**CHAROTAR UNIVERSITY OF SCIENCE & TECHNOLOGY**

**DEVANG PATEL INSTITUE OF ADVANCE TECHNOLOGY AND RESEARCH**

**NAME: Parth N Patel**

**ID: 19DCS098**

**SUBJECT: DBMS (Mini Project)**

**SUBJECT CODE: CE246**

**SEM: 4**

**PROJECT TITLE: BUS RESERVATION SYSTEM**

**DATA DICTIONARY:**

**TABLE NAME:** BUS\_INFO

|  |  |  |  |
| --- | --- | --- | --- |
| **COLUMN NAME** | **DATA TYPE** | **LENGTH** | **DESCRIPTION** |
| BUS\_ID | Number | 3 | For uniquely identify all buses |
| BUS\_NAME | Varchar2 | 10 | Name of bus |
| DRIVER\_ID | Number | 3 | Id of driver driving the bus |
| TOTAL\_SEATS | Number | 2 | Seats of bus |
|  |  |  |  |
| RESERVED\_SEATS | Number | 2 | Reserved seats in bus |

**TABLE NAME:** DRIVER

|  |  |  |  |
| --- | --- | --- | --- |
| **COLUMN NAME** | **DATA TYPE** | **LENGTH** | **DESCRPITION** |
| DRIVER\_ID | Number | 2 | ID of the driver |
| DRIVER\_NAME | Varchar2 | 20 | Name of driver |

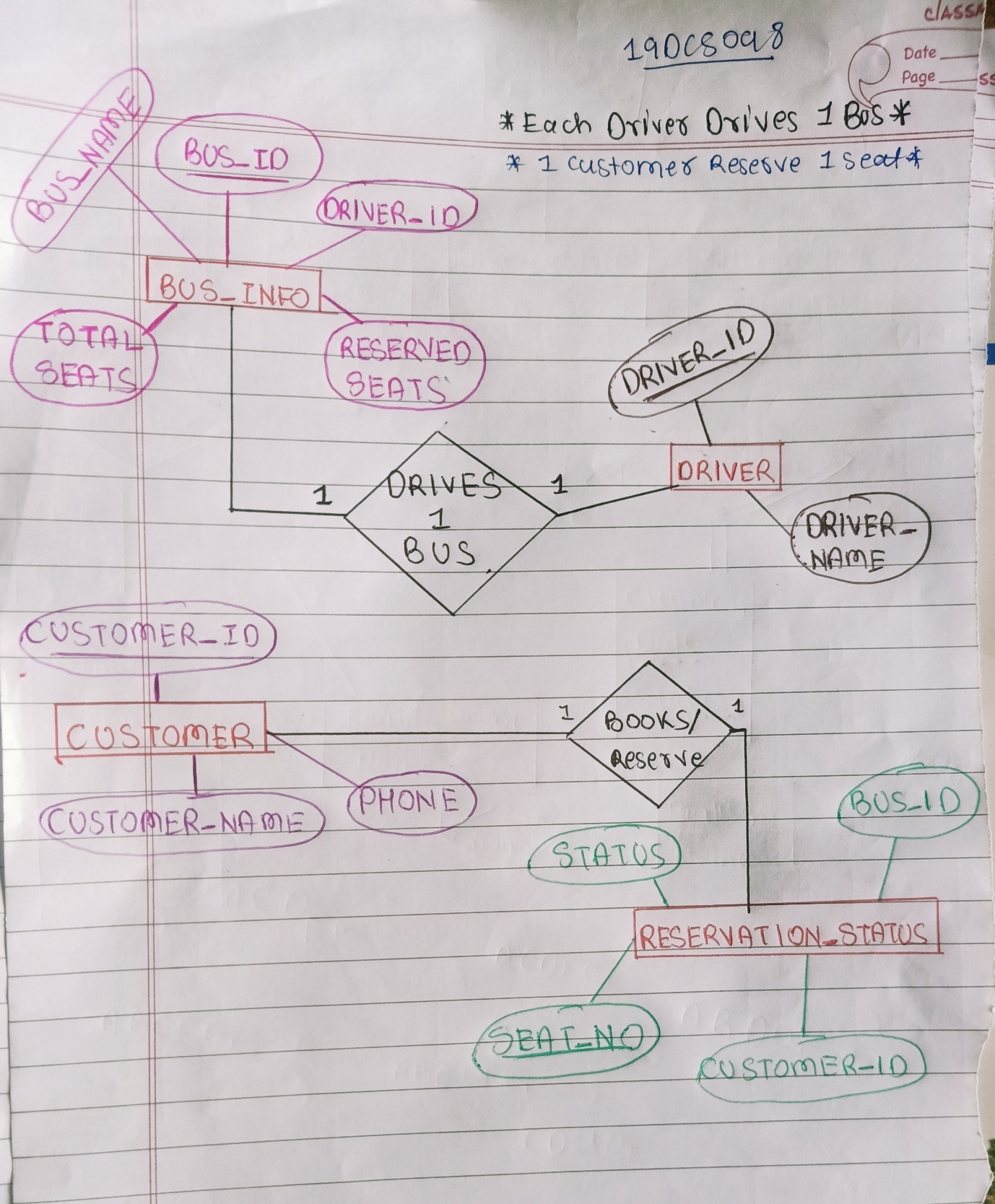
**TABLE NAME:** CUSTOMER

|  |  |  |  |
| --- | --- | --- | --- |
| **COLUMN NAME** | **DATA TYPE** | **LENGTH** | **DESCRIPTION** |
| CUSTOMER\_ID | Number | 3 | ID of customer |
| CUSTOMER\_NAME | Varchar2 | 20 | Name of customer |
| PHONE | Number | 10 | Phone number of customers |

**TABLE NAME**: RESERVATION\_STATUS

|  |  |  |  |
| --- | --- | --- | --- |
| **COLUMN NAME** | **DATA TYPE** | **LENGTH** | **DESCRIPTION** |
| SEAT\_NO | Number | 4 | Seat number in bus |
| CUSTOMER\_ID | Number | 3 | ID of customer travelling |
| BUS\_ID | Number | 3 | ID of BUS having seat number |
| STATUS | Char | 3 | We let know whether seat reserved or not |

**E-R DIAGRAM:**



**PROGRAM CODE:**

/\* CODE FOR CREATING TABLES \*/

CREATE TABLE bus\_info ( bus\_id NUMBER(3), bus\_name VARCHAR2(10), driver\_id NUMBER(3), total\_seats NUMBER(2), reserved\_seats NUMBER(2));

CREATE TABLE driver (driver\_id NUMBER(2), driver\_name VARCHAR2(20));

CREATE TABLE customer (customer\_id NUMBER(3), customer\_name VARCHAR2(20), phone NUMBER(10));

CREATE TABLE reservation\_status (seat\_no NUMBER(4), customer\_id NUMBER(3), bus\_id NUMBER(3), status CHAR(3));

/\* CODE FOR ASSIGNNING THE PRIMARY KEY TO TABLES \*/

ALTER TABLE bus\_info ADD CONSTRAINT pk\_bus\_id PRIMARY KEY(bus\_id);

ALTER TABLE customer ADD CONSTRAINT pk\_customer\_id PRIMARY KEY(customer\_id);

ALTER TABLE driver ADD CONSTRAINT pk\_driver\_id PRIMARY KEY(driver\_id);

ALTER TABLE reservation\_status ADD CONSTRAINT pk\_seat\_no PRIMARY KEY(seat\_no);

/\*ASSIGNNING THE FOREIGN KEY \*/

ALTER TABLE reservation\_status

ADD CONSTRAINTS fk\_customer\_id FOREIGN KEY(customer\_id) REFERENCES customer;

ALTER TABLE reservation\_status

ADD CONSTRAINTS fk\_bus\_id FOREIGN KEY(bus\_id) REFERENCES bus\_info;

ALTER TABLE bus\_info

ADD CONSTRAINTS fk\_driver\_id FOREIGN KEY(driver\_id) REFERENCES driver;

/\* CODE FOR INSERTING THE DATA IN DRIVER TABLE \*/

CREATE OR REPLACE PROCEDURE UPDATE\_DRIVER(id NUMBER, name VARCHAR2)

IS

BEGIN

IF(id>0) THEN

INSERT INTO driver(driver\_id,driver\_name) VALUES (id,name);

DBMS\_OUTPUT.PUT\_LINE('DRIVER INFO UPDATED');

ELSE

DBMS\_OUTPUT.PUT\_LINE('ERROR ENCOUNTERED');

END IF;

END

/\* CODE FOR INSERTING THE BUS DATA \*/

CREATE OR REPLACE PROCEDURE BUS(id NUMBER, busname VARCHAR2,seat NUMBER)

IS BEGIN

IF(id>0) THEN

INSERT INTO bus\_info(bus\_id,bus\_name,total\_seats,reserved\_seats) VALUES (id,busname,seat,0);

DBMS\_OUTPUT.PUT\_LINE('BUS INFO UPDATED');

ELSE

DBMS\_OUTPUT.PUT\_LINE('ERROR ENCOUNTERED');

END IF;

END;

/\* CODE FOR INSERTING THE CUSTOMER DATA \*/

CREATE OR REPLACE PROCEDURE CUSTOMER\_INFO(id NUMBER,name VARCHAR2,ph NUMBER)

IS BEGIN

IF(id>0) THEN

INSERT INTO customer(customer\_id,customer\_name,phone) VALUES (id,name,ph);

DBMS\_OUTPUT.PUT\_LINE('CUSTOMER INFORMATION UPDATED');

ELSE

DBMS\_OUTPUT.PUT\_LINE('ERROR ENCOUNTERED');

END IF;

END;

/\*CODE FOR INSERTING CUSTOMER DETAILS IN RESERVATION TABLE\*/

CREATE OR REPLACE PROCEDURE RESERVATION\_INFO(id NUMBER,b\_id NUMBER)

IS BEGIN

IF(id>0) THEN

INSERT INTO reservation\_status(seat\_no,bus\_id,status) VALUES (id,b\_id,'NO');

DBMS\_OUTPUT.PUT\_LINE('SEAT INFORMATION UPDATED');

ELSE

DBMS\_OUTPUT.PUT\_LINE('ERROR ENCOUNTERED');

END IF;

END;

/\* CODE FOR ASSIGNNING THE DRIVER TO HIS/HER BUS \*/

CREATE OR REPLACE PROCEDURE ASSIGN\_DRIVER(id NUMBER,d\_id NUMBER)

IS BEGIN

IF(id>0) THEN

UPDATE bus\_info SET driver\_id=d\_id WHERE bus\_id=id;

DBMS\_OUTPUT.PUT\_LINE('DRIVER ASSIGNED TO BUS');

ELSE

DBMS\_OUTPUT.PUT\_LINE('ERROR ENCOUNTERED');

END IF;

END;

/\*CODE FOR CONFIRMING THE RESERVATION \*/

CREATE OR REPLACE PROCEDURE CONFIRM\_RESERVATION(id NUMBER,cust\_id NUMBER,b\_id NUMBER)

IS BEGIN

IF(id>0) THEN

UPDATE reservation\_status SET customer\_id=cust\_id,bus\_id=b\_id,status='YES' WHERE seat\_no=id;

UPDATE bus\_info SET reserved\_seats=reserved\_seats+1 WHERE bus\_id=b\_id;

DBMS\_OUTPUT.PUT\_LINE('RESERVATION CONFIRMED');

ELSE

DBMS\_OUTPUT.PUT\_LINE('ERROR ENCOUNTERED');

END IF;

END;

/\*CODE FOR CANCELLING THE RESERVATION \*/CREATE OR REPLACE PROCEDURE DELETE\_RESERVATION(id NUMBER,b\_id NUMBER)

IS BEGIN

IF(id>0) THEN

UPDATE reservation\_status SET customer\_id=NULL,bus\_id=NULL,status='NO' WHERE seat\_no=id;

UPDATE bus\_info SET reserved\_seats=reserved\_seats-1 WHERE bus\_id=b\_id;

DBMS\_OUTPUT.PUT\_LINE('RESERVATION CANCELLED');

ELSE

DBMS\_OUTPUT.PUT\_LINE('ERROR ENCOUNTERED');

END IF;

END;

/\* CREATING VIEW FOR CUSTOMER TO SEE THE RESERVATION INFORMATION\*/

CREATE VIEW customer\_view AS

SELECT c.customer\_id,c.customer\_name,c.phone,r.seat\_no,r.bus\_id,r.status

FROM customer c INNER JOIN reservation\_status r ON c.customer\_id=r.customer\_id;

/\* CREATING VIEW FOR DRIVER AND BUS ASSIGNED TO THEM \*/

CREATE VIEW driver\_view AS

SELECT d.driver\_id,d.driver\_name,b.bus\_id,b.bus\_name

FROM driver D INNER JOIN bus\_info b ON d.driver\_id=b.driver\_id;

/\* CREATING VIEW FOR OWNER TO SEE WHICH DRIVER IS ASSIGNNED TO WHICH BUS \*/

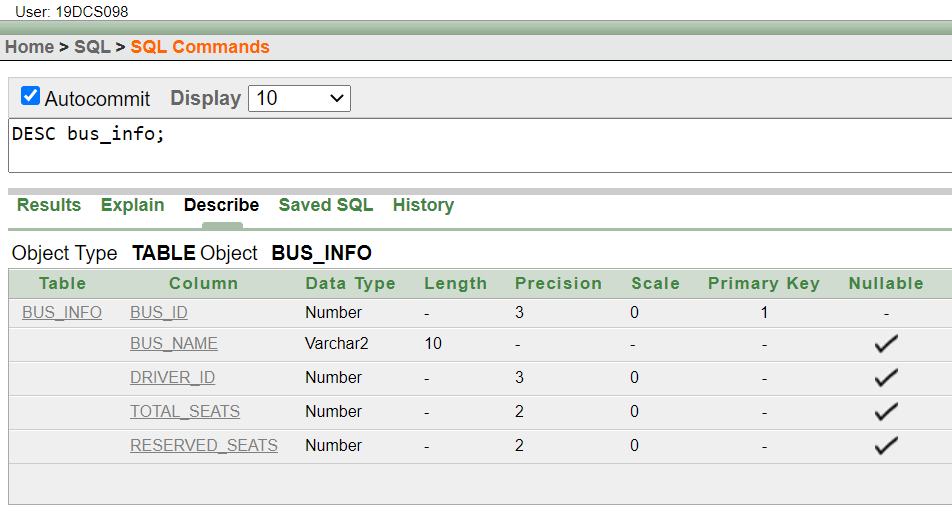
CREATE VIEW TRANSPORTER\_VIEW\_BUS AS

SELECT d.driver\_id,d.driver\_name,b.bus\_id,b.bus\_name,b.total\_seats,b.reserved\_seats

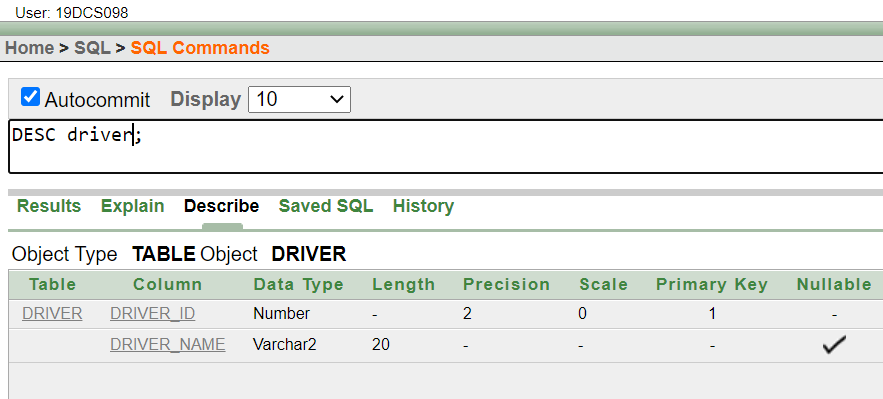
FROM driver d INNER JOIN bus\_info b ON d.driver\_id=b.driver\_id;

**IMPLEMENTATION SCREEN SHOTS:**

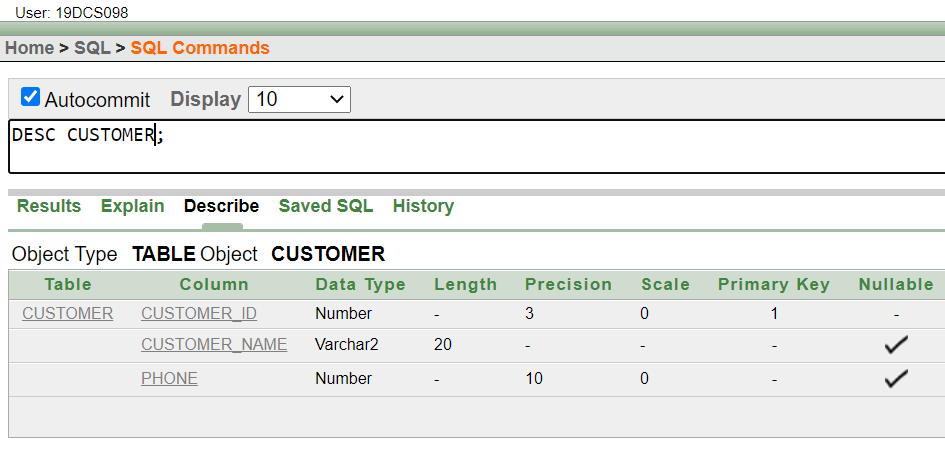
* **DETAILS OF BUS\_INFO**:



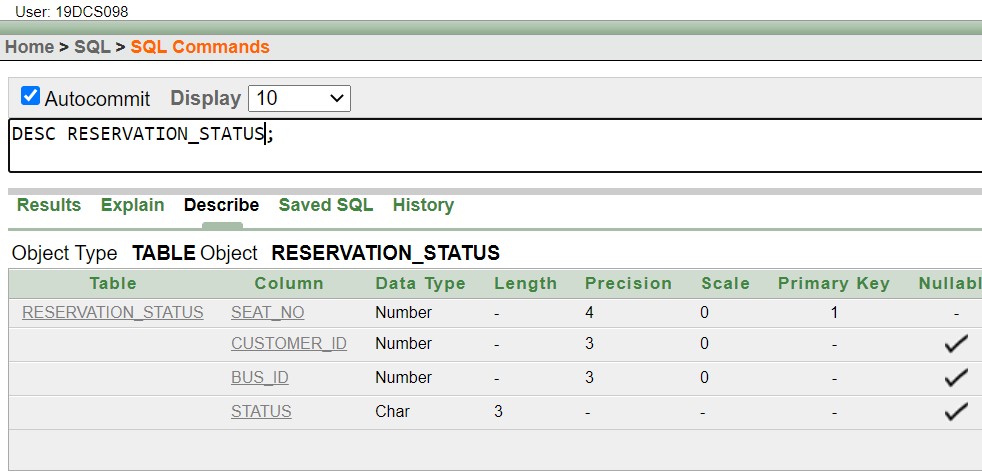
* **DETAILS OF DRIVER TABLE**:



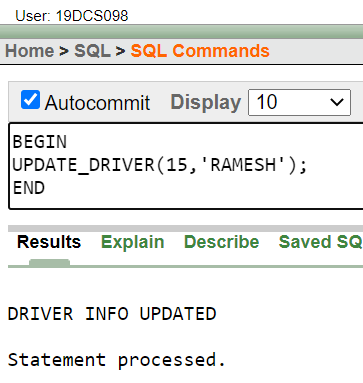
* **DETAILS OF CUSTOMER TABLE :**



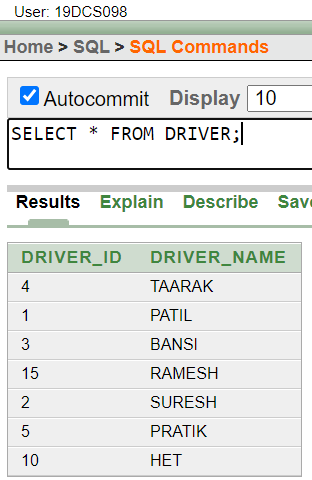
* **DETAILS OF RESERVATION\_STATUS TABLE**:



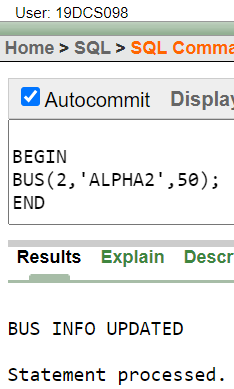
* **INSERTING DRIVER’S INFO USING PL/SQL PROCEDURE**:



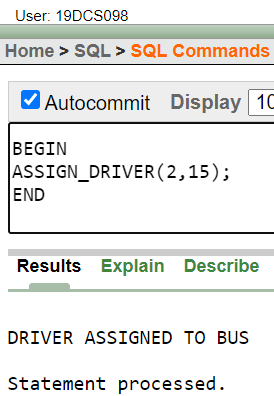
* **TABLE OF DRIVER AFTER UPPDATION:**



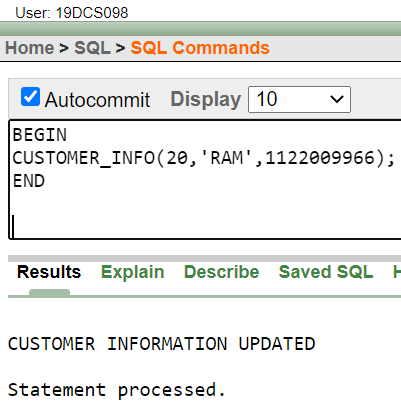
* **UPDATING BUS INFO USING PL/SQL PROCEDURE:**



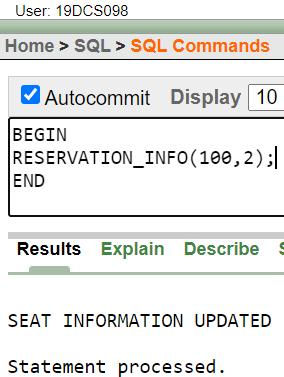
* **ASSIGN DRIVER TO HIS/HER BUS:**



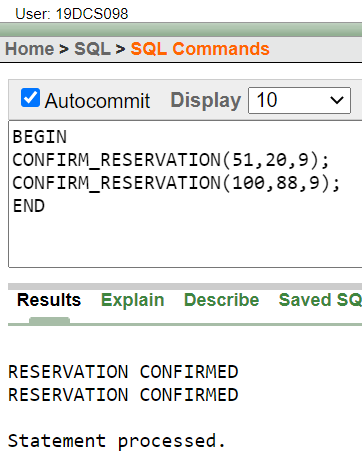
* **INSERTING DATA OF NEW CUSTOMER**



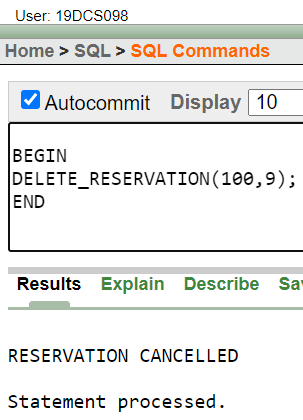
* **ASSIGNNING SEATS IN RESERVATION QUOTA**



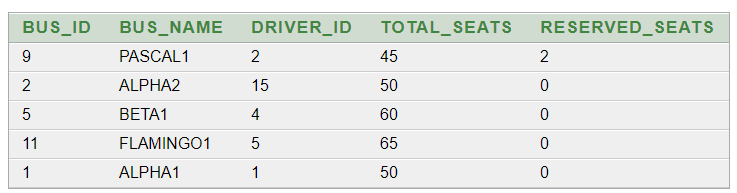
* **CONFIRMING THE RESERVATION**

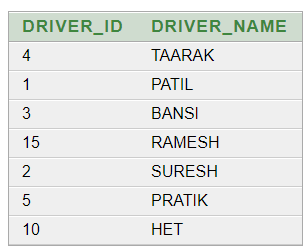


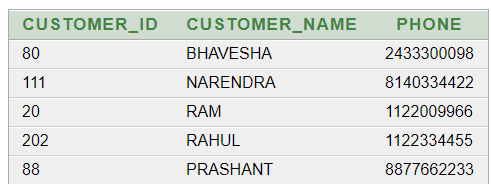
* **CANCEL THE RESERVATION**

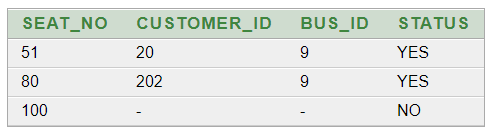


* **GLIMPSES OF ALL TABLES AFTER PERFORMING THE ABOVE OPERATIONS:**

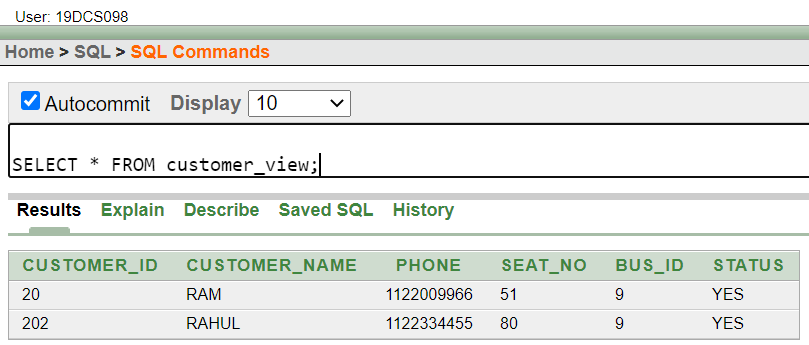




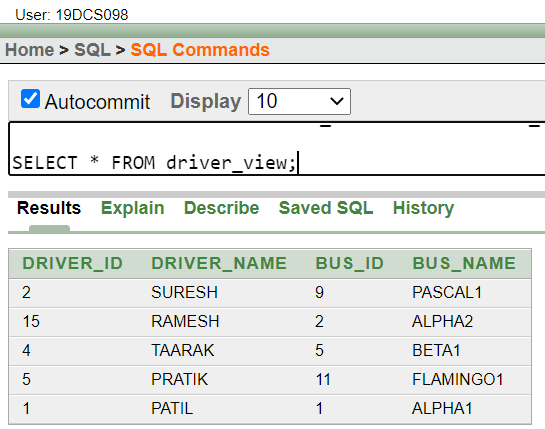




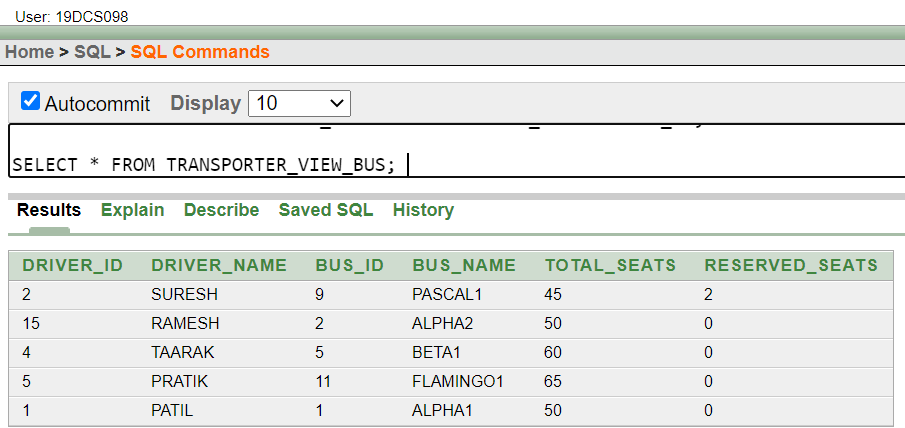
* **GLIMPSE OF VIEWS OF CUSTOMERS HAVING CONFIRMED SEATS**



* **GLIMPSE OF VIEW OF ASSIGNED DRIVER**



* **GLIMPSE OF OWNER**



**CONCLUSION:**

By doing this mini project we polished our skills in DBMS,PL/SQL,SQL and other concepts