
--Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM [A]
CREATE OR REPLACE FUNCTION check_balance(card_id_input VARCHAR2) RETURN INT IS
    card_balance INT;
BEGIN
    SELECT balance INTO card_balance
    FROM Card
    WHERE card_id = card_id_input;
    RETURN card_balance;
END;
```

The results pane shows the message "Statement processed." and a execution time of "0.08 seconds". The bottom right corner displays the Application Express version "2.1.0.0.39" and copyright information "Copyright © 1999, 2006, Oracle. All rights reserved."

- b) Write a query to create a Function to get passenger email by ticket ID

Ans:

--Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM
[A]

```
CREATE OR REPLACE FUNCTION get_psngr_email(tic_id_input VARCHAR2) RETURN VARCHAR2
IS
    psngr_email VARCHAR2(50);
BEGIN
    SELECT email INTO psngr_email
    FROM Psngr
    WHERE p_id = (SELECT p_id FROM Tic WHERE tic_id = tic_id_input);

    RETURN psngr_email;
END;
```

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The user is executing a SQL script to create a function named `get_psngr_email`. The function takes a parameter `tic_id_input` of type VARCHAR2 and returns a VARCHAR2 value. It uses a SELECT statement to find the email address of the passenger associated with the given ticket ID. The code is as follows:

```
--Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM [A]
CREATE OR REPLACE FUNCTION get_psngr_email(tic_id_input VARCHAR2) RETURN VARCHAR2 IS
    psngr_email VARCHAR2(50);
BEGIN
    SELECT email INTO psngr_email
    FROM Psngr
    WHERE p_id = (SELECT p_id FROM Tic WHERE tic_id = tic_id_input);

    RETURN psngr_email;
END;
```

After running the script, the message "Statement processed." is displayed. The execution time is 0.00 seconds. The language is set to en-gb. The application version is Application Express 2.1.0.0.39, and the copyright notice is from Oracle.

Stored procedure

- a) Write a query to create Procedure to update card balance

Ans:

-Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM [A]

```
CREATE OR REPLACE PROCEDURE update_balance(card_id_input VARCHAR2, amount INT) IS
BEGIN
    UPDATE Card
    SET balance = balance + amount
    WHERE card_id = card_id_input;

    END;
```

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The user is executing a SQL script to create a procedure named `update_balance`. The script is as follows:

```
--Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM [A]
CREATE OR REPLACE PROCEDURE update_balance(card_id_input VARCHAR2, amount INT) IS
BEGIN
    UPDATE Card
    SET balance = balance + amount
    WHERE card_id = card_id_input;
END;
```

After running the script, the results show 0 rows updated in 0.00 seconds. The interface includes tabs for Results, Explain, Describe, Saved SQL, and History. The bottom status bar indicates the language is en-gb and the application version is Application Express 2.1.0.0.39. Copyright © 1999, 2006, Oracle. All rights reserved.

- b) Write a query to create Procedure to mark a transaction as paid

Ans:

-Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM
[A]

```
CREATE OR REPLACE PROCEDURE mark_transaction_paid(trx_id_input VARCHAR2) IS
BEGIN
    UPDATE Trx
    SET status = 'paid'
    WHERE trx_id = trx_id_input;
END;
```

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL code is:

```
--Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM [A]
CREATE OR REPLACE PROCEDURE mark_transaction_paid(trx_id_input VARCHAR2) IS
BEGIN
  UPDATE Trx
  SET status = 'paid'
  WHERE trx_id = trx_id_input;
END;;
```

The results show:

0 row(s) updated.
0.00 seconds

Language: en-gb Application Express 2.1.0.0.39
Copyright © 1999, 2006, Oracle. All rights reserved.

Table-based record

- a) Write a query to find record for Card table

Ans:

-Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM [A]

```
DECLARE
  card_record Card%ROWTYPE;
BEGIN
  SELECT * INTO card_record FROM Card WHERE card_id = '1';
  DBMS_OUTPUT.PUT_LINE('Card ID: ' || card_record.card_id || ', Balance: ' ||
  card_record.balance);
END;
```

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL code entered is:

```
--Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM [A]
DECLARE
    card_record Card%ROWTYPE;
BEGIN
    SELECT * INTO card_record FROM Card WHERE card_id = '1';
    DBMS_OUTPUT.PUT_LINE('Card ID: ' || card_record.card_id || ', Balance: ' || card_record.balance);
END;
```

The results pane displays the output:

```
Card ID: 1, Balance: 2000
Statement processed.

0.00 seconds
```

At the bottom right, it shows Application Express 2.1.0.0.39 and Copyright © 1999, 2006, Oracle. All rights reserved.

- b) Write a query to find record for Passenger table

Ans:

--Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM
[A]

```
DECLARE
    psngr_record Psngr%ROWTYPE;
BEGIN
    SELECT * INTO psngr_record FROM Psngr WHERE p_id = 'p001';
    DBMS_OUTPUT.PUT_LINE('Passenger Name: ' || psngr_record.name || ', Email: ' || psngr_record.email);
END;
```

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL code entered is:

```
--Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM [A]
DECLARE
    psngr_record Psngr%ROWTYPE;
BEGIN
    SELECT * INTO psngr_record FROM Psngr WHERE p_id = 'p001';
    DBMS_OUTPUT.PUT_LINE('Passenger Name: ' || psngr_record.name || ', Email: ' || psngr_record.email);
END;]
```

The results pane shows the output: "Passenger Name: Asir, Email: asir@gmail.com". Below the results, it says "Statement processed." and "0.00 seconds". At the bottom right, it shows "Application Express 2.1.0.0.39" and "Copyright © 1999, 2006, Oracle. All rights reserved."

Explicit cursor

- a) Create a explicit cursor that can output all active cards from card table

Ans:

--Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM
[A]

```
DECLARE
    CURSOR active_cards IS
        SELECT card_id, type FROM Card WHERE balance > 0;
        card_data active_cards%ROWTYPE;
    BEGIN
        OPEN active_cards;
        LOOP
            FETCH active_cards INTO card_data;
            EXIT WHEN active_cards%NOTFOUND;
            DBMS_OUTPUT.PUT_LINE('Card ID: ' || card_data.card_id || ', Type: ' || card_data.type);
        END LOOP;
        CLOSE active_cards;
    END;
```

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL code is as follows:

```

-- Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM [A]

DECLARE
    CURSOR active_cards IS
        SELECT card_id, type FROM Card WHERE balance > 0;
    card_data active_cards%ROWTYPE;
BEGIN
    OPEN active_cards;
    LOOP
        FETCH active_cards INTO card_data;
        EXIT WHEN active_cards%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE('Card ID: ' || card_data.card_id || ', Type: ' || card_data.type);
    END LOOP;
    CLOSE active_cards;
END;

```

The results section shows the output of the query:

```

Card ID: 1, Type: MRT
Card ID: 2, Type: RAPID
Card ID: 5, Type: MRT

Statement processed.

0.00 seconds

```

At the bottom, the status bar indicates "Application Express 2.1.0.0.39" and the system clock shows "8:04 PM 12/12/2024".

- b) Create an explicit cursor that can output all Transaction ID and Amount with unpaid status from Trx table

Ans:

--Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM
[A]

```

DECLARE
    CURSOR unpaid_trx IS
        SELECT trx_id, amount FROM Trx WHERE status = 'unpaid';
    trx_data unpaid_trx%ROWTYPE;
BEGIN
    OPEN unpaid_trx;
    LOOP
        FETCH unpaid_trx INTO trx_data;
        EXIT WHEN unpaid_trx%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE('Transaction ID: ' || trx_data.trx_id || ', Amount: ' ||
        trx_data.amount);
    END LOOP;
    CLOSE unpaid_trx;
END;

```

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The code entered is:

```
--Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM [A]

DECLARE
  CURSOR unpaid_trx IS
    SELECT trx_id, amount FROM Trx WHERE status = 'unpaid';
  trx_data unpaid_trx%ROWTYPE;
BEGIN
  OPEN unpaid_trx;
  LOOP
    FETCH unpaid_trx INTO trx_data;
    EXIT WHEN unpaid_trx%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE('Transaction ID: ' || trx_data.trx_id || ', Amount: ' || trx_data.amount);
  END LOOP;
  CLOSE unpaid_trx;
END;
```

The results pane shows the output of the PL/SQL block:

```
Transaction ID: trx004, Amount: 100
Statement processed.

0.00 seconds
```

At the bottom right, it says "Application Express 2.1.0.0.39 Copyright © 1999, 2006, Oracle. All rights reserved."

Cursor-based record

- a) Create a record that can output ticket id and price from tic table

Ans:

--Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM
[A]

```
DECLARE
  CURSOR ticket_cursor IS
    SELECT tic_id, price FROM Tic;
  ticket_record ticket_cursor%ROWTYPE;
BEGIN
  OPEN ticket_cursor;
  LOOP
    FETCH ticket_cursor INTO ticket_record;
    EXIT WHEN ticket_cursor%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE('Ticket ID: ' || ticket_record.tic_id || ', Price: ' || ticket_record.price);
  END LOOP;
  CLOSE ticket_cursor;
END;
```

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL code entered is:

```
--Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM [A]

DECLARE
    CURSOR ticket_cursor IS
        SELECT tic_id, price FROM Tic;
    ticket_record ticket_cursor%ROWTYPE;
BEGIN
    OPEN ticket_cursor;
    LOOP
        FETCH ticket_cursor INTO ticket_record;
        EXIT WHEN ticket_cursor%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE('Ticket ID: ' || ticket_record.tic_id || ', Price: ' || ticket_record.price);
    END LOOP;
    CLOSE ticket_cursor;
END;]
```

The results pane displays the output of the PL/SQL block:

```
Ticket ID: t001, Price: 100
Ticket ID: t002, Price: 100
Ticket ID: t003, Price: 100
Ticket ID: t004, Price: 100
Ticket ID: t005, Price: 100
Statement processed.
```

At the bottom, the taskbar shows the date and time as 12/12/2024 at 8:23 PM.

- b) Create a record that can output Station id and name from station table

Ans:

--Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM
[A]

```
DECLARE
    CURSOR station_cursor IS
        SELECT stn_id, sttn_name FROM Station;
    station_record station_cursor%ROWTYPE;
BEGIN
    OPEN station_cursor;
    LOOP
        FETCH station_cursor INTO station_record;
        EXIT WHEN station_cursor%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE('Station ID: ' || station_record.stn_id || ', Name: ' || station_record.sttn_name);
    END LOOP;
    CLOSE station_cursor;
END;
```

The screenshot shows the Oracle Database Express Edition SQL Commands interface. A PL/SQL block is being run:

```
--Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM [A]

DECLARE
    CURSOR station_cursor IS
        SELECT stn_id, sttn_name FROM Station;
    station_record station_cursor%ROWTYPE;
BEGIN
    OPEN station_cursor;
    LOOP
        FETCH station_cursor INTO station_record;
        EXIT WHEN station_cursor%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE('Station ID: ' || station_record.stn_id || ', Name: ' || station_record.sttn_name);
    END LOOP;
    CLOSE station_cursor;
END;
```

The results of the execution are displayed below:

```
Station ID: s001, Name: Uttara north
Station ID: s002, Name: Uttara south
Station ID: s003, Name: motijheel
Station ID: s004, Name: agargaon
Station ID: s005, Name: mirpur 10

Statement processed.
```

Execution time: 0.00 seconds

Row level trigger

- a) Create a trigger to keep a payment log whenever payment method is changed.

Ans:

--Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM
[A]

```
CREATE TABLE payment_log (tic_id VARCHAR2(20),old_payment VARCHAR2(50),new_payment  
VARCHAR2(50),change_time TIMESTAMP);
```

```
CREATE OR REPLACE TRIGGER log_payment_changes
AFTER UPDATE OF payment ON Tic
FOR EACH ROW
BEGIN
    INSERT INTO payment_log (tic_id, old_payment, new_payment, change_time)
    VALUES (:OLD.tic_id, :OLD.payment, :NEW.payment, SYSTIMESTAMP);
END;
update tic set payment= 'CASH' where tic_id='t005'
select * from payment_log
```

The screenshot shows the Oracle Database Express Edition SQL Commands interface. In the SQL editor, a trigger named 'log_payment_changes' is created to log payment changes. The trigger is defined as follows:

```
--Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM [A]
CREATE TABLE payment_log (tic_id VARCHAR2(20),old_payment VARCHAR2(50),new_payment VARCHAR2(50),change_time TIMESTAMP);

CREATE OR REPLACE TRIGGER log_payment_changes
AFTER UPDATE OF payment ON Tic
FOR EACH ROW
BEGIN
    INSERT INTO payment_log (tic_id, old_payment, new_payment, change_time)
    VALUES (:OLD.tic_id, :OLD.payment, :NEW.payment, SYSTIMESTAMP);
END;
update tic set payment= 'CASH' where tic_id='t005'
select * from payment_log
```

The results of the query show one row inserted into the payment_log table:

TIC_ID	OLD_PAYMENT	NEW_PAYMENT	CHANGE_TIME
t005	BKASH	CASH	14-DEC-24 02:00:33.212000 AM

1 rows returned in 0.00 seconds

CSV Export

Application Express 2.1.0.0.39
Copyright © 1999, 2006, Oracle. All rights reserved.

Language: en-us

200 AM 14-Dec-24

- b) Create a trigger to stop the price from being negative.

Ans:

--Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM

[A]

CREATE OR REPLACE TRIGGER check_price_non_negative

BEFORE INSERT OR UPDATE ON Tic

FOR EACH ROW

BEGIN

IF :NEW.price < 0 THEN

dbms_output.put_line('Price cannot be negative.');

END IF;

END;

update tic set price=-100 where tic_id='t005'

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The user is SCOTT. In the SQL editor, a trigger named 'check_price_non_negative' is created:

```
--Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM [A]
CREATE OR REPLACE TRIGGER check_price_non_negative
BEFORE INSERT OR UPDATE ON Tic
FOR EACH ROW
BEGIN
    IF :NEW.price < 0 THEN
        dbms_output.put_line( 'Price cannot be negative.' );
    END IF;
END;
```

Then, a update statement is run:

```
update tic set price=-100 where tic_id='t005';
```

The results show an error message: "Price cannot be negative." and a success message: "1 row(s) updated." The execution time was 0.01 seconds.

At the bottom right, the status bar shows "Application Express 2.1.0.0.39" and "Copyright © 1999, 2006, Oracle. All rights reserved."

statement level trigger

- a) Create a trigger to give a discount everytime the price is more than 100.

Ans:

--Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM

[A]

CREATE OR REPLACE TRIGGER order_discount

BEFORE INSERT ON tic

FOR EACH ROW

BEGIN

IF :NEW.price > 100 THEN

:NEW.price := :NEW.price * 0.9;

END IF;

END;

```
insert into tic values('t005', 'MRT' , TIMESTAMP '2024-11-09 16:43:00',120, 'BKASH' , 'p005')
select * from tic
```

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL code entered is:

```
--Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM [A]
CREATE OR REPLACE TRIGGER order_discount
BEFORE INSERT ON tic
FOR EACH ROW
BEGIN
  IF :NEW.price > 100 THEN
    :NEW.price := :NEW.price * 0.9;
  END IF;
END;

insert into tic values('t005', 'MRT', TIMESTAMP '2024-11-09 16:43:00', 120, 'BKASH', 'p005')
select * from tic;
```

The results section displays the following table:

TIC_ID	TIC_TYPE	ISSUE_TYM	PRICE	PAYMENT	P_ID
t005	MRT	09-NOV-24 04:43:00.000000 PM	108	BKASH	p005
t001	MRT	09-AUG-24 02:30:00.000000 PM	100	BKASH	p001
t002	RAPID	09-SEP-24 03:30:00.000000 PM	100	BKASH	p002
t003	REGULAR	09-OCT-24 12:35:00.000000 PM	100	CASH	p003
t004	REGULAR	09-DEC-24 10:40:00.000000 AM	100	CASH	p004

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

Application Express 2.1.0.0.39
Copyright © 1999, 2006, Oracle. All rights reserved.

b) Create a trigger

Ans:

--Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM [A]

```
create table psngr_audit_log(audit_time timestamp,operation varchar2(250),details
varchar2(250))
```

```
CREATE OR REPLACE TRIGGER log_psngr_inserts
```

```
AFTER INSERT ON psngr
```

```
BEGIN
```

```
  INSERT INTO psngr_audit_log (audit_time, operation, details)
```

```
    VALUES (SYSTIMESTAMP, 'INSERT', 'New passenger records inserted.');
```

```
END
```

```
insert into psngr values('p007', 'Isty', 500089087, 177773789, 'isty@gmail.com', 20, 'Male', 'ctg', ,005)
```

```
select * from psngr_audit_log
```

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL code entered is:

```
--Metro rail management system, 10th semester, ADVANCE DATABASE MANAGEMENT SYSTEM [A]
create table psngr_audit_log(audit_time timestamp,operation varchar2(250),details varchar2(250))
CREATE OR REPLACE TRIGGER log_psngr_insert
AFTER INSERT ON psngr
BEGIN
  INSERT INTO psngr_audit_log (audit_time, operation, details)
  VALUES (SYSTIMESTAMP, 'INSERT', 'New passenger records inserted.');
END
insert into psngr values('p007', 'Isty', 500089087, 177773789, 'isty@gmail.com', 20, 'Male', '5tr', 005)

select * from psngr_audit_log
```

The results section shows a single row in the audit log:

AUDIT_TIME	OPERATION	DETAILS
14-DEC-24 02:17:22.861000 AM	INSERT	New passenger records inserted

1 rows returned in 0.00 seconds

CSV Export

Application Express 2.1.0.0.39
Copyright © 1999, 2006, Oracle. All rights reserved.

Package

- a) Create a package that contains a procedure which can update passenger email id.

Ans:

```
CREATE OR REPLACE PACKAGE psngr_pkg IS
  PROCEDURE update_email(p_id_input VARCHAR2, new_email VARCHAR2);
END psngr_pkg;
```

```
CREATE OR REPLACE PACKAGE BODY psngr_pkg IS
  PROCEDURE update_email(p_id_input VARCHAR2, new_email VARCHAR2) IS
  BEGIN
    UPDATE Psngr
    SET email = new_email
    WHERE p_id = p_id_input;
  END;
END psngr_pkg;
```

```

CREATE OR REPLACE PACKAGE psngr_pkg IS
  PROCEDURE update_email(p_id_input VARCHAR2, new_email VARCHAR2);
END psngr_pkg;

CREATE OR REPLACE PACKAGE BODY psngr_pkg IS
  PROCEDURE update_email(p_id_input VARCHAR2, new_email VARCHAR2) IS
  BEGIN
    UPDATE Psngr
    SET email = new_email
    WHERE p_id = p_id_input;
  END;
END psngr_pkg;

```

Results Explain Describe Saved SQL History

Package created.

0.03 seconds

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

```

CREATE OR REPLACE PACKAGE psngr_pkg IS
  PROCEDURE update_email(p_id_input VARCHAR2, new_email VARCHAR2);
END psngr_pkg;

CREATE OR REPLACE PACKAGE BODY psngr_pkg IS
  PROCEDURE update_email(p_id_input VARCHAR2, new_email VARCHAR2) IS
  BEGIN
    UPDATE Psngr
    SET email = new_email
    WHERE p_id = p_id_input;
  END;
END psngr_pkg;

```

Results Explain Describe Saved SQL History

Package Body created.

0.01 seconds

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

- b) Create a package that contains a procedure which can update trx status.

Ans:

```

CREATE OR REPLACE PACKAGE trx_pkg IS
  PROCEDURE update_status(trx_id_input VARCHAR2, new_status VARCHAR2);
END trx_pkg;

```

```

CREATE OR REPLACE PACKAGE BODY trx_pkg IS
  PROCEDURE update_status(trx_id_input VARCHAR2, new_status VARCHAR2) IS

```

```

BEGIN
  UPDATE Trx
  SET status = new_status
  WHERE trx_id = trx_id_input;
END;
END trx_pkg;

```

The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands window, the following PL/SQL code is being run:

```

CREATE OR REPLACE PACKAGE trx_pkg IS
  PROCEDURE update_status(trx_id_input VARCHAR2, new_status VARCHAR2);
END trx_pkg;

CREATE OR REPLACE PACKAGE BODY trx_pkg IS
  PROCEDURE update_status(trx_id_input VARCHAR2, new_status VARCHAR2) IS
  BEGIN
    UPDATE Trx
    SET status = new_status
    WHERE trx_id = trx_id_input;
  END;
END trx_pkg;

```

The results pane shows the message "Package created." and a execution time of "0.00 seconds". The bottom status bar indicates the application version is "Application Express 2.1.0.00.39" and the copyright year is "Copyright © 1999, 2006, Oracle. All rights reserved.".

The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands window, the following PL/SQL code is being run:

```

CREATE OR REPLACE PACKAGE trx_pkg IS
  PROCEDURE update_status(trx_id_input VARCHAR2, new_status VARCHAR2);
END trx_pkg;

CREATE OR REPLACE PACKAGE BODY trx_pkg IS
  PROCEDURE update_status(trx_id_input VARCHAR2, new_status VARCHAR2) IS
  BEGIN
    UPDATE Trx
    SET status = new_status
    WHERE trx_id = trx_id_input;
  END;
END trx_pkg;

```

The results pane shows the message "Package Body created." and a execution time of "0.01 seconds". The bottom status bar indicates the application version is "Application Express 2.1.0.00.39" and the copyright year is "Copyright © 1999, 2006, Oracle. All rights reserved.".

Exception handling

- a) How can you handle a situation where a query retrieves no rows (e.g., no passenger is found with a specific (p_id) and display an appropriate message?

Ans:

```
DECLARE
    v_name VARCHAR2(50);
    v_email VARCHAR2(50);
BEGIN
    SELECT name, email INTO v_name, v_email FROM psngr WHERE p_id = 'p010';
    DBMS_OUTPUT.PUT_LINE('Passenger Name: ' || v_name || ', Email: ' || v_email);
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('No passenger found with the provided ID.');
END;
```

The screenshot shows the Oracle Application Express SQL Commands interface. The SQL code entered is:

```
DECLARE
    v_name VARCHAR2(50);
    v_email VARCHAR2(50);
BEGIN
    SELECT name, email INTO v_name, v_email FROM psngr WHERE p_id = 'p010';
    DBMS_OUTPUT.PUT_LINE('Passenger Name: ' || v_name || ', Email: ' || v_email);
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('No passenger found with the provided ID.');
END;
```

The results pane shows the output:

```
No passenger found with the provided ID.
```

Statement processed.

0.00 seconds

At the bottom, the status bar indicates: Application Express 2.1.0.00.39, Copyright © 1999, 2006, Oracle. All rights reserved.

- b) How can you handle a divide-by-zero error when calculating an average balance in a scenario where the total number of cards might be zero?

Ans:

```
DECLARE
    v_total_balance NUMBER := 1000;
    v_card_count NUMBER := 0;
    v_average_balance NUMBER;
BEGIN
    v_average_balance := v_total_balance / v_card_count;
    DBMS_OUTPUT.PUT_LINE('Average Balance: ' || v_average_balance);
EXCEPTION
```

```

WHEN ZERO_DIVIDE THEN
    DBMS_OUTPUT.PUT_LINE('Error: Division by zero is not allowed.');
END;

```

The screenshot shows the Oracle Application Express SQL Commands interface. The user is SCOTT. The code entered is:

```

DECLARE
    v_total_balance NUMBER := 1000;
    v_card_count NUMBER := 0;
    v_average_balance NUMBER;
BEGIN
    v_average_balance := v_total_balance / v_card_count;
    DBMS_OUTPUT.PUT_LINE('Average Balance: ' || v_average_balance);
EXCEPTION
    WHEN ZERO_DIVIDE THEN
        DBMS_OUTPUT.PUT_LINE('Error: Division by zero is not allowed.');
END;
/

```

The results show the error message:

```

Error: Division by zero is not allowed.
Statement processed.
0.01 seconds

```

At the bottom, it says "Language: en-us" and "Application Express 2.1.0.00.39 Copyright © 1999, 2006, Oracle. All rights reserved". The system tray at the bottom right shows the date and time.

Implicit Locking

- a) How can you lock specific rows in the station table to prevent other users from updating or deleting them while ensuring you can update them?

Ans:

```

SELECT sttn_name, loc
FROM station
WHERE stn_id = 's001'
FOR UPDATE;

```

User SCOTT

Home > SQL > SQL Commands

Autocommit Display 50 Save Run

```
SELECT sttn_name, loc
FROM station
WHERE stn_id = 's001'
FOR UPDATE;
```

Results Explain Describe Saved SQL History

STTN_NAME	LOC
Uttara north	uttara

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

Application Express 2.1 0.00.39
Copyright © 1999, 2006, Oracle. All rights reserved.

Language: en-us

Type here to search

AW01 -1.24% 11:20 PM 1/27/2025

- b) How can you ensure no other user modifies a specific route while updating its name in the route table?

Ans:

UPDATE route

SET rou_name = 'New Route Name'
WHERE rou_id = 'rou001';

User SCOTT

Home > SQL > SQL Commands

Autocommit Display 50 Save Run

```
SELECT sttn_name, loc
FROM station
WHERE stn_id = 's001'
FOR UPDATE;
```

```
UPDATE route
SET rou_name = 'New Route Name'
WHERE rou_id = 'rou001';
```

Results Explain Describe Saved SQL History

1 row(s) updated.

0.00 seconds

Application Express 2.1 0.00.39
Copyright © 1999, 2006, Oracle. All rights reserved.

Language: en-us

Type here to search

AW01 -1.24% 11:20 PM 1/27/2025

Explicit Locking

- a) How can you lock the entire station table to prevent all modifications, allowing only SELECT operations?

Ans:

LOCK TABLE station IN EXCLUSIVE MODE NOWAIT;

The screenshot shows the Oracle Application Express SQL Commands interface. The code entered is:

```
DBMS_OUTPUT.PUT_LINE('Passenger Name: ' || v_name || ', Email: ' || v_email);
EXCEPTION
  WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE('No passenger found with the provided ID.');
END;

LOCK TABLE station IN EXCLUSIVE MODE NOWAIT;
```

The results section shows the message "Statement processed." and a duration of "0.00 seconds". At the bottom, it displays the operating system taskbar with various icons and the date/time "1/27/2025 11:17 PM".

- b) How can you allow multiple users to read or update rows in the train table while preventing others from locking the table exclusively?

Ans:

LOCK TABLE train IN ROW SHARE MODE;

The screenshot shows the Oracle Application Express SQL Commands interface. The code entered is identical to the previous one, but the last line is changed to:

```
LOCK TABLE train IN ROW SHARE MODE;
```

The results section shows the message "Statement processed." and a duration of "0.00 seconds". At the bottom, it displays the operating system taskbar with various icons and the date/time "1/27/2025 11:18 PM".

Relational Algebra

- a) Retrieve all passenger names and the card type associated with them.

Relational Algebra: $\pi_{\text{name}, \text{type}}(\text{psngr} \bowtie \text{Card})$

Query: SELECT p.name, c.type FROM psngr p JOIN Card c ON p.card_id = c.card_id;

```
SELECT p.name, c.type
FROM psngr p
JOIN Card c ON p.card_id = c.card_id;
```

NAME	TYPE
Asir	MRT
Afshan	RAPID
Suma	REGULAR
Preo	REGULAR
Iisty	MRT

5 rows returned in 0.01 seconds [CSV Export](#)

- b) Find the ticket type and price for all tickets purchased via 'BKASH'.

Relational Algebra: $\pi_{\text{tic_type}, \text{price}}(\sigma_{\text{payment}='BKASH'}(\text{Tic}))$

Query: SELECT tic_type, price FROM Tic WHERE payment = 'BKASH';

```
SELECT tic_type, price FROM Tic WHERE payment = 'BKASH';
```

TIC_TYPE	PRICE
MRT	100
RAPID	100
MRT	100

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

- c) List all staff members who work in the 'morning' shift along with their assigned station names.

Relational Algebra: $\pi_{s_name, stn_name}(\sigma_{shift='morning'}(Staff \bowtie Station))$

Query: SELECT s.s_name, st.stn_name FROM Staff s JOIN Station st ON s.assign_stnid = st.stn_id WHERE s.shift = 'morning';

```
SELECT s.s_name, st.stn_name FROM Staff s JOIN Station st ON s.assign_stnid = st.stn_id WHERE s.shift = 'morning';

Results
S_NAME STNN_NAME
farha Uttara north
arina Uttara south
rina motjheel
nisha Uttara north

3 rows returned in 0.00 seconds CSV Export
```

- d) Retrieve the names of routes that start from stations in 'uttara'.

Relational Algebra: $\pi_{rou_name}(\sigma_{loc='uttara'}(Station) \bowtie Route)$

Query: SELECT r.rou_name FROM Route r JOIN Station s ON r.startstn_id = s.stn_id WHERE s.loc = 'uttara';

```
SELECT r.rou_name FROM Route r JOIN Station s ON r.startstn_id = s.stn_id WHERE s.loc = 'uttara';

Results
ROU_NAME
uttara north-motjheel
uttara south-motjheel
uttara south-mirpur 10
uttara north-mirpur 10

4 rows returned in 0.02 seconds CSV Export
```

- e) Find the passengers who have unpaid transactions.

Relational Algebra: $\pi_{\text{name}, \text{status}}(\sigma_{\text{status}=\text{'unpaid'}}(\text{Trx} \bowtie \text{Psngr}))$

Query: `SELECT p.name, t.status FROM Psngr p JOIN Trx t ON p.p_id = t.p_id WHERE t.status = 'unpaid';`

The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands window, the following query is entered:

```
SELECT p.name, t.status FROM Psngr p JOIN Trx t ON p.p_id = t.p_id WHERE t.status = 'unpaid';
```

The results pane displays a single row of data:

NAME	STATUS
Pno	unpaid

1 rows returned in 0.02 seconds [CSV Export](#)

At the bottom right, the status bar shows "Application Express 2.1 0.00.39" and "Copyright © 1999, 2008, Oracle. All rights reserved."

Conclusion

The Metro Rail Management System is designed to revolutionize metro operations by delivering a streamlined, efficient, and user-centric platform. It addresses critical operational areas such as ticketing, scheduling, route management, and staff coordination. By integrating real-time updates, intuitive UI design, and robust data management features, the system ensures seamless experience for both passengers and administrators.

The project focuses on providing:

Enhanced passenger convenience in ticketing and journey planning. Operational efficiency for metro administrators through effective management tools. Scalable and accessible system that can be adapted to diverse metro networks. This system bridges the gap between passenger needs and operational demands, offering a sustainable solution to modernize metro station management.