**ASSIGNMENT NO.4**

**Name – Indranil Bain**

**Enrollment NO. – 2020CSB039**

**Branch – Computer Science and Technology**

**Group - GX**

**Subject – DBMS Laboratory**

**Q1)**

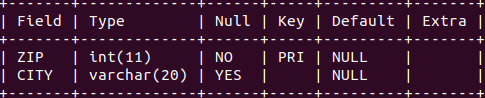
**<A. Creation of tables:>**

**ZIPCODES Table**

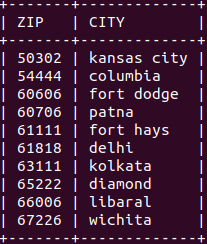
create table ZIPCODES(

-> ZIP int primary key,

-> CITY varchar(20));



**After insertion of values:-**



**EMPLOYEES Table**

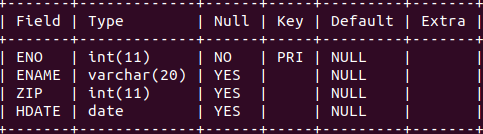
create table EMPLOYEES(

-> ENO int primary key,

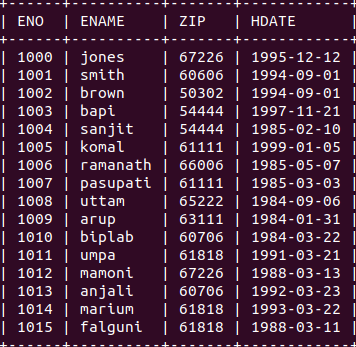
-> ENAME varchar(20),

-> ZIP int references ZIPCODES(ZIP),

-> HDATE date);

****

**After insertion of values:-**



**PARTS Table**

create table PARTS(

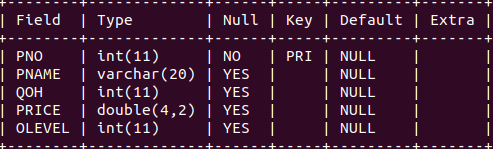
-> PNO int primary key,

-> PNAME varchar(20),

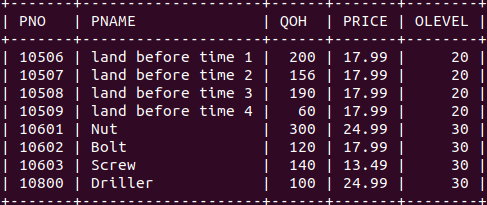
-> QOH int,

-> PRICE double(4,2),

-> OLEVEL int);



**insertion of values:-**

****

**CUSTOMERS Table**

create table CUSTOMERS(

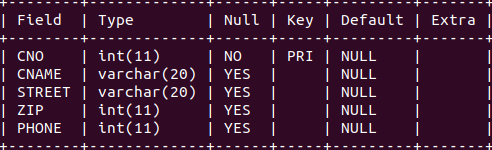
-> CNO int primary key,

-> CNAME varchar(20),

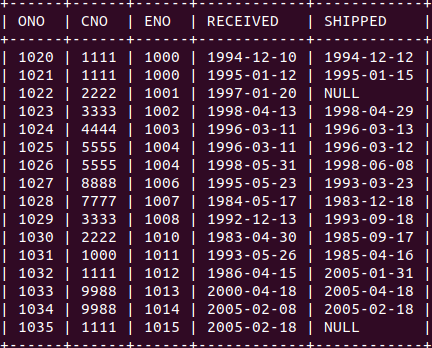
-> STREET varchar(20),

-> ZIP int references ZIPCODES(ZIP),

-> PHONE date);



**After insertion of values:-**

****

**ORDERS Table**

create table ORDERS(

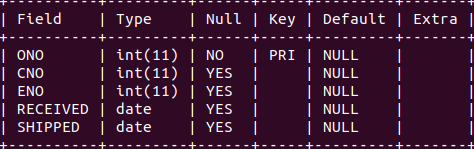
-> ONO int primary key,

-> CNO int references CUSTOMERS(CNO),

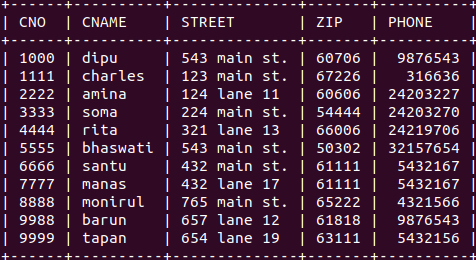
-> ENO int references EMPLOYEES(ENO),

-> RECEIVED date,

-> SHIPPED date);

****

**After insertion of values:-**



**ODETAILS Table**

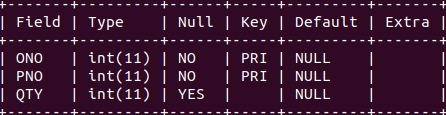
create table ODETAILS(

-> ONO int references ORDERS(ONO),

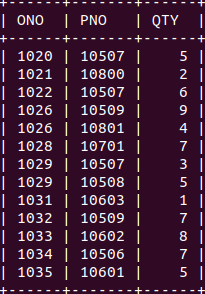
-> PNO int references PARTS(PNO),

-> QTY int,

-> primary key(ONO,PNO));

****

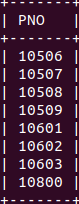
**After insertion of values:-**

****

**<B. Queries >**

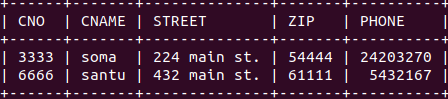
**1) Get PNO values for PARTS for which ORDERS have been placed.**

SELECT DISTINCT(PNO) FROM ODETAILS;

****

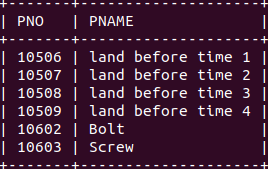
**2) Get all the details of CUSTOMERS whose name has the beginning letter ‘s’.**

SELECT \* FROM CUSTOMERS WHERE CNAME LIKE 'S%';

****

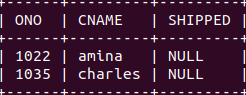
**3) Get PNO and PNAME values of PARTS that are priced less than 19.99.**

SELECT PNO, PNAME FROM PARTS WHERE PRICE < 19.99;



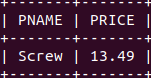
**4) Get the ONO, CNAME and SHIPPED values for CUSTOMERS whose orders have not yet been shipped.**

SELECT ONO, CNAME, SHIPPED FROM ORDERS,CUSTOMERS WHERE ORDERS.CNO=CUSTOMERS.CNO AND SHIPPED IS NULL;



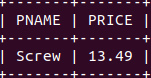
**5) Get PNAME and PRICE value from PARTS with the lowest PRICE.**

SELECT PNAME, PRICE FROM PARTS WHERE PRICE = (SELECT MIN(PRICE) FROM PARTS);



**6) Get the PNAME and PRICE values of PARTS that cost less than the least expensive ‘land before time’ part.**

SELECT PNAME, PRICE FROM PARTS WHERE PRICE < (SELECT MIN(PRICE) FROM PARTS WHERE PNAME LIKE 'LAND BEFORE TIME%');



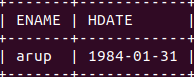
**7) Get the ENO values of EMPLOYEES from ‘Fort Dodge’.**

SELECT ENO FROM EMPLOYEES, ZIPCODES WHERE ZIPCODES.ZIP=EMPLOYEES.ZIP AND CITY='fort dodge';



**8) Get the ENAME and HDATE of the EMPLOYEES who was hired on the earliest date.**

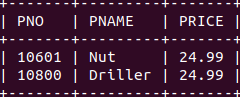
SELECT ENAME, HDATE FROM EMPLOYEES WHERE HDATE = (SELECT MIN(HDATE) FROM EMPLOYEES);



**9) Retrieve the PNO, PNAME and PRICE of PARTS with price greater than**

**20.00 in an ascending order of PNO.**

SELECT PNO, PNAME, PRICE FROM PARTS WHERE PRICE > 20.00 ORDER BY PNO;

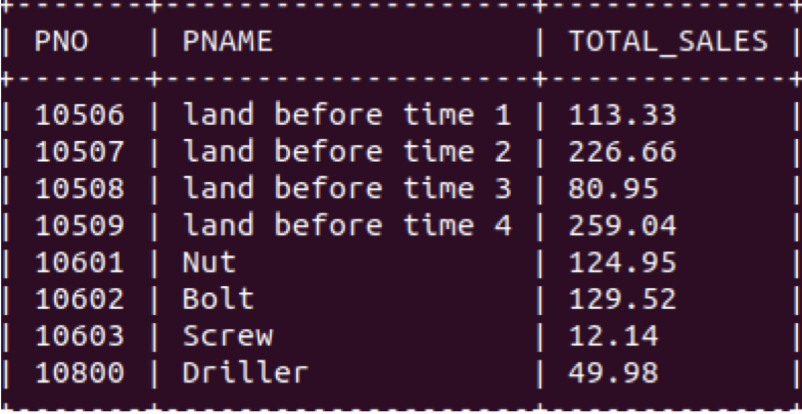


**10) For each PARTS get PNO and PNAME values along with total sales in**

**details.**

SELECT PNO, PNAME, FORMAT(SUM(PRICE\*QTY), 2) AS TOTAL\_SALES FROM ODETAILS NATURAL JOIN PARTS GROUP BY PNO;

**AFTER UPDATION**

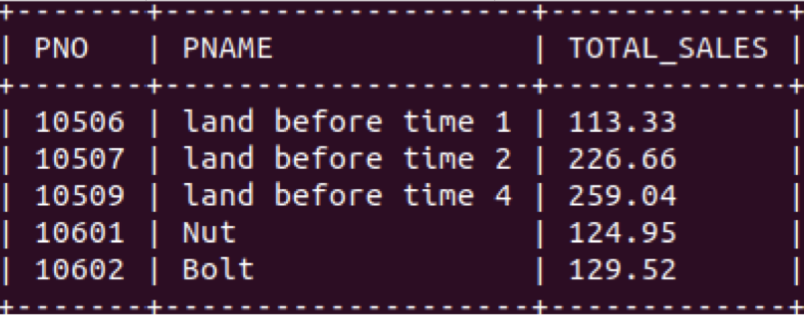
****

**11) For each PARTS get PNO and PNAME values along with total sales in**

**rupees but only when the total sales exceeds 100.53.**

SELECT PNO, PNAME, FORMAT(SUM(price\*QTY), 2) AS TOTAL\_SALES FROM ODETAILS NATURAL JOIN PARTS GROUP BY PNO HAVING TOTAL\_SALES > 100.53;

**AFTER UPDATION**



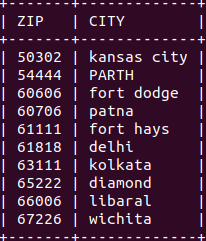
**12) Change the name of the CITY ‘columbia’ to ‘parth’.**

UPDATE ZIPCODES

-> SET CITY='PARTH'

-> WHERE CITY='COLUMBIA';

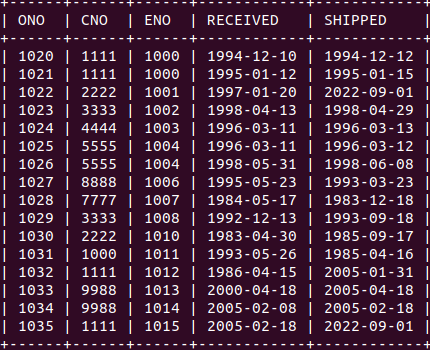
**AFTER UPDATION**



**13) Update all the null valued SHIPPED dates to the current date.**

UPDATE ORDERS SET SHIPPED=(SELECT SYSDATE() FROM DUAL) WHERE SHIPPED IS NULL;

**AFTER UPDATION**

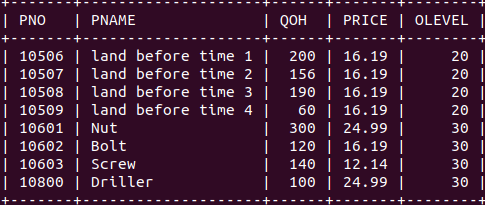


**14) Decrease the PRICE of all PARTS that cost less than 24.00 by 10%.**

UPDATE PARTS

-> SET PRICE=PRICE-0.1\*PRICE

-> WHERE PRICE < 24.00;

**AFTER UPDATION **

**15) Set the ‘QOH’ value of those PARTS whose current ‘QOH’ value is less**

**than 100 to the maximum ‘QOH’ value present in the table.**

UPDATE PARTS

-> SET QOH = (SELECT MAX(QOH)

-> FROM (SELECT QOH FROM PARTS) AS T) -> WHERE QOH < 100;

**AFTER UPDATION**

