ID: 171-15-8596

Section: E

Lab: 4

1.

Customer(customerNo, name, email)

Orders(orderNo, orderDate, handingCost, customerNo)

Includes(unitPrice, quantity, orderNo, productNo)

Products(productNo, productName, listPrice)

2.

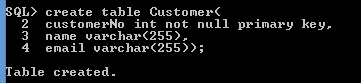
SQL> create table Customer(

2 customerNo int not null primary key,

3 name varchar(255),

4 email varchar(255));

Table created.



SQL> create table Orders(

2 orderNo int not null primary key,

3 orderDate date,

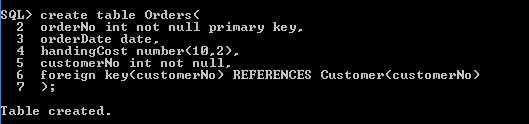
4 handingCost number(10,2),

5 customerNo int not null,

6 foreign key(customerNo) REFERENCES Customer(customerNo)

7 );

Table created.



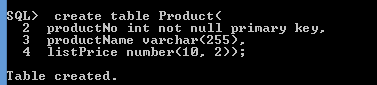
SQL> create table Product(

2 productNo int not null primary key,

3 productName varchar(255),

4 listPrice number(10, 2));

Table created.



SQL> create table Includes(

2 unitPrice number(10,2),

3 unitQuantity number(10,2),

4 orderNo int not null,

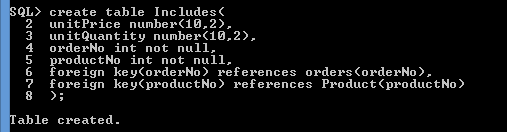
5 productNo int not null,

6 foreign key(orderNo) references orders(orderNo),

7 foreign key(productNo) references Product(productNo)

8 );

Table created.



3.

SQL> insert all

2 into Customer(customerNo, name, email) values(1, 'Hridoy', 'Hridoy@mail.com

')

3 into Customer(customerNo, name, email) values(2, 'Hasan', 'Hasan@mail.com')

4 into Customer(customerNo, name, email) values(3, 'Monisha', 'Monisha@mail.c

om')

5 into Customer(customerNo, name, email) values(4, 'Ahamed', 'Ahamed@mail.com

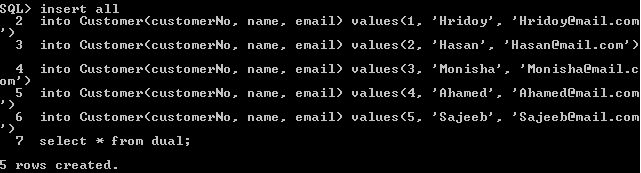
')

6 into Customer(customerNo, name, email) values(5, 'Sajeeb', 'Sajeeb@mail.com

')

7 select \* from dual;

5 rows created.



SQL> insert all

2 into Product(productNo, productName, listPrice)values(1, 'pen', 10)

3 into Product(productNo, productName, listPrice)values(2, 'pencil', 9)

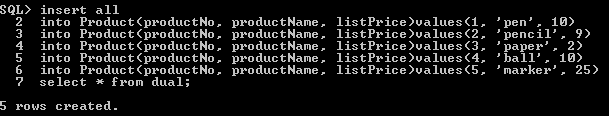
4 into Product(productNo, productName, listPrice)values(3, 'paper', 2)

5 into Product(productNo, productName, listPrice)values(4, 'ball', 10)

6 into Product(productNo, productName, listPrice)values(5, 'marker', 25)

7 select \* from dual;

5 rows created.



SQL> insert all

2 into Orders(orderNo, orderDate, handingCost, customerNo) values(1, TO\_DATE

('2018/12/16 01:00:00', 'yyyy/mm/dd hh:mi:ss'), 100, 1)

3 into Orders(orderNo, orderDate, handingCost, customerNo) values(2, TO\_DATE

('2018/12/15 02:00:00', 'yyyy/mm/dd hh:mi:ss'), 120, 2)

4 into Orders(orderNo, orderDate, handingCost, customerNo) values(3, TO\_DATE

('2018/12/17 03:00:00', 'yyyy/mm/dd hh:mi:ss'), 130, 3)

5 into Orders(orderNo, orderDate, handingCost, customerNo) values(4, TO\_DATE

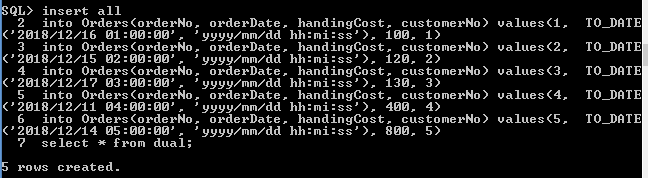
('2018/12/11 04:00:00', 'yyyy/mm/dd hh:mi:ss'), 400, 4)

6 into Orders(orderNo, orderDate, handingCost, customerNo) values(5, TO\_DATE

('2018/12/14 05:00:00', 'yyyy/mm/dd hh:mi:ss'), 800, 5)

7 select \* from dual;

5 rows created.



SQL> insert all

2 into Includes(unitPrice, unitQuantity, orderNo, productNo)values(100, 200,

1, 1)

3 into Includes(unitPrice, unitQuantity, orderNo, productNo)values(9, 300, 2,

2)

4 into Includes(unitPrice, unitQuantity, orderNo, productNo)values(2, 400, 3,

3)

5 into Includes(unitPrice, unitQuantity, orderNo, productNo)values(10, 100, 4

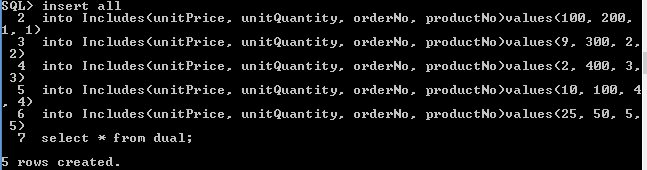
, 4)

6 into Includes(unitPrice, unitQuantity, orderNo, productNo)values(25, 50, 5,

5)

7 select \* from dual;

5 rows created.

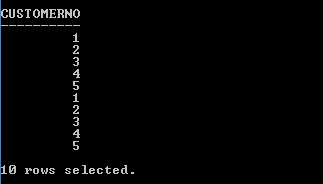


4.

SQL> select customerNo from Customer

2 UNION ALL

3 select customerNo from Orders;



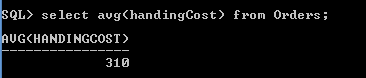
5.

SQL> select avg(handingCost) from Orders;

AVG(HANDINGCOST)

----------------

310



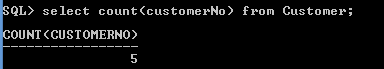
6.

SQL> select count(customerNo) from Customer;

COUNT(CUSTOMERNO)

-----------------

5



7.

SQL> select sum(handingCost), max(handingCost), min(handingCost) from Orders;

SUM(HANDINGCOST) MAX(HANDINGCOST) MIN(HANDINGCOST)

---------------- ---------------- ----------------

1550 800 100



8.

SQL has many built-in functions for performing processing on string or numeric data. Some of them are count(), avg(), min(), max(), sum, char() etc.

9.

DDL is used to create the database schema and DML is used to populate and manipulate database.

DDL is not classified further and DML is further classified as Procedural and Non-Procedural DMLs.

DDL commands are CREATE, ALTER, DROP, TRUNCATE AND COMMENT and RENAME, etc. DML commands are SELECT, INSERT, UPDATE, DELETE, MERGE, CALL, etc.