AMERICAN INTERNATIONAL UNIVERSITY BANGLADESH (AIUB)

FACULTY OF SCIENCE & TECHNOLOGY

Course Title

INTRODUCTION TO DATABASE (2108)

Semester: SPRING 2023-2024

Section: [V]

TITLE

The Online Train Reservation System

Supervised By

Noboranjan Dey

Submitted By: Group no: 03

Name	ID
1-ROMAN SHEIKH RAZAN	**23-50460-1**
2- MD. MOBASHIR TAJUARE PARTHO	**23-50006-1**
3-NISHI BHOWMICK	**23-50919-1**
4-MD RAFIT	**23-50418-1**

TABLE OF CONTENTS

Introduction

The Online Train Reservation System offers a convenient solution for booking train tickets, eliminating the need for physical counters. Users can register securely, access real-time train schedules and seat availability, select seats, and make payments through a secure gateway. E-tickets are generated instantly upon successful booking, and users can manage their profiles. The system also includes an admin dashboard for managing schedules, seats, user accounts, and generating reports.

Case Study / Scenario

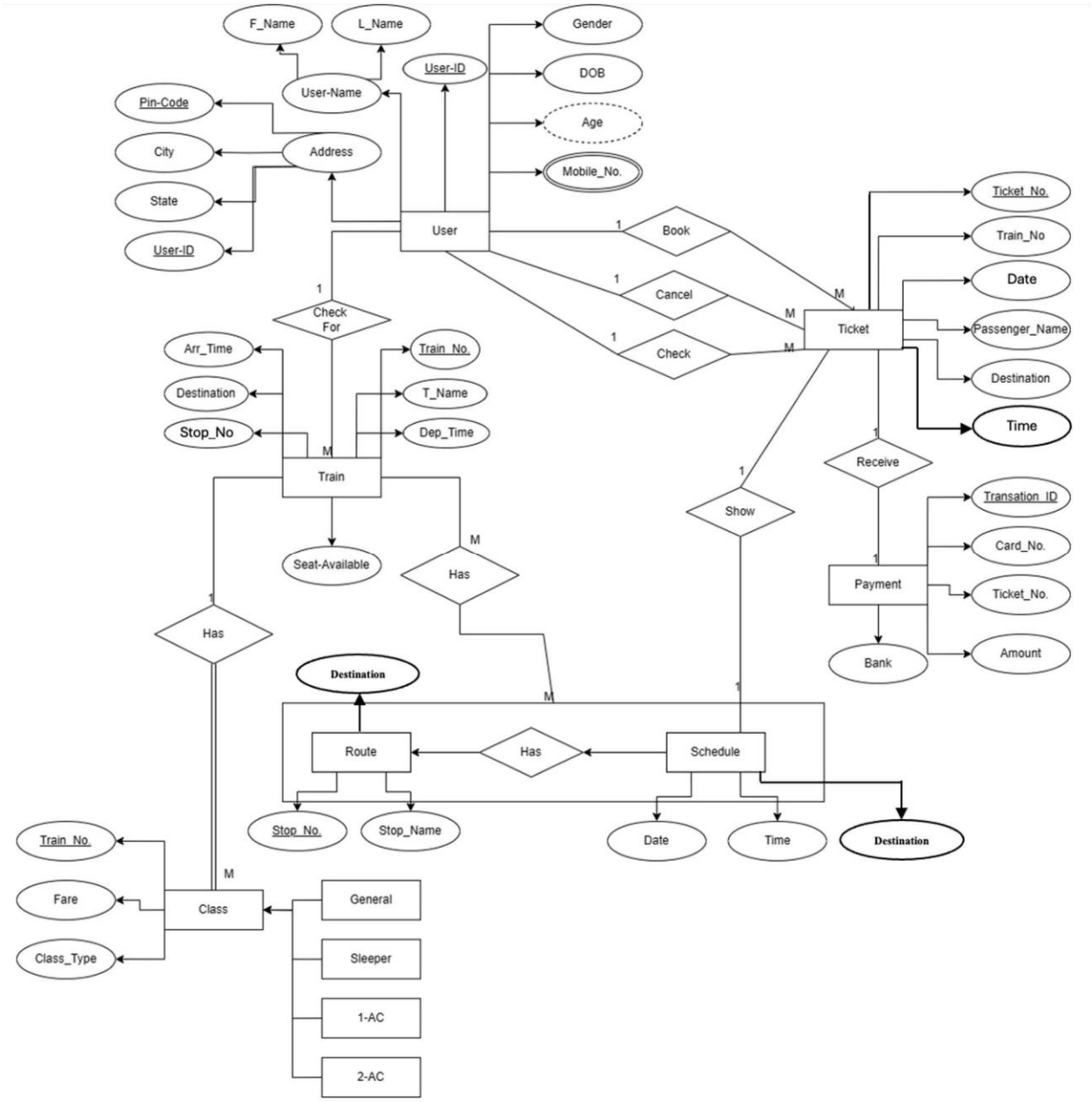
StudentID1: 23-50460-1 Name: ROMAN SHEIKH RAZAN	StudentID3: 23-50919-1 Name: NISHI BHOWMICK
StudentID2: 23-50006-1 Name:MD.MOBASHIR TAJUARE PARTHO	StudentID4: 23-50418-1 Name: MD RAFIT
CO2: Understand the fundamental concepts underlying database systems and gain hands-on experience with ER diagram Case study	
PO-c2: Develop process for complex computer science and engineering problems considering cultural and societal factors.	Marks

The Online Train Reservation System

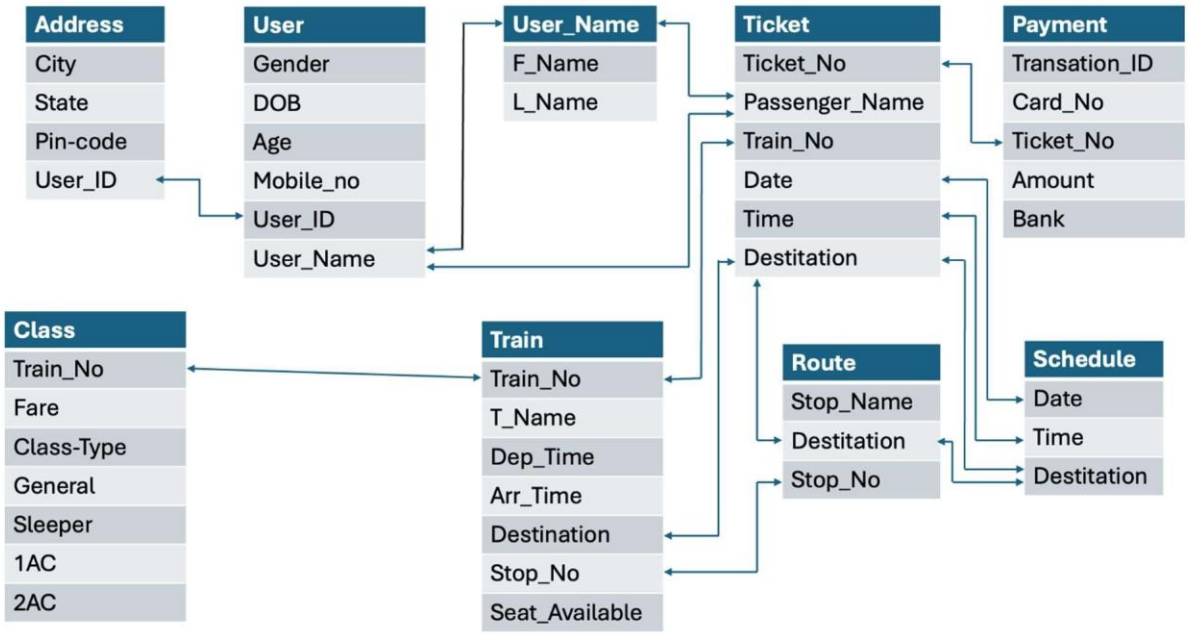
In The Online Train Reservation System, a user may book many tickets. One ticket may be booked by exactly one user. Every user is identified by each User ID. The system also stores Date of Birth, Gender, Username, Address, Age and Mobile number. Age is the derived Attribute. Every user address is composed of city, state, Pin code and User ID. Every username is composed of first name and last name. Every user has a multivalued attribute as mobile number. User can check the ticket. Every ticket is identified by each ticket no. The system also stores train no, Datetime, passenger name and destination. A user may cancel many tickets. One ticket may be cancelled by exactly at least one User. Each ticket received a payment. Payment is identified by a transition ID. The system also stores amounts, card number, bank and ticket number. Every ticket shows schedule. The schedule store date, time and destination. Every Route has a specific schedule. The route stores the stop number, stop name and destination. A user can check for train information. The train is identified by train no. The system also stores train name, Dep time, Distance, seat available, arrive time, destination and stop number. Every train has a route and schedule. Every train has many classes. The system store class type, fare and train no. Class can be specialized in general, sleeper, 1AC and 2AC .

ER Diagram

The Online Train Reservation System



Scheme diagram:



Normalization

Normalization:

Check For Relation: One to Many

UNF: Gender, DOB, Age, Mobile No, User ID, F_Name, L_name, Adress, PinCode, City, State, UserID,Arr_Time, Destination, Stop_No, Train_NO, T_name, Dep_Time,Seat-Available

1NF: Gender, DOB, Age, Mobile No, User ID, F_Name, L_name, Adress, PinCode, City, State,Arr_Time, Destination, Stop_No, Train_NO, T_name, Dep_Time, Seat-Available

2NF:

1st: Gender, DOB, Age, Mobile No, User ID, F_Name, L_name, Adress, PinCode, City, State

2nd: Mobile No, User ID

3rd: Arr_Time, Destination, Stop_No, Train_NO, T_name, Dep_Time, Seat-Available, User ID

3NF:

1st: Gender, DOB, Age, User ID, F_Name, , PinCode

2nd: Mobile No, User ID

3rd: Arr_Time, Destination, Stop_No, Train_NO, T_name, Dep_Time, Seat-Available, User ID

4th: F_Name, L_name

5th: Adress, PinCode, City, State,

Book (Cancel,Check) Relation: One to Many

UNF: Gender, DOB, Age, Mobile No, User ID, F_Name, L_name, Adress, PinCode, City, State , Ticket_NO, Train_NO, Date, Passenger_Name, Destination, Time

1NF: Gender, DOB, Age, Mobile No, User ID, F_Name, L_name, Adress, PinCode, City, State , Ticket_NO, Train_NO, Date, Passenger_Name, Destination, Time

2NF:

1st: Gender, DOB, Age, Mobile No, User ID, F_Name, L_name, Adress, PinCode, City, State

2nd: Ticket_NO, Train_NO, Date, Passenger_Name, Destination, Time, UserID 3rd:

Mobile No, User ID

3NF:

1st: Gender, DOB, Age, Mobile No, User ID, F_Name, Adress, PinCode

2nd: Ticket_NO, Train_NO, Date, Passenger_Name, Destination, Time, UserID

3rd: Mobile No, User ID

4th: F_Name, L_name

5th: PinCode, City, State

Receive Relation: One to One
UNF: Ticket_No,Train_No,Date,Passanger_Name,Destination,Time,Transection_ID,Card_No, Ticket_no,Amount,Bank

1NF: Ticket_No,Train_No,Date,Passanger_Name,Destination,Time,Transection_ID,Card_No, Ticket_no,Amount,Bank

2NF:

1st: Ticket_No,Train_No,Date,Passanger_Name,Destination,Time

2nd: Transection_ID,Card_No,Ticket_no,Amount,Bank

3NF: 1st: Ticket_No,Train_No,Date,Passanger_Name,Destination,Time

2nd: Transection_ID,Card_No,Ticket_no,Amount,Bank

Show Relation:One to One
UNF: Ticket_No,Train_No,Date,Passanger_Name,Destination,Time,Date,Time,Destination 1NF: Ticket_No,Train_No,Date,Passanger_Name,Destination,Time

2NF:

1st: Ticket_No,Train_No,Date,Passanger_Name

2nd: Date, Destination,Time

3rd: Date , Ticket_No

3NF:

1st: Ticket_No,Train_No,Date,Passanger_Name

2nd: Date, Destination,Time

3rd: Date , Ticket_No

Has Relation:

UNF: Date, Time, Destination ,StopNO, Stop_name

1NF: Date, Time, Destination ,StopNO, Stop_name

2NF:

1st: Date, Time, Desition

2nd: StopNO, Stop_name

3rd: Date , StopNO

3NF:

1st: Date, Time, Desition

2nd: StopNO, Stop_name

3rd: Desition, StopNO

Has Relation: Many to Many

UNF: Arr_Time, Destination, Stop_No, Train_NO, T_name, Dep_Time, Seat-Available, Date, Time, Destination ,StopNO, Stop_name

1NF: Arr_Time, Destination, Stop_No, Train_NO, T_name, Dep_Time, Seat-Available, Date, Time, Destination ,StopNO, Stop_name

2NF:
1st: Arr_Time, Destination, Stop_No, Train_NO, T_name, Dep_Time, Seat-Available 2nd:

Date, Time, Destination ,StopNO, Stop_name, Train_NO

3NF:
1st: Arr_Time, Destination, Stop_No, Train_NO, T_name, Dep_Time, Seat-Available 2nd:

Date, Time, Destination ,StopNO, Stop_name, Train_NO

Has Relation: One to Many
UNF: Arr_Time, Destination, Stop_No, Train_NO, T_name, Dep_Time, Seat-Available,Train_No,Fare,Class_Type,General, Sleeper,1-AC,2-AC

1NF: Arr_Time, Destination, Stop_No, Train_NO, T_name, Dep_Time,

Seat_Available,Train_No,Fare,Class_Type,General, Sleeper,1-AC,2-AC

2NF:

1st: Arr_Time, Destination, Stop_No, Train_No, T_name, Dep_Time,

Seat_Available

2nd: Train_No,Fare,Class_Type,General, Sleeper,1-AC,2-AC

3rd: Train_No, Class_Type

3NF:

1st: Arr_Time, Destination, Stop_No, Train_No, T_name, Dep_Time,

Seat_Available

2nd: Train_No,Fare,Class_Type,General, Sleeper,1-AC,2-AC

3rd: Train_No, Class_Type

Finalization

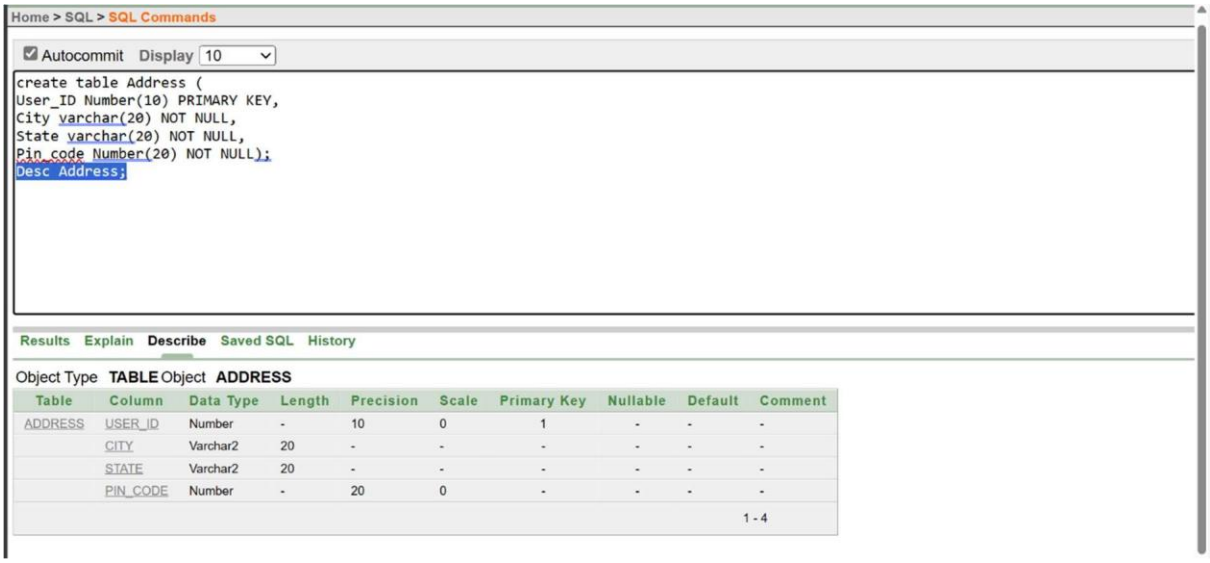
Finalization:

1. Gender, DOB, Age, User ID, F_Name, , PinCode
2. Mobile No, User ID
3. Arr_Time, Destination, Stop_No, Train_NO, T_name, Dep_Time, Seat-Available, User ID
4. F_Name, L_name
5. Adress, PinCode, City, State,
6. Gender, DOB, Age, Mobile No, User ID, F_Name, Adress, PinCode
7. Ticket_NO, Train_NO, Date, Passenger_Name, Destination, Time, UserID
8. ~~Mobile No, User ID~~
9. ~~F_Name, L_name~~
10. ~~PinCode, City, State~~
11. Ticket_No,Train_No,Date,Passanger_Name,Destination,Time
12. Transection_ID,Card_No,Ticket_no,Amount,Bank
13. ~~Ticket_No,Train_No,Date,Passanger_Name~~
14. Date, Destination,Time
15. Date , Ticket_No
16. ~~Date, Time, Desition~~
17. StopNO, Stop_name
18. Destination, Stop_NO
19. ~~Arr_Time, Destination, Stop_No, Train_NO, T_name, Dep_Time, Seat-Available~~
20. Date, Time, Destination ,StopNO, Stop_name, Train_NO
21. ~~Arr_Time, Destination, Stop_No, Train_No, T_name, Dep_Time, Seat_Available~~
22. Train_No,Fare,Class_Type,General, Sleeper,1-AC,2-AC
23. Train_No, Class_Type

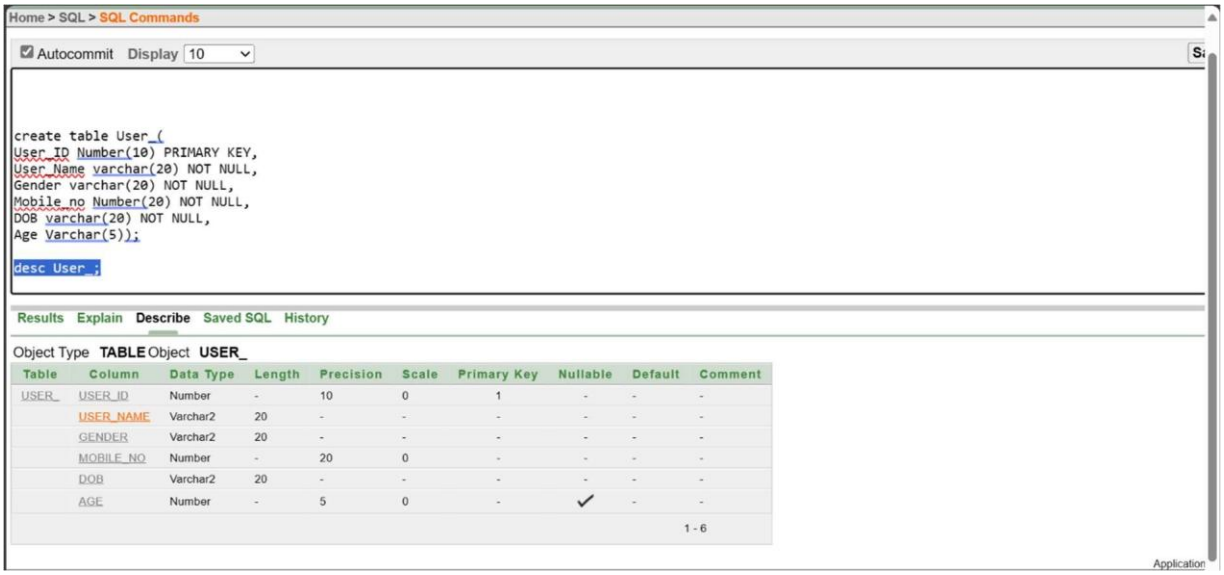
Table Creation (DDL Operations)

StudentID1: 23-50460-1 Name: ROMAN SHEIKH RAZAN	StudentID3: 23-50919-1 Name:NISHI BHOWMICK
StudentID2: 23-50006-1 Name: MD.MOBASHIR TAJUARE PARTHO	StudentID4: 23-50418-1 Name: MD RAFIT
CO4: Creating DML, DDL using Oracle and connection with ODBC/JDBC for existing JAVA application	
PO-e-2: Use modern engineering and IT tools for prediction and modeling of complex computer science and engineering problem	Marks

1. Adress Table Creation and Descrived:



2. User_ Table Creation and Descrived:



3. User_Name Table Creation and Descrived:

Home > SQL > SQL Commands

Autocommit Display 10

```
create table User_Name(
 F_Name varchar(20),
L_Name varchar(20));

desc User_Name;
```

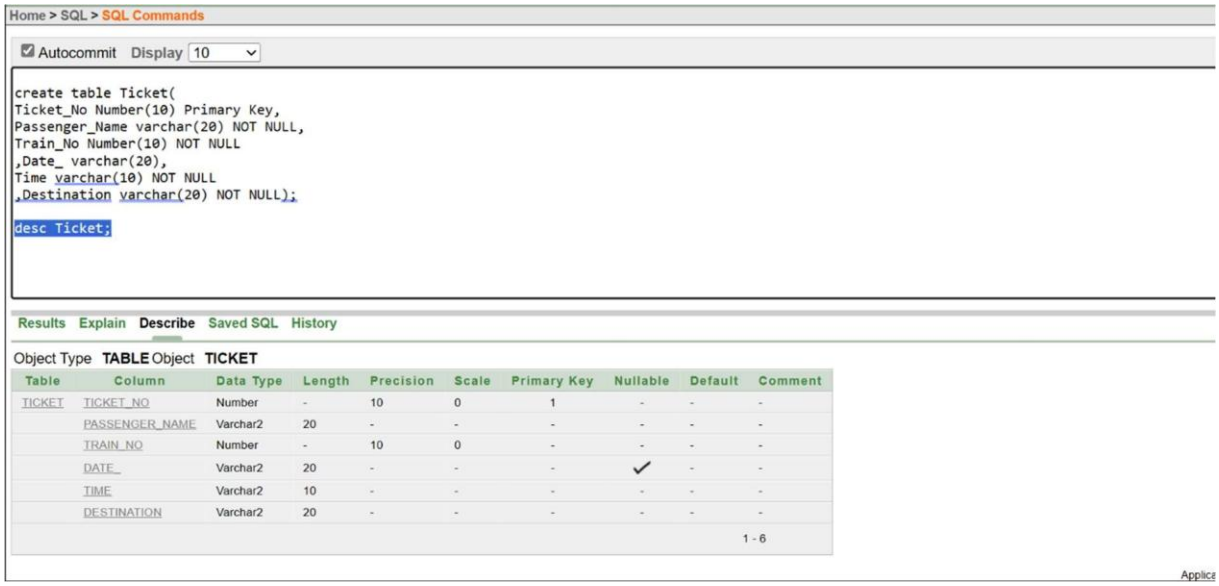
Results Explain Describe Saved SQL History

Object Type TABLE Object USER_NAME

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
USER_NAME	F_NAME	Varchar2	20	-	-	-	✓	-	-
	L_NAME	Varchar2	20	-	-	-	✓	-	-

1 - 2

4. Ticket Table Creation and Descrived:



Home > SQL > SQL Commands

Autocommit Display 10

```
create table Ticket(
Ticket_No Number(10) Primary Key,
Passenger_Name varchar(20) NOT NULL,
Train_No Number(10) NOT NULL
,Date_ varchar(20),
Time varchar(10) NOT NULL
,Destination varchar(20) NOT NULL);

desc Ticket;
```

Results Explain Describe Saved SQL History

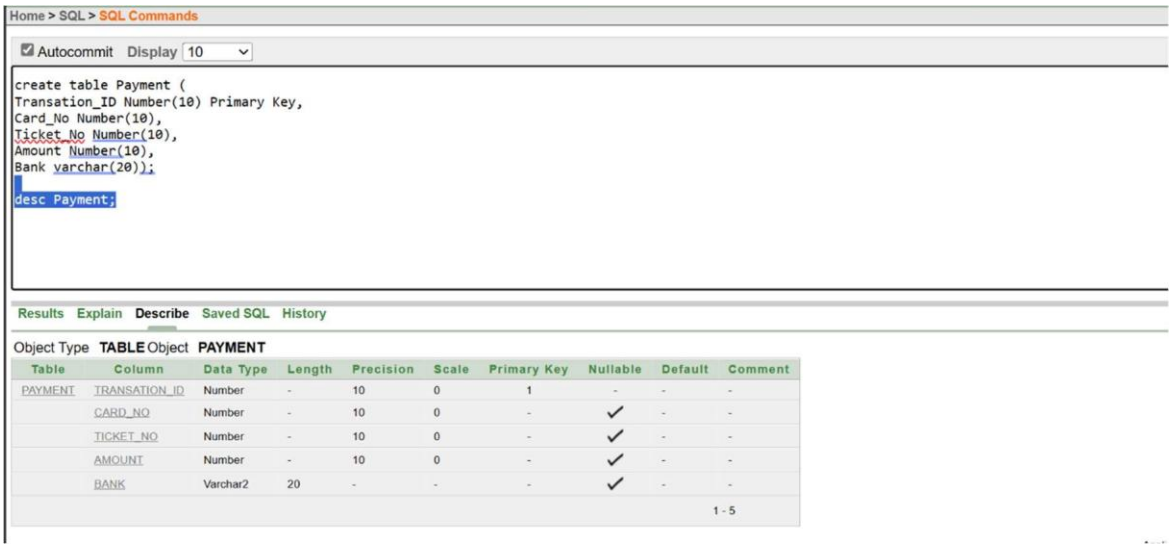
Object Type TABLE Object TICKET

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
TICKET	TICKET_NO	Number	-	10	0	1	-	-	-
	PASSENGER_NAME	Varchar2	20	-	-	-	-	-	-
	TRAIN_NO	Number	-	10	0	-	-	-	-
	DATE_	Varchar2	20	-	-	-	✓	-	-
	TIME	Varchar2	10	-	-	-	-	-	-
	DESTINATION	Varchar2	20	-	-	-	-	-	-

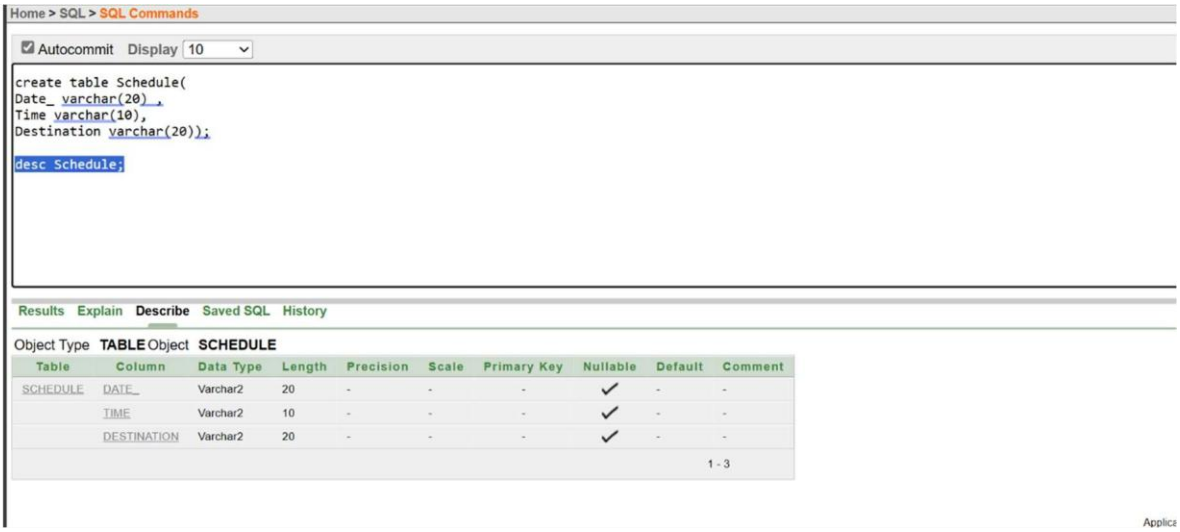
1 - 6

Applica

5. Payment Table Creation and Descrived:

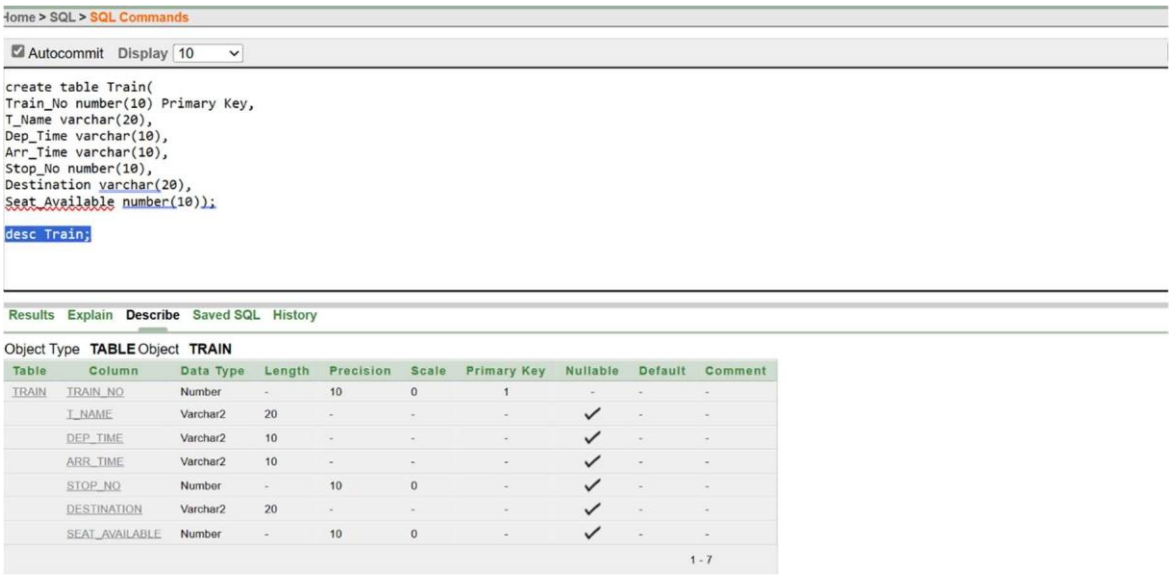


6. Schedule Table Creation and Descrived:

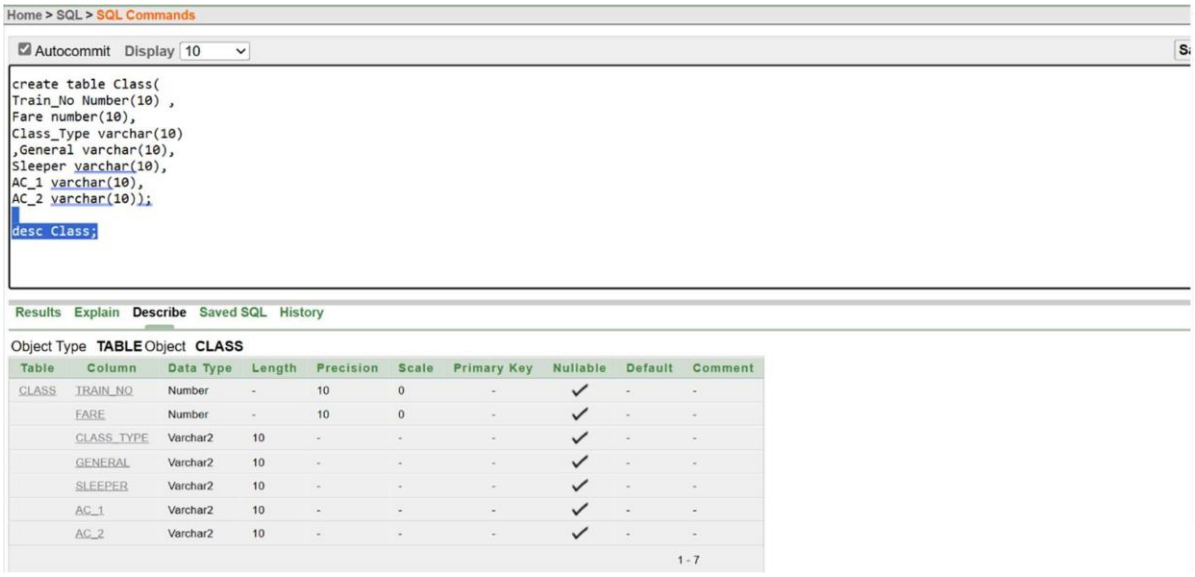


7. Route Table Creation and Descrived:

8. Train Table Creation and Descrived:



9. Class Table Creation and Descrived:



Inserted Values in the tables

1. Address value Insertion insert into Address values (1775,'DHAKA','DHAKA',1100); insert into Address values (1855,'NATOR','RAJSHAHI',1400); insert into Address values (1900,'SUNAMGONJ','SYLHET',1300); insert into Address values (1943,'COXBAZAR','CHITTAGONG',1900); insert into Address values (1628,'NOWPARA','RANGPUR',1500); insert into Address values (2015,'BEANPOLE','KHULNA',1200); insert into Address values (2189,'KUAKATA','BARISAL',2000);

```
select*from Address;
```

Results Explain Describe Saved SQL History

USER_ID	CITY	STATE	PIN_CODE
1775	DHAKA	DHAKA	1100
1855	NATOR	RAJSHAHI	1400
1900	SUNAMGONJ	SYLHET	1300
1943	COXBAZAR	CHITTAGONG	1900
1628	NOWPARA	RANGPUR	1500
2015	BEANPOLE	KHULNA	1200
2189	KUAKATA	BARISAL	2000

7 rows returned in 0.00 seconds CSV Export

2. User_ value Insertion insert into User_ values (1775,'PARTHO','MALE',01842471248,'15-NOV-2001',0); insert into User_ values (1855,'HRID','MALE',01954762318,'27-APR-2003',0); insert into User_ values (1900,'NISHI','FEMALE',01624359765,'11-JUN-2002',0); insert into User_ values (1943,'RAFIT','MALE',01357845623,'31-FEB-2001',0); insert into User_ values (1628,'RAZON','MALE',01524851953,'23-AUG-2000',0); insert into User_ values (2015,'MUNIRA','FEMALE',01424568752,'05-DEC-

2004',0); insert into User_ values (2189,'MAHI','FEMALE',01775612056,'20-MAY-2003',0);

```
select*from User_;
```

Results Explain Describe Saved SQL History

USER_ID	USER_NAME	GENDER	MOBILE_NO	DOB	AGE
1775	MD.PARTHO	MALE	1842471248	2001-11-15	0
1855	HRID HOSSAIN	MALE	1954762318	2003-04-27	0
1900	NISHI BHOWMICK	FEMALE	1624359765	2002-06-11	0
1943	TISHAN RAFIT	MALE	1357845623	2001-02-11	0
1628	RS RAZON	MALE	1524851953	2000-08-23	0
2015	MUNIRA MISHU	FEMALE	1424568752	2004-12-05	0
2189	MRS.MAHI	FEMALE	1775612056	2003-05-20	0

7 rows returned in 0.00 seconds CSV Export

3. User_Name value Insertion

Insert into User_Name values ('MD.','PARTHO');

Insert into User_Name values ('HRID','HOSSAIN');

Insert into User_Name values ('NISHI','BHOWMICK');

Insert into User_Name values ('TISHAN','RAFIT');

Insert into User_Name values ('RS','RAZON');

Insert into User_Name values ('MUNIRA','MISHU');

Insert into User_Name values ('MRS.','MAHI');

```
SELECT* FROM User_Name;
```

Results Explain Describe Saved SQL History

F_NAME	L_NAME
MD.	PARTHO
HRID	HOSSAIN
NISHI	BHOWMICK
TISHAN	RAFIT
RS	RAZON
MUNIRA	MISHU
MRS.	MAHI

7 rows returned in 0.00 seconds CSV Export

4. Ticket value Insertion

Insert into Ticket values (12568,'MD.PARTHO',5525,'15-MAY-2024','10:30','BARISAL');

Insert into Ticket values (22504,'HRID HOSSAIN',5530,'30-MAY-2024','5:30','CHITTAGONG');

Insert into Ticket values (35126,'NISHI BHOWMICK',5533,'05-MAY-2024','7:00','RAJSHAHI');

Insert into Ticket values (45129,'TISHAN RAFIT',5533,'05-MAY-2024','7:00','RAJSHAHI');

Insert into Ticket values (54261,'RS RAZON',5530,'30-MAY-2024','5:30','CHITTAGONG');

Insert into Ticket values (67842,'MUNIRA MISHU',5511,'23-MAY-2024','1.30','DHAKA');

Insert into Ticket values (86499,'MRS.MAHI',5525,'15-MAY-2024','10:30','BARISAL');

```
SELECT* FROM Ticket;
```

Results Explain Describe Saved SQL History

TICKET_NO	PASSENGER_NAME	TRAIN_NO	DATE_	TIME	DESTINATION
12568	MD.PARTHO	5525	15-MAY-2024	10:30	BARISAL
86499	MRS.MAHI	5525	15-MAY-2024	10:30	BARISAL
67842	MUNIRA MISHU	5511	23-MAY-2024	1.30	DHAKA
54261	RS RAZON	5530	30-MAY-2024	5:30	CHITTAGONG
22504	HRID HOSSAIN	5530	30-MAY-2024	5:30	CHITTAGONG
35126	NISHI BHOWMICK	5533	05-MAY-2024	7:00	RAJSHAHI
45129	TISHAN RAFIT	5533	05-MAY-2024	7:00	RAJSHAHI

7 rows returned in 0.00 seconds CSV Export

5. Payment value Insertion

Insert into Payment values (1005621,8845,12568,600,'ISLAMIC');

Insert into Payment values (1005489,6616,22504,800,'BRAC');

Insert into Payment values (1005945,4434,35126,1000,'CITY');

Insert into Payment values (1005750,8899,45129,1000,'ISLAMIC');

Insert into Payment values (1005012,4489,54261,800,'CITY');

Insert into Payment values (1005500,6636,67842,900,'BRAC');

Insert into Payment values (1005244,1191,86499,600,'DHAKA');

```
SELECT* FROM Payment;
```

Results Explain Describe Saved SQL History

TRANSATION_ID	CARD_NO	TICKET_NO	AMOUNT	BANK
1005621	8845	12568	600	ISLAMIC
1005489	6616	22504	800	BRAC
1005945	4434	35126	1000	CITY
1005750	8899	45129	1000	ISLAMIC
1005012	4489	54261	800	CITY
1005500	6636	67842	900	BRAC
1005244	1191	86499	600	DHAKA

7 rows returned in 0.00 seconds CSV Export

6. Schedule value Insertion

Insert into Schedule values ('15-MAY-2024','10:30','BARISAL');

Insert into Schedule values ('30-MAY-2024','5:30','CHITTAGONG');

Insert into Schedule values ('05-MAY-2024','7:00','RAJSHAHI');

Insert into Schedule values ('23-MAY-2024','1.30','DHAKA');

```
SELECT* FROM Schedule;
```

Results Explain Describe Saved SQL History

DATE_	TIME	DESTINATION
15-MAY-2024	10:30	BARISAL
30-MAY-2024	5:30	CHITTAGONG
05-MAY-2024	7:00	RAJSHAHI
23-MAY-2024	1.30	DHAKA

4 rows returned in 0.00 seconds CSV Export

7. Route value Insertion

Insert into Route values ('DHAKA',101,'BARISAL');

Insert into Route values ('KHULNA',102,'CHITTAGONG');

Insert into Route values ('BARISAL',103,'RAJSHAHI');

Insert into Route values ('CHITTAGONG',104,'DHAKA');

```
SELECT* FROM Route;
```

Results Explain Describe Saved SQL History

STOP_NAME	STOP_NO	DESTINATION
DHAKA	101	BARISAL
KHULNA	102	CHITTAGONG
BARISAL	103	RAJSHAHI
CHITTAGONG	104	DHAKA

4 rows returned in 0.00 seconds CSV Export

8. Train value Insertion

Insert into Train values (5525,'VOIVO EXPRESS','10:30','5:30',101,'BARISAL',60);

Insert into Train values (5530,'DURONTO EXPRESS','5:30','12:45',102,'CHITTAGONG',90);

Insert into Train values (5533,'HIGHFIVE EXPRESS','7:00','2:15',103,'RAJSHAHI',45); Insert into Train values (5511,'MIDNIGHT EXPRESS','1:30','7:15',104,'DHAKA',02);

```
SELECT* FROM Train;
```

Results Explain Describe Saved SQL History

TRAIN_NO	T_NAME	DEP_TIME	ARR_TIME	STOP_NO	DESTINATION	SEAT_AVAILABLE
5525	VOIVO EXPRESS	10:30	5:30	101	BARISAL	60
5530	DURONTO EXPRESS	5:30	12:45	102	CHITTAGONG	90
5533	HIGHFIVE EXPRESS	7:00	2:15	103	RAJSHAHI	45
5511	MIDNIGHT EXPRESS	1:30	7:15	104	DHAKA	2

4 rows returned in 0.00 seconds CSV Export

9. Class value Insertion

Insert into Class values (5525,600,'A','yes',' ',",' ');

Insert into Class values (5525,1200,'A',' ','yes',' ',' ');

Insert into Class values (5525,800,'A',' ',' ','yes',' ');

Insert into Class values (5525,1600,'A',' ',' ',' ','yes');

Insert into Class values (5525,600,'B','yes',' ',",' ');

Insert into Class values (5525,1200,'B',' ','yes',' ',' '); Insert into Class values (5525,800,'B',' ',' ','yes',' ');

Insert into Class values (5525,1600,'B',' ',' ',' ','yes');

```
Insert into Class values (5530,800,'A','yes',' ','','' );
Insert into Class values (5530,1600,'A',' ','yes',' ',' ');
Insert into Class values (5530,1000,'A',' ',' ','yes',' ');
Insert into Class values (5530,2000,'A',' ',' ',' ','yes');
Insert into Class values (5530,800,'B','yes',' ','','' );
Insert into Class values (5530,1600,'B',' ','yes',' ',' '); Insert
into Class values (5530,1000,'B',' ',' ','yes',' '); Insert into
Class values (5530,2000,'B',' ',' ',' ','yes');
Insert into Class values (5533,1000,'A','yes',' ','','' );
Insert into Class values (5533,2000,'A',' ','yes',' ',' ');
Insert into Class values (5533,1200,'A',' ',' ','yes',' ');
Insert into Class values (5533,2400,'A',' ',' ',' ','yes'); Insert
into Class values (5533,1000,'B','yes',' ','','' ); Insert into
Class values (5533,2000,'B',' ','yes',' ',' ');

Insert into Class values (5533,1200,'B',' ',' ','yes',' '); Insert
into Class values (5533,2400,'B',' ',' ',' ','yes');
Insert into Class values (5511,900,'A','yes',' ','','' );
Insert into Class values (5511,1800,'A',' ','yes',' ',' ');
Insert into Class values (5511,1100,'A',' ',' ','yes',' ');
Insert into Class values (5511,2200,'A',' ',' ',' ','yes');
Insert into Class values (5511,900,'B','yes',' ','','' );
Insert into Class values (5511,1800,'B',' ','yes',' ',' '); Insert
into Class values (5511,1100,'B',' ',' ','yes',' '); Insert into
Class values (5511,2200,'B',' ',' ',' ','yes');
```

```
SELECT* FROM class;
```

Results Explain Describe Saved SQL History

TRAIN_NO	FARE	CLASS_TYPE	GENERAL	SLEEPER	AC_1	AC_2
5525	600	A	yes		-	
5525	1200	A		yes		
5525	800	A			yes	
5525	1600	A				yes
5525	600	B	yes		-	
5525	1200	B		yes		
5525	800	B			yes	
5525	1600	B				yes
5530	800	A	yes		-	
5530	1600	A		yes		
5530	1000	A			yes	
5530	2000	A				yes
5530	800	B	yes		-	
5530	1600	B		yes		
5530	1000	B			yes	
5530	2000	B				yes
5533	1000	A	yes		-	
5533	2000	A		yes		
5533	1200	A			yes	
5533	2400	A				yes
5533	1000	B	yes		-	
5533	2000	B		yes		
5533	1200	B			yes	
5533	2400	B				yes
5511	900	A	yes		-	
5511	1800	A		yes		
5511	1100	A			yes	
5511	2200	A				yes
5511	900	B	yes		-	
5511	1800	B		yes		
5511	1100	B			yes	
5511	2200	B				yes

32 rows returned in 0.00 seconds CSV Export

Query Test in DB

A) Simple queary

Q. Show User Name ,Id and Date of Birth (DOB) where mobile_no = 01424568752

```
select User_ID,User_Name,DOB from User_ where mobile_no =(01424568752);
```

Results Explain Describe Saved SQL History

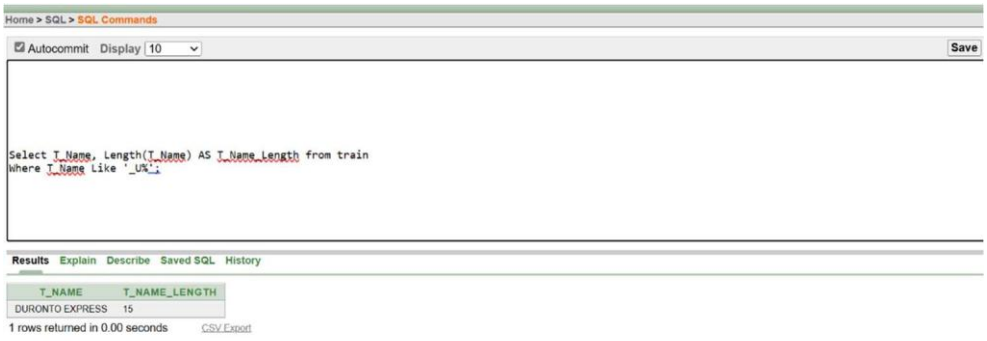
USER_ID	USER_NAME	DOB
2015	MUNIRA MISHU	2004-12-05

1 rows returned in 0.02 seconds CSV Export

B) Single-row function

Q. Find the User_Name where the Letter 'R' in 1st position from User_Name table

Q. Get the length of T_Name form Train table where the Letter 'u' in 2nd position



C) Multiple row function

Q. Show Train_No, Class_Type, Average fare by Train_No & Class_Type from Class

```
select Train_NO, Class_Type, AVG(Fare) from class
group by Train_NO , Class_Type
order by Train_NO;
```

Results Explain Describe Saved SQL History

TRAIN_NO	CLASS_TYPE	AVG(FARE)
5511	A	1500
5511	B	1500
5525	A	1050
5525	B	1050
5530	A	1350
5530	B	1350
5533	A	1650
5533	B	1650

8 rows returned in 0.00 seconds CSV Export

Q. Show total amount each of all Type Bank name form Payment table

```
select Bank,Sum(Amount) from Payment group by Bank;
```

Results Explain Describe Saved SQL History

BANK	SUM(AMOUNT)
ISLAMIC	1600
BRAC	1700
DHAKA	600
CITY	1800

4 rows returned in 0.02 seconds CSV Export

Q. Find the Age of User from User table

```
UPDATE User_
SET Age = FLOOR((SYSDATE - TO_DATE(DOB, 'YYYY-MM-DD')) / 365);
select * from User_;
```

Results Explain Describe Saved SQL History

USER_ID	USER_NAME	GENDER	MOBILE_NO	DOB	AGE
1775	MD.PARTHO	MALE	1842471248	2001-11-15	22
1855	HRID HOSSAIN	MALE	1954762318	2003-04-27	21
1900	NISHI BHOWMICK	FEMALE	1624359765	2002-06-11	21
1943	TISHAN RAFIT	MALE	1357845623	2001-02-11	23
1628	RS RAZON	MALE	1524851953	2000-08-23	23
2015	MUNIRA MISHU	FEMALE	1424568752	2004-12-05	19
2189	MRS.MAHI	FEMALE	1775612056	2003-05-20	20

7 rows returned in 0.00 seconds CSV Export

Subquery:

Q. Select city and state of the address associated with the user_ID 1855?

```
Select City, State
From Address
Where Pin_code = (Select Pin_code From Address Where User_ID = 1855);
```

Results Explain Describe Saved SQL History

CITY	STATE
NATOR	RAJSHAHI

1 rows returned in 0.00 seconds CSV Export

Q. Select user name and gender of the user whose mobile number matches that of the user ID 2189, and whose date of birth is later than that of the user ID 2015?

```
Select User_Name, Gender
From User_
Where Mobile_no = ( Select Mobile_no From User_ Where User_id = 2189)
And DOB > ( Select DOB From User_ Where User_id = 2015);
```

Results Explain Describe Saved SQL History

USER_NAME	GENDER
MRS.MAHI	FEMALE

1 rows returned in 0.00 seconds CSV Export

Q.Select all ticket details for train_no greater than any train_no going to Chittagong, excluding those going to Chittagong?

```
Select * From Ticket
Where Train_no > Any ( Select Train_no From Ticket Where Destination = 'CHITTAGONG')
And Destination <> 'CHITTAGONG';
```

Results Explain Describe Saved SQL History

TICKET_NO	PASSENGER_NAME	TRAIN_NO	DATE_	TIME	DESTINATION
35126	NISHI BHOWMICK	5533	05-MAY-2024	7:00	RAJSHAHI
45129	TISHAN RAFIT	5533	05-MAY-2024	7:00	RAJSHAHI

2 rows returned in 0.00 seconds CSV Export

Q.Select all the details of classes where the fare is higher than the average fare of each class type?

```
Select * From Class
Where Fare > All ( Select avg(Fare) From Class Group By Class_type );
```

Results Explain Describe Saved SQL History

TRAIN_NO	FARE	CLASS_TYPE	GENERAL	SLEEPER	AC_1	AC_2
5530	1600	A		yes		
5530	1600	B		yes		
5530	2000	B				yes
5533	2000	A		yes		
5533	2400	A				yes
5533	2400	B				yes
5511	1800	A		yes		
5511	2200	A				yes
5511	1800	B		yes		
5511	2200	B				yes

10 rows returned in 0.00 seconds CSV Export

Joining:

Q. Select passenger name, ticket number, date of the ticket, date of the schedule, time of the schedule, and destination from the Ticket and Schedule tables?

```
SELECT T.Passenger_name, T.Ticket_no, T.Date_, S.Date_, S.Time, S.Destination
FROM Ticket T, Schedule S;
```

Results Explain Describe Saved SQL History

PASSENGER_NAME	TICKET_NO	DATE_	DATE_	TIME	DESTINATION
MD.PARTHO	12568	15-MAY-2024	15-MAY-2024	10:30	BARISAL
HRID HOSSAIN	22504	30-MAY-2024	15-MAY-2024	10:30	BARISAL
NISHI BHOWMICK	35126	05-MAY-2024	15-MAY-2024	10:30	BARISAL
RS RAZON	54261	30-MAY-2024	15-MAY-2024	10:30	BARISAL
MUNIRA MISHU	67842	23-MAY-2024	15-MAY-2024	10:30	BARISAL
MRS.MAHI	86499	15-MAY-2024	15-MAY-2024	10:30	BARISAL
TISHAN RAFIT	45129	05-MAY-2024	15-MAY-2024	10:30	BARISAL
MD.PARTHO	12568	15-MAY-2024	30-MAY-2024	5:30	CHITTAGONG
HRID HOSSAIN	22504	30-MAY-2024	30-MAY-2024	5:30	CHITTAGONG
NISHI BHOWMICK	35126	05-MAY-2024	30-MAY-2024	5:30	CHITTAGONG
More than 10 rows available. Increase rows selector to view more rows.					

10 rows returned in 0.00 seconds CSV Export

Q.Select user name, user ID, ticket number, train number, and destination for tickets where the passenger name matches the user name?

```
Select User_.User_name , User_.User_id , Ticket.Ticket_no , Ticket.Train_no , ticket.Destination
From User_ , Ticket
Where User_.User_name = Ticket.Passenger_name;
```

Results Explain Describe Saved SQL History

USER_NAME	USER_ID	TICKET_NO	TRAIN_NO	DESTINATION
MD.PARTHO	1775	12568	5525	BARISAL
HRID HOSSAIN	1855	22504	5530	CHITTAGONG
NISHI BHOWMICK	1900	35126	5533	RAJSHAHI
RS RAZON	1628	54261	5530	CHITTAGONG
MUNIRA MISHU	2015	67842	5511	DHAKA
MRS.MAHI	2189	86499	5525	BARISAL
TISHAN RAFIT	1943	45129	5533	RAJSHAHI

7 rows returned in 0.00 seconds CSV Export

Q. Select ticket number, passenger name, train name, and train number for tickets where the destination matches the destination of the corresponding train?

```
Select T.Ticket_no, T.Passenger_name, Tr.T_name , Tr.Train_no
From Ticket T , Train Tr
Where T.Destination(+) = Tr.Destination;
```

Results Explain Describe Saved SQL History

TICKET_NO	PASSENGER_NAME	T_NAME	TRAIN_NO
12568	MD.PARTHO	VOIVO EXPRESS	5525
22504	HRID HOSSAIN	DURONTO EXPRESS	5530
35126	NISHI BHOWMICK	HIGHFIVE EXPRESS	5533
54261	RS RAZON	DURONTO EXPRESS	5530
67842	MUNIRA MISHU	MIDNIGHT EXPRESS	5511
86499	MRS.MAHI	VOIVO EXPRESS	5525
45129	TISHAN RAFIT	HIGHFIVE EXPRESS	5533

7 rows returned in 0.00 seconds CSV Export

Q. Select passenger name with the phrase "TRAVEL IN" and the destination of the place they are traveling to, for tickets where the passenger and place have the same ticket number?

```
Select Passenger.Passenger_name ||' '|| 'TRAVEL IN' ||' '||place.Destination
From Ticket Passenger , Ticket place
Where Passenger.Ticket_no = Place.Ticket_no;
```

Results Explain Describe Saved SQL History

PASSENGER.PASSENGER_NAME||''||'TRAVELIN'||''||PLACE.DESTINATION
MD.PARTHO TRAVEL IN BARISAL
HRID HOSSAIN TRAVEL IN CHITTAGONG
NISHI BHOWMICK TRAVEL IN RAJSHAHI
RS RAZON TRAVEL IN CHITTAGONG
MUNIRA MISHU TRAVEL IN DHAKA
MRS.MAHI TRAVEL IN BARISAL
TISHAN RAFIT TRAVEL IN RAJSHAHI

7 rows returned in 0.00 seconds CSV Export

Simple View

Q. Create a view named as Train_information where train_No 5530 will be shown over the coloums T_Name,Stop_No,Seat_Available

```
create view Train_information as
select T_Name,Stop_No,Seat_Available
From Train
where Train_No =5530;
```

Fig : Simple View Creation Command

```
desc Train_information
```

Results Explain Describe Saved SQL History

Object Type VIEW Object TRAIN_INFORMATION

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
TRAIN_INFORMATION	T_NAME	Varchar2	20	-	-	-	✓	-	-
	STOP_NO	Number	-	10	0	-	✓	-	-
	SEAT_AVAILABLE	Number	-	10	0	-	✓	-	-

1 - 3

Fig : Description of the Simple View

```
select* from Train_information;
```

Results Explain Describe Saved SQL History

T_NAME	STOP_NO	SEAT_AVAILABLE
DURONTO EXPRESS	102	90

1 rows returned in 0.00 seconds CSV Export

Fig : Result of the Simple View As A whole Table

Complex View:

Q. Create a view named as Payment_information will be shown the total amount spend by each bank for train ticket to different destination

```
CREATE VIEW PaymentSummaryView AS
SELECT
    p.Bank,
    t.Destination,
    SUM(p.Amount) AS TotalAmount
FROM
    Payment p ,Ticket t
where
     p.Ticket_No = t.Ticket_No
GROUP BY
    p.Bank, t.Destination;
```

Fig : Complex View Creation Command

```
desc PaymentSummaryView
```

Results Explain Describe Saved SQL History

Object Type VIEW Object PAYMENTSUMMARYVIEW

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PAYMENTSUMMARYVIEW	BANK	Varchar2	20	-	-	-	✓	-	-
	DESTINATION	Varchar2	20	-	-	-	-	-	-
	TOTALAMOUNT	Number	-	-	-	-	✓	-	-

1 - 3

Fig : Description of the Complex View

```
select* from PaymentSummaryView;
```

Results | Explain | Describe | Saved SQL | History

BANK	DESTINATION	TOTALAMOUNT
ISLAMIC	BARISAL	600
CITY	RAJSHAHI	1000
CITY	CHITTAGONG	800
ISLAMIC	RAJSHAHI	1000
DHAKA	BARISAL	600
BRAC	DHAKA	900
BRAC	CHITTAGONG	800

7 rows returned in 0.00 seconds CSV Export

Fig : Result of the Complex View As A whole Table

Description of a Successful DB connection

1. First I install all the software and jar files.

- Install Inteliji Idea
- Install Xampp Server
- Download mysql-connector-j-8.4.0.jar

I open Xampp server from Xampp control panel then start Apache and MySQL module then open phpMyAdmin. There I create a Database by name "TableName" and create a table name "TableName" then I insert Some values in it.

3. After creating Database and table I open Inteliji Idea and create a project there I add the mysql-connector-j-8.4.0.jar as library. Then inside the main file I write code to connect my Database that I create trough the Xampp. There I print our addresss table columns.

1- Rafit:

```
C:\Users\brish\.jdks\openjdk-22.0.
15-MAY-2024 10:30 BARISAL
30-MAY-2024 5:30 CHITTAGONG
05-MAY-2024 7:00 RAJSHAHI
23-MAY-2024 1:30 DHAKAA
```

```
class demo {
blic static void main(String[] args) {
 String url = "jdbc:mysql://localhost:3306/Railway";
 String username = "root";
 String password = "";
try {
    Class.forName( className: "com.mysql.cj.jdbc.Driver");

    Connection connection = DriverManager.getConnection(url , username , password);

    Statement statement = connection.createStatement();

    ResultSet resultSet = statement.executeQuery( sql: "select * from schedule");

    while(resultSet.next())
    {
        System.out.println(resultSet.getString( columnIndex: 1)+" "+resultSet.getString( columnIndex: 2)+" "+resultSet.getString( columnIndex: 3))
    }
    connection.close();

 } catch (Exception e) {
    System.out.println("ERROR FOUND");
    System.out.println(e);
 }
```

DATE_	TIME	DESTINATION
15-MAY-2024	10:30	BARISAL
30-MAY-2024	5:30	CHITTAGONG
05-MAY-2024	7:00	RAJSHAHI
23-MAY-2024	1:30	DHAKAA

2- Razan:

USER_ID	CITY	STATE	PIN_CODE
1775	DHAKA	DHAKA	1100
1855	NATOR	RAJSHAHI	1400
1900	SUNAMGONJ	SYLHET	1300
1775	DHAKA	DHAKA	1100
1855	NATOR	RAJSHAHI	1400
1900	SUNAMGONJ	SYLHET	1300

Main.java NIRJHOR.java demo.java

Reader Mode

```java
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;

Rename usages
public class demo {
    public static void main(String[] args) {
        String url = "jdbc:mysql://localhost:3306/RAZAN";
        String username = "root";
        String password = "";
        try {
            Class.forName( className: "com.mysql.cj.jdbc.Driver");

            Connection connection = DriverManager.getConnection(url , username , password);

            Statement statement = connection.createStatement();

            ResultSet resultSet = statement.executeQuery( sql: "SELECT * FROM ADDRESS");

            while(resultSet.next()) {
                int userId = resultSet.getInt( columnLabel: "USER_ID");
                String city = resultSet.getString( columnLabel: "CITY");
                String state = resultSet.getString( columnLabel: "STATE");
                int pinCode = resultSet.getInt( columnLabel: "PIN_CODE");

                System.out.println(userId + " " + city + " " + state + " " + pinCode);
            }
            connection.close();
```

```
C:\Users\brish\.jdks\openjdk-22.0.1\b
1775 DHAKA DHAKA 1100
1855 NATOR RAJSHAHI 1400
1900 SUNAMGONJ SYLHET 1300
1775 DHAKA DHAKA 1100
1855 NATOR RAJSHAHI 1400
1900 SUNAMGONJ SYLHET 1300
```

3-Mobashir:

```
C:\Users\brish\.jdks\openjdk-22.0.1\bin\java.exe
1005621 8845 12568 600
1005489 6616 22504 800
1005621 8845 12568 600
1005489 6616 22504 800

Process finished with exit code 0
```

```java
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;

Rename usages
public class demo {
    public static void main(String[] args) {
        String url = "jdbc:mysql://localhost:3306/mobashir";
        String username = "root";
        String password = "";
        try {
            Class.forName( className: "com.mysql.cj.jdbc.Driver");

            Connection connection = DriverManager.getConnection(url , username , password);

            Statement statement = connection.createStatement();

            ResultSet resultSet = statement.executeQuery( sql: "SELECT * FROM payment");

            while(resultSet.next()) {
                int transId = resultSet.getInt( columnLabel: "Trans_id");
                int cardNo = resultSet.getInt( columnLabel: "Card_No");
                int ticketNo = resultSet.getInt( columnLabel: "Ticket_No");
                int amount = resultSet.getInt( columnLabel: "Amount");

                System.out.println(transId + " " + cardNo + " " + ticketNo + " " + amount);
            }
            connection.close();
```

Trans_id	Card_No	Ticket_No	Amount	Bank
1005621	8845	12568	600	Islamic
1005489	6616	22504	800	Brac
1005621	8845	12568	600	Islamic
1005489	6616	22504	800	Brac

4-Bhowmick:

Ticket_No	Pass_Name	Train_No
12568	MD. Partho	5525
86499	Mrs Mahi	5525
12568	MD. Partho	5525
86499	Mrs Mahi	5525

```
C:\Users\brish\.jdks\openjdk-22.0.1\bin\java.exe "-javaagent:
5525 MD. Partho 12568
5525 Mrs Mahi 86499
5525 MD. Partho 12568
5525 Mrs Mahi 86499

Process finished with exit code 0
```

```java
import java.sql.Statement;

public class demo {
    public static void main(String[] args) {
        String url = "jdbc:mysql://localhost:3306/BHOWMICK";
        String username = "root";
        String password = "";
        try {
            Class.forName( className: "com.mysql.cj.jdbc.Driver");

            Connection connection = DriverManager.getConnection(url , username , password);

            Statement statement = connection.createStatement();

            ResultSet resultSet = statement.executeQuery( sql: "SELECT * FROM ticket");

            while(resultSet.next()) {
                int trainNo = resultSet.getInt( columnLabel: "Train_No");
                String passName = resultSet.getString( columnLabel: "Pass_Name");
                int ticketNo = resultSet.getInt( columnLabel: "Ticket_No");

                System.out.println(trainNo + " " + passName + " " + ticketNo);
            }
            connection.close();

        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

Conclusion

In conclusion, the proposed Online Train Reservation System offers a comprehensive solution to the challenges faced by passengers and railway authorities in traditional ticket booking processes. By leveraging modern technology and the internet, the system enhances the booking experience for passengers by providing convenience, accessibility, and efficiency. With features such as user registration, real-time train schedules, seat selection, secure payment gateways, and e-ticket generation, passengers can easily plan and book their journeys from anywhere at any time. Moreover, the system's administrative dashboard enables efficient management of train schedules, seat inventories, user accounts, and financial transactions. Overall, the Online Train Reservation System represents a significant step towards improving the railway travel experience for passengers while optimizing operational processes for railway authorities.