

A  
Project Report  
On  
**Bus Reservation System**

Developed By

**Shah Harsh-Department of IT, DD University**  
**Sinha Shubham-Department of IT, DD University**

**Guided By**  
**Internal Guide:**  
**Prof. Sunil Vithlani**  
**Department of Information Technology**  
**Faculty of Technology**  
**DD University**



**Department of Information Technology**  
**Faculty of Technology, Dharmsinh Desai University**  
**College Road, Nadiad-387001**  
**October-2019**

---



# DHARMSINH DESAI UNIVERSITY

NADIAD-387001, GUJARAT



## CERTIFICATE

This is to certify that the project entitled “Bus Reservation System” is a bonafide report of the work carried out by,

- 1) Mr. Shah Harsh, Student ID No: 17ITUOS113
- 2) Mr. Sinha Shubham, Student ID No: 17ITUOS098

of Department of Information Technology, semester V, under the guidance and supervision for the subject Database Management System. They were involved in Project training during academic year 2019-2020.

Prof. Sunil Vithlani  
(Project Guide)  
Department of Information Technology,  
Faculty of Technology,  
Dharmsinh Desai University, Nadiad  
Date:

Prof. Vipul Dabhi  
Head , Department of Information Technology,  
Faculty of Technology,  
Dharmsinh Desai University, Nadiad  
Date:

# ACKNOWLEDGEMENT

The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely privileged to have got this all along the completion of our project. All that we have done is only due to such supervision and assistance and we would not forget to thank them.

We respect and thank Prof. Sunil Vithlani for providing us an opportunity to do the project work in DBMS and giving us all support and guidance which made us complete the project duly. We are extremely thankful to him for providing such a nice support, guidance by taking keen interest in our project, although he had busy schedule managing the lectures.

We would also like to express our special thanks of gratitude to our Head Prof. Vipul Dabhi who gave us the golden opportunity to do this wonderful project on the topic: Bus Reservation System. We would also like to thank him for including such things in curriculum making it more interesting and useful practically.

Finally, we would like to thank each and every person who were there around us helping more or less in our project and keeping us motivated to work hard and complete the project. We would also thank them for inspiring us and sharing their ideas and views to make this project a success.

Yours sincerely,

Shah Harsh (IT-105)

Sinha Shubham (IT-110)

# TABLE OF CONTENTS

<b>I. Certificate.....</b>	<b>I</b>
<b>II. Acknowledgement.....</b>	<b>II</b>
<b>1. SYSTEM OVERVIEW .....</b>	<b>1</b>
1.1 Current system.....	1
1.2 Objectives of the Proposed System.....	1
1.3 Advantages of the Proposed system (over current) .....	2
<b>2. E-R DIAGRAM.....</b>	<b>3</b>
2.1 Entities.....	3
2.2 Relationships.....	3
2.3 Mapping Constraints.....	3
<b>3. DATA DICTIONARY .....</b>	<b>4</b>
<b>4. SCHEMA DIAGRAM.....</b>	<b>6</b>
<b>5. DATABASE IMPLEMENTION .....</b>	<b>7</b>
5.1 Create Schema.....	7
5.2 Insert Data values .....	9
5.3 Queries (Based on functions, group by, having, joins, sub query etc.) .....	15
5.4 PL/SQL Blocks (Procedures and Functions) .....	17
5.5 Views.....	20
5.6 Triggers .....	23
5.7 Cursors. ....	25
<b>6. FUTURE ENHANCEMENTS OF THE SYSTEM .....</b>	<b>27</b>
<b>7. BIBLIOGRAPHY .....</b>	<b>28</b>



# 1. SYSTEM OVERVIEW

## 1.1 Current System

- The work was done manually.
- Those who wanted to inquire about the Bus Type, Fare, Seat Availability, Facilities etc. had to walk to the office.
- More officials were required to be present at office to manage the customers.
- Officials had to be present at office even during vacations away from the family.
- To change the time table or update the information of existing bus, it took time for implementation.

## 1.2 Objectives of the Proposed System

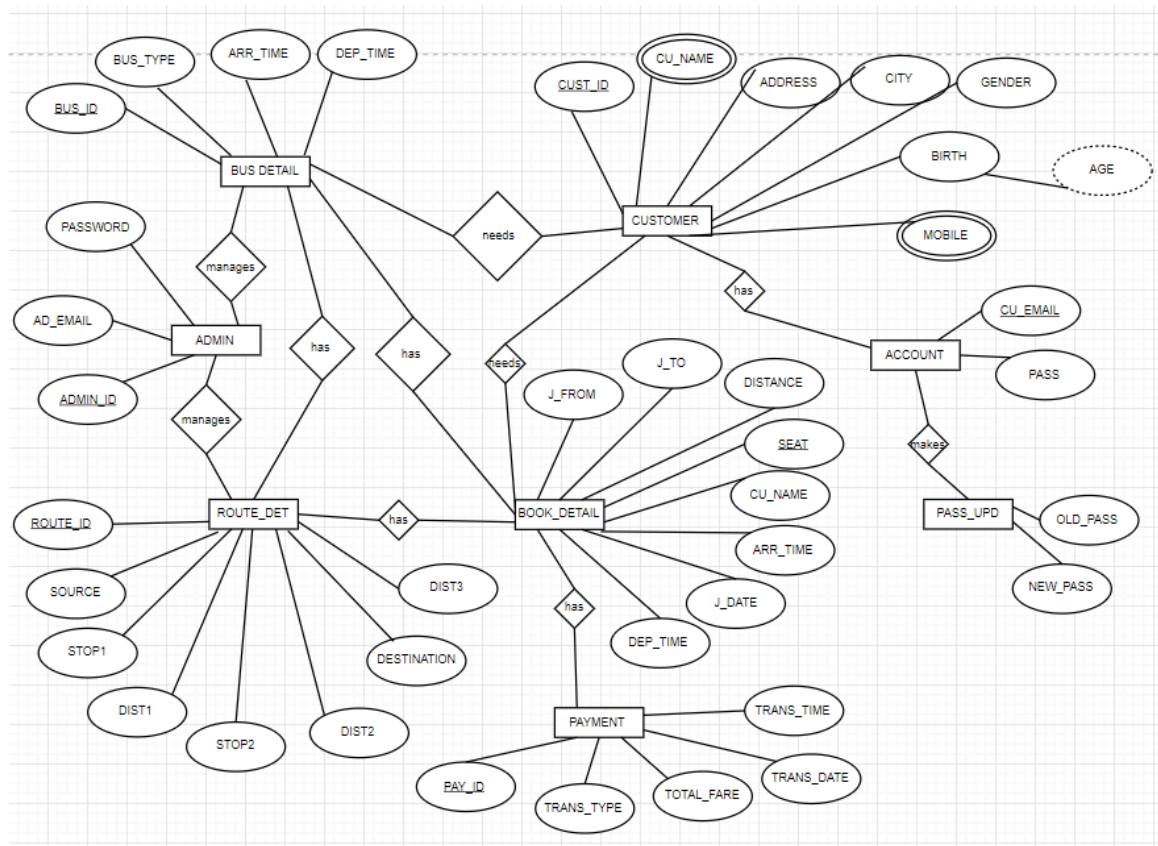
- To reduce the time of standing in queue for booking tickets.
- To take the system digital and make it centralized so that the booking is updated instantly.
- To remove the pen-paper system so that no information is misplaced.
- To reduce the manual labour.
- To update time table online so that it can be implemented sooner.

### 1.3 Advantages of proposed system

- Ticket can be booked hassle-free without waiting in a queue.
- For bus inquiry one doesn't have to visit the office, he can inquire it online.
- One can select seat of his/her choice unlike earlier where they were allotted the seat by the officials.
- Customer data is stored which reduces the time for filling details during booking.
- Admin can change the bus details and route details easily.
- Ticket can be generated on mobile, so no trouble for print-outs.
- Officials can also enjoy their vacation as the database can be managed online from anywhere.



## 2.E-R DIAGRAM



# 1.DATA DICTIONARY

## CUSTOMER:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CUSTOMER	CUST_ID	Varchar2	10	-	-	1	-	-	-
	CU_NAME	Varchar2	15	-	-	-	-	-	-
	ADDRESS	Varchar2	100	-	-	-	✓	-	-
	CITY	Varchar2	15	-	-	-	-	-	-
	GENDER	Varchar2	10	-	-	-	-	-	-
	BIRTH	Date	7	-	-	-	✓	-	-
	MOBILE	Number	-	10	0	-	✓	-	-
1 - 7									

## ACCOUNT:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ACCOUNT	CU_EMAIL	Varchar2	20	-	-	1	-	-	-
	PASS	Varchar2	20	-	-	-	-	-	-
	CUST_ID	Varchar2	10	-	-	-	-	-	-
1 - 3									

## PASS\_UPD:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PASS_UPD	CU_EMAIL	Varchar2	20	-	-	-	-	-	-
	OLD_PASS	Varchar2	20	-	-	-	-	-	-
	NEW_PASS	Varchar2	20	-	-	-	-	-	-
1 - 3									

## ADMIN:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ADMIN	ADMIN_ID	Varchar2	10	-	-	1	-	-	-
	AD_EMAIL	Varchar2	30	-	-	-	-	-	-
	PASSWORD	Varchar2	20	-	-	-	-	-	-
1 - 3									

## ROUTE\_DET:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ROUTE_DET	ROUTE_ID	Varchar2	5	-	-	1	-	-	-
	SOURCE	Varchar2	20	-	-	-	-	-	-
	STOP1	Varchar2	20	-	-	-	✓	-	-
	DIST1	Number	-	6	2	-	✓	-	-
	STOP2	Varchar2	20	-	-	-	✓	-	-
	DIST2	Number	-	6	2	-	✓	-	-
	DESTINATION	Varchar2	20	-	-	-	-	-	-
	DIST3	Number	-	5	2	-	✓	-	-
1 - 8									

## BUS\_DETAIL:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
<u>BUS_DETAIL</u>	<u>BUS_ID</u>	Varchar2	10	-	-	1	-	-	-
	<u>BUS_TYPE</u>	Varchar2	20	-	-	-	-	-	-
	<u>ROUTE_ID</u>	Varchar2	5	-	-	-	-	-	-
	<u>ARR_TIME</u>	Varchar2	10	-	-	-	-	-	-
	<u>DEP_TIME</u>	Varchar2	10	-	-	-	-	-	-
1 - 5									

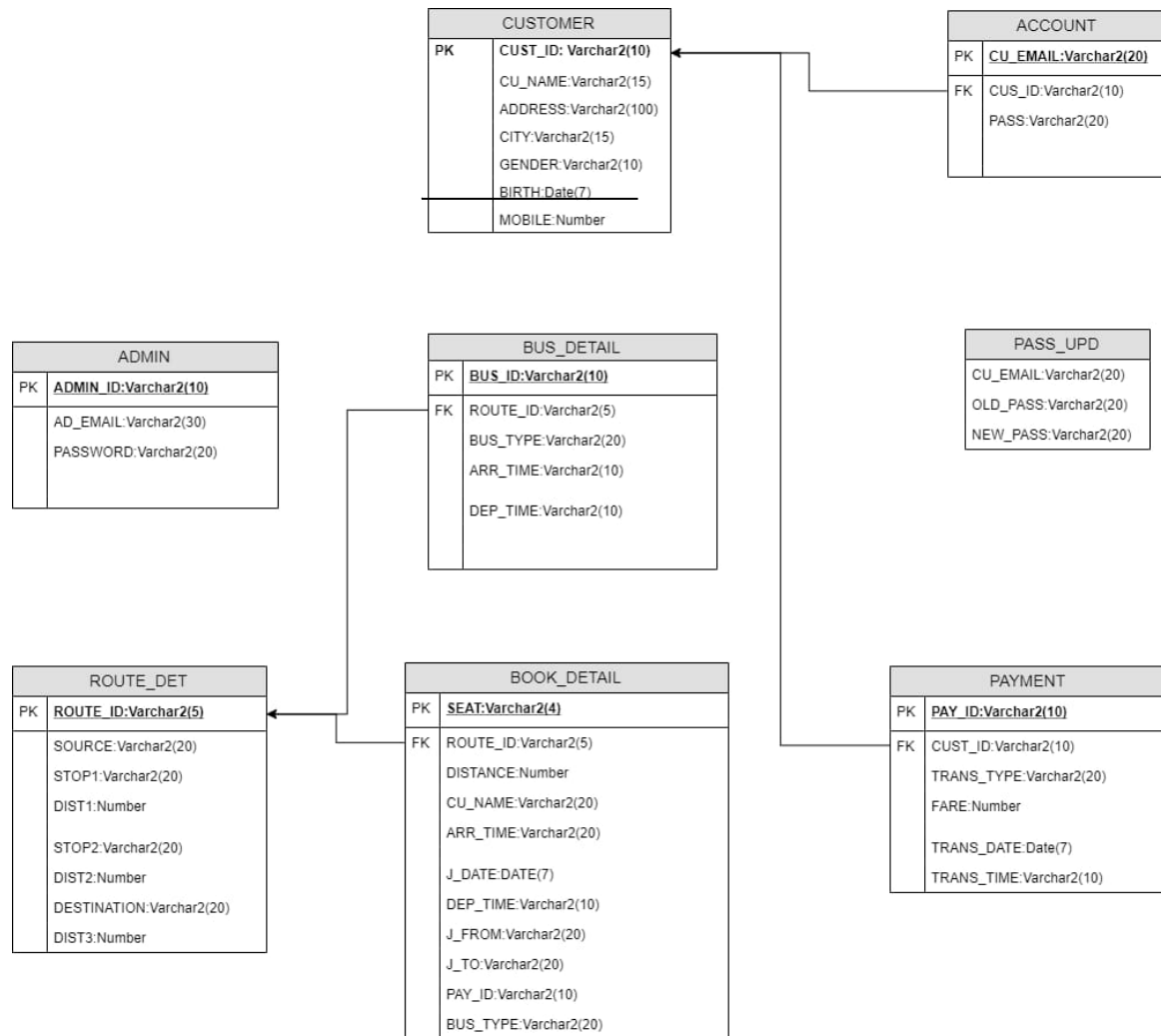
## PAYMENT:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
<u>PAYMENT</u>	<u>PAY_ID</u>	Varchar2	10	-	-	1	-	-	-
	<u>TRANS_TYPE</u>	Varchar2	20	-	-	-	-	-	-
	<u>CUST_ID</u>	Varchar2	10	-	-	-	-	-	-
	<u>FARE</u>	Number	-	6	2	-	✓	-	-
	<u>TRANS_DATE</u>	Date	7	-	-	-	-	-	-
	<u>TRANS_TIME</u>	Varchar2	10	-	-	-	-	-	-
1 - 6									

## BOOK\_DETAIL:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
<u>BOOK_DETAIL</u>	<u>DISTANCE</u>	Number	-	6	2	-	-	-	-
	<u>SEAT</u>	Varchar2	4	-	-	-	-	-	-
	<u>CU_NAME</u>	Varchar2	20	-	-	-	-	-	-
	<u>ARR_TIME</u>	Varchar2	20	-	-	-	-	-	-
	<u>J_DATE</u>	Date	7	-	-	-	-	-	-
	<u>DEP_TIME</u>	Varchar2	10	-	-	-	-	-	-
	<u>J_FROM</u>	Varchar2	20	-	-	-	-	-	-
	<u>J_TO</u>	Varchar2	20	-	-	-	-	-	-
	<u>PAY_ID</u>	Varchar2	10	-	-	-	-	-	-
	<u>BUS_TYPE</u>	Varchar2	20	-	-	-	-	-	-
	<u>ROUTE_ID</u>	Varchar2	5	-	-	-	-	-	-
	<u>BOOK_ID</u>	Varchar2	10	-	-	1	-	-	-
1 - 12									

## 4. SCHEMA DIAGRAM



## 5. DATABASE IMPLEMENTATION

### 5.1 CREATE SCHEMA

Create Customer Table:

```
CREATE TABLE "CUSTOMER"  
(  
    "CUST_ID" VARCHAR2(10),  
    "CU_NAME" VARCHAR2(15),  
    "ADDRESS" VARCHAR2(100),  
    "CITY" VARCHAR2(15),  
    "GENDER" VARCHAR2(10),  
    "BIRTH" DATE,  
    "MOBILE" NUMBER(10,0),  
    PRIMARY KEY ("CUST_ID") ENABLE  
)
```

Create Account Table:

```
CREATE TABLE "ACCOUNT"  
(  
    "CU_EMAIL" VARCHAR2(20),  
    "PASS" VARCHAR2(20),  
    "CU_ID" VARCHAR2(10),  
    PRIMARY KEY ("CU_EMAIL") ENABLE,  
    FOREIGN KEY ("CU_ID")  
        REFERENCES "CUSTOMER" ("CUST_ID") ENABLE  
)
```

Create Admin Table:

```
CREATE TABLE "ADMIN"  
(  
    "ADMIN_ID" VARCHAR2(10),  
    "AD_EMAIL" VARCHAR2(30),  
    "PASSWORD" VARCHAR2(20),  
    PRIMARY KEY ("ADMIN_ID") ENABLE  
)
```

Create Payment Table:

```
CREATE TABLE "PAYMENT"  
  
(  
    "PAY_ID" VARCHAR2(10),  
    "TRANS_TYPE" VARCHAR2(20),  
    "CUST_ID" VARCHAR2(10),  
    "FARE" NUMBER(6,2),  
    "TRANS_DATE" DATE,  
    "TRANS_TIME" VARCHAR2(10),  
    PRIMARY KEY ("PAY_ID") ENABLE,
```

```
FOREIGN KEY ("CUST_ID")  
REFERENCES "CUSTOMER" ("CUST_ID") ENABLE
```

```
)  
Create Route Detail Table:
```

```
CREATE TABLE "ROUTE_DET"  
(  
    "ROUTE_ID" VARCHAR2(5),  
    "SOURCE" VARCHAR2(20),  
    "STOP1" VARCHAR2(20),  
    "DIST1" NUMBER(6,2),  
    "STOP2" VARCHAR2(20),  
    "DIST2" NUMBER(6,2),  
    "DESTINATION" VARCHAR2(20),  
    "DIST3" NUMBER(5,2),  
    PRIMARY KEY ("ROUTE_ID") ENABLE  
)
```

```
Create Book Detail Table:
```

```
CREATE TABLE "BOOK_DETAIL"  
(  
    "DISTANCE" NUMBER(6,2),  
    "SEAT" VARCHAR2(4),  
    "CU_NAME" VARCHAR2(20),  
    "ARR_TIME" VARCHAR2(20),  
    "J_DATE" DATE,  
    "DEP_TIME" VARCHAR2(10),  
    "J_FROM" VARCHAR2(20),  
    "J_TO" VARCHAR2(20),  
    "PAY_ID" VARCHAR2(10),  
    "BUS_TYPE" VARCHAR2(20),  
    "ROUTE_ID" VARCHAR2(5),  
    PRIMARY KEY ("SEAT") ENABLE,  
    FOREIGN KEY ("PAY_ID")  
        REFERENCES "PAYMENT" ("PAY_ID") ENABLE,  
    FOREIGN KEY ("ROUTE_ID")  
        REFERENCES "ROUTE_DET" ("ROUTE_ID") ENABLE  
)
```

```
Create Bus Detail Table:
```

```
CREATE TABLE "BUS_DETAIL"  
(  
    "BUS_ID" VARCHAR2(10),  
    "BUS_TYPE" VARCHAR2(20),  
    "ROUTE_ID" VARCHAR2(5),  
    "ARR_TIME" VARCHAR2(10),  
    "DEP_TIME" VARCHAR2(10),  
    PRIMARY KEY ("BUS_ID") ENABLE,  
    FOREIGN KEY ("ROUTE_ID")  
        REFERENCES "ROUTE_DET" ("ROUTE_ID") ENABLE  
)
```

Create Password Update Table:

```
CREATE TABLE "PASS_UPD"
(
    "CU_EMAIL" VARCHAR2(20),
    "OLD_PASS" VARCHAR2(20),
    "NEW_PASS" VARCHAR2(20),
    FOREIGN KEY ("CU_EMAIL")
        REFERENCES "ACCOUNT" ("CU_EMAIL") ENABLE
)
```

## 5.2 INSERT DATA VALUES

Insert values into Customer Table:

```
INSERT INTO CUSTOMER(CUST_ID,CU_NAME,CITY,GENDER) VALUES
('C101','HARSH','ANKLESHWAR','M');
```

```
INSERT INTO CUSTOMER(CUST_ID,CU_NAME,CITY,GENDER) VALUES
('C102','SHUBHAM','RAJKOT','M');
```

```
INSERT INTO CUSTOMER(CUST_ID,CU_NAME,CITY,GENDER) VALUES
('C103','PUJA','ANAND','F');
```

```
INSERT INTO CUSTOMER(CUST_ID,CU_NAME,CITY,GENDER) VALUES
('C104','CHITRA','ANKLESHWAR','F');
```

```
INSERT INTO CUSTOMER(CUST_ID,CU_NAME,CITY,GENDER) VALUES
('C105','DHRUV','VADODARA','M');
```

```
SELECT * FROM CUSTOMER;
```

CUST_ID	CU_NAME	ADDRESS	CITY	GENDER	BIRTH	MOBILE
c101	Harsh	-	Ankleshwar	M	-	-
c102	Shubham	-	Rajkot	M	-	-
c103	Puja	-	Anand	F	-	-
c104	Chitra	-	Ankleshwar	F	-	-
c105	Dhruv	-	Vadodara	M	-	-

Insert Values in Account Table:

```
INSERT INTO ACCOUNT(CU_EMAIL,PASS,CU_ID) VALUES
('ABC@XYZ.COM','XXXXXX','C101');
```

```
INSERT INTO ACCOUNT(CU_EMAIL,PASS,CU_ID) VALUES
('BCD@XYZ.COM','HXGCS','C102');
```

```
INSERT INTO ACCOUNT(CU_EMAIL,PASS,CU_ID) VALUES  
( 'CDE@XYZ.COM','XTJSFRS','C103');
```

```
INSERT INTO ACCOUNT(CU_EMAIL,PASS,CU_ID) VALUES  
( 'DEF@XYZ.COM','YVXJGFOJG','C104');
```

```
INSERT INTO ACCOUNT(CU_EMAIL,PASS,CU_ID) VALUES  
( 'ABC@XYZ.COM','TEAJBKJF','C105');
```

```
SELECT * FROM ACCOUNT;
```

CU_EMAIL	PASS	CU_ID
abc@xyz.com	xxxxxx	c101
bcd@xyz.com	hxgcs	c102
cde@xyz.com	xtjsfrs	c103
def@xyz.com	yvxjgfojg	c104
efg@xyz.com	teajbkjf	c105

Insert into Password Update Table:

```
INSERT INTO PASS_UPD(CU_EMAIL,OLD_PASS,NEW_PASS) VALUES(  
'ABC@XYZ.COM','XXXXXX','ABCDEF');
```

```
INSERT INTO PASS_UPD(CU_EMAIL,OLD_PASS,NEW_PASS) VALUES(  
'ABC@XYZ.COM','HXGCS','BCDEFG');
```

```
INSERT INTO PASS_UPD(CU_EMAIL,OLD_PASS,NEW_PASS) VALUES(  
'ABC@XYZ.COM','XTJSFRS','CDEFGH');
```

```
INSERT INTO PASS_UPD(CU_EMAIL,OLD_PASS,NEW_PASS) VALUES(  
'ABC@XYZ.COM','YVXJGFOJG','DEFGHI');
```

```
INSERT INTO PASS_UPD(CU_EMAIL,OLD_PASS,NEW_PASS) VALUES(  
'ABC@XYZ.COM','TEAJBKJF','EFGHIJ');
```



```
SELECT * FROM PASS_UPD;
```

CU_EMAIL	OLD_PASS	NEW_PASS
abc@xyz.com	xxxxxx	abcdef
bcd@xyz.com	hxgcs	bcdefg
cde@xyz.com	xtjsfrs	cdefgh
def@xyz.com	yvxjgfojg	defghi
efg@xyz.com	teajbkjf	efghij

Insert values into Menu Table:

```
INSERT INTO ROUTE_DET
(ROUTE_ID,SOURCE,STOP1,DIST1,STOP2,DIST2,DESTINATION,DIST3)
VALUES('R103','AHMEDABAD','VADODARA',100,'BHARUCH',190,'SURAT',280);
```

```
INSERT INTO ROUTE_DET
(ROUTE_ID,SOURCE,STOP1,DIST1,STOP2,DIST2,DESTINATION,DIST3)
VALUES('R104','SURAT','BHARUCH',90,'VADODARA',180,'AHMEDABAD',280);
```

```
INSERT INTO ROUTE_DET
(ROUTE_ID,SOURCE,STOP1,DIST1,STOP2,DIST2,DESTINATION,DIST3)
VALUES('R105','SURAT','BHARUCH',90,'VADODARA',180,'AHMEDABAD',280);
```

```
INSERT INTO ROUTE_DET
(ROUTE_ID,SOURCE,STOP1,DIST1,STOP2,DIST2,DESTINATION,DIST3)
VALUES('R101','RAJKOT','CHOTILA',46,'SURENDRANAGAR',106,'AHMEDABAD',
,215);
```

```
INSERT INTO ROUTE_DET
(ROUTE_ID,SOURCE,STOP1,DIST1,STOP2,DIST2,DESTINATION,DIST3)
VALUES('R102','AHMEDABAD','SURENDRAGAR',121,'CHOTILA',169,'RAJKOT',2
16);
```

```
INSERT INTO ROUTE_DET
(ROUTE_ID,SOURCE,STOP1,DIST1,STOP2,DIST2,DESTINATION,DIST3)
VALUES('R106',
'AHMEDABAD','SURENDRAGAR',121,'CHOTILA',169,'RAJKOT',216);
```

```
INSERT INTO ROUTE_DET
(ROUTE_ID,SOURCE,STOP1,DIST1,STOP2,DIST2,DESTINATION,DIST3)
VALUES('R107',
'RAJKOT','CHOTILA',46,'SURENDRANAGAR',106,'AHMEDABAD',215);
```

```
INSERT INTO ROUTE_DET
(ROUTE_ID,SOURCE,STOP1,DIST1,STOP2,DIST2,DESTINATION,DIST3)
VALUES('R108','AHMEDABAD','VADODARA',100,'BHARUCH',190,'SURAT',280);
```

```
SELECT * FROM ROUTE_DET;
```

ROUTE_ID	SOURCE	STOP1	DIST1	STOP2	DIST2	DESTINATION	DIST3
r103	Ahmedabad	Vadodara	100	Bharuch	190	Surat	280
r104	Surat	Bharuch	90	Vadodara	180	Ahmedabad	280
r105	Surat	Bharuch	90	Vadodara	180	Ahmedabad	280
r101	Rajkot	Chotila	46	Surendranagar	106	Ahmedabad	215
r102	Ahmedabad	Surendranagar	121	Chotila	169	Rajkot	216
r106	Ahmedabad	Surendranagar	121	Chotila	169	Rajkot	216
r107	Rajkot	Chotila	46	Surendranagar	106	Ahmedabad	215
r108	Ahmedabad	Vadodara	100	Bharuch	190	Surat	280

Insert values into Payment Table:

```
INSERT INTO ADMIN (ADMIN_ID,AD_EMAIL,PASSWORD) VALUES
('A101','HSG@XYZ.COM','VCSGSG');
```

```
INSERT INTO ADMIN (ADMIN_ID,AD_EMAIL,PASSWORD) VALUES
('A102','DFV@XYZ.COM','GDUKET');
```

```
INSERT INTO ADMIN (ADMIN_ID,AD_EMAIL,PASSWORD) VALUES
('A103','HJD@XYZ.COM','KJFJWD');
```

```
SELECT * FROM PAYMENT;
```

ADMIN_ID	AD_EMAIL	PASSWORD
a101	hsg@xyz.com	vcsgsg
a102	dfv@xyz.com	gdutek
a103	hjd@xyz.com	kjfgwd

Insert Values into Bus Detail Table:

```
INSERT INTO BUS_DETAIL
(BUS_ID,BUS_TYPE,ROUTE_ID,ARR_TIME,DEP_TIME) VALUES
('B101','SEATER','R101','8:00 AM','8:05 AM');
```

```
INSERT INTO BUS_DETAIL
(BUS_ID,BUS_TYPE,ROUTE_ID,ARR_TIME,DEP_TIME) VALUES
('B102','SIEEPER','R107','8:00 AM','8:05 AM');
```

```
INSERT INTO BUS_DETAIL
(BUS_ID,BUS_TYPE,ROUTE_ID,ARR_TIME,DEP_TIME) VALUES
('B103','SLEEPER','R106','9:00 AM','9:05 AM');
```

```
INSERT INTO BUS_DETAIL
(BUS_ID,BUS_TYPE,ROUTE_ID,ARR_TIME,DEP_TIME) VALUES
('B104','SEATER','R102','9:00 AM','9:05 AM');
```

```
INSERT INTO BUS_DETAIL
(BUS_ID,BUS_TYPE,ROUTE_ID,ARR_TIME,DEP_TIME) VALUES
('B105','SEATER','R103','7:00 AM','7:05 AM');
```

```
INSERT INTO BUS_DETAIL
(BUS_ID,BUS_TYPE,ROUTE_ID,ARR_TIME,DEP_TIME) VALUES
('B106','SLEEER','R108','7:00 AM','7:05 AM');
```

```
INSERT INTO BUS_DETAIL
(BUS_ID,BUS_TYPE,ROUTE_ID,ARR_TIME,DEP_TIME) VALUES
('B107','SLEEPER','R105','8:30 AM','8:35 AM');
```

```
INSERT INTO BUS_DETAIL
(BUS_ID,BUS_TYPE,ROUTE_ID,ARR_TIME,DEP_TIME) VALUES
('B108','SEATER','R104','8:30 AM','8:35 AM');
```

```
SELECT * FROM BUS_DETAIL;
```

BUS_ID	BUS_TYPE	ROUTE_ID	ARR_TIME	DEP_TIME
b101	Seater	r101	8:00 am	8:05 am
b102	Sleeper	r107	8:00 am	8:05 am
b103	Sleeper	r106	9:00 am	9:05 am
b104	Seater	r102	9:00 am	9:05 am
b105	Seater	r103	7:00 am	7:05 am
b106	Sleaper	r108	7:00 am	7:05 am
b107	Sleaper	r105	8:30 am	8:35 am
b108	Seater	r104	8:30 am	8:35am

Insert values into Payment Table:

```
INSERT INTO PAYMENT
(PAY_ID,TRANS_TYPE,CUST_ID,TRANS_DATE,TRANS_TIME)
VALUES('P101','DEBIT CARD','C103','09-SEP-19','14:48:35');
```

INSERT INTO PAYMENT

(PAY\_ID,TRANS\_TYPE,CUST\_ID,TRANS\_DATE,TRANS\_TIME)

VALUES('P102','CREDIT CARD','C105','16-SEP-19','05:29:24');

INSERT INTO PAYMENT

(PAY\_ID,TRANS\_TYPE,CUST\_ID,TRANS\_DATE,TRANS\_TIME)

VALUES('P103','NET BANKING','C102','06-SEP-19','16:24:35');

INSERT INTO PAYMENT

(PAY\_ID,TRANS\_TYPE,CUST\_ID,TRANS\_DATE,TRANS\_TIME)

VALUES('P104','UPI','C104','24-SEP-19','14:53:04');

INSERT INTO PAYMENT

(PAY\_ID,TRANS\_TYPE,CUST\_ID,TRANS\_DATE,TRANS\_TIME)

VALUES('P105','PAYTM','C101','24-SEP-19','13:28:23');

SELECT \* FROM PAYMENT;

PAY_ID	TRANS_TYPE	CUST_ID	FARE	TRANS_DATE	TRANS_TIME
p101	debit card	c103	-	09-SEP-19	18:48:35
p102	credit card	c105	-	16-SEP-19	05:29:24
p103	net banking	c102	-	06-SEP-19	16:24:35
p104	upi	c104	-	24-SEP-19	14:53:04
p105	paytm	c101	-	24-SEP-19	13:28:23

Insert values into Booking Detail Table:

INSERT INTO BOOK\_DETAIL

(DISTANCE,SEAT,CU\_NAME,ARR\_TIME,J\_DATE,DEP\_TIME,J\_FROM,J\_TO,PAY\_ID,BUS\_TYPE,ROUTE\_ID) VALUES(106,6,'SHUBHAM','08:00 AM','08-SEP-19','08:05 AM','RAJKOT','SURENDRANAGAR','P103','SLEEPER','R107');

INSERT INTO BOOK\_DETAIL

(DISTANCE,SEAT,CU\_NAME,ARR\_TIME,J\_DATE,DEP\_TIME,J\_FROM,J\_TO,PAY\_ID,BUS\_TYPE,ROUTE\_ID) VALUES(190,10,'PUJA','07:00 AM','10-SEP-19','07:05 AM','AHMEDABAD','BHARUCH','P101','SLEEPER','R108');

INSERT INTO BOOK\_DETAIL

(DISTANCE,SEAT,CU\_NAME,ARR\_TIME,J\_DATE,DEP\_TIME,J\_FROM,J\_TO,PAY\_ID,BUS\_TYPE,ROUTE\_ID) VALUES(100,15,'DHRUV','08:30 AM','20-SEP-19','08:35 AM','VADODARA','AHMEDABAD','P102','SEATER','R104');

INSERT INTO BOOK\_DETAIL

(DISTANCE,SEAT,CU\_NAME,ARR\_TIME,J\_DATE,DEP\_TIME,J\_FROM,J\_TO,PAY\_ID,BUS\_TYPE,ROUTE\_ID) VALUES(100,15,'DHRUV','08:30 AM','20-SEP-19','08:35 AM','VADODARA','AHMEDABAD','P102','SEATER','R104');

```
_ID,BUS_TYPE,ROUTE_ID) VALUES(190,24,'CHITRA','08:30 AM','29-SEP-19','08:35 AM','BHARUCH','AHMEDABAD','P104','SLEEPER','R105');
```

```
INSERT INTO BOOK_DETAIL
```

```
(DISTANCE,SEAT,CU_NAME,ARR_TIME,J_DATE,DEP_TIME,J_FROM,J_TO,PAY_ID,BUS_TYPE,ROUTE_ID) VALUES(180,4,'HARSH','08:30 AM','30-SEP-19','08:35 AM','SURAT','VADODARA','P105','SLEEPER','R105');
```

```
SELECT *FROM BOOK_DETAIL
```

DISTANCE	SEAT	CU_NAME	ARR_TIME	J_DATE	DEP_TIME	J_FROM	J_TO	PAY_ID	BUS_TYPE	ROUTE_ID	BOOK_ID
106	6	Shubham	08:00 AM	08-SEP-19	08:05 AM	Rajkot	Surendranagar	p103	Sleeper	r107	B101
190	10	Puja	07:00 AM	10-SEP-19	07:05 AM	Ahmedabad	Bharuch	p101	Sleeper	r108	B102
100	15	Dhruv	08:30 AM	20-SEP-19	08:35 AM	Vadodara	Ahmedabad	p102	Seater	r104	B103
190	24	Chitra	08:30 AM	29-SEP-19	08:35 AM	Bharuch	Ahmedabad	p104	Sleeper	r105	B104
180	4	Harsh	08:30 AM	30-SEP-19	08:35 AM	Surat	Vadodara	p105	Sleeper	r105	B105
216	25	Shubham	9:05 am	29-SEP-19	9:05 am	Ahmedabad	Rajkot	p106	Seater	r102	B106

## 5.3 QUERIES

### AGGREGATE FUNCTION

1.COUNT NUMBER OF CUSTOMERS WHICH ARE MALE:

```
SELECT COUNT(CUST_ID) AS MALE_CUSTOMERS FROM CUSTOMER WHERE GENDER='M';
```

MALE_CUSTOMERS
3

2. FIND THE MINIMUM DISTANCE FROM BOOKING DETAIL:

```
SELECT MIN(DISTANCE) FROM BOOK_DETAIL
```

MIN(DISTANCE)
100

### JOINS

1.RETRIVE CUSTOMER DETAIL AND PAYMENT DETAIL:

```
SELECT *FROM CUSTOMER C LEFT JOIN PAYMENT P ON C.CUST_ID=P.CUST_ID
```

CUST_ID	CU_NAME	ADDRESS	CITY	GENDER	BIRTH	MOBILE	PAY_ID	TRANS_TYPE	CUST_ID	FARE	TRANS_DATE	TRANS_TIME
c103	Puja	-	Anand	F	-	-	p101	debit card	c103	-	09-SEP-19	18:48:35
c105	Dhruv	-	Vadodara	M	-	-	p102	credit card	c105	-	16-SEP-19	05:29:24
c102	Shubham	-	Rajkot	M	-	-	p103	net banking	c102	-	06-SEP-19	16:24:35
c104	Chitra	-	Ankleshwar	F	-	-	p104	upi	c104	-	24-SEP-19	14:53:04
c101	Harsh	-	Ankleshwar	M	-	-	p105	paytm	c101	-	24-SEP-19	13:28:23

SUBQUERY

1.RETRIEVE ALL DETAILS OF CUSTOMER WHO HAS MADE PAYMENT ON A PARTICULAR DATE.

```
SELECT *FROM CUSTOMER C WHERE C.CU_NAME=(SELECT CU_NAME
FROM BOOK_DETAIL B WHERE B.PAY_ID=(SELECT PAY_ID FROM PAYMENT
P WHERE P.TRANS_DATE=('16-SEP-19')));
```

CUST_ID	CU_NAME	ADDRESS	CITY	GENDER	BIRTH	MOBILE
c105	Dhruv	-	Vadodara	M	-	-

GROUP BY AND ORDER BY

1.DISPLAY NUMBER OF BUSES ON PARTICULAR ROUTE:

```
SELECT ROUTE_ID,COUNT(SOURCE) FROM ROUTE_DET GROUP BY
ROUTE_ID ORDER BY ROUTE_ID
```

ROUTE_ID	COUNT(SOURCE)
r101	1
r102	1
r103	1
r104	1
r105	1
r106	1
r107	1
r108	1

2.DISPLAY ROUTE DETAIL TABLE ORDER BY A ROUTE ID:

```
SELECT *FROM ROUTE_DET ORDER BY ROUTE_ID ASC
```

ROUTE_ID	SOURCE	STOP1	DIST1	STOP2	DIST2	DESTINATION	DIST3
r101	Rajkot	Chotila	46	Surendranagar	106	Ahmedabad	215
r102	Ahmedabad	Surendranagar	121	Chotila	169	Rajkot	216
r103	Ahmedabad	Vadodara	100	Bharuch	190	Surat	280
r104	Surat	Bharuch	90	Vadodara	180	Ahmedabad	280
r105	Surat	Bharuch	90	Vadodara	180	Ahmedabad	280
r106	Ahmedabad	Surendranagar	121	Chotila	169	Rajkot	216
r107	Rajkot	Chotila	46	Surendranagar	106	Ahmedabad	215
r108	Ahmedabad	Vadodara	100	Bharuch	190	Surat	280

### 3.COUNT THE NO. OF CUSTOMERS WHO BOOKED TICKET:

SELECT SEAT,COUNT(CU\_NAME) FROM BOOK\_DETAIL GROUP BY SEAT;

SEAT	COUNT(CU_NAME)
10	1
15	1
24	1
4	1
6	1

### EXTRACT

#### 1.EXTRACT THE DATE FROM THE PAYMENT TABLE.D

SELECT PAY\_ID,EXTRACT(DAY FROM TRANS\_DATE) FROM PAYMENT;

PAY_ID	EXTRACT(DAYFROMTRANS_DATE)
p101	9
p102	16
p103	6
p104	24
p105	24

## 5.4 PL/SQL BLOCKS

### 5.4.1 PROCEDURES

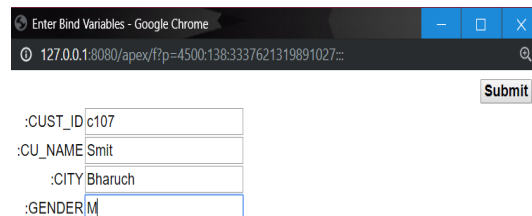
## 1. CREATE A PROCEDURE TO REGISTER A NEW CUSTOMER.

```
CREATE OR REPLACE PROCEDURE INSERT_INT0(C IN
CUSTOMER.CUST_ID%TYPE,CN IN CUSTOMER.CU_NAME%TYPE,CI
IN CUSTOMER.CITY%TYPE,G IN CUSTOMER.GENDER%TYPE)
IS
BEGIN
```

```
INSERT INTO CUSTOMER
("CUST_ID","CU_NAME","CITY","GENDER")VALUES(C,CN,CI,G);
END;
```

```
DECLARE
A CUSTOMER.CUST_ID%TYPE;
B CUSTOMER.CU_NAME%TYPE;
C CUSTOMER.CITY%TYPE;
D CUSTOMER.GENDER%TYPE;
```

```
BEGIN
A:=CUST_ID;
B:=CU_NAME;
C:=CITY;
D:=GENDER;
INSERT_INT0(A,B,C,D);
END;
```



Procedure created.

## 2. CREATE A PROCEDURE TO BOOK A TICKET.

```
CREATE OR REPLACE PROCEDURE BOOK_TICKET(PAY_ID_VAR
PAYMENT.PAY_ID%TYPE, TRANS_TYPE_VAR
PAYMENT.TRANS_TYPE%TYPE, DIST_VAR
BOOK_DETAIL.DISTANCE%TYPE, CUST_ID_VAR
PAYMENT.CUST_ID%TYPE,
CU_NAME_VAR BOOK_DETAIL.CU_NAME%TYPE,SEAT_VAR
BOOK_DETAIL.SEAT%TYPE, J_FROM_VAR
BOOK_DETAIL.J_FROM%TYPE, J_TO_VAR
BOOK_DETAIL.J_TO%TYPE, J_DATE_VAR
BOOK_DETAIL.J_DATE%TYPE, ARR_TIME_VAR
```



```
BOOK_DETAIL.ARR_TIME%TYPE, DEP_TIME_VAR
BOOK_DETAIL.DEP_TIME%TYPE, ROUTE_ID_VAR
BOOK_DETAIL.ROUTE_ID%TYPE, BUS_TYPE_VAR
BOOK_DETAIL.BUS_TYPE%TYPE)

IS

BEGIN

INSERT INTO PAYMENT("PAY_ID", "TRANS_TYPE",
"CUST_ID", "FARE", "TRANS_DATE", "TRANS_TIME")
VALUES(PAY_ID_VAR, TRANS_TYPE_VAR, CUST_ID_VAR, 100, '25-
MAR-2019', '33');

INSERT INTO BOOK_DETAIL("DISTANCE", "SEAT", "CU_NAME",
"ARR_TIME", "J_DATE", "DEP_TIME", "J_FROM", "J_TO", "PAY_ID",
"BUS_TYPE", "ROUTE_ID") VALUES

(DIST_VAR, SEAT_VAR, CU_NAME_VAR, ARR_TIME_VAR,
J_DATE_VAR, DEP_TIME_VAR, J_FROM_VAR, J_TO_VAR,
PAY_ID_VAR, BUS_TYPE_VAR, ROUTE_ID_VAR);

END;

BEGIN

BOOK_TICKET('p107', 'credit card', 100, 'c101', 'Harsh', 35, 'Ahmedabad',
'Vadodara', '13-OCT-19', '07:00 AM', '07:05 AM', 'r108', 'Sleeper');

END;
```

Procedure created.

## 5.4.2 FUNCTIONS

1. CREATE A FUNCTION WHICH DISPLAYS BOOKING DETAILS OF CUSTOMER.

```
create or replace function getdisc(M varchar2) RETURN NUMBER
IS
disc number;
A number;
BEGIN
Select FARE into disc from PAYMENT where PAY_ID=M;
A:=0.05*disc;
```

```
disc:=disc-A;  
RETURN disc;  
END;
```

Function created.

## 2. CREATE A FUNCTION TO DISPLAY FARE.

```
CREATE OR REPLACE FUNCTION CAL_FARE(D IN  
BOOK_DETAIL.DISTANCE%TYPE,B IN BOOK_DETAIL.BUS_TYPE%TYPE)  
RETURN NUMBER IS  
FARE NUMBER(4) :=0;  
BEGIN  
IF B='Sleeper' THEN  
FARE:= 3*D;  
ELSIF B='Seater' THEN  
FARE:= 2*D;  
ELSE  
DBMS_OUTPUT.PUT_LINE('ENTER VALID BUS TYPE: (Sleeper/Seater)');  
END IF;  
RETURN FARE;  
END;
```

Function created.

## 5.5 VIEWS

### 1. CREATE A VIEW TO DISPLAY CITY OF THE CUSTOMERS.

```
CREATE VIEW CUST_CITY AS (SELECT CUST_ID,(BIRTH || CITY)"CITY" FROM  
CUSTOMER);  
SELECT * FROM CUST_CITY;
```

View created.

CUST_ID	CITY
c101	Ankleshwar
c102	Rajkot
c103	Anand
c104	Ankleshwar
c105	Vadodara

2.CREATE VIEW TO DISPLAY BOOKING DETAIL WITH THEIR TRANSACTION DATE,TRANSACTION TYPE, JOURNEY DATE AND PAYMENT ID.

```
CREATE VIEW JOURNEY AS(SELECT
P.TRANS_DATE,P.TRANS_TYPE,B.J_DATE,P.PAY_ID FROM PAYMENT
P,BOOK_DETAIL B WHERE P.PAY_ID=B.PAY_ID);
SELECT * FROM JOURNEY;
```

View created.

TRANS_DATE	TRANS_TYPE	J_DATE	PAY_ID
06-SEP-19	net banking	08-SEP-19	p103
09-SEP-19	debit card	10-SEP-19	p101
16-SEP-19	credit card	20-SEP-19	p102
24-SEP-19	upi	29-SEP-19	p104
24-SEP-19	paytm	30-SEP-19	p105

## EXCEPTION(INBUILT)

1. DISPLAY INFORMATION OF CUSTOMER IF EXIST ELSE HANDLE THE NO\_DATA\_FOUND EXCEPTION.

```
DECLARE
ID CUSTOMER.CUST_ID%TYPE;
CN CUSTOMER.CU_NAME%TYPE;
CI CUSTOMER.CITY%TYPE;
G CUSTOMER.GENDER%TYPE;

BEGIN
ID:=:ENTER_ID;
```

```
SELECT CU_NAME,CITY,GENDER INTO CN,CI,G FROM CUSTOMER WHERE
CUST_ID=ID;
```

```
DBMS_OUTPUT.PUT_LINE('CUSTOMER DETAILS :' || ID);
```

```
DBMS_OUTPUT.PUT_LINE('CUSTOMER_NAME :' || CN);
```

```
DBMS_OUTPUT.PUT_LINE('CITY: ' || CI);
```

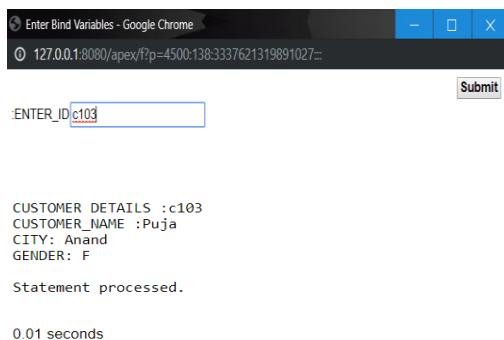
```
DBMS_OUTPUT.PUT_LINE('GENDER: ' || G);
```

```
EXCEPTION
```

```
WHEN NO_DATA_FOUND THEN
```

```
DBMS_OUTPUT.PUT_LINE('ENTER VALID CUSTOMER ID');
```

```
END
```



Enter Bind Variables - Google Chrome  
127.0.0.1:8080/apex/f?p=4500:138:3337621319891027::

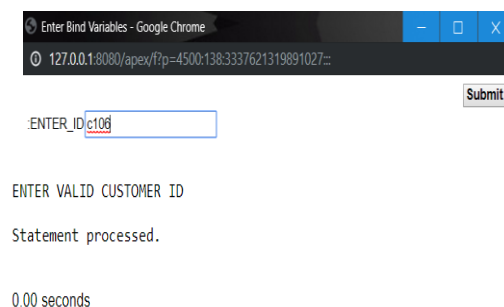
Submit

ENTER\_ID c103

CUSTOMER DETAILS :c103  
CUSTOMER\_NAME :Puja  
CITY: Anand  
GENDER: F

Statement processed.

0.01 seconds



Enter Bind Variables - Google Chrome  
127.0.0.1:8080/apex/f?p=4500:138:3337621319891027::

Submit

ENTER\_ID c108

ENTER VALID CUSTOMER ID

Statement processed.

0.00 seconds

## EXCEPTION(USER DEFINED)

2.CHECK WHETHER A PARTICULAR OFFER IS VALID ELSE GENERATE EXCEPTION.

Declare

```
S route_det.source%type;
```

```
R route_det.route_id%type;
```

```

D ROUTE_DET.DESTINATION%TYPE;
T BUS_DETAIL.BUS_TYPE%TYPE;
B bus_detail.bus_id%type;
R VALID EXCEPTION;
BEGIN
S:=:S;
D:=:D;
T:=:T;
select route_id into R from
route_det
where source=S AND DESTINATION=D ;
select bus_id into B from bus_detail where route_id=R and bus_type=T;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('ROUTE NOT FOUND!');
END;
```

Enter Bind Variables - Google Chrome

127.0.0.1:8080/apex/f?p=4500:138:3993425780693393::

Submit

:S Ahmedabad

:D Vadodara

:T sleeper

ROUTE NOT FOUND!

Statement processed.

## 5.6 TRIGGERS

1. CREATE TRIGGER TO DISPLAY BOOKING DETAILS AFTER THE NEW BOOKING IS DONE.

```

create or replace trigger new_booked after insert on Book_detail for each row
declare
begin
DBMS_OUTPUT.PUT_LINE('Trigger Fired');
DBMS_OUTPUT.PUT_LINE('Pay: '||:NEW.Pay_id);
DBMS_OUTPUT.PUT_LINE('Seat: '||:NEW.Seat);
DBMS_OUTPUT.PUT_LINE('Name: '|| :NEW.Cu_Name);
DBMS_OUTPUT.PUT_LINE('Route: '|| :NEW.Route_ID);
```

```

DBMS_OUTPUT.PUT_LINE('Distance: ' || :NEW.Distance);
DBMS_OUTPUT.PUT_LINE('Arrival: ' || :NEW.Arr_Time);
DBMS_OUTPUT.PUT_LINE('Journey: ' || :NEW.J_Date);
DBMS_OUTPUT.PUT_LINE('Departure: ' || :NEW.Dep_Time);
DBMS_OUTPUT.PUT_LINE('From: ' || :NEW.J_From);
DBMS_OUTPUT.PUT_LINE('To: ' || :NEW.J_to);
DBMS_OUTPUT.PUT_LINE('Type: ' || :NEW.Bus_Type);
END;

```

```

Trigger Fired
Pay: p106
Seat: 25
Name: Shubham
Route: r102
Distance: 216
Arrival: 9:05 am
Journey: 29-SEP-19
Departure: 9:05 am
From: Ahmedabad
To: Rajkot
Type: Seater

```

Trigger created.

```
1 row(s) inserted.
```

## 2. CREATE TRIGGER INSERT, UPDATE OR DELETE A ROUTE.

CREATE OR REPLACE TRIGGER ROUTE\_IUD AFTER INSERT OR UPDATE OR DELETE ON ROUTE\_DET FOR EACH ROW  
DECLARE

BEGIN

IF UPDATING THEN

```

DBMS_OUTPUT.PUT_LINE(:OLD.ROUTE_ID||' ROUTE UPDATED FROM : ' ||
:OLD.SOURCE || '->' || :OLD.STOP1 || '(' || :OLD.DIST1 || ')' || '->' || :OLD.STOP2 || '(' ||
:OLD.DIST2 || ')' || '->' || :OLD.DESTINATION|| '(' || :OLD.DIST3 || ')' || ' TO: ' ||
:NEW.SOURCE || '->' || :NEW.STOP1 || '(' || :NEW.DIST1 || ')' || '->' || :NEW.STOP2 || '(' ||
|| :NEW.DIST2 || ')' || '->' || :NEW.DESTINATION|| '(' || :NEW.DIST3 || ')');

```

ELSIF INSERTING THEN

```

DBMS_OUTPUT.PUT_LINE(:NEW.ROUTE_ID||' ROUTE ADDED ' ||
:NEW.SOURCE || '->' || :NEW.STOP1 || '(' || :NEW.DIST1 || ')' || '->' || :NEW.STOP2 || '(' ||
|| :NEW.DIST2 || ')' || '->' || :NEW.DESTINATION|| '(' || :NEW.DIST3 || ')');

```

ELSIF DELETING THEN

```

DBMS_OUTPUT.PUT_LINE(:OLD.ROUTE_ID||' ROUTE REMOVED' ||
:OLD.SOURCE || '->' || :OLD.STOP1 || '(' || :OLD.DIST1 || ')' || '->' || :OLD.STOP2 || '(' ||
:OLD.DIST2 || ')' || '->' || :OLD.DESTINATION|| '(' || :OLD.DIST3 || ')');

```

END IF;

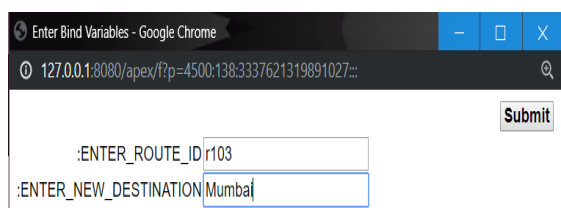
END;

```
TRIGGER FIRED  
CID:C4  
CNAME:ROHIT SHAH  
CONTACT:9846572477  
EMAILID:rohitsshah@gmail.com  
STREET:B11 SWETA COLONY  
CITY:JUNAGADH
```

## 5.7 CURSORS

### 1.IMPLICIT CURSOR FOR UPDATING THE DESTINATION OF EXISTING ROUTE.

```
DECLARE  
ID VARCHAR2(20);  
DEST VARCHAR2(20);  
BEGIN  
ID:=:ENTER_ROUTE_ID;  
DEST:=:ENTER_NEW_DESTINATION;  
UPDATE ROUTE_DET SET DESTINATION=DEST WHERE ROUTE_ID=ID;  
IF SQL%FOUND THEN  
DBMS_OUTPUT.PUT_LINE('ROUTE HAS CHANGED DESTINATION');  
END IF;  
IF SQL%NOTFOUND THEN  
DBMS_OUTPUT.PUT_LINE('ROUTE NOT FOUND');  
END IF;  
END;
```



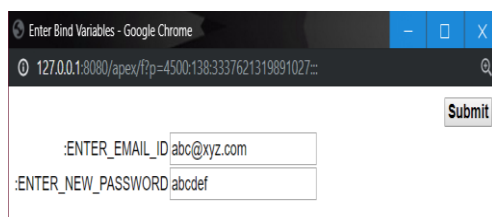
ROUTE HAS CHANGED DESTINATION

Statement processed.

### 2.EXPLICIT CURSOR TO UPDATE PASSWORD OF EXISTING CUSTOMER.

```
DECLARE  
EMAIL VARCHAR2(20);  
PASSWORD VARCHAR2(20);
```

```
BEGIN
EMAIL:=:ENTER_EMAIL_ID;
PASSWORD:=:ENTER_NEW_PASSWORD;
UPDATE PASS_UPD SET NEW_PASS=PASSWORD WHERE CU_EMAIL=EMAIL;
IF SQL%FOUND THEN
DBMS_OUTPUT.PUT_LINE('PASSWORD IS UPDATED TO '||PASSWORD||' FOR
CU_EMAILD ' || EMAIL);
END IF;
IF SQL%NOTFOUND THEN
DBMS_OUTPUT.PUT_LINE('USER NOT FOUND');
END IF;
END;
```



PASSWORD IS UPDATED TO abcdef FOR CU\_EMAILD abc@xyz.com

Statement processed.

0.03 seconds



## **6. FUTURE ENHANCEMENTS OF THE SYSTEM**

- The System can be expanded by adding more number of restaurants in state/country.
- The tracking facility can be added to the system through which the customer can track the order through GPS.
- The details of delivery can be more enhanced.
- The automatic generation of CID,RID and MID can be implemented.
- The policy of compensation of late delivery or damaged food items can be added.
- The number of offers can be increased.

The above mentioned points are the enhancements which can be done to increase the applicability and usage of this project.

## 7. BIBLIOGRAPHY

Books:

- Database System Concepts  
By:- Henry F.Korth and A.Silberschatz
- PL/SQL Programming  
By:- Ivan Bayross
- The Complete Reference  
By:- George Koch

References: [www.W3Schools.com/sql](http://www.W3Schools.com/sql)