AMERICAN
INTERNATIONAL
UNIVERSITY-
BANGLADESH
where leaders are created

Group Project

On

Ticket Management System

Course Name: Advance Database Management System

Section: A

Contributed By:

ID	NAME	Contribution
18-36702-1	Fahad Mollah	20%
18-36674-1	Anisuzzzaman	20%
18-36272-1	Samiya Rahman	20%
18-38852-3	Arifur Rahman	20%
19-39349-1	Md. Tamim Hossen Sefat	20%

Contents:

Introduction:

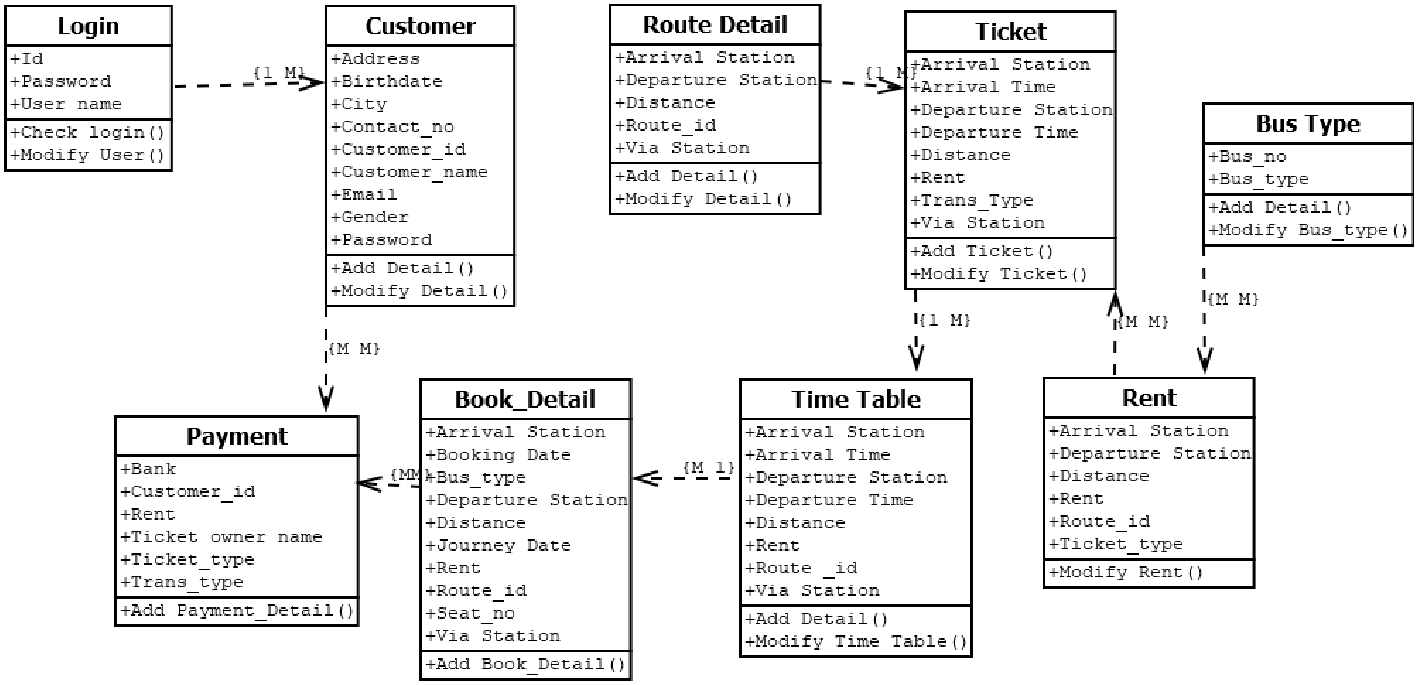
In this project we are working on to create the database of Ticket Management, where a customer can buy advance departure ticket of a bus and can select bus type like Ac and Non Ac bus. A customer can make his/her payment through different payment method. They can also see the bus route before the departure of the bus.

Project Proposal:

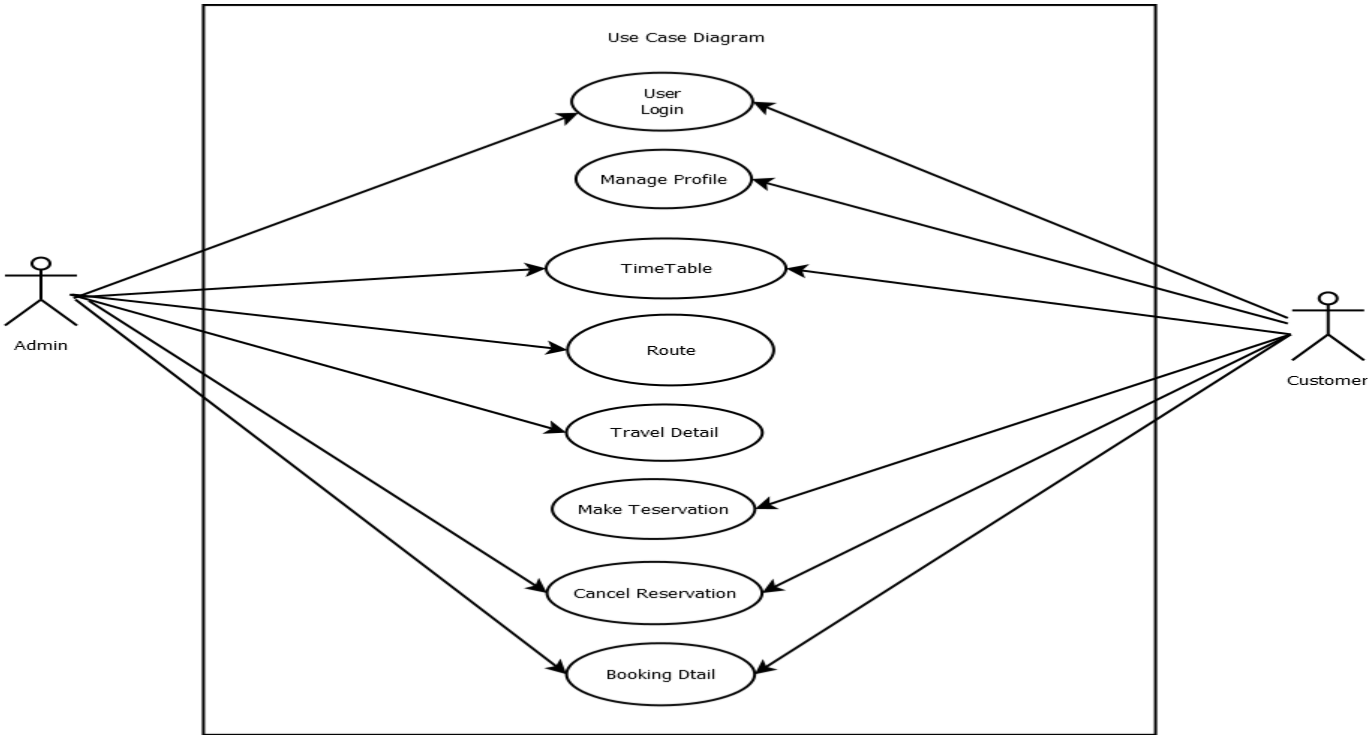
We are proposing this project, because now a days there are many people who love to travel different bus, they fell hustle to book ticket. By our project A customer can buy a ticket before the day of their arrival so it will be very beneficial of people. And this will be profitable for us as well.

This is a ticket management system where a customer has a unique customer Id, name, email, Address, Date Of birth, password, city name, gender and contact no who get service of bus. The bus service has place of departure, place of arrival, via place, journey date, rent, Time of departure, time of arrival, distance and booking data. A customer can update his/her password and to update the password it requires customer id password, customer id and new password. A customer's email, customer id and message are stored as contact in the database. The service has time table and book details where time table has arrival time, departure time, departure place, route id and arrival place and book details have seat no, distance, result id. The bus on service has bus type and its unique id. The service's payment methods are also stored in the database where payment has payment id, customer id as the customer pay, transection type and rent. The service has route details and ticket info where route details have route id and ticket has transection type. The service has also an admin who maintain the whole process and the admin has a unique admin id, email, password and name.

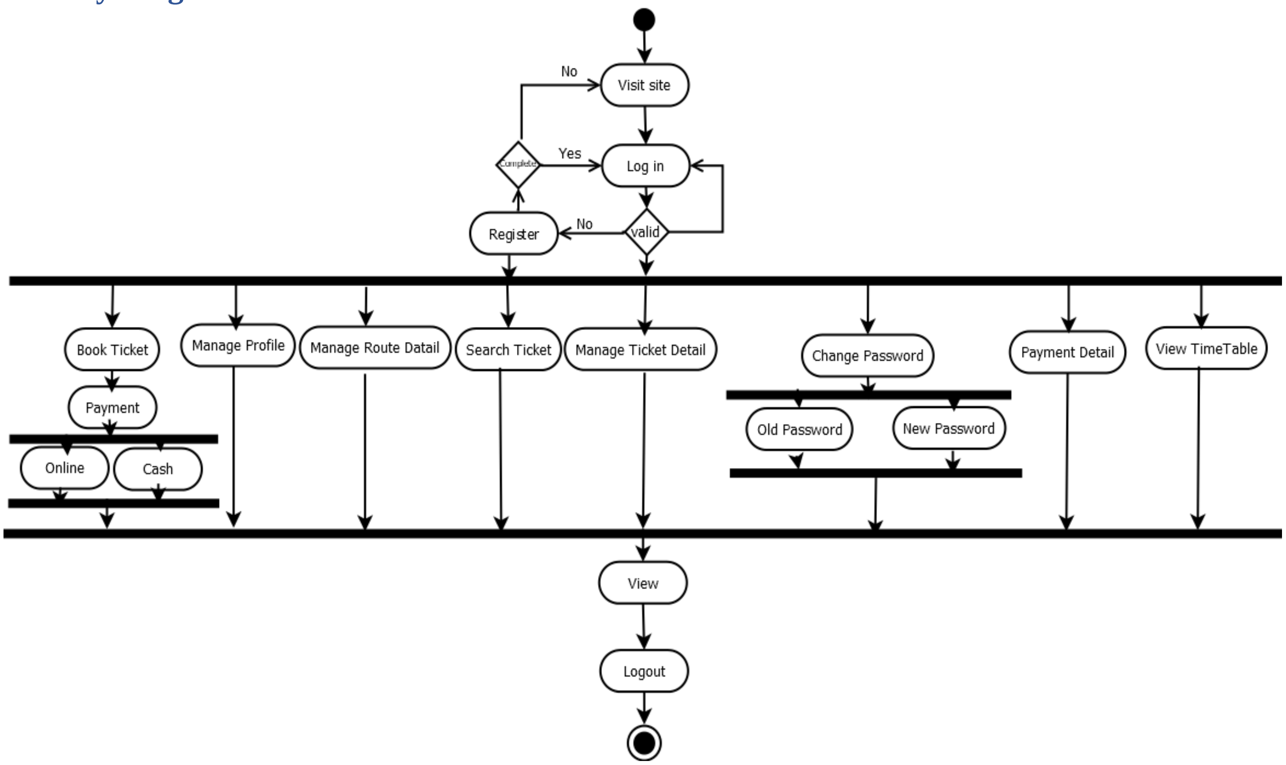
Class Diagram:



Use Case Diagram

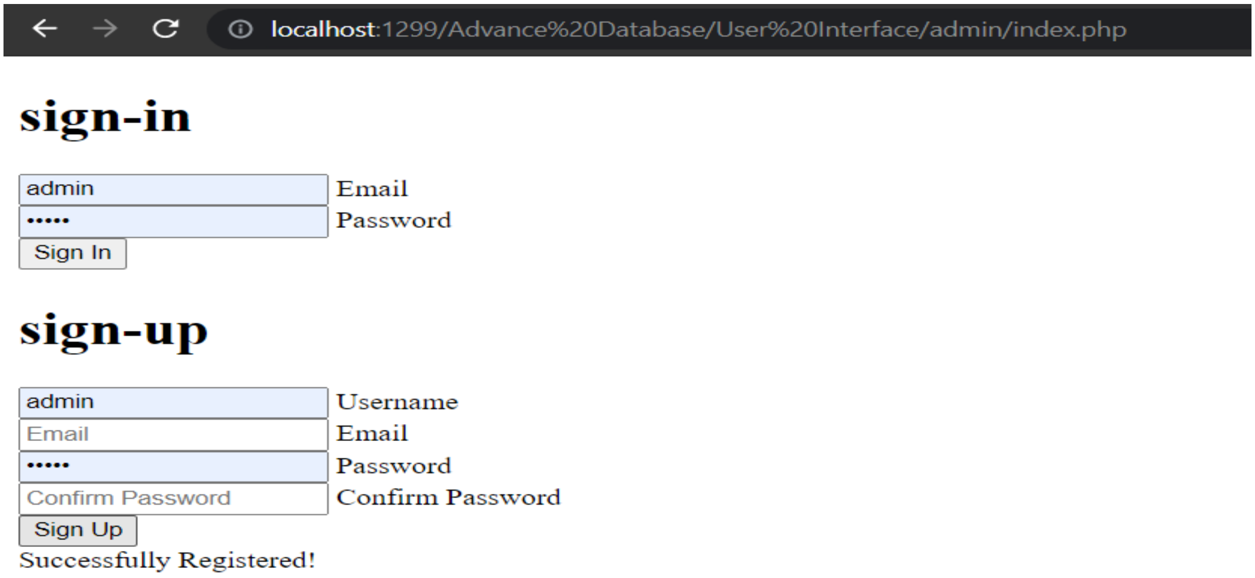


Activity Diagram:



User Interface:

Admin



localhost:1299/Advance%20Database/User%20Interface/admin/index.php

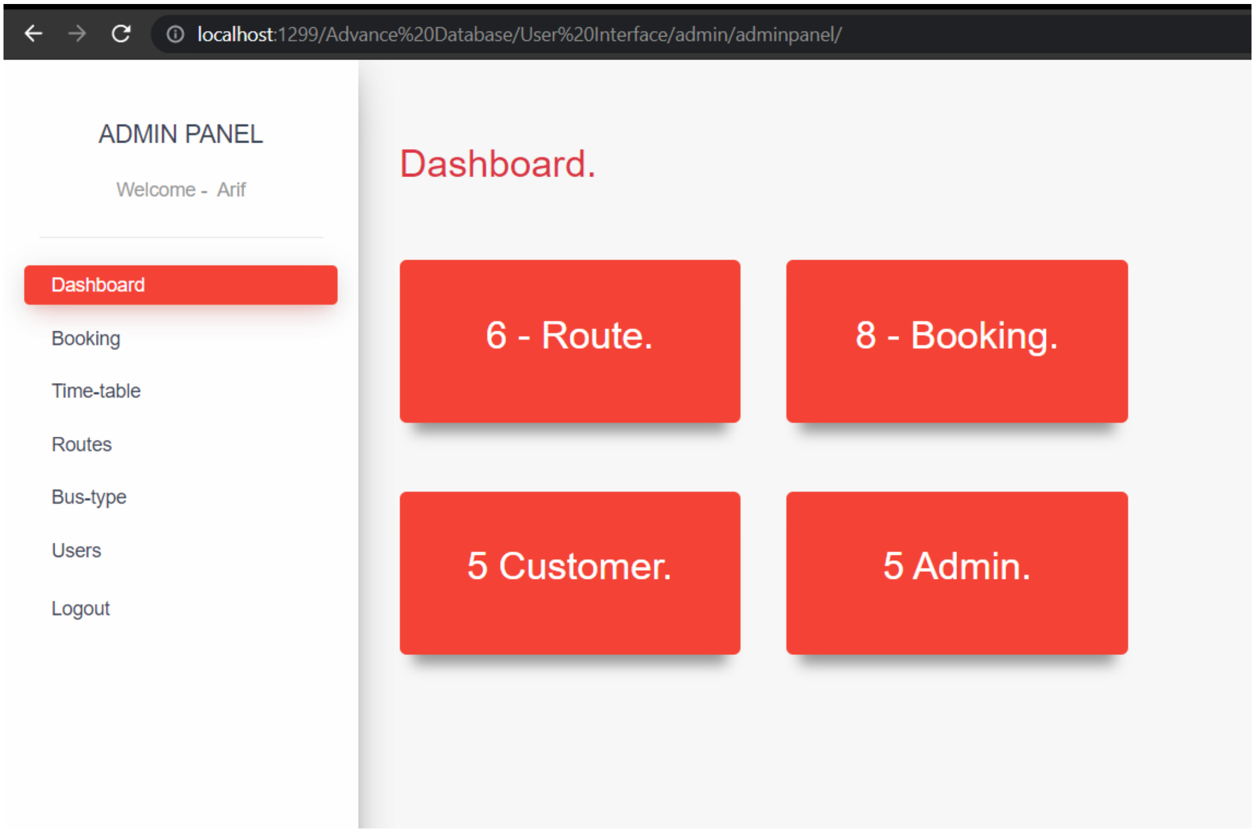
sign-in

admin Email
••••• Password

sign-up

admin Username
Email Email
••••• Password
Confirm Password Confirm Password

Successfully Registered!



localhost:1299/Advance%20Database/User%20Interface/admin/adminpanel/

ADMIN PANEL

Welcome - Arif

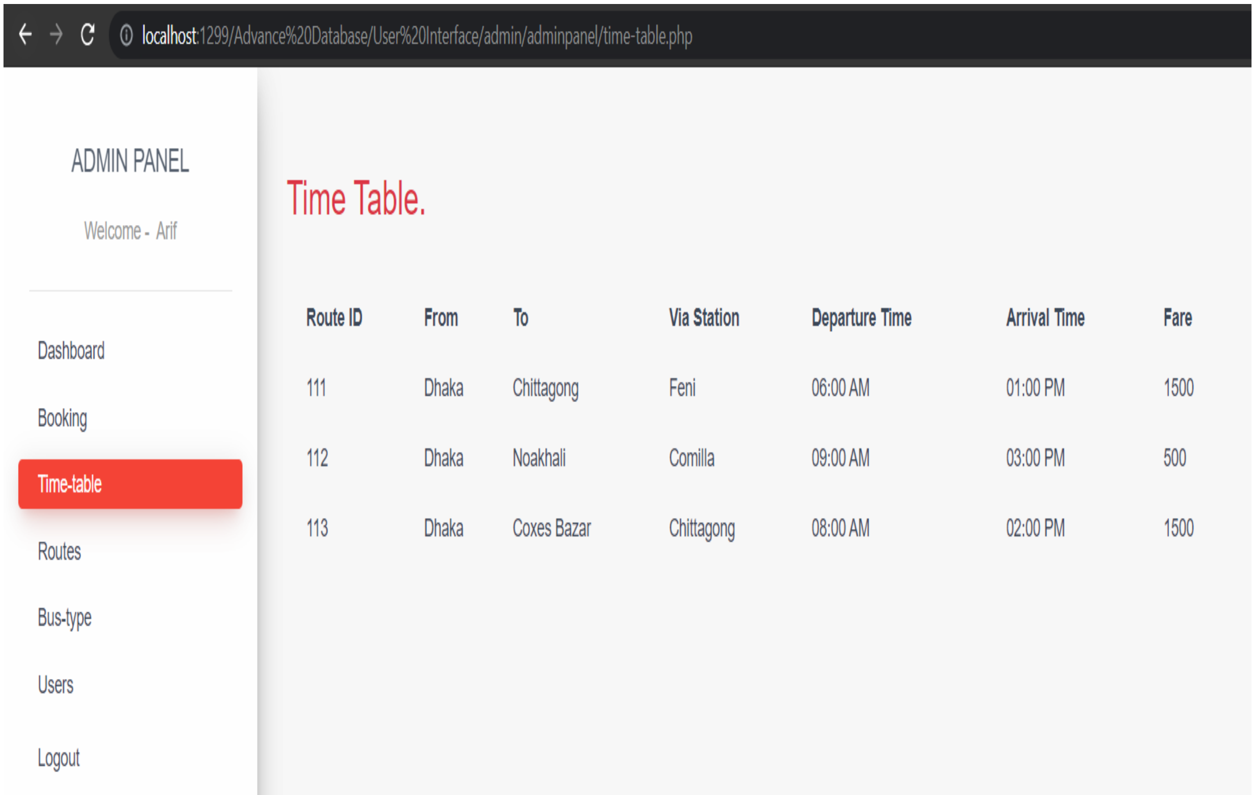
Dashboard.

6 - Route.

8 - Booking.

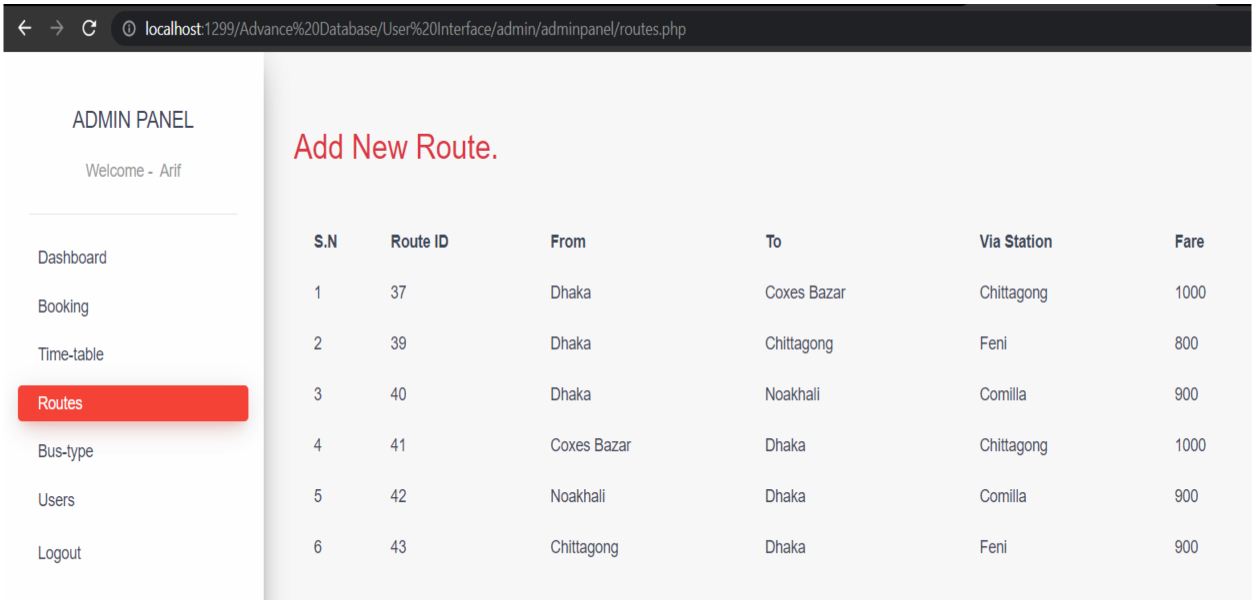
5 Customer.

5 Admin.


localhost:1299/Advance%20Database/User%20Interface/admin/adminpanel/time-table.php
ADMIN PANEL
Welcome - Arif
Dashboard
Booking
Time-table
Routes
Bus-type
Users
Logout

Time Table.

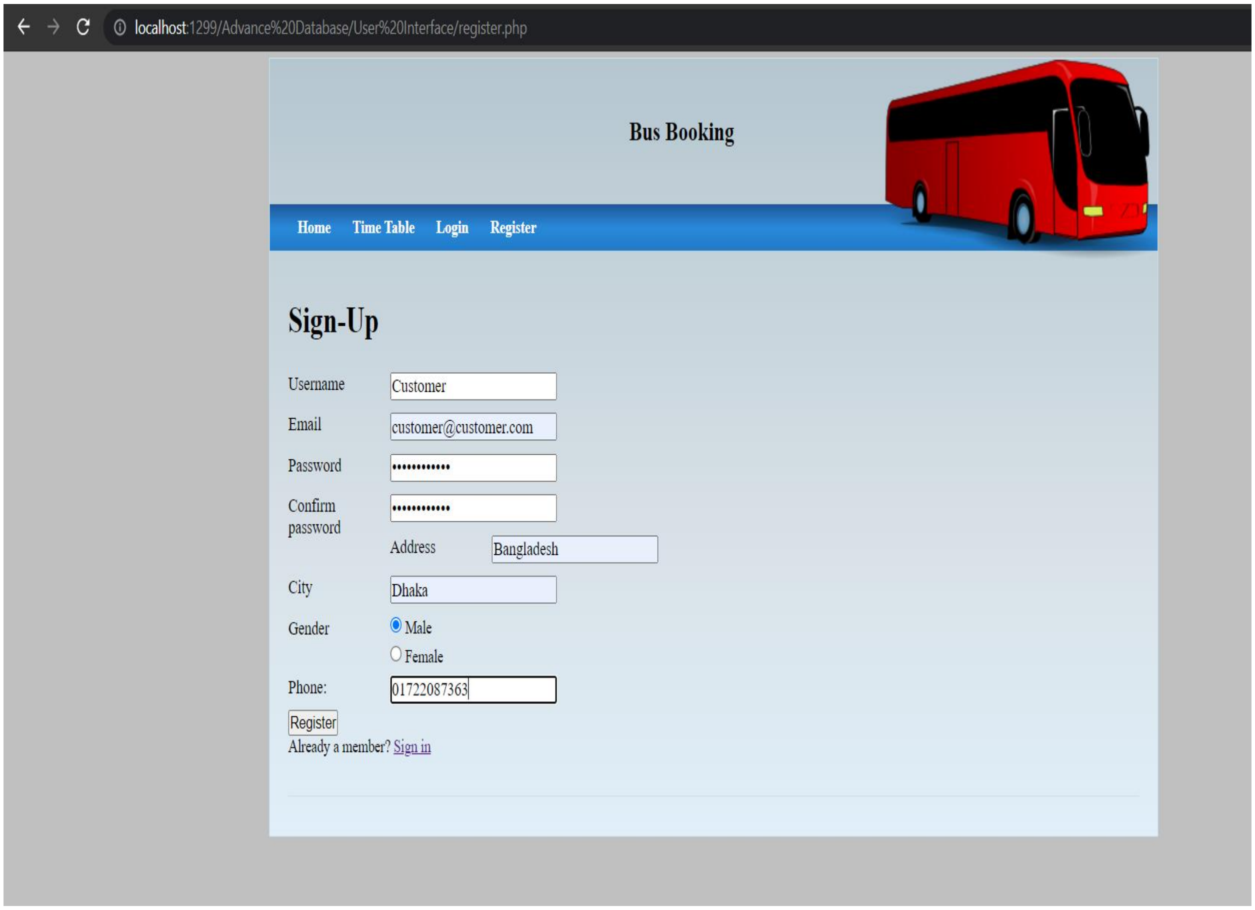
Route ID	From	To	Via Station	Departure Time	Arrival Time	Fare
111	Dhaka	Chittagong	Feni	06:00 AM	01:00 PM	1500
112	Dhaka	Noakhali	Comilla	09:00 AM	03:00 PM	500
113	Dhaka	Coxes Bazar	Chittagong	08:00 AM	02:00 PM	1500


localhost:1299/Advance%20Database/User%20Interface/admin/adminpanel/routes.php
ADMIN PANEL
Welcome - Arif
Dashboard
Booking
Time-table
Routes
Bus-type
Users
Logout

Add New Route.

S.N	Route ID	From	To	Via Station	Fare
1	37	Dhaka	Coxes Bazar	Chittagong	1000
2	39	Dhaka	Chittagong	Feni	800
3	40	Dhaka	Noakhali	Comilla	900
4	41	Coxes Bazar	Dhaka	Chittagong	1000
5	42	Noakhali	Dhaka	Comilla	900
6	43	Chittagong	Dhaka	Feni	900

<u>User</u>



localhost:1299/Advance%20Database/User%20Interface/register.php

Bus Booking

Sign-Up

Username: Customer

Email: customer@customer.com

Password: ••••••••••••

Confirm password: ••••••••••••

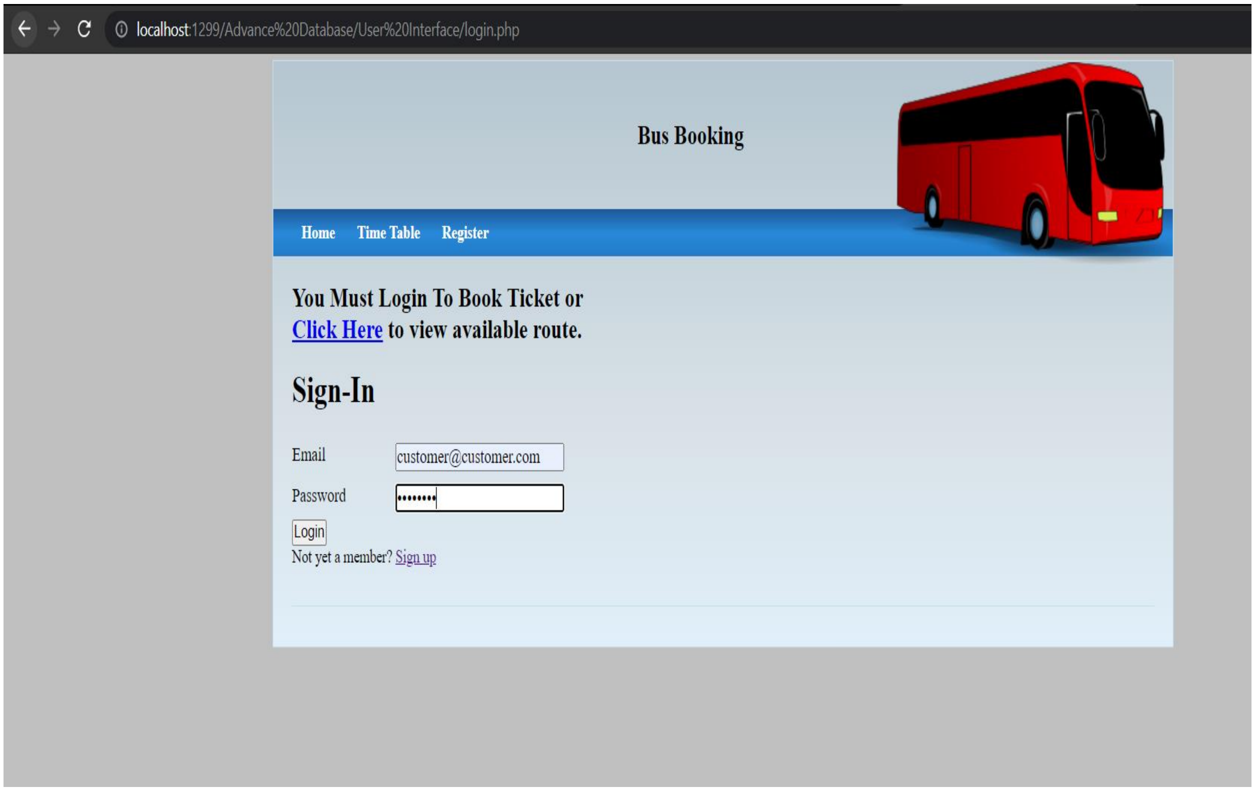
Address: Bangladesh

City: Dhaka

Gender: ● Male ○ Female

Phone: 01722087363

Already a member? Sign in



localhost:1299/Advance%20Database/User%20Interface/login.php

Bus Booking

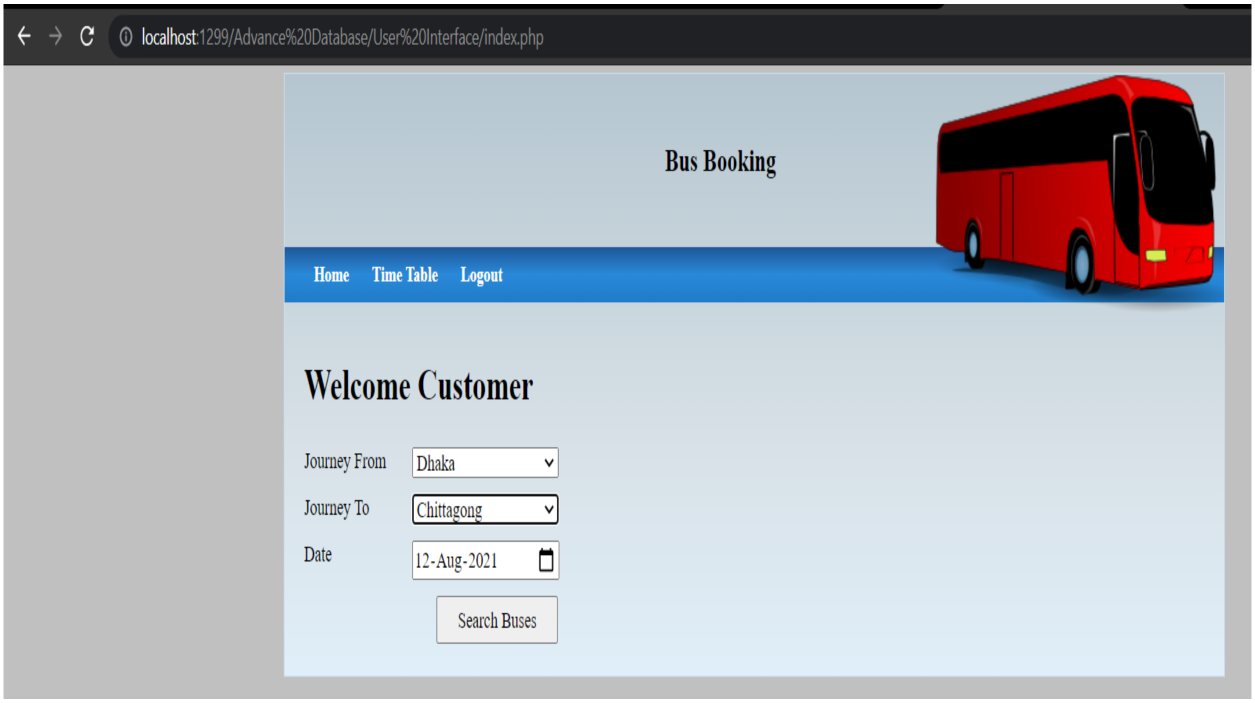
You Must Login To Book Ticket or
Click Here to view available route.

Sign-In

Email: customer@customer.com

Password: ••••••••

Not yet a member? Sign up

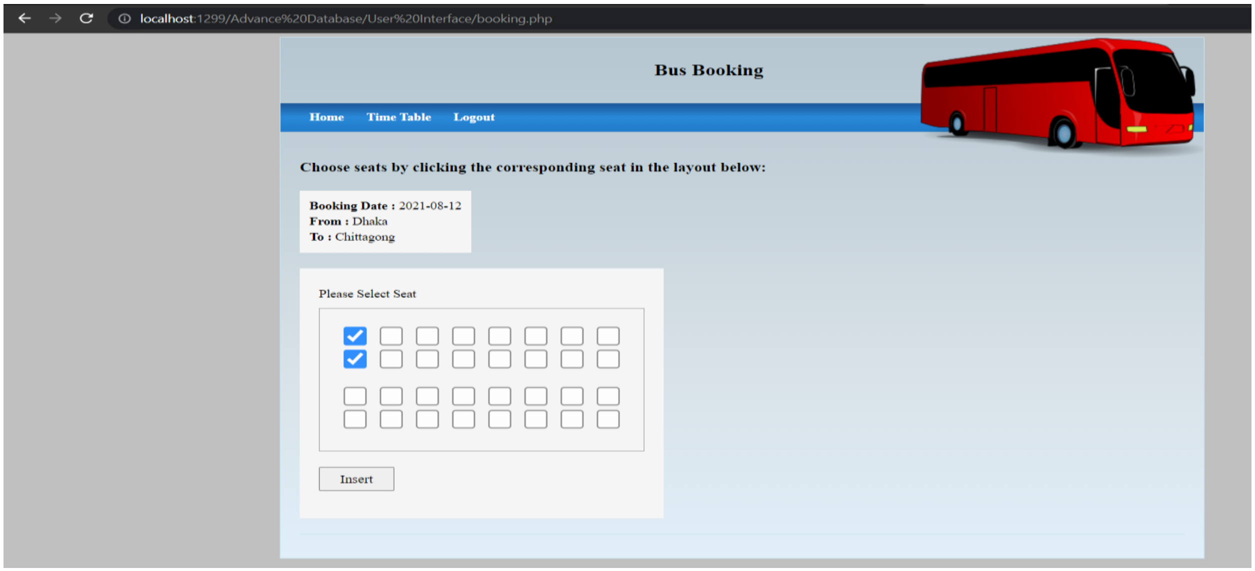

localhost:1299/Advance%20Database/User%20Interface/index.php
Bus Booking
Home
Time Table
Logout

Welcome Customer

Journey From: Dhaka

Journey To: Chittagong

Date: 12-Aug-2021



localhost:1299/Advance%20Database/User%20Interface/booking.php

Bus Booking

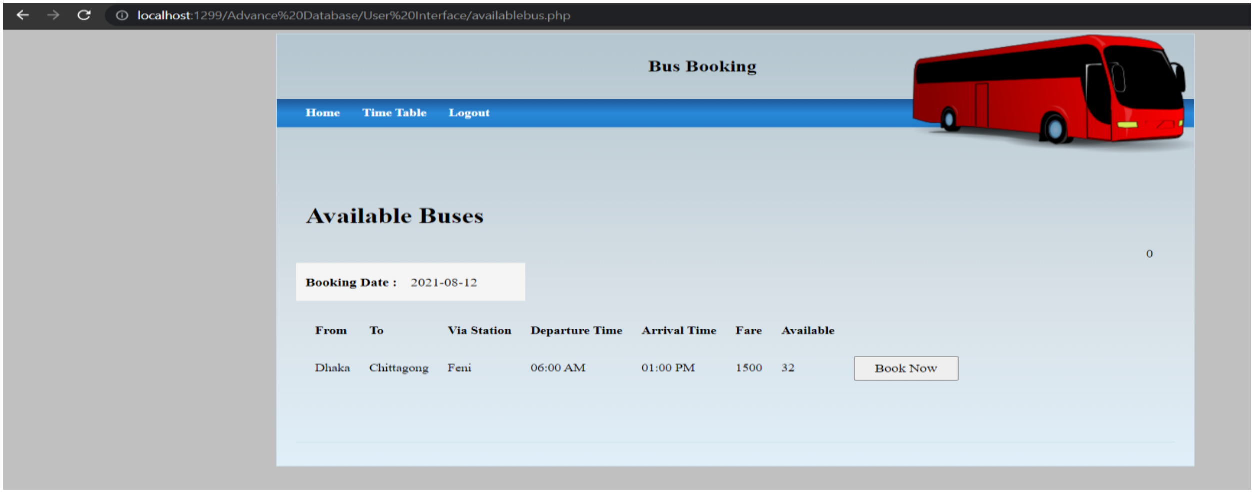
Choose seats by clicking the corresponding seat in the layout below:

Booking Date : 2021-08-12
From : Dhaka
To : Chittagong

Please Select Seat

☑ □ □ □ □ □ □ □
☑ □ □ □ □ □ □ □

□ □ □ □ □ □ □ □
□ □ □ □ □ □ □ □



localhost:1299/Advance%20Database/User%20Interface/availablebus.php

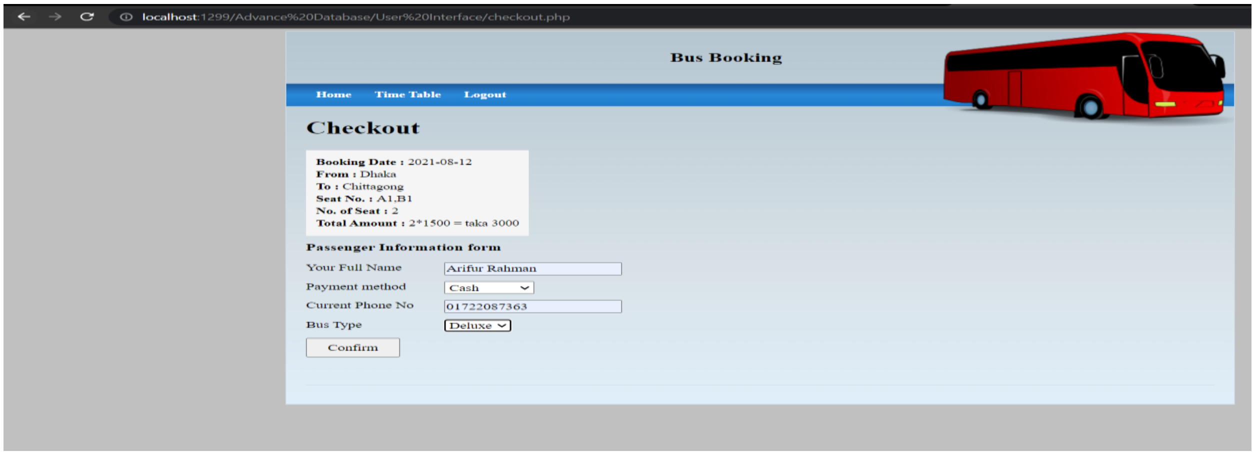
Bus Booking

Available Buses

0

Booking Date : 2021-08-12

From	To	Via Station	Departure Time	Arrival Time	Fare	Available	
Dhaka	Chittagong	Feni	06:00 AM	01:00 PM	1500	32	Book Now



localhost:1299/Advance%20Database/User%20Interface/checkout.php

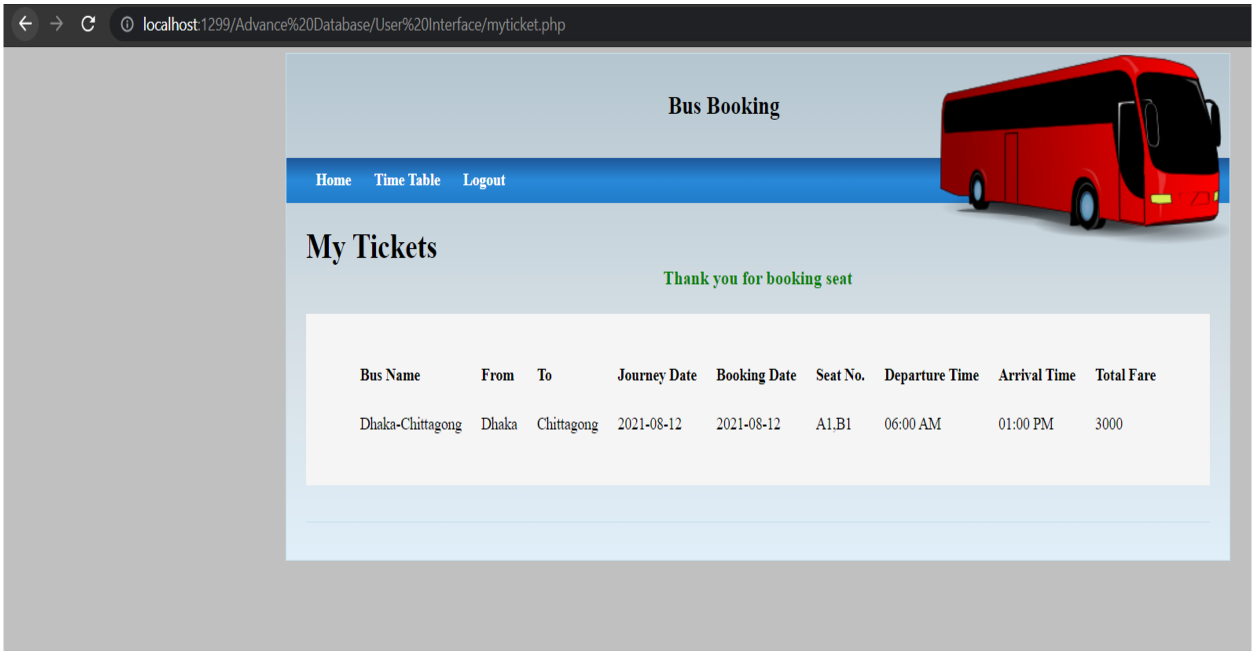
Bus Booking

Checkout

Booking Date : 2021-08-12
From : Dhaka
To : Chittagong
Seat No. : A1,B1
No. of Seat : 2
Total Amount : 2*1500 = taka 3000

Passenger Information form

Your Full Name: Arifur Rahman
Payment method: Cash
Current Phone No: 01722087363
Bus Type: Deluxe


localhost:1299/Advance%20Database/User%20Interface/myticket.php
Bus Booking
Home
Time Table
Logout

My Tickets

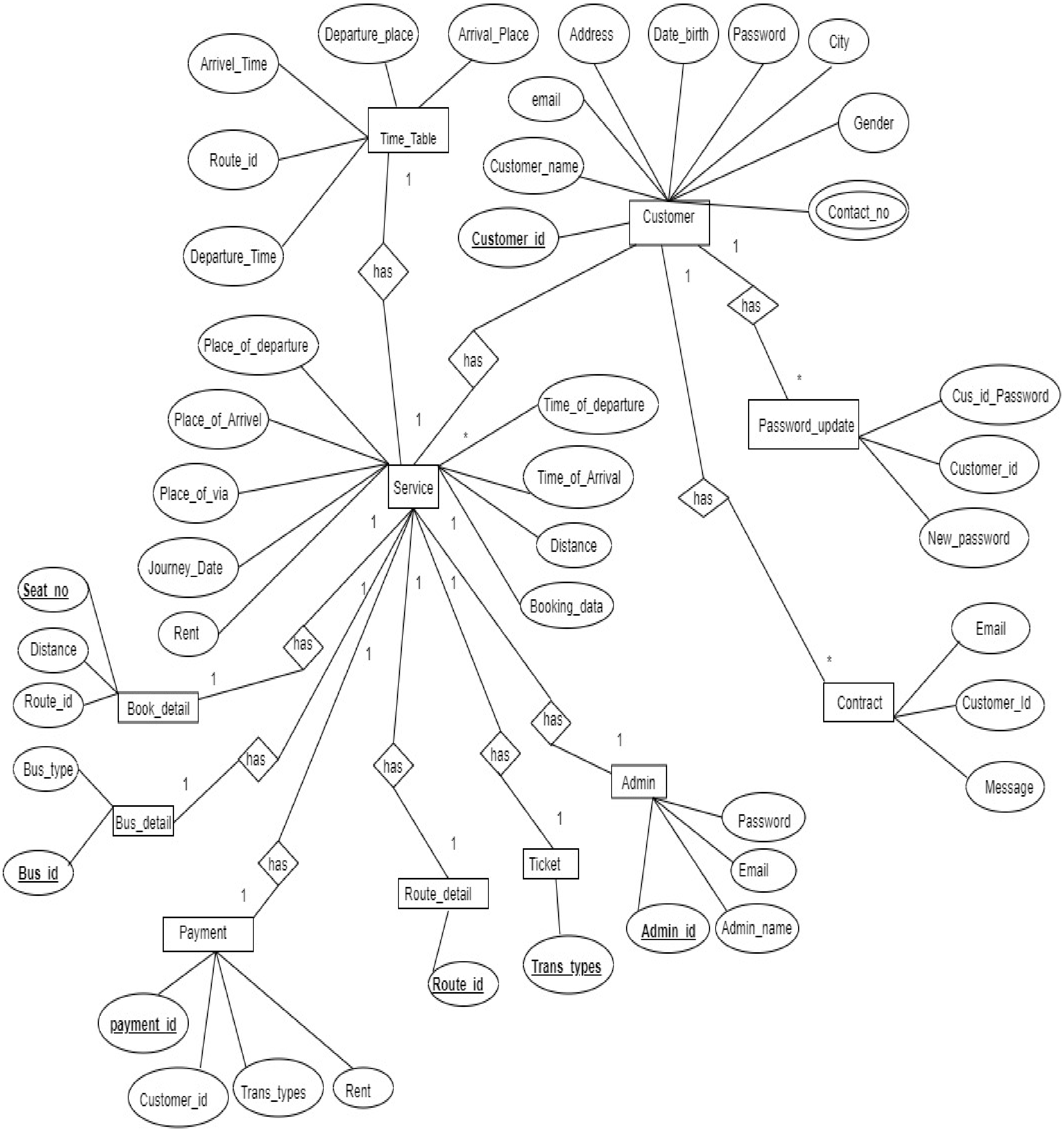
Thank you for booking seat

Bus Name	From	To	Journey Date	Booking Date	Seat No.	Departure Time	Arrival Time	Total Fare
Dhaka-Chittagong	Dhaka	Chittagong	2021-08-12	2021-08-12	A1,B1	06:00 AM	01:00 PM	3000

Scenario Description:

This scenario description is about ticket management system. This is a ticket management system where a customer buys a ticket or booking a ticket. We have many type of entities and multiple relationships among them. Each customer has an unique customer Id. Customer's data which is Customer name, email, Address, Date Of birth, password, city name, gender and contact no are also stored in the system. This system has many services. This system provides many services to a customer and there is one to many relationships is Customer entity with Service entity. The bus service has place of departure, place of arrival, via place, journey date, rent, Time of departure, time of arrival, distance and booking data. A customer can update his/her password and to update the password it requires customer id password, customer id and new password. A customer's email, customer id and message are stored as contact in the database. The multi-value attribute has in Customer entity. The entity Customer has bind with contact entity with foreign key. The service has a time table and book details. Time table data which is arrival time, departure place, departure time, route id is primary key and book details have primary key is seat no , distance, and route id is primary key. The bus on service has bus type and its unique id. The service's payment methods are also stored in the database where payment has primary key payment id, customer id as the customer pay, transection type and rent. The service has a route detail and a ticket info where route details have route id and ticket has transection type. one to one relationship is available time table, Password, book detail, bus detail, route detail, Ticket, Admin, Payment with service entity. The service has also an admin who maintain the whole process and the admin has a unique admin id, email, password and name.

ER Diagram


Departure_place
Arrival_Place
Address
Date_birth
Password
City
Arrivel_Time
email
Gender
Time_Table
Route_id
Customer_name
Customer
Contact_no
Customer_id
Departure_Time
1
has
1
1
has
has
Place_of_departure
*
Cus_id_Password
Time_of_departure
Password_update
Place_of_Arrivel
1
*
Customer_id
Service
Time_of_Arrival
has
Place_of_via
New_password
1
1
Distance
Journey_Date
1
1
1
Booking_data
Seat_no
Rent
Email
1
has
*
Distance
1
Contract
Customer_Id
Route_id
Book_detail
has
1
Message
Bus_type
has
has
has
Admin
1
Bus_detail
1
Password
Bus_id
has
Ticket
Email
1
Route_detail
Payment
Admin_id
Admin_name
Route_id
Trans_types
payment_id
Customer_id
Trans_types
Rent

Normalization:

#Customer-Sevice: (S_ID , A_Seat , B_Seat , W_Seat , Customer_Id, Customer_name , Phone ,Email, T_Status , Gender , Seat_No , U_ID , Password)

1NF: Phone is a multivalued attribute

2NF: S_ID , A_Seat , B_Seat , W_Seat

Customer_Id, Customer_name, Phone, Email, B_Status, Seat_No

U_ID , Password

3NF : S_ID , A_Seat , B_Seat , W_Seat

Customer_Id, Customer_name, Phone, Email, B_Status, Seat_No

U_ID , Password

Tables Customer-Sevice:

1. S_ID , A_Seat , B_Seat , W_Seat , Customer_Id , U_ID
2. Customer_Id, Customer_name, Phone, Email, B_Status, Seat_No
3. U_ID , Password
4. Customer_Id, Phone (Composite Primary Key)

Book details-Service: (S_ID, S_Name, R_No, R_name , Arr_Time, Dep_Time)

1NF: No Multivalued attribute

2NF: S_ID, S_Name

R_No, R_name , Arr_Time, Dep Time

3NF: S_ID, S_Name

R_No , R_name

AD_T , Arr_Time , Dep_Time

Tables From Book details-Service:

1. S_ID, S_Name

2. R_No , R_name , AD_T

3. AD_T , Arr_Time , Dep_time

4. RS_N , R_No , S_ID

Service-Time Table: **(**Bus_ID , Bus_Name , Av_Class , R_No , R_name , Arr_Time , Dep_Time)

1NF: No Multivalued Attribute

2NF: Bus_ID , B_Name , Bus_Type, Av_Class

R_No , R_name

3NF : Bus_ID , Bus_Name , Bus_Type, Av_Class

R_No , R_name

AD_T , Arr_Time , Dep_time

Final Table of Service-Time Table:

1. Bus_ID , Bus_Name , Av_Class, Bus_Type , AD_T , R_No
2. R_No , R_Name
3. AD_T , Arr_Time , Dep_Time

Service-Bus Details:

(Bus_ID, Bus_Name, AV_Class, Status_ID,A_Seat,B_Seat,W_Seat)

1NF: No Multivalued Attribute.

2NF: Bus_ID, Bus_Name, AV_Class,

S_ID, A_Seat, B_Seat,W_Seat

3NF: Bus_ID, Bus_Name, AV_Class

S_ID, A_Seat, B_Seat,W_Seat

Table from Service-Time Table:

1. Bus_ID, Bus_Name, AV_Class, Status_ID
2. S_ID, A_Seat, B_Seat,W_Seat

Final Tables after Normalization :

1. S_ID , A_Seat , B_Seat , W_Seat , Customer_ID, U_ID -> STATUS
2. Customer_ID , Customer_name , B_status , Email, Gender , Seat_No -> PASSENGER
3. U_ID , Password -> USER
4. Customer_ID, Phone (Composite Primary Key) -> CONTACT
5. S_ID, S_Name -> Bus_Station
6. R_No , R_name , AD_T -> ROUTE
7. AD_T , Arr_Time , Dep_time -> TIME
8. RS_N , R_No , S_ID -> R_S
9. Bus_ID , Bus_Name , Av_Class , Bus_Type, AD_T , R_No -> Bus
10. Bus_ID, Bus_Name, AV_Class, Bus_Type, Status_ID -> Bus_STATUS

Schema Diagram:

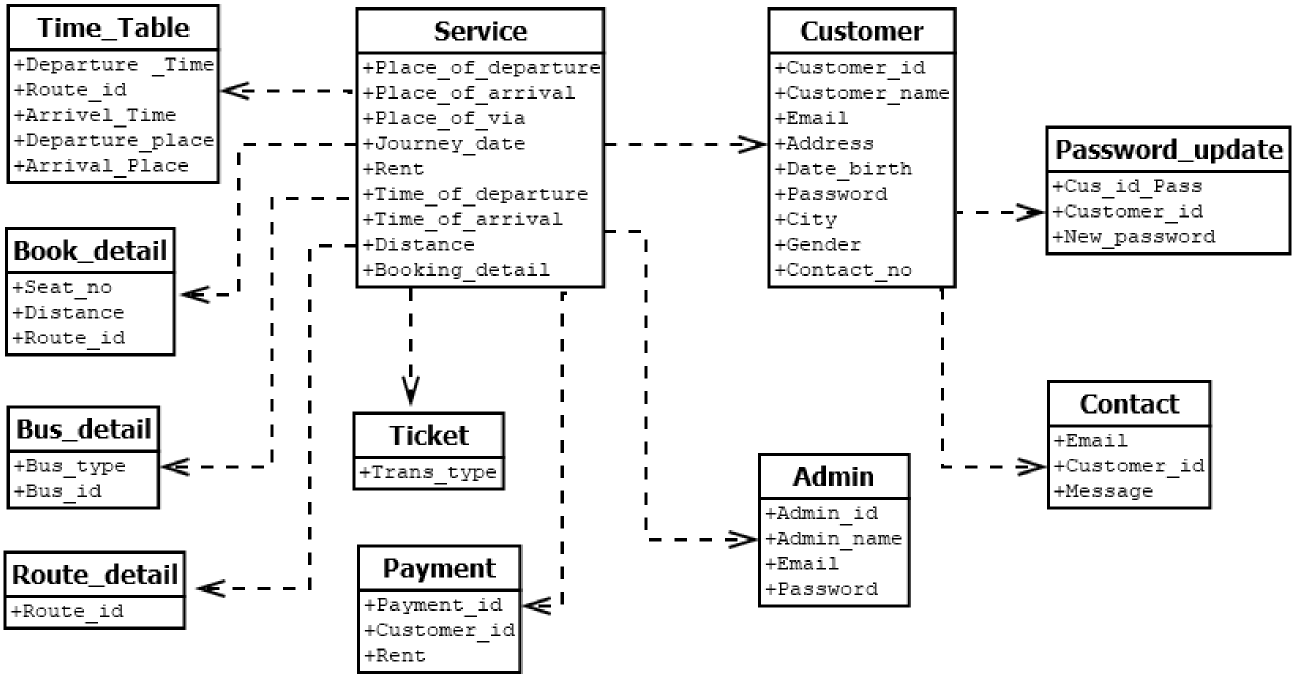


Table Creation

Admin

Create table Admin

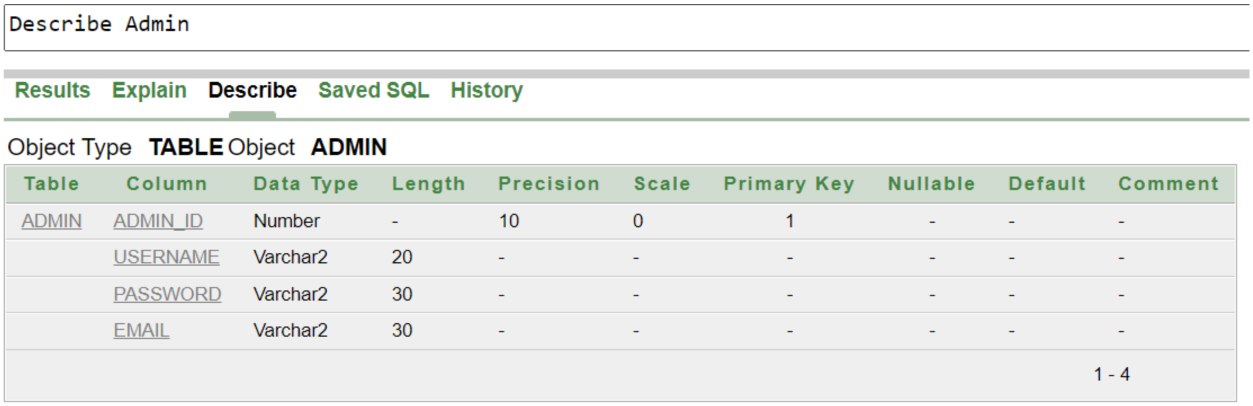
(Admin_id number(10) CONSTRAINT Admin_pk PRIMARY KEY,

Username varchar2(20) NOT NULL,

Password varchar2(30) NOT NULL,

Email varchar2(30) NOT NULL

);



```
Describe Admin
```

Object Type **TABLE** Object **ADMIN**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ADMIN	ADMIN_ID	Number	-	10	0	1	-	-	-
	USERNAME	Varchar2	20	-	-	-	-	-	-
	PASSWORD	Varchar2	30	-	-	-	-	-	-
	EMAIL	Varchar2	30	-	-	-	-	-	-

1 - 4

Customer

Create table Customer

(Customer_id number(10) CONSTRAINT Customer_pk PRIMARY KEY,

Username varchar2(20) NOT NULL,

Password varchar2(30) NOT NULL,

Address varchar2(60) NOT NULL,

City varchar2(20) NOT NULL,

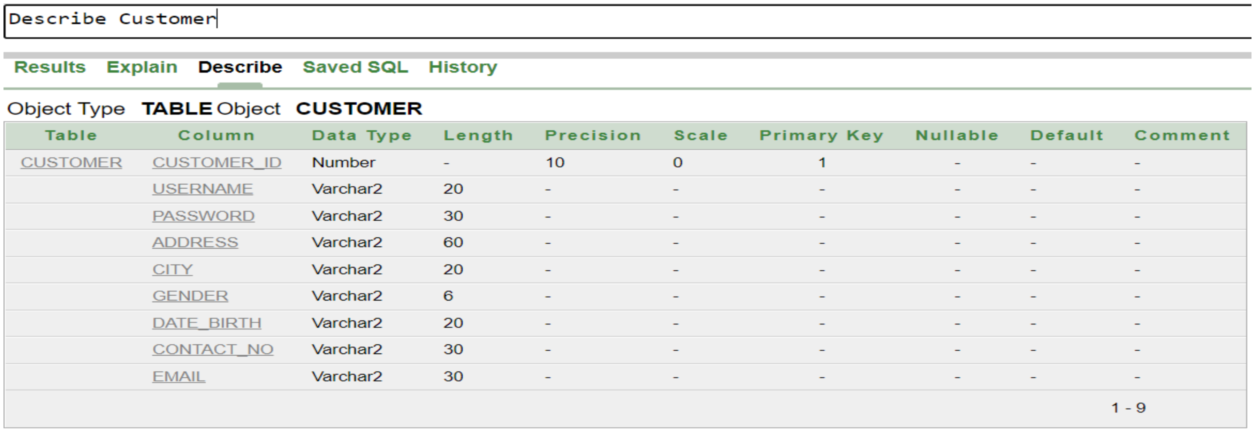
Gender varchar2(6) NOT NULL,

Date_birth varchar2(20) NOT NULL,

Contact_No varchar2(30) NOT NULL,

Email varchar2(30) NOT NULL

);



```
Describe Customer
```

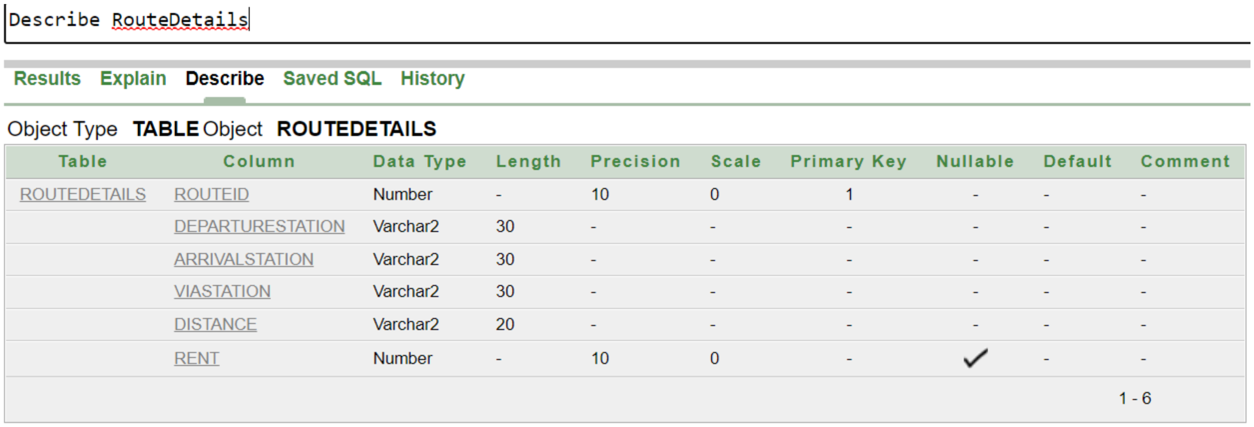
Object Type **TABLE** Object **CUSTOMER**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CUSTOMER	CUSTOMER_ID	Number	-	10	0	1	-	-	-
	USERNAME	Varchar2	20	-	-	-	-	-	-
	PASSWORD	Varchar2	30	-	-	-	-	-	-
	ADDRESS	Varchar2	60	-	-	-	-	-	-
	CITY	Varchar2	20	-	-	-	-	-	-
	GENDER	Varchar2	6	-	-	-	-	-	-
	DATE_BIRTH	Varchar2	20	-	-	-	-	-	-
	CONTACT_NO	Varchar2	30	-	-	-	-	-	-
	EMAIL	Varchar2	30	-	-	-	-	-	-

1 - 9

RouteDetails

```
Create table RouteDetails

(routeId number(10) CONSTRAINT RouteDetails_pk PRIMARY KEY,

departureStation varchar2(30)  NOT NULL ,

arrivalStation varchar2(30) NOT NULL,

viaStation varchar2(30) NOT NULL,

distance varchar2(20) NOT NULL,

Rent number(10) CONSTRAINT FK_rent REFERENCES Rent

);
```



```
Describe RouteDetails
```

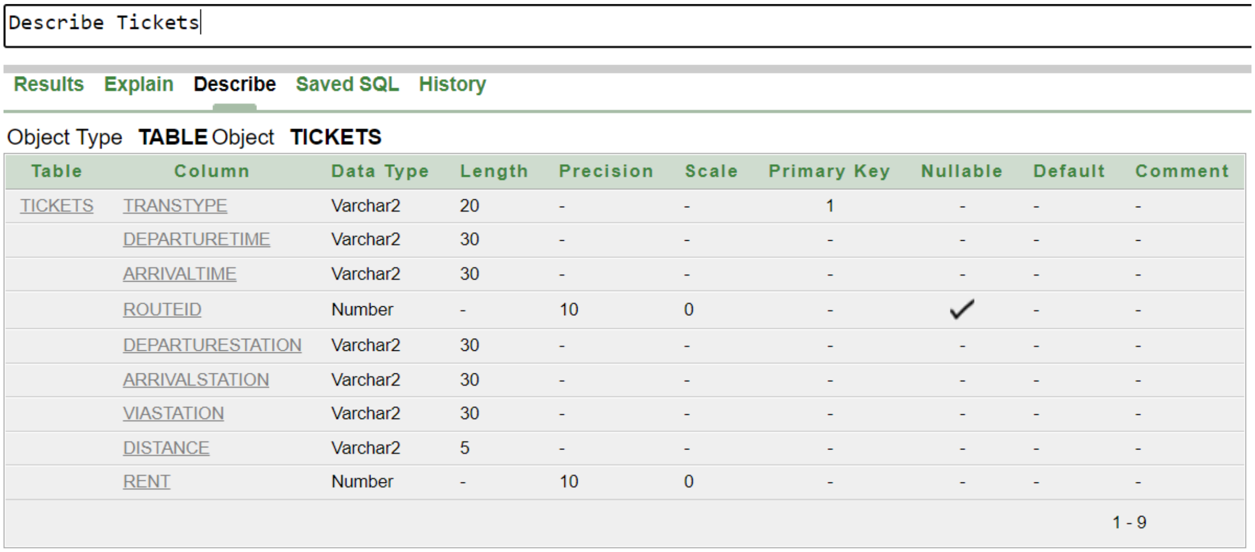
Object Type **TABLE** Object **ROUTEDETAILS**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ROUTEDETAILS	ROUTEID	Number	-	10	0	1	-	-	-
	DEPARTURESTATION	Varchar2	30	-	-	-	-	-	-
	ARRIVALSTATION	Varchar2	30	-	-	-	-	-	-
	VIASTATION	Varchar2	30	-	-	-	-	-	-
	DISTANCE	Varchar2	20	-	-	-	-	-	-
	RENT	Number	-	10	0	-	✓	-	-

1 - 6

Tickets

```
Create table Tickets

(TransType varchar2(20) CONSTRAINT PK_Tickets PRIMARY KEY,

departureTime varchar2(30) NOT NULL,

arrivalTime varchar2(30) NOT NULL,

routeId number(10) CONSTRAINT FK_departureStation REFERENCES RouteDetails,

departureStation varchar2(30) NOT NULL,

arrivalStation varchar2(30) NOT NULL,

viaStation varchar2(30) NOT NULL,

distance varchar2(5) NOT NULL,

Rent number(10) NOT NULL

);
```



```
Describe Tickets
```

Object Type **TABLE** Object **TICKETS**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
TICKETS	TRANSTYPE	Varchar2	20	-	-	1	-	-	-
	DEPARTURETIME	Varchar2	30	-	-	-	-	-	-
	ARRIVALTIME	Varchar2	30	-	-	-	-	-	-
	ROUTEID	Number	-	10	0	-	✓	-	-
	DEPARTURESTATION	Varchar2	30	-	-	-	-	-	-
	ARRIVALSTATION	Varchar2	30	-	-	-	-	-	-
	VIASTATION	Varchar2	30	-	-	-	-	-	-
	DISTANCE	Varchar2	5	-	-	-	-	-	-
	RENT	Number	-	10	0	-	-	-	-

1 - 9

Rent

Create table Rent

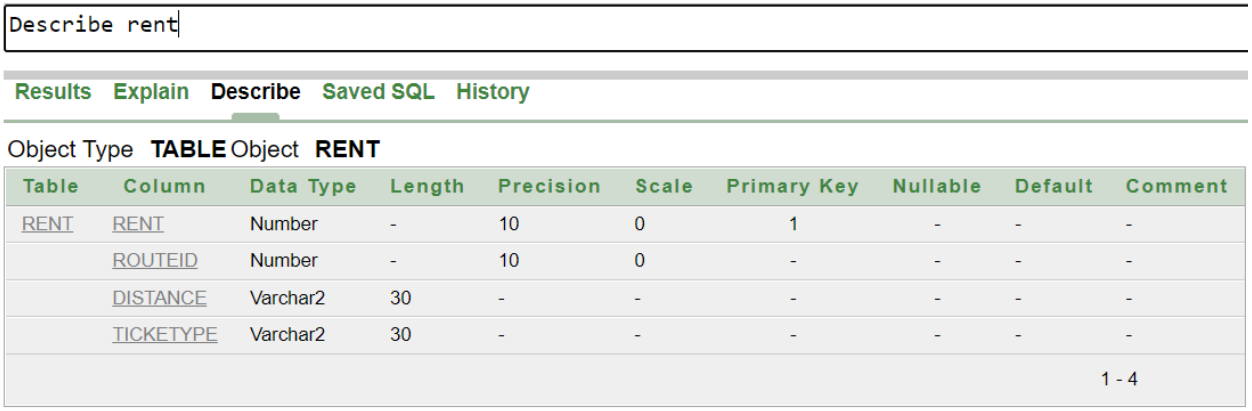
(Rent number(10) CONSTRAINT Rent_pk PRIMARY KEY,

RouteId number(10) NOT NULL,

Distance varchar2(30) NOT NULL,

TickeType varchar2(30) NOT NULL

);



```
Describe rent
```

Object Type **TABLE** Object **RENT**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
RENT	RENT	Number	-	10	0	1	-	-	-
	ROUTEID	Number	-	10	0	-	-	-	-
	DISTANCE	Varchar2	30	-	-	-	-	-	-
	TICKETYPE	Varchar2	30	-	-	-	-	-	-

1 - 4

Payment

Create table Payment

(transactionNO number(10) CONSTRAINT Payment_pk PRIMARY KEY,

Customer_id number(10) CONSTRAINT FK_Customer_id REFERENCES Customer,

owner_Name varchar2(30) NOT NULL,

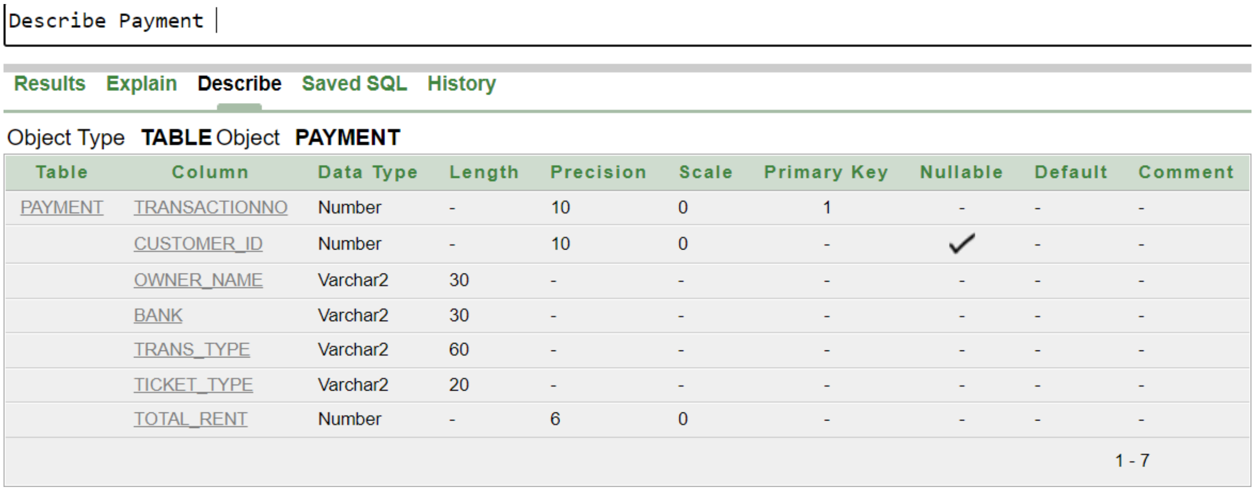
Bank varchar2(30) NOT NULL,

trans_type varchar2(60) NOT NULL,

ticket_type varchar2(20) NOT NULL,

Total_Rent number(6) NOT NULL

);



```
Describe Payment
```

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

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PAYMENT	TRANSACTIONNO	Number	-	10	0	1	-	-	-
	CUSTOMER_ID	Number	-	10	0	-	✓	-	-
	OWNER_NAME	Varchar2	30	-	-	-	-	-	-
	BANK	Varchar2	30	-	-	-	-	-	-
	TRANS_TYPE	Varchar2	60	-	-	-	-	-	-
	TICKET_TYPE	Varchar2	20	-	-	-	-	-	-
	TOTAL_RENT	Number	-	6	0	-	-	-	-

1 - 7

BookDetail

Create table BookDetail

(Seat_No varchar2(3) CONSTRAINT BookDetail_pk PRIMARY KEY,

RouteId number(10) ,

Journey_Date varchar2(30) NOT NULL,

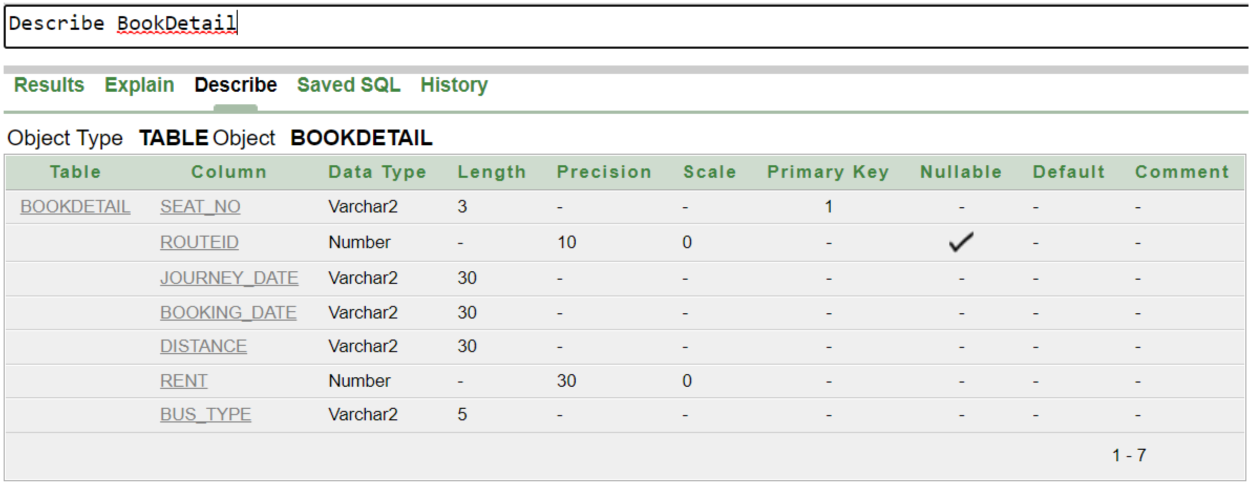
Booking_Date varchar2(30) NOT NULL,

Distance varchar2(30) NOT NULL,

Rent number(30) NOT NULL,

Bus_type varchar2(5) NOT NULL

);



```
Describe BookDetail
```

Object Type **TABLE** Object **BOOKDETAIL**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
BOOKDETAIL	SEAT_NO	Varchar2	3	-	-	1	-	-	-
	ROUTEID	Number	-	10	0	-	✓	-	-
	JOURNEY_DATE	Varchar2	30	-	-	-	-	-	-
	BOOKING_DATE	Varchar2	30	-	-	-	-	-	-
	DISTANCE	Varchar2	30	-	-	-	-	-	-
	RENT	Number	-	30	0	-	-	-	-
	BUS_TYPE	Varchar2	5	-	-	-	-	-	-

1 - 7

TimeTable

Create table TimeTable

(TimeTableId number(10) CONSTRAINT TimeTable_pk PRIMARY KEY,

routeid number(10)Not NULL,

departureStation varchar2(30) NOT NULL,

arrivalStation varchar2(30) NOT NULL,

viaStation varchar2(30) NOT NULL,

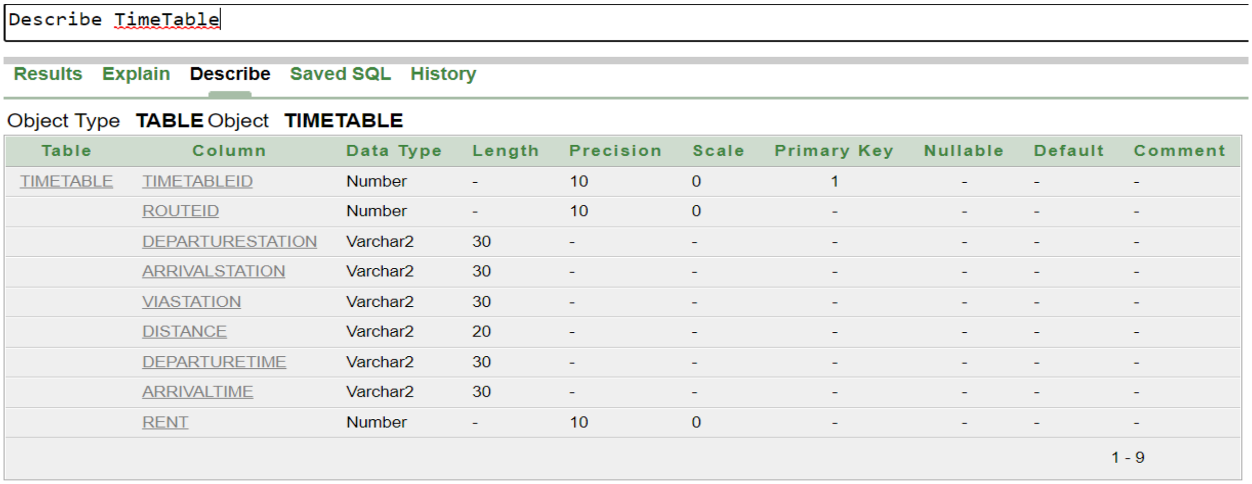
distance varchar2(20) NOT NULL,

departureTime varchar2(30) NOT NULL,

arrivalTime varchar2(30) NOT NULL,

rent number(10) NOT NULL

);



```
Describe TimeTable
```

Object Type **TABLE** Object **TIMETABLE**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
TIMETABLE	TIMETABLEID	Number	-	10	0	1	-	-	-
	ROUTEID	Number	-	10	0	-	-	-	-
	DEPARTURESTATION	Varchar2	30	-	-	-	-	-	-
	ARRIVALSTATION	Varchar2	30	-	-	-	-	-	-
	VIASTATION	Varchar2	30	-	-	-	-	-	-
	DISTANCE	Varchar2	20	-	-	-	-	-	-
	DEPARTURETIME	Varchar2	30	-	-	-	-	-	-
	ARRIVALTIME	Varchar2	30	-	-	-	-	-	-
	RENT	Number	-	10	0	-	-	-	-

1 - 9

Data Insertion:

Admin

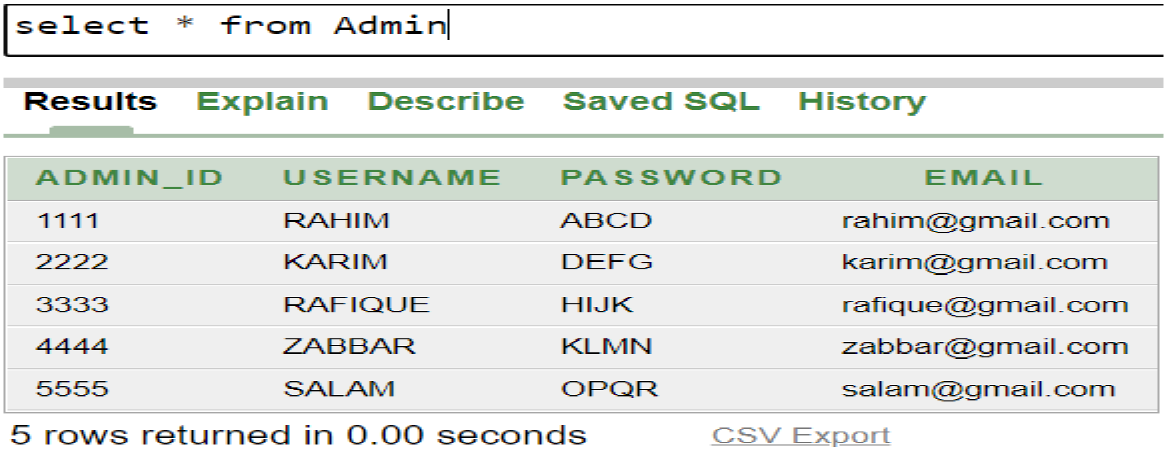
Insert into Admin values ('1111','RAHIM','ABCD','rahim@gmail.com')

Insert into Admin values ('2222','KARIM','DEFG','karim@gmail.com')

Insert into Admin values ('3333','RAFIQUE','HIJK','rafique@gmail.com')

Insert into Admin values ('4444','ZABBAR','KLMN','zabbar@gmail.com')

Insert into Admin values ('5555','SALAM','OPQR','salam@gmail.com')



```
select * from Admin
```

ADMIN_ID	USERNAME	PASSWORD	EMAIL
1111	RAHIM	ABCD	rahim@gmail.com
2222	KARIM	DEFG	karim@gmail.com
3333	RAFIQUE	HIJK	rafique@gmail.com
4444	ZABBAR	KLMN	zabbar@gmail.com
5555	SALAM	OPQR	salam@gmail.com

5 rows returned in 0.00 seconds

Customer

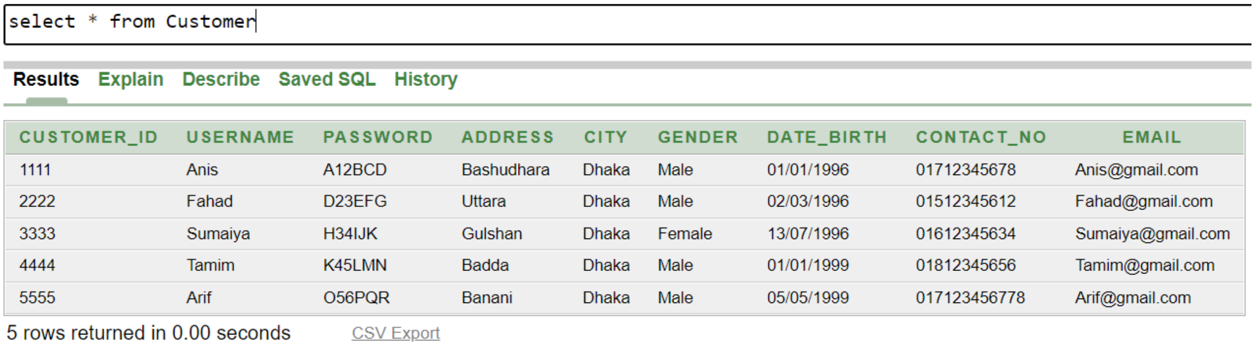
Insert into Customer values ('1111','Anis','A12BCD','Bashudhara','Dhaka','Male','01/01/1996','01712345678','Anis@gmail.com')

Insert into Customer values ('2222','Fahad','D23EFG','Uttara','Dhaka','Male','02/03/1996','01512345612','Fahad@gmail.com')

Insert into Customer values ('3333','Sumaiya','H34IJK','Gulshan','Dhaka','Female','13/07/1996','01612345634','Sumaiya@gmail.com')

Insert into Customer values ('4444','Tamim','K45LMN','Badda','Dhaka','Male','01/01/1999','01812345656','Tamim@gmail.com')

Insert into Customer values ('5555','Arif','O56PQR','Banani','Dhaka','Male','05/05/1999','017123456778','Arif@gmail.com')



```
select * from Customer
```

CUSTOMER_ID	USERNAME	PASSWORD	ADDRESS	CITY	GENDER	DATE_BIRTH	CONTACT_NO	EMAIL
1111	Anis	A12BCD	Bashudhara	Dhaka	Male	01/01/1996	01712345678	Anis@gmail.com
2222	Fahad	D23EFG	Uttara	Dhaka	Male	02/03/1996	01512345612	Fahad@gmail.com
3333	Sumaiya	H34IJK	Gulshan	Dhaka	Female	13/07/1996	01612345634	Sumaiya@gmail.com
4444	Tamim	K45LMN	Badda	Dhaka	Male	01/01/1999	01812345656	Tamim@gmail.com
5555	Arif	O56PQR	Banani	Dhaka	Male	05/05/1999	017123456778	Arif@gmail.com

5 rows returned in 0.00 seconds

RouteDetails

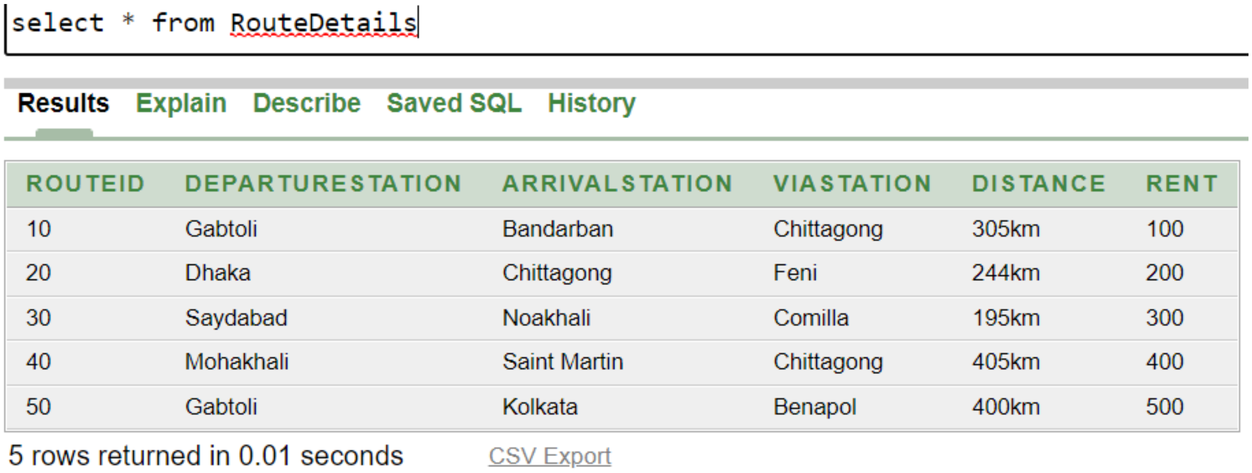
Insert into RouteDetails values ('10','Gabtoli','Bandarban','Chittagong','305km','100')

Insert into RouteDetails values ('20','Dhaka','Chittagong','Feni','244km','200')

Insert into RouteDetails values ('30','Saydabad','Noakhali','Comilla','195km','300')

Insert into RouteDetails values ('40','Mohakhali','Saint Martin','Chittagong','405km','400')

Insert into RouteDetails values ('50','Gabtoli','Kolkata','Benapol','400km','500')



```
select * from RouteDetails
```

ROUTEID	DEPARTURESTATION	ARRIVALSTATION	VIASTATION	DISTANCE	RENT
10	Gabtoli	Bandarban	Chittagong	305km	100
20	Dhaka	Chittagong	Feni	244km	200
30	Saydabad	Noakhali	Comilla	195km	300
40	Mohakhali	Saint Martin	Chittagong	405km	400
50	Gabtoli	Kolkata	Benapol	400km	500

5 rows returned in 0.01 seconds

Tickets

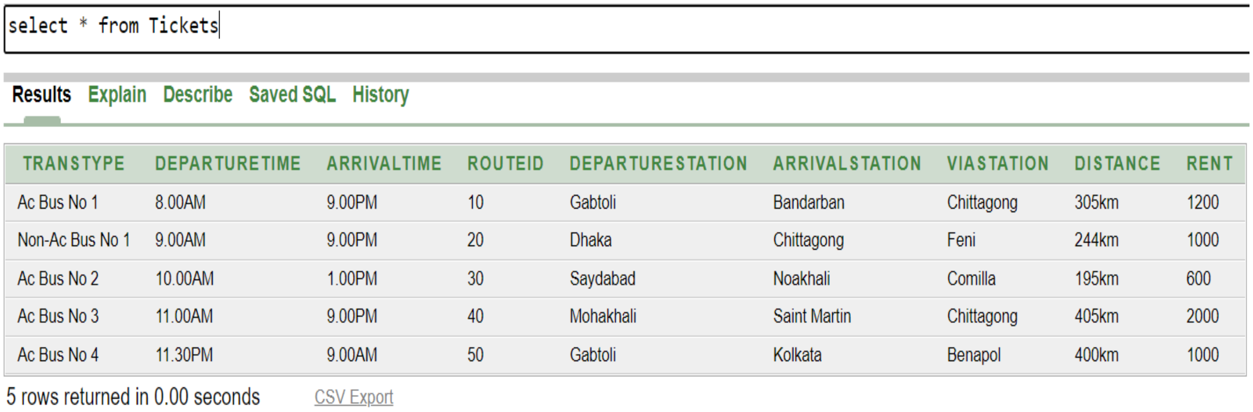
Insert into Tickets values ('Ac Bus No 1','8.00AM','9.00PM','10','Gabtoli','Bandarban','Chittagong','305km','1200')

Insert into Tickets values ('Non-Ac Bus No 1','9.00AM','9.00PM','20','Dhaka','Chittagong','Feni','244km','1000')

Insert into Tickets values ('Ac Bus No 2','10.00AM','1.00PM','30','Saydabad','Noakhali','Comilla','195km','600')

Insert into Tickets values ('Ac Bus No 3','11.00AM','9.00PM','40','Mohakhali','Saint Martin','Chittagong','405km','2000')

Insert into Tickets values ('Ac Bus No 4','11.30PM','9.00AM','50','Gabtoli','Kolkata','Benapol','400km','1000')



```
select * from Tickets
```

TRANSTYPE	DEPARTURETIME	ARRIVALTIME	ROUTEID	DEPARTURESTATION	ARRIVALSTATION	VIASTATION	DISTANCE	RENT
Ac Bus No 1	8.00AM	9.00PM	10	Gabtoli	Bandarban	Chittagong	305km	1200
Non-Ac Bus No 1	9.00AM	9.00PM	20	Dhaka	Chittagong	Feni	244km	1000
Ac Bus No 2	10.00AM	1.00PM	30	Saydabad	Noakhali	Comilla	195km	600
Ac Bus No 3	11.00AM	9.00PM	40	Mohakhali	Saint Martin	Chittagong	405km	2000
Ac Bus No 4	11.30PM	9.00AM	50	Gabtoli	Kolkata	Benapol	400km	1000

5 rows returned in 0.00 seconds

Rent

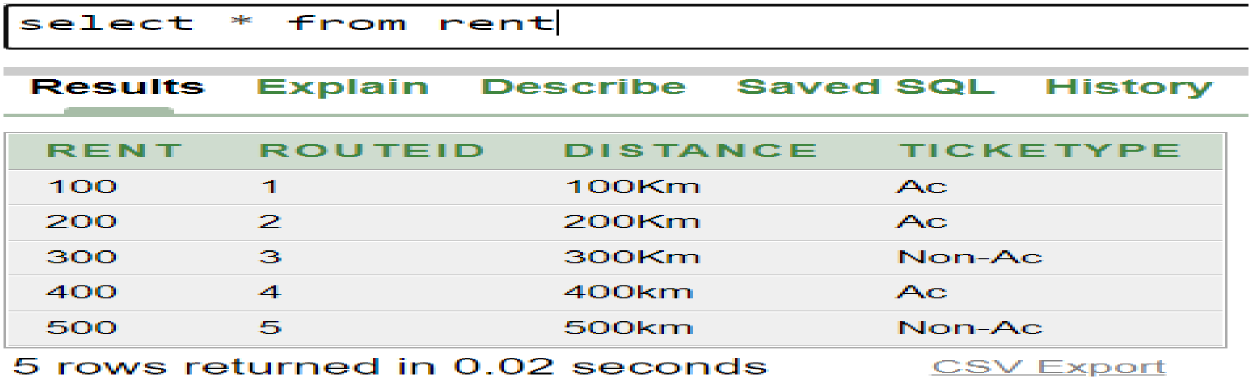
Insert into Rent values ('100','001','100Km','Ac')

Insert into Rent values ('200','002','200Km','Ac')

Insert into Rent values ('300','003','300Km','Non-Ac')

Insert into Rent values ('400','004','400km','Ac')

Insert into Rent values ('500','005','500km','Non-Ac')



```
select * from rent
```

RENT	ROUTEID	DISTANCE	TICKETYPE
100	1	100Km	Ac
200	2	200Km	Ac
300	3	300Km	Non-Ac
400	4	400km	Ac
500	5	500km	Non-Ac

5 rows returned in 0.02 seconds

Payment

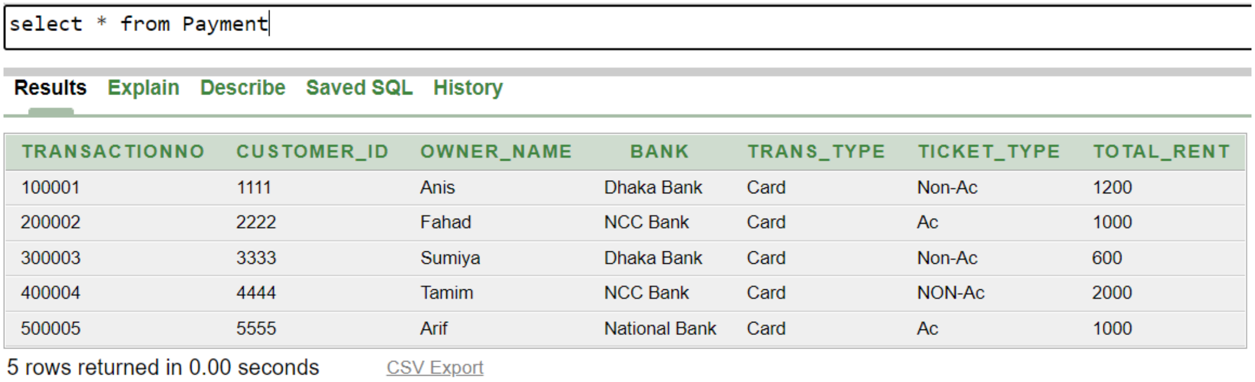
Insert into Payment values ('100001','1111','Anis','Dhaka Bank','Card','Non-Ac','1200')

Insert into Payment values ('200002','2222','Fahad','NCC Bank','Card','Ac','1000')

Insert into Payment values ('300003','3333','Sumiya','Dhaka Bank','Card','Non-Ac','600')

Insert into Payment values ('400004','4444','Tamim','NCC Bank','Card','NON-Ac','2000')

Insert into Payment values ('500005','5555','Arif','National Bank','Card','Ac','1000')



```
select * from Payment
```

TRANSACTIONNO	CUSTOMER_ID	OWNER_NAME	BANK	TRANS_TYPE	TICKET_TYPE	TOTAL_RENT
100001	1111	Anis	Dhaka Bank	Card	Non-Ac	1200
200002	2222	Fahad	NCC Bank	Card	Ac	1000
300003	3333	Sumiya	Dhaka Bank	Card	Non-Ac	600
400004	4444	Tamim	NCC Bank	Card	NON-Ac	2000
500005	5555	Arif	National Bank	Card	Ac	1000

5 rows returned in 0.00 seconds

BookDetail

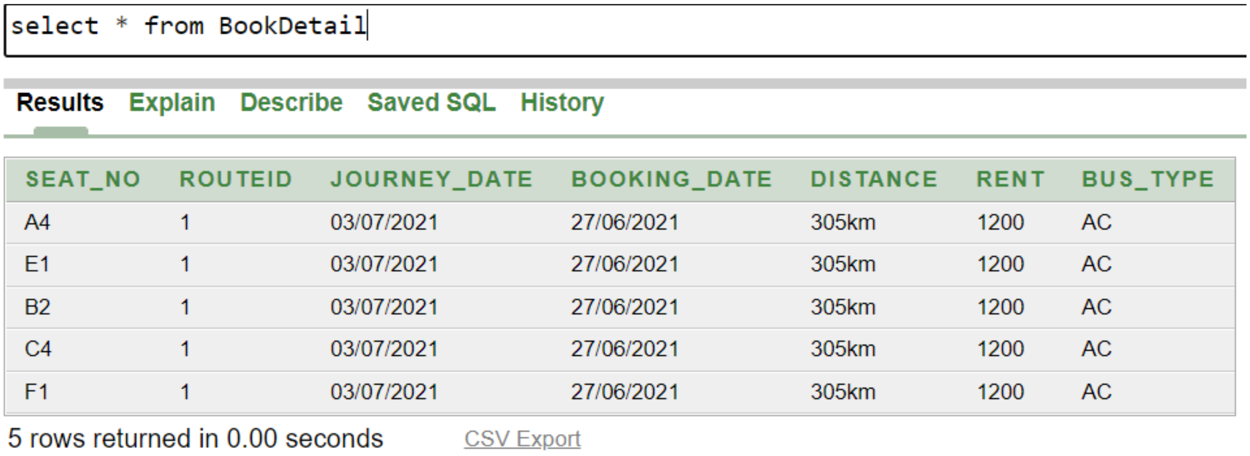
Insert into BookDetail values ('A4','001','03/07/2021','27/06/2021','305km','1200','AC')

Insert into BookDetail values ('E1','001','03/07/2021','27/06/2021','305km','1200','AC')

Insert into BookDetail values ('B2','001','03/07/2021','27/06/2021','305km','1200','AC')

Insert into BookDetail values ('C4','001','03/07/2021','27/06/2021','305km','1200','AC')

Insert into BookDetail values ('F1','001','03/07/2021','27/06/2021','305km','1200','AC')



```
select * from BookDetail
```

SEAT_NO	ROUTEID	JOURNEY_DATE	BOOKING_DATE	DISTANCE	RENT	BUS_TYPE
A4	1	03/07/2021	27/06/2021	305km	1200	AC
E1	1	03/07/2021	27/06/2021	305km	1200	AC
B2	1	03/07/2021	27/06/2021	305km	1200	AC
C4	1	03/07/2021	27/06/2021	305km	1200	AC
F1	1	03/07/2021	27/06/2021	305km	1200	AC

5 rows returned in 0.00 seconds

TimeTable

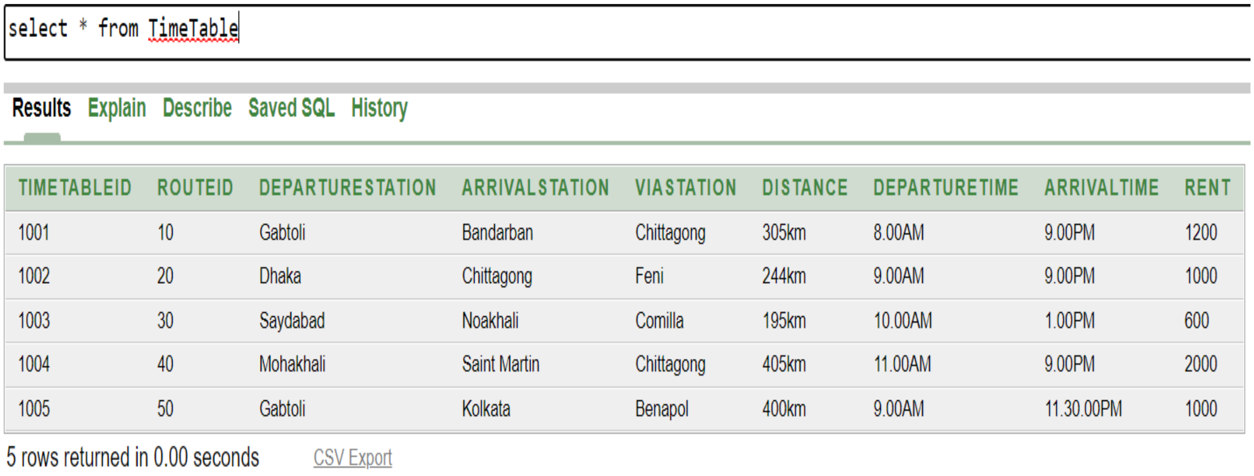
Insert into TimeTable values ('1001','10','Gabtoli','Bandarban','Chittagong','305km','8.00AM','9.00PM','1200')

Insert into TimeTable values ('1002','20','Dhaka','Chittagong','Feni','244km','9.00AM','9.00PM','1000')

Insert into TimeTable values ('1003','30','Saydabad','Noakhali','Comilla','195km','10.00AM','1.00PM','600')

Insert into TimeTable values ('1004','40','Mohakhali','Saint Martin','Chittagong','405km','11.00AM','9.00PM','2000')

Insert into TimeTable values ('1005','50','Gabtoli','Kolkata','Benapol','400km','9.00AM','11.30.00PM','1000')



```
select * from TimeTable
```

TIMETABLEID	ROUTEID	DEPARTURESTATION	ARRIVALSTATION	VIASTATION	DISTANCE	DEPARTURETIME	ARRIVALTIME	RENT
1001	10	Gabtoli	Bandarban	Chittagong	305km	8.00AM	9.00PM	1200
1002	20	Dhaka	Chittagong	Feni	244km	9.00AM	9.00PM	1000
1003	30	Saydabad	Noakhali	Comilla	195km	10.00AM	1.00PM	600
1004	40	Mohakhali	Saint Martin	Chittagong	405km	11.00AM	9.00PM	2000
1005	50	Gabtoli	Kolkata	Benapol	400km	9.00AM	11.30.00PM	1000

5 rows returned in 0.00 seconds

Sequences

1. CREATE SEQUENCE SQ_Admin_ID START WITH 1111

INCREMENT BY 1

MAXVALUE 99999

NOCACHE NOCYCLE

2. CREATE SEQUENCE SQ_Customer_ID START WITH 1

INCREMENT BY 1

MAXVALUE 9999

NOCACHE NOCYCLE

3. CREATE SEQUENCE SQ_Route_ID START WITH 10

INCREMENT BY 1

MAXVALUE 9999

NOCACHE NOCYCLE

4. CREATE SEQUENCE SQ_Ticket_ID START WITH 1000

INCREMENT BY 1

MAXVALUE 10000

NOCACHE NOCYCLE

5. CREATE SEQUENCE SQ_Rent START WITH 200

INCREMENT BY 1

MAXVALUE 5000

NOCACHE NOCYCLE

6. CREATE SEQUENCE SQ_ transactionNO START WITH 10000

INCREMENT BY 1

MAXVALUE 9000000

NOCACHE NOCYCLE

7. CREATE SEQUENCE SQ_SeatNo START WITH 1

INCREMENT BY 1

MAXVALUE 60

NOCACHE NOCYCLE

8. CREATE SEQUENCE SQ_TimeTable_ID START WITH 1000

INCREMENT BY 1

MAXVALUE 9999

NOCACHE NOCYCLE

Index:

1. CREATE INDEX RouteDetails_IDX ON RouteDetails (RD_NAME);

2. CREATE INDEX RouteDetails _IDX ON RouteDetails (T_NAME);

3. CREATE INDEX Rent_IDX ON Rent (R_NAME);

4. CREATE INDEX Payment_IDX ON Payment (Payment _NAME);

5. CREATE INDEX BookDetails_IDX ON BookDetails (BD_NAME);

6. CREATE INDEX TimeTable_IDX ON TimeTable (TT_NAME);

7. CREATE INDEX Customer_IDX ON Customer (C_NAME)

8. CREATE INDEX Admin_IDX ON Admin (A_NAME)

Roles:

CREATE ROLE Arif;

GRANT CREATE SESSION, ALTER SESSION, CREATE TABLE, CREATE CLUSTER, CREATE SYNONYM, CREATE VIEW,

CREATE SEQUENCE, CREATE DATABASE LINK, CREATE PROCEDURE, CREATE TRIGGER, CREATE TYPE, CREATE

OPERATOR, CREATE INDEXTYPE, BACKUP ANY TABLE,SELECT ANY TABLE, CREATE ANY TABLE, CREATE ANY INDEX,

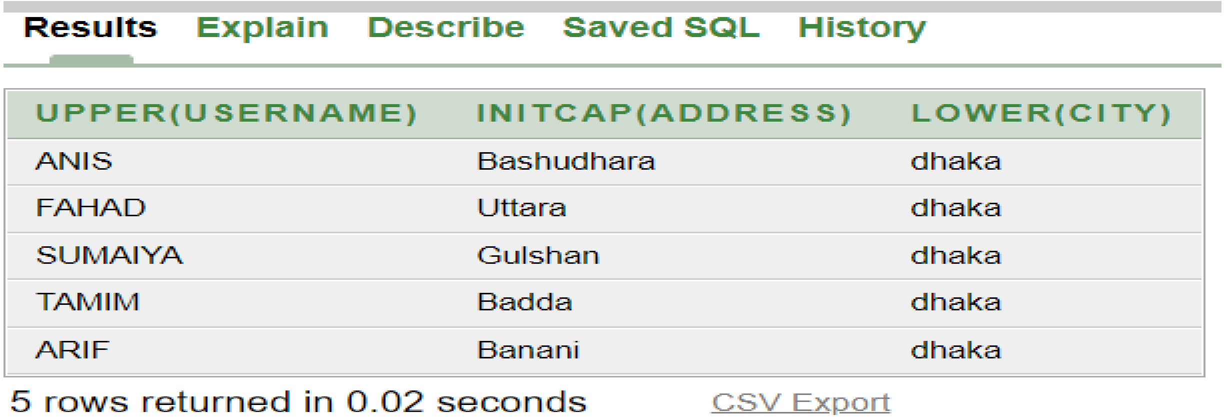
ALTER ANY INDEX, ALTER ANY INDEXTYPE, DROP ANY INDEX, DROP ANY INDEXTYPE TO Arif;

SQL Query Writing:

Single-Row_function

1.Show the username in upper latter, 1st word capital latter of address and city in all small latter from customer table

SELECT UPPER (username), INITCAP (address), LOWER (city) FROM Customer

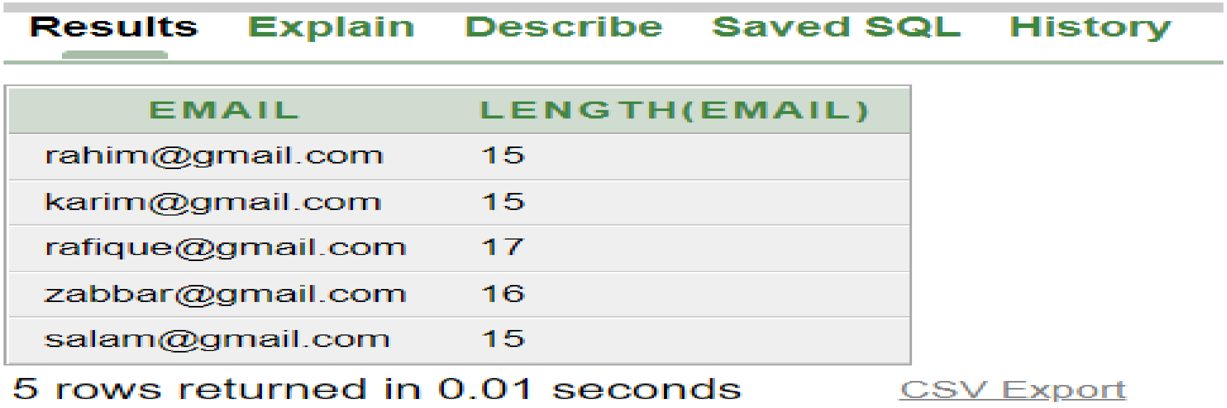


UPPER(USERNAME)	INITCAP(ADDRESS)	LOWER(CITY)
ANIS	Bashudhara	dhaka
FAHAD	Uttara	dhaka
SUMAIYA	Gulshan	dhaka
TAMIM	Badda	dhaka
ARIF	Banani	dhaka

5 rows returned in 0.02 seconds

2.

>>SELECT email, length(email) from Admin;

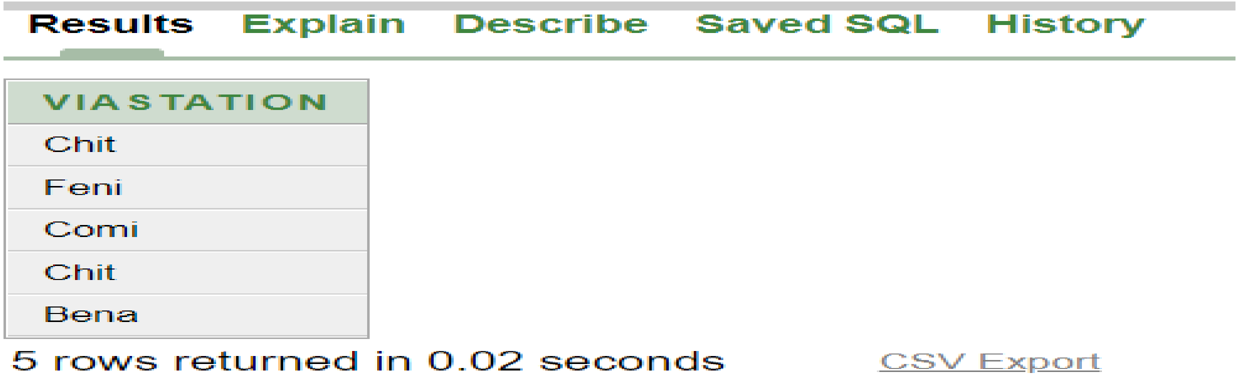


EMAIL	LENGTH(EMAIL)
rahim@gmail.com	15
karim@gmail.com	15
rafique@gmail.com	17
zabbar@gmail.com	16
salam@gmail.com	15

5 rows returned in 0.01 seconds

3. Display the 1st 4 latter of viastation from tickets table

SELECT SUBSTR(viastation,1,4) as viastation FROM Tickets;



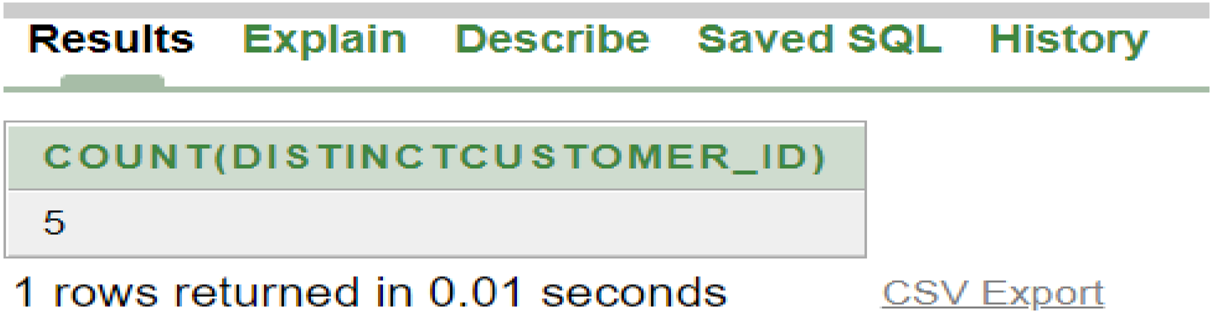
VIASTATION
Chit
Feni
Comi
Chit
Bena

5 rows returned in 0.02 seconds

Group_function

1. Display total number of customers in the database

>> SELECT COUNT (DISTINCT customer_ID) FROM customer;

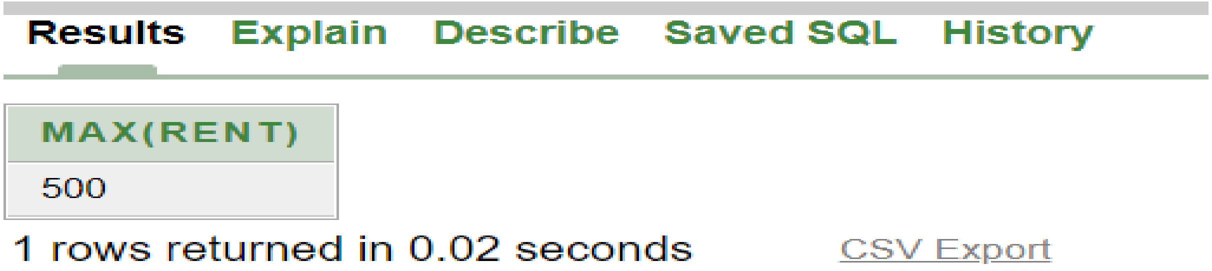


COUNT(DISTINCTCUSTOMER_ID)
5

1 rows returned in 0.01 seconds

2. Show the Maximum rent from rent

>>SELECT MAX(rent) FROM rent;



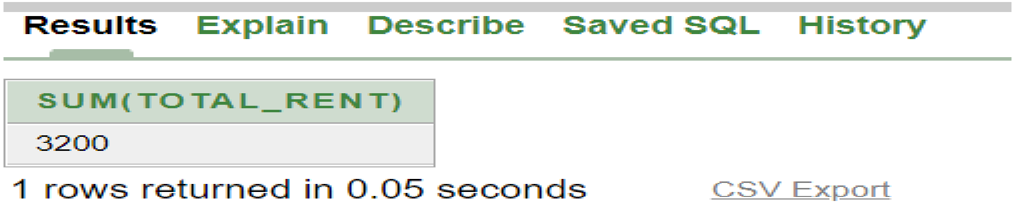
MAX(RENT)
500

1 rows returned in 0.02 seconds

3. Display total number rent where

>> SELECT SUM(Total_rent) FROM payment

WHERE Total_rent>1000;



SUM(TOTAL_RENT)
3200

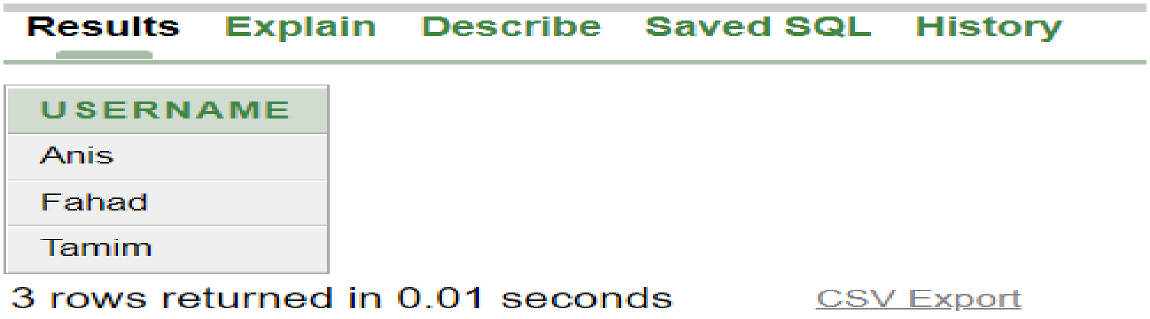
1 rows returned in 0.05 seconds

Sub-query

1. Display the user name of customer where customer's birthday come before Arif's birthday from customer table

>>select username from customer

where date_birth<(select date_birth from customer where username ='Arif');



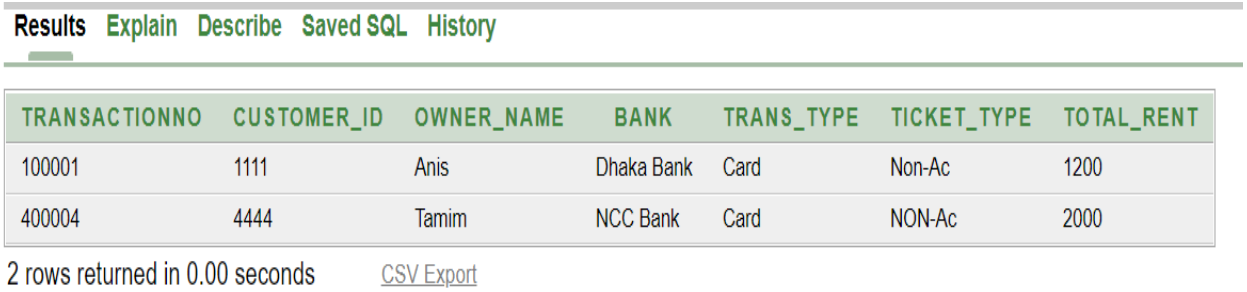
USERNAME
Anis
Fahad
Tamim

3 rows returned in 0.01 seconds

2. Display all the the owner who pay more total rent than Arif from payment table

>>select * from payment

where total_rent>(select total_rent from payment where owner_name='Arif');



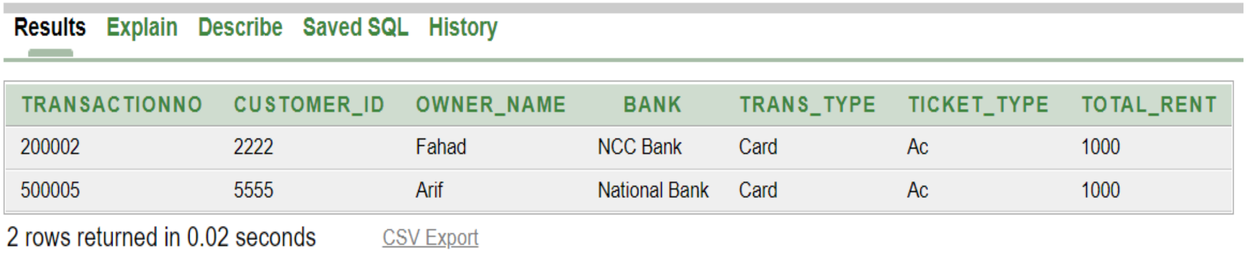
TRANSACTIONNO	CUSTOMER_ID	OWNER_NAME	BANK	TRANS_TYPE	TICKET_TYPE	TOTAL_RENT
100001	1111	Anis	Dhaka Bank	Card	Non-Ac	1200
400004	4444	Tamim	NCC Bank	Card	NON-Ac	2000

2 rows returned in 0.00 seconds

3. Display the table where total rent is same as the owner Arif

>> select * from payment

where total_rent=(select total_rent from payment where owner_name='Arif');



TRANSACTIONNO	CUSTOMER_ID	OWNER_NAME	BANK	TRANS_TYPE	TICKET_TYPE	TOTAL_RENT
200002	2222	Fahad	NCC Bank	Card	Ac	1000
500005	5555	Arif	National Bank	Card	Ac	1000

2 rows returned in 0.02 seconds

joining

View

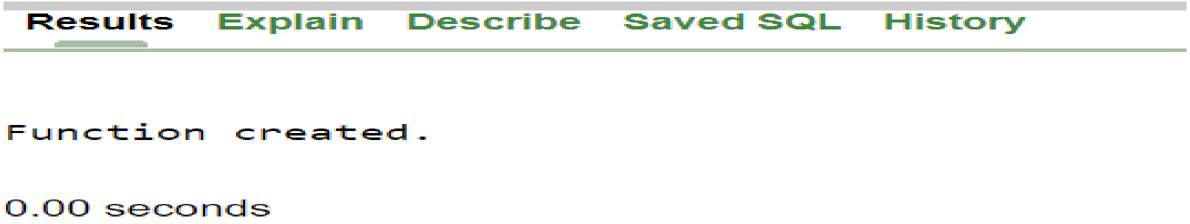
Synonym

PL/SQL Query Writing:

Function

1. Create a function where TimeTable table return the Bus Schedule.

```
>>CREATE OR REPLACE FUNCTION BusSchedule
RETURN number IS
total number(5) := 0;
BEGIN
SELECT count(*) into total
FROM TimeTable;
RETURN total;
END;
```

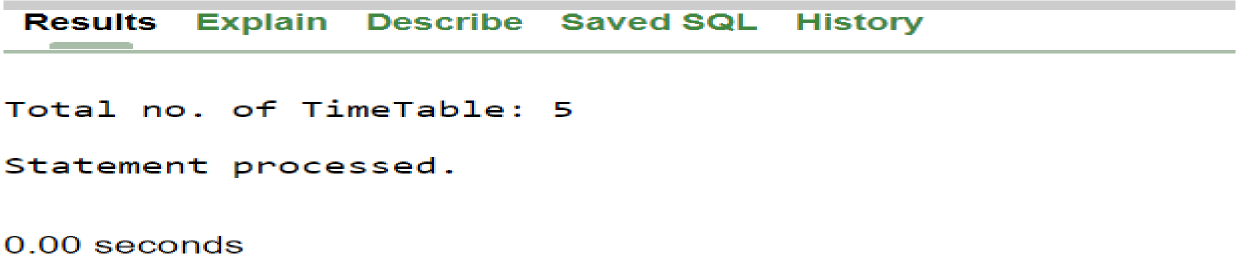


```
Function created.

0.00 seconds
```

2.Show the total number of of timetable

```
>> DECLARE
 t number(5);
BEGIN
t := BusSchedule ();
dbms_output.put_line('Total no. of TimeTable: ' || t);
```

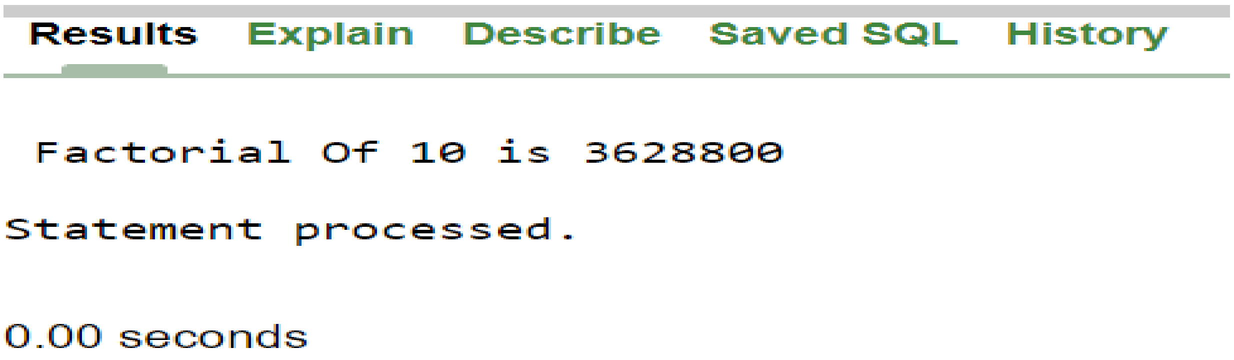
```
            END;
```



```
Total no. of TimeTable: 5

Statement processed.

0.00 seconds
```

3. Calculate the factorial of the number “10” using function

```
>> DECLARE
             n number;
            factorial number;
FUNCTION fact(x number)
RETURN number
IS
             f number;
BEGIN
            IF x=0 THEN
             f := 1;
            ELSE
             f := x * fact(x-1);
             END IF;
RETURN f;
END;

BEGIN
             n:= 10;
            factorial := fact(n);
             dbms_output.put_line(' Factorial of '|| n|| ' is ' || factorial);
END;
```

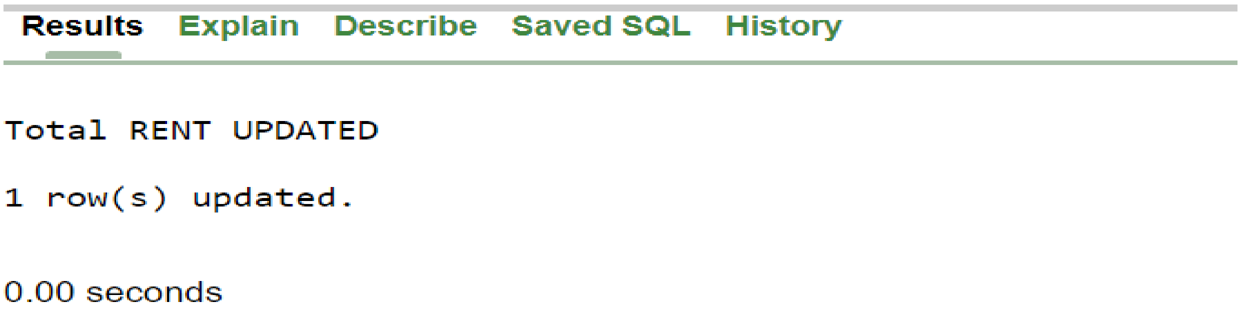


```
Factorial Of 10 is 3628800

Statement processed.

0.00 seconds
```

Procedure

1. Create a procedure to change the total_rent column to 2000 from payment table

```
>> DECLARE
  VAR_ED varchar2(20);
PROCEDURE rentUpdate(ED IN NUMBER) IS
BEGIN
UPDATE payment SET TOTAL_RENT=ED;
END;
BEGIN
VAR_ED:= 2000;
rentUpdate(VAR_ED);
dbms_output.put_line('Total RENT UPDATED ');
END;
```

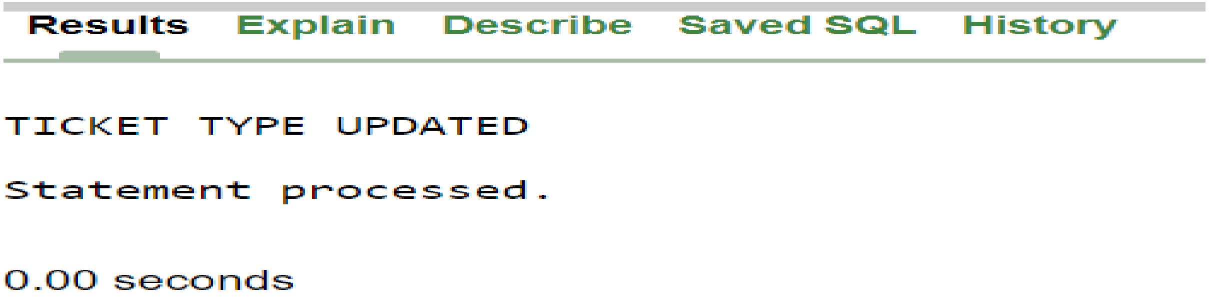


```
Total RENT UPDATED

1 row(s) updated.

0.00 seconds
```

2. Create procedure to change ticket type column to AC from rent table

```
>> DECLARE
VAR_ED varchar2(20);
```

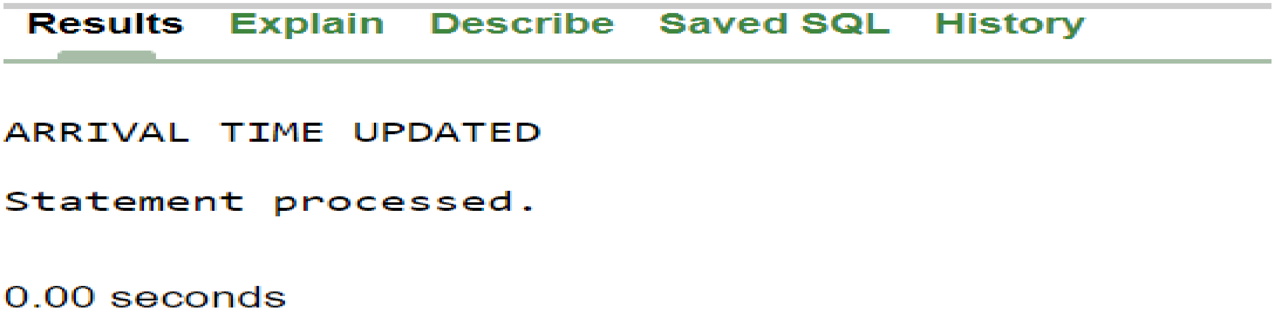
```
PROCEDURE ticketUpdate(ED IN VARCHAR2) IS
BEGIN
UPDATE rent SET TICKETYPE=ED;
END;
BEGIN
VAR_ED:= 'AC';
ticketUpdate(VAR_ED);
dbms_output.put_line('Total RENT UPDATED ');
END;
```



```
TICKET TYPE UPDATED

Statement processed.

0.00 seconds
```

3. Create a procedure to change the arrivaltime column of timetable time to “12/00 PM”

```
>> DECLARE
VAR_ED varchar2(20);
PROCEDURE arrivaltimeUpdate(ED IN VARCHAR2) IS
BEGIN
UPDATE timetable SET ARRIVALTIME=ED;
END;
BEGIN
VAR_ED:= '12.00 PM';
arrivaltimeUpdate(VAR_ED);
dbms_output.put_line('ARRIVAL TIME UPDATED ');
END;
```

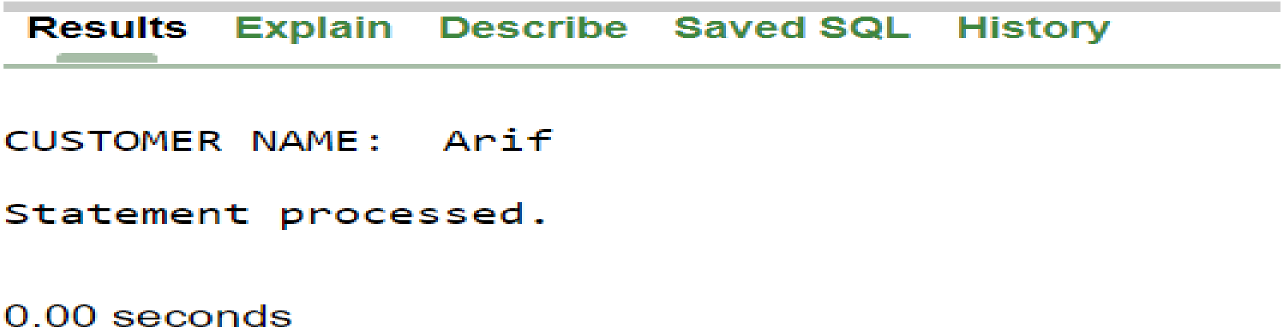


```
ARRIVAL TIME UPDATED

Statement processed.

0.00 seconds
```

<u>Record</u>

1. Create a record that can show output of username of the customer whose customer_id=5555

```
>> DECLARE
     CUSTOMER_NAME CUSTOMER%ROWTYPE;
     BEGIN
          SELECT * INTO CUSTOMER_NAME FROM CUSTOMER
          WHERE CUSTOMER_ID=5555;
          DBMS_OUTPUT.PUT_LINE('CUSTOMER NAME: '||' '
          ||CUSTOMER_NAME.USERNAME);
     END;
```

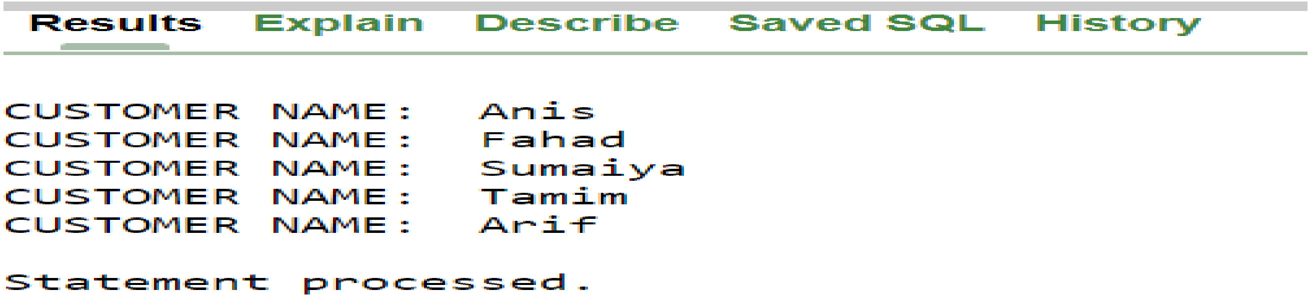


```
CUSTOMER NAME:  Arif

Statement processed.

0.00 seconds
```

2. Create a record that can show all the customer name inside the customer table

```
>> DECLARE
          CUSTOMER_NAME CUSTOMER%ROWTYPE;
BEGIN
```

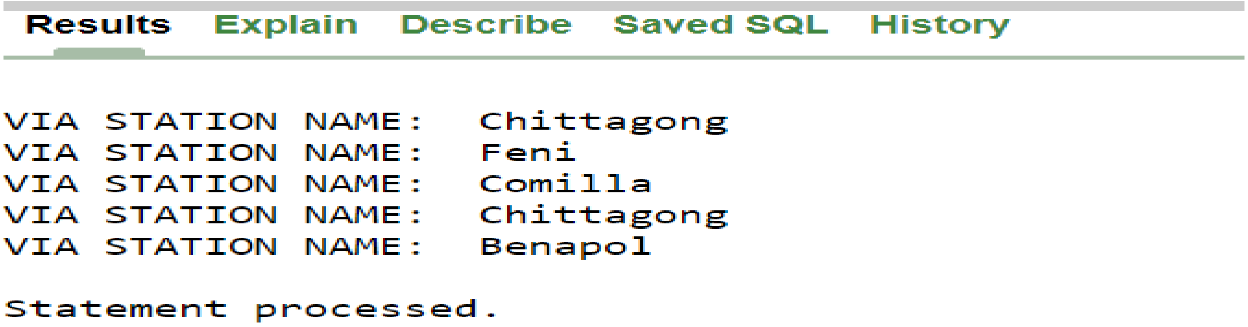
```
        FOR CUSTOMER_NAME IN(SELECT * FROM CUSTOMER)
LOOP
        DBMS_OUTPUT.PUT_LINE('CUSTOMER NAME: '||'
        '||CUSTOMER_NAME.USERNAME);
END LOOP;
END;
```



```
CUSTOMER NAME:  Anis
CUSTOMER NAME:  Fahad
CUSTOMER NAME:  Sumaiya
CUSTOMER NAME:  Tamim
CUSTOMER NAME:  Arif

Statement processed.
```

3. Create a record that can show the via station name from the ticket tables

```
>> DECLARE
        vStation TICKETS%ROWTYPE;
BEGIN
        FOR vStation    IN(SELECT * FROM TICKETS)
LOOP
        DBMS_OUTPUT.PUT_LINE('VIA STATION NAME: '||'
        '||vStation.VIASTATION);
END LOOP;
END;
```

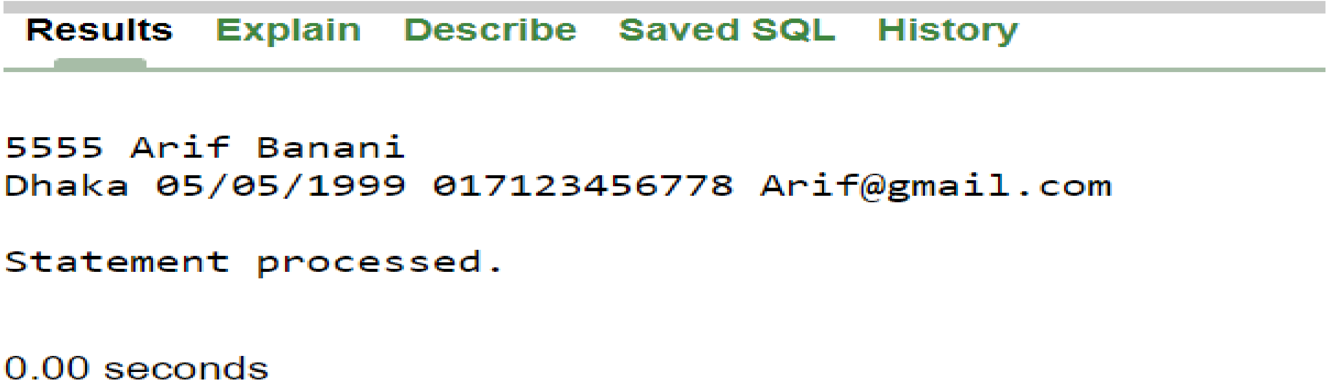


```
VIA STATION NAME:  Chittagong
VIA STATION NAME:  Feni
VIA STATION NAME:  Comilla
VIA STATION NAME:  Chittagong
VIA STATION NAME:  Benapol

Statement processed.
```

Cursor

1. Show the customer_id, username, address, city, date_birth, contact_no, and email from customer table where username= “Arif”

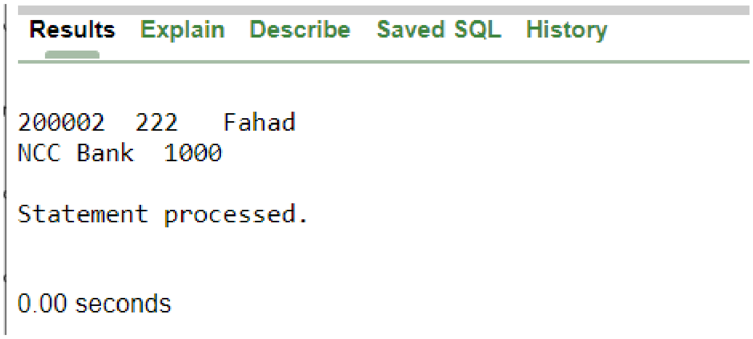
```
>> declare
cursor c_customer is
select * from customer where username='Arif';
rec_customer customer%rowtype;
begin
open c_customer;
fetch c_customer into rec_customer;
dbms_output.put_line(rec_customer.customer_id||' '||rec_customer.username||'
'||rec_customer.address||'
'||rec_customer.city||' '||rec_customer.date_birth||' '||rec_customer.contact_no||'
'||rec_customer.email);
close c_customer;
end;
```



```
5555 Arif Banani
Dhaka 05/05/1999 017123456778 Arif@gmail.com

Statement processed.


0.00 seconds
```

2. Show the TransactionNo, Customer_id, ownername, bank, total_rent from Payment table where owner_name = "Fahad"

```
>> declare
cursor c_payment is
select * from payment where owner_name='Fahad';
rec_payment payment%rowtype;
begin
open c_payment;
fetch c_payment into rec_payment;
dbms_output.put_line(rec_payment.TransactionNo||' '||rec_payment.customer_id||'
'||rec_payment.ownername||'
'||rec_payment.bank||' '||rec_payment.Total_rent);
close c_payment;
end;
```

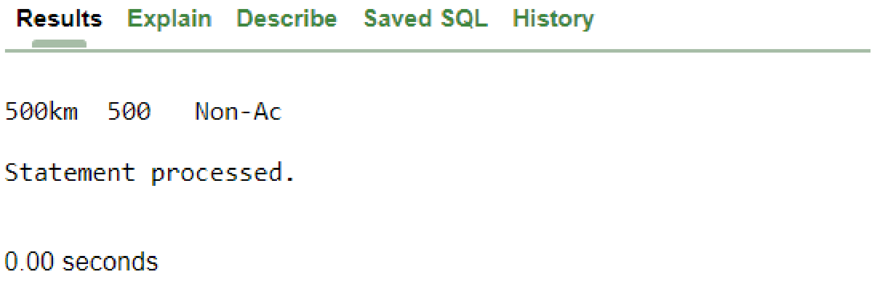


```
200002  222   Fahad
NCC Bank  1000

Statement processed.


0.00 seconds
```

3. Show the Distance, Rent and Ticket Type for RouteID ‘5’ from Rent Table.

```
>> declare
cursor c_rent is
select * from rent where RouteID='5';
rec_rent rent%rowtype;
begin
open c_rent;
fetch c_rent into rec_rent;
```

```
dbms_output.put_line(rec_rent.DISTANCE||' '||rec_RENT.RENT||' '||rec_RENT.TICKETYPE);
close c_RENT;
end;
```

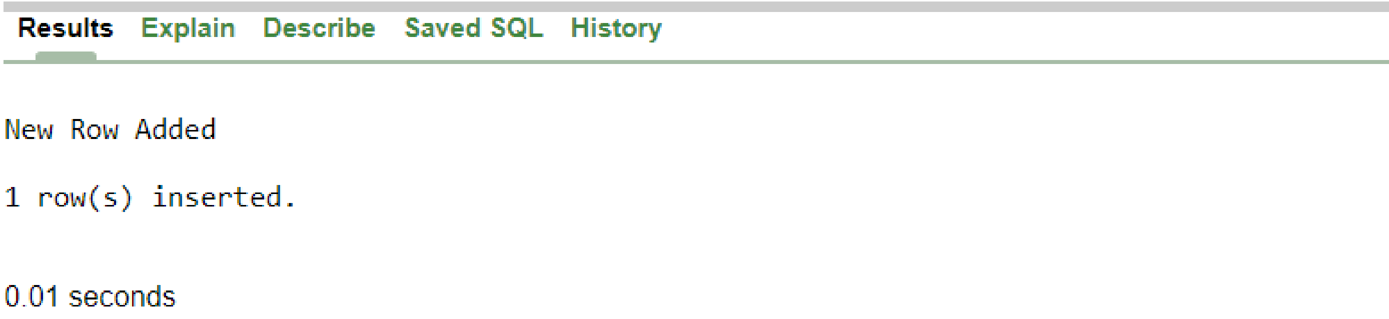


```
500km  500   Non-Ac

Statement processed.

0.00 seconds
```

Trigger

1. Create a trigger in such a way that whenever a new row is inserted into the BookDetail table an output ‘New Row Added’ is generated

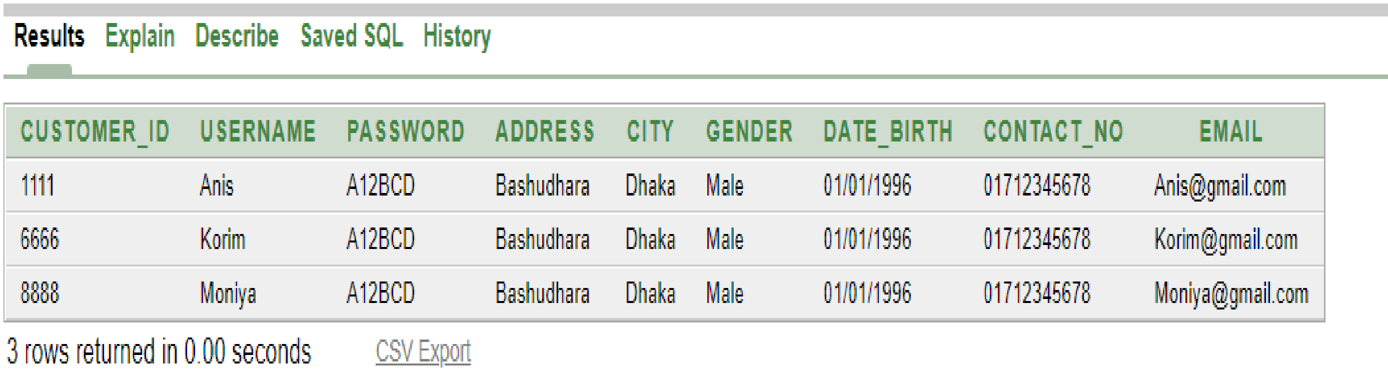
```
CREATE OR REPLACE TRIGGER display_insertion_changes
AFTER INSERT ON BookDetail
FOR EACH ROW

BEGIN
   dbms_output.put_line('New Row Added');
END;
/

INSERT INTO BookDetail VALUES
('C9','009','03/07/2021','27/06/2021','305km','1400','NonAC');
```



```
New Row Added

1 row(s) inserted.


0.01 seconds
```

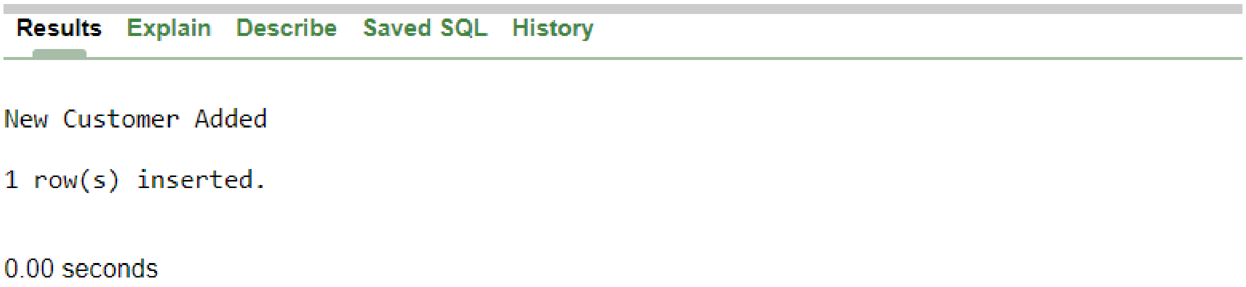
2. Create a trigger in such a way that whenever a new row is inserted into the Customer table an output 'New Customer Added' is generated

```
CREATE OR REPLACE TRIGGER customer_add
AFTER INSERT ON Customer
FOR EACH ROW
BEGIN
  dbms_output.put_line('New Customer Added');
END;
/
Select * from Customer;
INSERT INTO Customer VALUES
('1111','Anis','A12BCD','Bashudhara','Dhaka','Male','01/01/1996','01712345678','Anis@gmail.com')
INSERT INTO TimeTable VALUES
('6666','Korim','A12BCD','Bashudhara','Dhaka','Male','01/01/1996','01712345678','Korim@gmail.com')
INSERT INTO TimeTable VALUES
('8888','Moniya','A12BCD','Bashudhara','Dhaka','Male','01/01/1996','01712345678','Moniya@gmail.com')
```



CUSTOMER_ID	USERNAME	PASSWORD	ADDRESS	CITY	GENDER	DATE_BIRTH	CONTACT_NO	EMAIL
1111	Anis	A12BCD	Bashudhara	Dhaka	Male	01/01/1996	01712345678	Anis@gmail.com
6666	Korim	A12BCD	Bashudhara	Dhaka	Male	01/01/1996	01712345678	Korim@gmail.com
8888	Moniya	A12BCD	Bashudhara	Dhaka	Male	01/01/1996	01712345678	Moniya@gmail.com

3 rows returned in 0.00 seconds



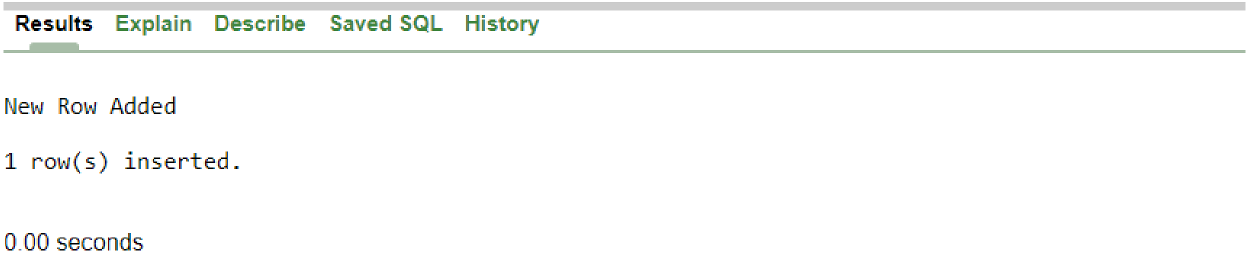
```
New Customer Added

1 row(s) inserted.
```

0.00 seconds

3. Create a trigger in such a way that whenever a new row is inserted into the TimeTable table an output ‘New Row Added’ is generated

```
 CREATE OR REPLACE TRIGGER display_rent_changes
AFTER INSERT ON TimeTable
FOR EACH ROW
WHEN (NEW.Rent > 800)

BEGIN
  dbms_output.put_line('New Row Added');
END;
/
```

```
INSERT INTO TimeTable VALUES
('1001','10','Gabtoli','Bandarban','Chittagong','305km','8.00AM','9.00PM','1200');
```



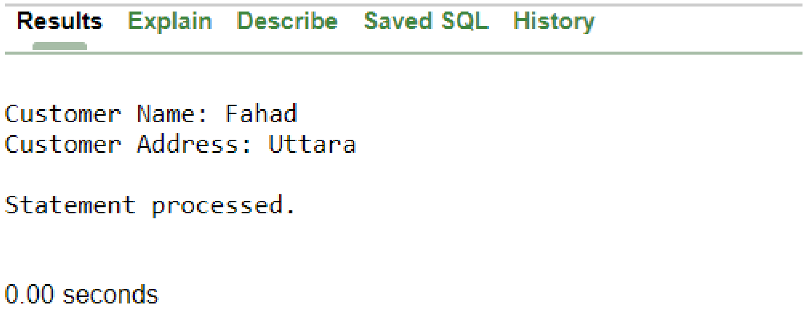
```
New Row Added

1 row(s) inserted.

0.00 seconds
```

<u>Package:</u>

1. **Create a package to display te customer name and address where customer id is '2222'**

---Package Specification

```
CREATE OR REPLACE PACKAGE CUSTOMER AS

  PROCEDURE display_cname(C_ID CUSTOMER.CUSTOMER_ID%type);

  PROCEDURE display_ADD(C_ID CUSTOMER.CUSTOMER_ID%type);

END CUSTOMER;

---package body

CREATE OR REPLACE PACKAGE BODY CUSTOMER AS


  PROCEDURE display_cname(C_ID CUSTOMER.CUSTOMER_ID%type) IS

  c_nam CUSTOMER.CUSTOMER_ID%type;
  BEGIN
    SELECT USERNAME INTO c_nam
    FROM customer
    WHERE CUSTOMER_ID = C_ID;
    dbms_output.put_line('CUSTOMER Name: '|| C_nam);
  END display_cname;

PROCEDURE display_add(C_ID CUSTOMER.CUSTOMER_ID%type) IS
  C_ADD CUSTOMER.ADDRESS%TYPE;
  BEGIN
    SELECT ADDRESS INTO C_ADD
    FROM CUSTOMER
    WHERE CUSTOMER_ID = C_ID;
    dbms_output.put_line('CUSTOMER ADDRESS: '|| C_ADD);
  END display_add;
END CUSTOMER;
/
---Using the package
begin
CUSTOMER.display_cname('2222');
CUSTOMER.display_add('2222');
end
```

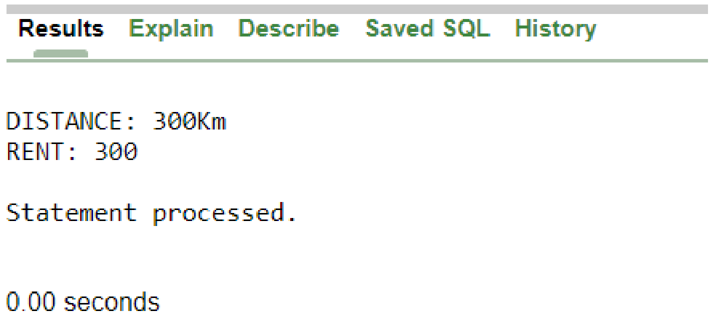


```
Customer Name: Fahad
Customer Address: Uttara

Statement processed.
```

0.00 seconds

2. **Create a pl/sql package that shows the rent and distance of RouteID 3 from rent table.**

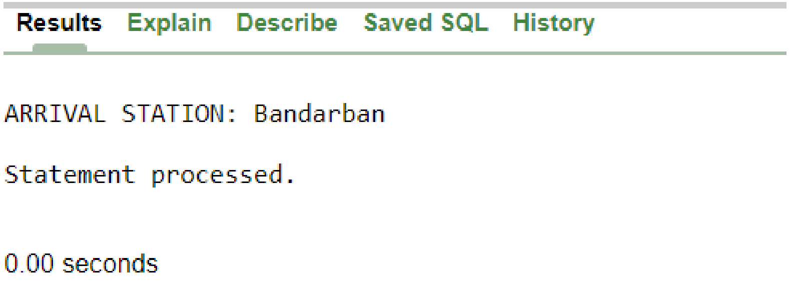
```
---package Specification
CREATE OR REPLACE PACKAGE RENT AS
  PROCEDURE display_ROUTEID(R_ID RENT.ROUTEIDID%type);
  PROCEDURE display_RENT(R_ID RENT.ROUTEIDID%type);
END RENT;
---Package Body
CREATE OR REPLACE PACKAGE BODY RENT AS

  PROCEDURE display_DISTANCE(R_ID RENT.ROUTEIDID%type) IS
  DISTANCE RENT.ROUTEIDID%type;

  BEGIN
    SELECT USERNAME INTO distance
    FROM rent
    WHERE routeID = R_ID;
    dbms_output.put_line('Distance: '|| distance);
  END display_DISTANCE;

PROCEDURE display_rent(R_ID  RENT.ROUTEIDID%type) IS
  rent RENT.RENT%type;
  BEGIN
    SELECT RENT INTO RENT
    FROM RENT
    WHERE ROUTEID = RENT;
    dbms_output.put_line('RENT: '|| RENT);
  END display_RENT;
END RENT;
/

---Using the Package
begin
RENT.display_DISTANCE('3');
RENT.display_RENT('3');
end
```



```
DISTANCE: 300Km
RENT: 300

Statement processed.


0.00 seconds
```

3. **Create a pl/sql program that shows the Arrival Station of Routeid 10 from thr RouteDetails Table.**

```
---Package Specification

CREATE OR REPLACE PACKAGE RouteDetais AS
  PROCEDURE display_ROUTEID(R_ID RENT.ROUTEIDID%type);

END RENT;
---Package Body
CREATE OR REPLACE PACKAGE BODY RouteDetais AS

  PROCEDURE display_ARRIVALSTATION(R_ID RouteDetais.ROUTEID%type) IS
  ARRIVALSTATION RouteDetais.ROUTEID%type;
  BEGIN
    SELECT ROUTEID INTO ARRIVALSTATION
    FROM ROUTEDETAILS
    WHERE routeID = R_ID;
    dbms_output.put_line('ARRIVAL STATION: '|| ARRIVALSTATION);
  END display_ARRIVALSTATION;

END ROUTEDETAILS;
/


---Using the Package
begin
ROUTEDETAILS.display_ARRIVALSTATION('10');
end
```



```
ARRIVAL STATION: Bandarban

Statement processed.

0.00 seconds
```

Conclusion

Nowadays, bus agencies are taking important role in transportation, and to make reservation reliable they need a strong system that they will make reservation easier, faster and safer. This project designed to meet requirements of a bus reservation system. It has been developed in XHTML, PHP, CSS, JAVASCRIPT and database has been built in Oracle. By using this application, the company can provide reservation services and information to their customers without the limitation of office hours or manpower. Not only does it let customers book trips around the clock from any location with an internet connection but it is also designed for use by the company to internally manage their business processes; minimizing human errors and overcoming difficulties and problems that arose in the previous system.