**NAME : KALYAN SAHA**

**ROLL : 2151219**

**DEPARTMENT : CSE**

**SECTION : A**

**AUTONOMY ROLL NO : 12621001059**

**SUBJECT : SQL**

**PRODUCT TABLE :**

CREATE TABLE PRODUCT (

PRODUCTNO INT PRIMARY KEY,

PRODUCTDESC VARCHAR(30) NOT NULL,

PRODUCTFINISH CHAR(10),

UNITPRICE DECIMAL(10,2) CHECK (unitprice > 0),

QYTONHAND NUMERIC(3) DEFAULT 0 );

INSERT INTO PRODUCT VALUES (1, 'End table', 'Cherry', 175.00, 8);

INSERT INTO PRODUCT VALUES (2,'Coffee table', 'Birch', 200.00, 4);

INSERT INTO PRODUCT VALUES (3,'Computer desk', 'Oak', 375.00, 5);

INSERT INTO PRODUCT VALUES (4,'Entertainment Center', 'Maple', 650.00, 3);

INSERT INTO PRODUCT (PRODUCTNO, PRODUCTDESC ,PRODUCTFINISH, UNITPRICE) VALUES (5,'Writer''s Desk', 'Oak', 325.00);

-- INSERT INTO PRODUCT VALUES (5,'Writer''s Desk', 'Oak', 325.00, 0);

INSERT INTO PRODUCT VALUES (6,'8-Drawer dresser', 'Birch', 750.00, 5);

INSERT INTO PRODUCT VALUES (7,'48" Bookcase', 'Walnut', 150.00, 5);

INSERT INTO PRODUCT VALUES (8,'48" Bookcase', 'Oak', 175.00, 2);

INSERT INTO PRODUCT VALUES (9,'96" Bookcase', 'Walnut', 225.00, 4);

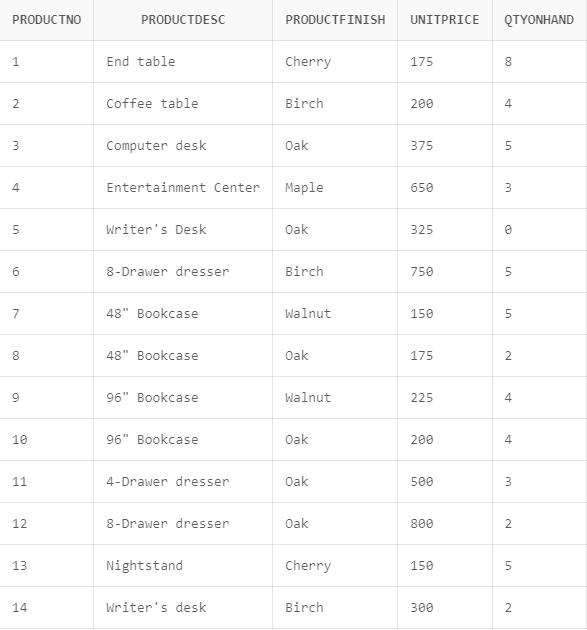
INSERT INTO PRODUCT VALUES (10,'96" Bookcase', 'Oak', 200.00, 4);

INSERT INTO PRODUCT VALUES (11,'4-Drawer dresser', 'Oak', 500.00, 3);

INSERT INTO PRODUCT VALUES (12,'8-Drawer dresser', 'Oak', 800.00, 2);

INSERT INTO PRODUCT VALUES (13,'Nightstand', 'Cherry', 150.00, 5);

INSERT INTO PRODUCT VALUES (14,'Writer''s desk', 'Birch', 300.00, 2);



**CUSTOMER TABLE:**

CREATE TABLE CUSTOMER (

customerid int,

customername varchar(20) NOT NULL,

caddress varchar(30),

ccity char(15),

cstate char(2),

czip char(6) NOT NULL,

PRIMARY KEY (customerid),

UNIQUE (customername));

INSERT INTO CUSTOMER VALUES (1, 'Contemporary Casuals' , '1355 S. Avenue' , 'Kolkata' , 'WB' , 700019);

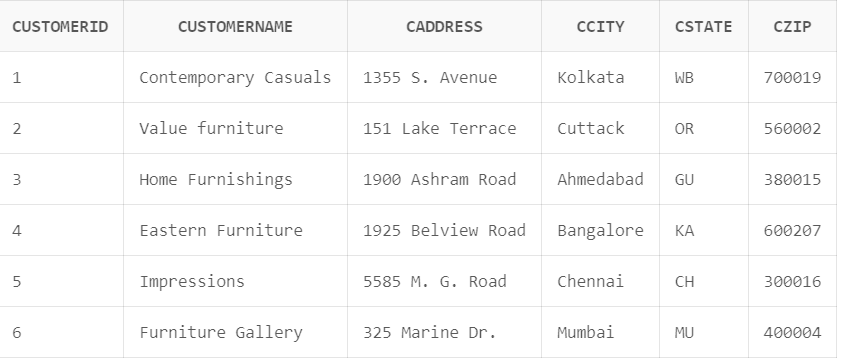
INSERT INTO CUSTOMER VALUES (2, 'Value furniture' , '151 Lake Terrace' , 'Cuttack' , 'OR' , 560002);

INSERT INTO CUSTOMER VALUES (3, 'Home Furnishings' , '1900 Ashram Road', 'Ahmedabad' , 'GU' , 380015);

INSERT INTO CUSTOMER VALUES (4, 'Eastern Furniture' , '1925 Belview Road', 'Bangalore' , 'KA' , 600207);

INSERT INTO CUSTOMER VALUES (5, 'Impressions' , '5585 M. G. Road' , 'Chennai', 'CH' , 300016);

INSERT INTO CUSTOMER VALUES (6, 'Furniture Gallery' , '325 Marine Dr.' , 'Mumbai' , 'MU' , 400004);



**ORDERS :**

create table orders(

orderno int,

orderdate date,

customerid int,

CONSTRAINT PKORDERS

PRIMARY KEY (orderno),

CONSTRAINT FKORDERS

FOREIGN KEY(customerid) REFERENCES CUSTOMER(customerid));

INSERT INTO ORDERS VALUES (1, '29-OCT-2021', 2);

INSERT INTO ORDERS VALUES (2, '11-JAN-2021', 3);

INSERT INTO ORDERS VALUES (3, '15-OCT-2021', 1);

INSERT INTO ORDERS VALUES (4, '04-OCT-2021', 6);

INSERT INTO ORDERS VALUES (5, '02-NOV-2021', 4);

****

**REQUESTS:**

CREATE TABLE REQUESTS(

ORDERNO INT,

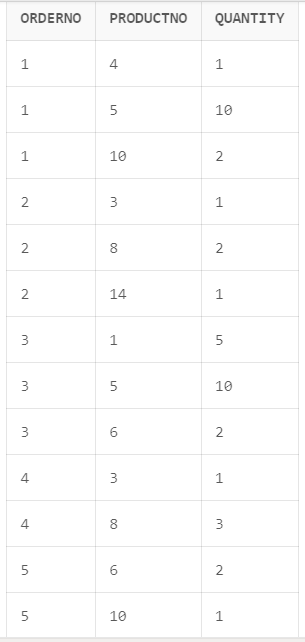
PRODUCTNO INT,

QUANTITY INT NOT NULL,

FOREIGN KEY(PRODUCTNO) REFERENCES PRODUCT(PRODUCTNO),

FOREIGN KEY(ORDERNO) REFERENCES ORDERS(ORDERNO)

)

INSERT INTO REQUESTS VALUES(1, 4, 1);

INSERT INTO REQUESTS VALUES(1, 5, 10);

INSERT INTO REQUESTS VALUES(1, 10, 2);

INSERT INTO REQUESTS VALUES(2, 3, 1);

INSERT INTO REQUESTS VALUES(2, 8, 2);

INSERT INTO REQUESTS VALUES(2, 14, 1);

INSERT INTO REQUESTS VALUES(3, 1, 5);

INSERT INTO REQUESTS VALUES(3, 5, 10);

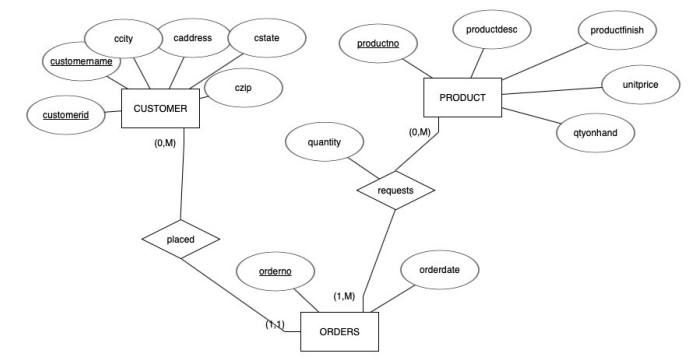
INSERT INTO REQUESTS VALUES(3, 6, 2);

INSERT INTO REQUESTS VALUES(4, 3, 1);

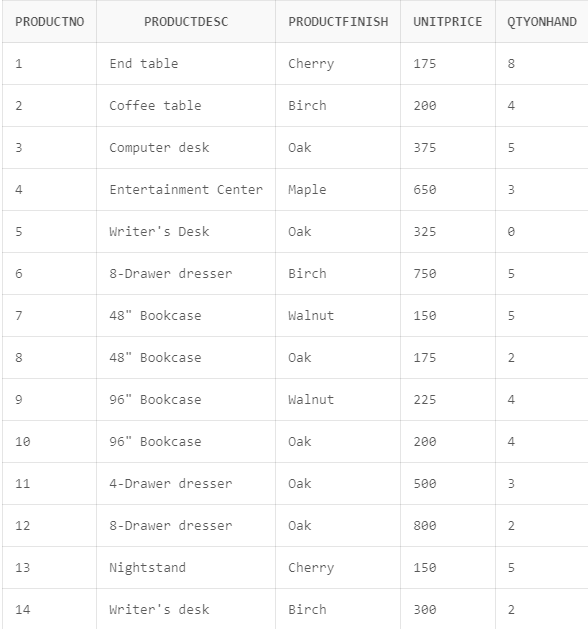
INSERT INTO REQUESTS VALUES(4, 8, 3);

INSERT INTO REQUESTS VALUES(5, 6, 2);

INSERT INTO REQUESTS VALUES(5, 10, 1);

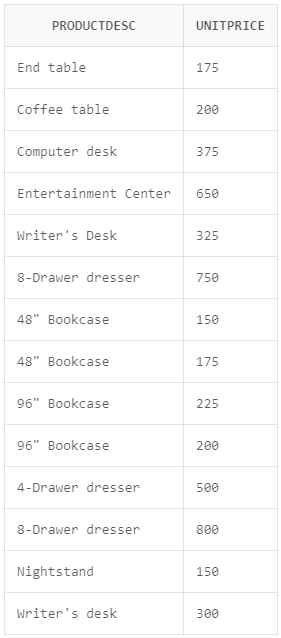


**C. Answer the queries:**

1. List all products

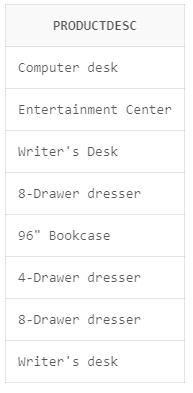
SELECT \*

FROM PRODUCT;

2. List all product names and prices (unitprice)

SELECT PRODUCTDESC, UNITPRICE

FROM PRODUCT;

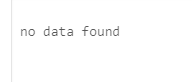
3. List all product names whose product prices > Rs.200

SELECT PRODUCTDESC

FROM PRODUCT

WHERE UNITPRICE>200;

4. List all product names whose product prices > Rs.200 and qty\_on\_hand > 5



SELECT PRODUCTDESC

FROM PRODUCT

WHERE UNITPRICE>200 AND QTYONHAND >5;

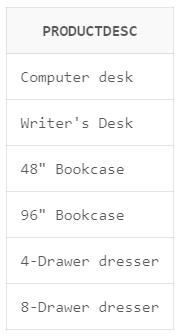
5. List all product names whose product prices > Rs.200 and qty\_on\_hand between 1 and 5



SELECT PRODUCTDESC

FROM PRODUCT

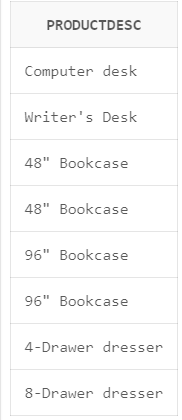
WHERE UNITPRICE>200 AND QTYONHAND BETWEEN 1 AND 5;

6. List all products with finish = "Oak"

SELECT PRODUCTDESC

FROM PRODUCT

WHERE PRODUCTFINISH = 'Oak';

7. List all products with finish = "Oak" or "Walnut"

SELECT PRODUCTDESC

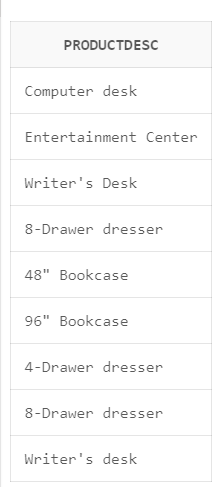
FROM PRODUCT

WHERE PRODUCTFINISH = 'Oak' OR PRODUCTFINISH = 'Walnut';

SELECT PRODUCTDESC

FROM PRODUCT

WHERE PRODUCTFINISH IN('Oak','Walnut');

8. List all products with finish = "Oak" or unitprice >= 300

SELECT PRODUCTDESC

FROM PRODUCT

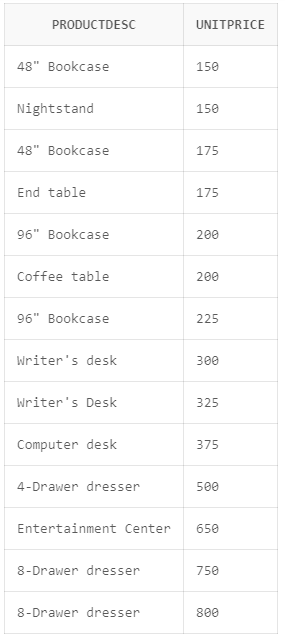
WHERE PRODUCTFINISH = 'Oak' OR UNITPRICE >= 300;

9. List all product names and prices arranged by prices

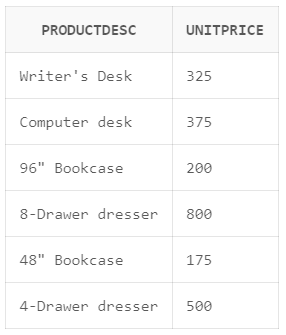
SELECT PRODUCTDESC, UNITPRICE

FROM PRODUCT

ORDER BY UNITPRICE;



10. List all product names and prices whose finish is "Oak" arranged by names in reverse order



SELECT PRODUCTDESC, UNITPRICE

FROM PRODUCT

WHERE PRODUCTFINISH = 'Oak'

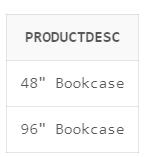
ORDER BY PRODUCTDESC DESC;

11. List all distinct Bookcases.

select distinct productdesc

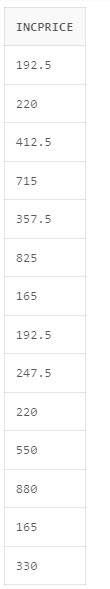
from product

where productdesc like '%Bookcase';



12. List the product prices if they are increased by 10%.

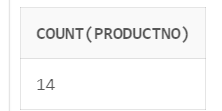
select unitprice\*1.1 as incprice

from product; 

13. How many products are there?

select count(productno)

from product;



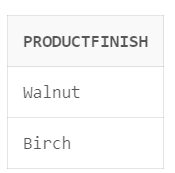
14. For every product finish, find out the total quantity of products

select sum(qtyonhand)

from product

group by productfinish;

15. List every product finish having unit price > 200 and average of qtyonhand greater than 3



select productfinish

from product

where unitprice>200

group by productfinish

having avg(qtyonhand) >3;

16. List product name(s) having maximum quantity on hand

select productdesc

from product

where qtyonhand=any(select max(qtyonhand)

from product);



17. Find out value of inventory for every product finish

select sum(qtyonhand\*unitprice) as inventory

from product

group by productfinish;

18. List all products whose unitprice is same as some made of ‘Oak’

select productdesc

from product

where productfinish <> 'Oak' and unitprice = any(

select unitprice

from product

where productfinish='Oak');

19. List product name(s) having qty on hand more than the average qty on hand

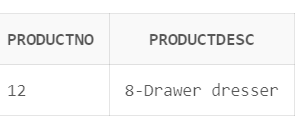
select productdesc

from product

where qtyonhand > (select avg(qtyonhand)

from product);

20. List products whose unit price is greater than every unitprice of different productfinish



SELECT P1.PRODUCTNO, P1.PRODUCTDESC

FROM PRODUCT P1

WHERE P1.UNITPRICE > ALL (

SELECT P2.UNITPRICE

FROM PRODUCT P2

WHERE P2.PRODUCTFINISH <> P1.PRODUCTFINISH

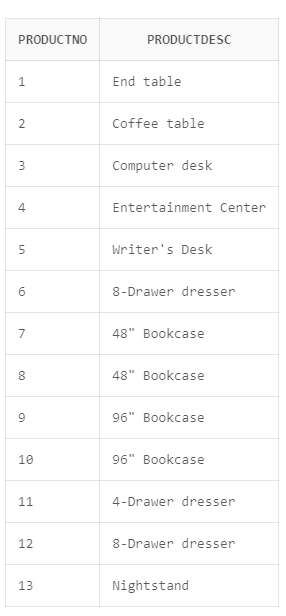
);

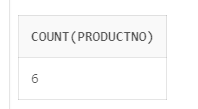
21. List all products which are either made of “Oak” or which has qtyonhand > 2

select productno,productdesc

from product

where productfinish='Oak' or qtyonhand>2;



22. How many products are there whose finish is of "Oak"?

select count(productno)

from product

group by productfinish

having productfinish='Oak';



23. List every productfinish whose total qtyonhand is greater than 3.

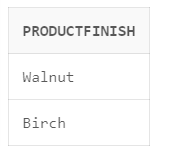
select productfinish

from product

group by productfinish

having sum(qtyonhand)>3;

24. List every productfinish having unit price > 200 and average of qtyonhand greater than 3 arranged in reversed order of productfinish.



select productfinish

from product

where unitprice>200

group by productfinish

having avg(qtyonhand)>3

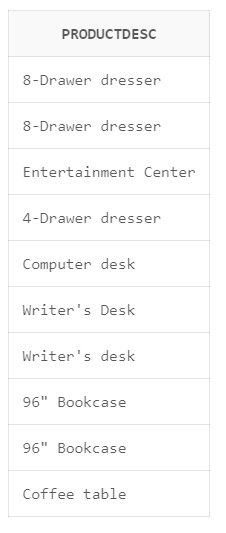
order by productfinish desc;

25. Find out the total value of inventory (unitprice \* qtyonhand) summed over all products

SELECT SUM(UNITPRICE \* QTYONHAND) AS TOTAL\_INVENTORY\_VALUE

FROM PRODUCT;

26. List all products whose unitprice is more than unitprice of products made of ‘Oak”



select productdesc

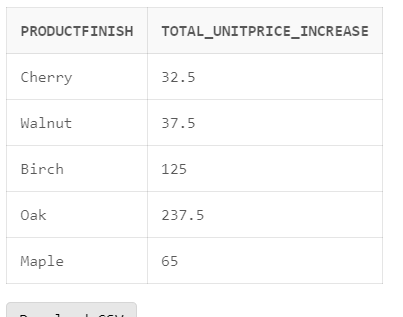
from product

where unitprice > any(select unitprice

from product

where productfinish = 'Oak');

27. If the unitprice is increased by 10% for every product, find out the total increase of unitprice for every productfinish (without updating tables)

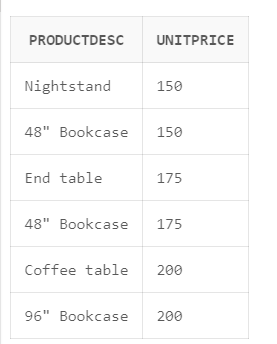


SELECT PRODUCTFINISH, SUM(UNITPRICE \* 0.10) AS TOTAL\_UNITPRICE\_INCREASE

FROM PRODUCT

GROUP BY PRODUCTFINISH;

28. List all products whose unitprice is same as some other product.



select p.productdesc,p.unitprice

from product p

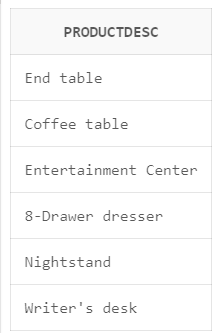
where p.unitprice =

any(select q.unitprice

from product q

where p.productno<>q.productno)

order by 2;

29. List the products which are not made of “Oak” or “Walnut”

select productdesc

from product

where productfinish not in('Oak','Walnut');

30. List names of products whose finish is ‘Oak’ and having minimum qtyonhand



select productdesc

from product

where productfinish='Oak'

and qtyonhand = (select min(qtyonhand)

from product);

31.Is the following SQL statement valid? What output will be displayed?

SELECT \*

FROM CUSTOMER,CUSTOMER;

=> not valid

=> column ambiguously defined

32. What will be the output for

SELECT \*

FROM CUSTOMER, CUSTOMER;

=> column ambiguously defined

33. What will be the output for

SELECT \*

FROM CUSTOMER C1, CUSTOMER C2;

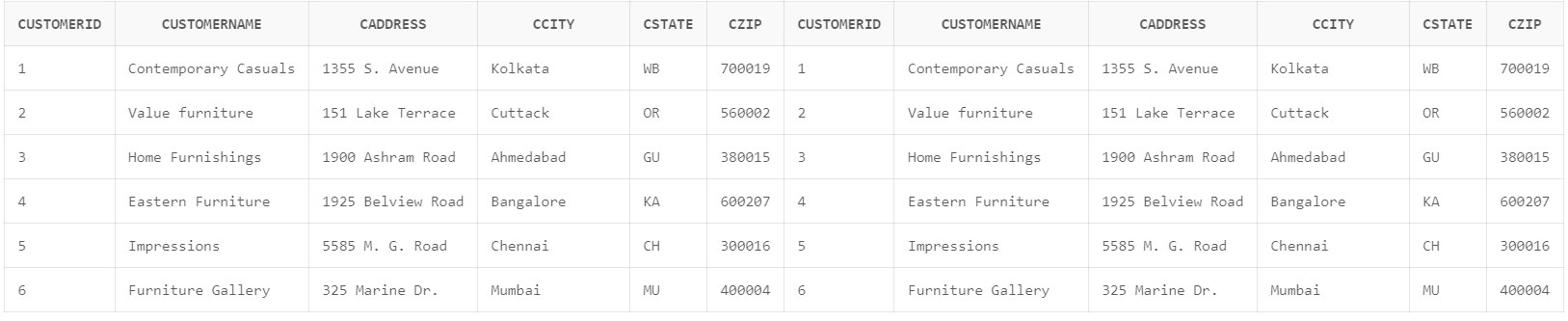
=>cardesion product

34. What will be the output for

SELECT \*

FROM CUSTOMER C1, CUSTOMER C2

WHERE C1.CUSTOMERID=C2.CUSTOMERID;

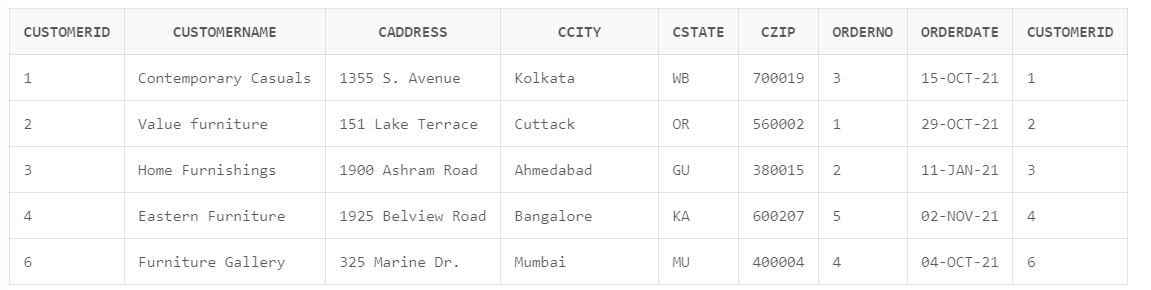


35. What will be the output for

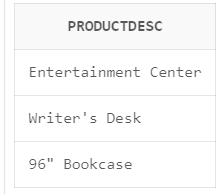
SELECT \*

FROM CUSTOMER C, ORDERS O

WHERE C.CUSTOMERID = O.CUSTOMERID;



=> customers who have placed order



36.List all product descriptions ordered in orderno1

SELECT PRODUCTDESC

FROM PRODUCT NATURAL JOIN REQUESTS

WHERE ORDERNO = 1;



37.List every request where qty ordered is more than the average quantity ordered for the product

SELECT r.ORDERNO, r.PRODUCTNO, r.QUANTITY

FROM REQUESTS r

WHERE r.QUANTITY > (

SELECT AVG(QUANTITY)

FROM REQUESTS

WHERE PRODUCTNO = r.PRODUCTNO

);

38.List the names of all customers who have placed an order

select c.customername,c.customerid

from customer c right join orders o

on c.customerid=o.customerid;

39.List the product descriptions which are ordered by customer named "Contemporary Casuals"

select p.productdesc

from product p,customer c,requests r,orders o

where p.productno=r.productno

and c.customerid=o.customerid

and r.orderno=o.orderno

and c.customername='Contemporary Casuals';

40.List the order dates for every order made by customer "Value Furniture"

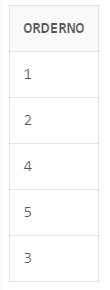
select o.orderdate

from orders o inner join customer c

on c.customerid=o.customerid

and c.customername='Value Furniture';

41.List every order which includes a request for a product made of "Oak"



SELECT DISTINCT o.orderno

FROM ORDERS o

INNER JOIN REQUESTS r ON o.orderno = r.orderno

INNER JOIN PRODUCT p ON r.PRODUCTNO = p.PRODUCTNO

WHERE p.PRODUCTFINISH = 'Oak';

42.List every order which includes request only for products made of “Oak”

SELECT DISTINCT o.orderno

FROM ORDERS o

JOIN REQUESTS r ON o.orderno = r.orderno

JOIN PRODUCT p ON r.productno = p.productno

WHERE p.productfinish = 'Oak'

AND NOT EXISTS (

SELECT 1

FROM REQUESTS r2

JOIN PRODUCT p2 ON r2.productno = p2.productno

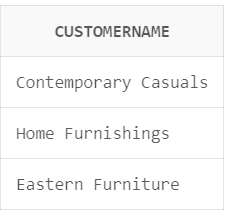
WHERE r2.orderno = o.orderno

AND p2.productfinish != 'Oak'

);



43.List the names of customers who have ordered a product of “Birch” finish

SELECT DISTINCT c.customername

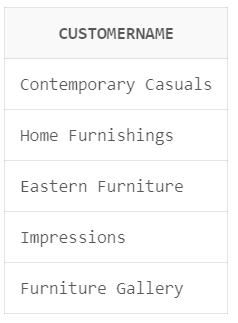
FROM CUSTOMER c

INNER JOIN ORDERS o ON c.customerid = o.customerid

INNER JOIN REQUESTS r ON o.orderno = r.orderno

INNER JOIN PRODUCT p ON r.PRODUCTNO = p.PRODUCTNO

WHERE p.PRODUCTFINISH = 'Birch';

44.List the names of customers who have not ordered any product of “Maple” finish

SELECT DISTINCT c.customername

FROM CUSTOMER c

LEFT JOIN ORDERS o ON c.customerid = o.customerid

LEFT JOIN REQUESTS r ON o.orderno = r.orderno

LEFT JOIN PRODUCT p ON r.PRODUCTNO = p.PRODUCTNO

AND p.PRODUCTFINISH = 'Maple'

WHERE p.PRODUCTNO IS NULL;



45.List the total quantity ordered of every product finish

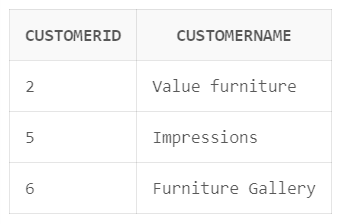
select sum(r.quantity)

from product p

left join requests r

on r.productno=p.productno

group by productfinish;

46.List the customers who have not ordered products made of “Birch”

SELECT DISTINCT c.customerid, c.customername

FROM CUSTOMER c

WHERE NOT EXISTS (

SELECT 1

FROM ORDERS o

JOIN REQUESTS r ON o.orderno = r.orderno

JOIN PRODUCT p ON r.productno = p.productno

WHERE o.customerid = c.customerid

AND p.productfinish = 'Birch'

);

47.List quantity on hand and quantity ordered for every product

select p.productno,p.QYTONHAND,sum(r.quantity)

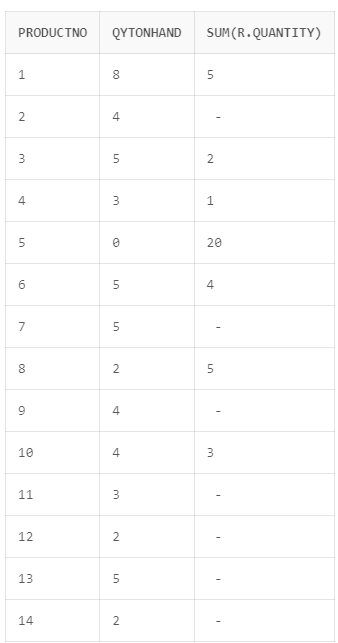
from product p

left join requests r on p.productno=r.productno

left join orders o on o.orderno=r.orderno

group by p.productno,p.qytonhand

order by 1;



48.List the products which have not been ordered

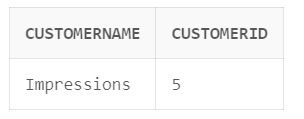
select p.productno

from product p

left join requests r on r.productno=p.productno

where r.productno is null;

49.List the customers who have not placed any order

select c.customername,c.customerid

from customer c

left join orders o

on c.customerid=o.customerid

where o.customerid is null;

50.List the products whose total ordered quantity is more than qtyonhand.

SELECT p.PRODUCTNO, p.PRODUCTDESC, p.QYTONHAND, SUM(r.QUANTITY) AS TotalOrderedQuantity

FROM PRODUCT p

LEFT JOIN REQUESTS r ON p.PRODUCTNO = r.PRODUCTNO

GROUP BY p.PRODUCTNO, p.PRODUCTDESC, p.QYTONHAND

HAVING SUM(r.QUANTITY) > p.QYTONHAND;

