Equijoin is called as _____.

Select one:

- a. Simple Join ✔
- b. Self Join
- c. Equal Join
- d. Outer Join

Your answer is correct.

The correct answer is: Simple Join

Consider the below orders table:

Column Name	Datatype	Constraint
order_id	Number	PK
Order_date	Date	
Order_mode	varchar	
Customer_id	Number	
Order_total	Number(8,2)	

There is only one customer with the CUST_LAST_NAME column having value Roberts. Which INSERT statement should be used to add a row into the ORDERS table for the customer whose CUST_LAST_NAME is Roberts and CREDIT_LIMIT is 600?

Select one:

- a. INSERT INTO orders (order_id ,order _date,order_mode , (SELECT customer_id FROM customers
  WHERE cust_last_name ='Roberts' AND
  credit_limit =600) , order _total)
  VALUES (1,'10-mar-2007','direct' ,& customer_id,1000);
- b. INSERT INTO orders (order_id ,order _date,order_mode , (SELECT customer_id FROM customers
  WHERE cust_last_name ='Roberts' AND
  credit_limit =600) , order _total)
  VALUES (1,'10-mar-2007','direct' ,& &customer_id,1000);
- c. INSERT INTO(SELECT o.order_id,o.order,o.order_mode,c.customer_id,o.order_total
  FROM orders o, customers c WHERE o.customer_id = c.customer_id AND c.cust_last_name
  ='Roberts' and c.credit_limit =600)
  VALUES (1,'10-mar-2007','direct' ,& &customer_id,1000);
- d. INSERT INTO orders VALUES(1,'10-mar-2007','direct', (SELECT customer_id FROM ✔
  customers WHERE cust_last_name ='Roberts' AND
  credit_limit =600) ,1000);

Your answer is correct.

The correct answer is: INSERT INTO orders VALUES(1,'10-mar-2007','direct', (SELECT customer_id FROM customers WHERE cust_last_name ='Roberts' AND
credit_limit =600) ,1000);

Joining a table to itself is called as _____.

Select one:

- a. Self Join ✔
- b. Equi Join
- c. Non Equi Join
- d. Outer Join

Your answer is correct.

The correct answer is: Self Join

The _____ join produces the cross product of two tables.

Select one:

- a. Outer
- b. Cross ✔
- c. Equi
- d. Self

Your answer is correct.

The correct answer is: Cross

_____ is used to retrieve records that do not meet the join condition

Select one:

- a. Non Equi Join
- b. Outer Join ✔
- c. Self Join
- d. Equi Join

Your answer is correct.

The correct answer is: Outer Join

To delete rows from the SALES table, where the PROMO_NAME column in the PROMOTIONS table has either 'blowout sale' or 'everyday low price' as values. Which DELETE statements are valid? (Choose all that apply.)

Select one or more:

a.
```
DELETE
FROM sales
WHERE promo_id = (SELECT promo_id
FROM promotions
WHERE promo_name = 'blowout sale')
OR promo_id = (SELECT promo_id
FROM promotions
WHERE promo_name = 'everyday low price');
```

b.
```
DELETE
FROM sales
WHERE promo_id = (SELECT promo_id
FROM promotions
WHERE promo_name = 'blowout sale')
AND promo_id = (SELECT promo_id
FROM promotions
WHERE promo_name = 'everyday low price');
```

c.
```
DELETE
FROM sales
WHERE promo_id IN (SELECT promo_id
FROM promotions
WHERE promo_name IN ('blowout sale','everyday low price'));
```

d.
```
DELETE
FROM sales
WHERE promo_id IN (SELECT promo_id
FROM promotions
WHERE promo_name = 'blowout sale')
OR promo_name = 'everyday low price';
```

Your answer is correct.

The correct answers are: DELETE
FROM sales
WHERE promo_id = (SELECT promo_id
FROM promotions
WHERE promo_name = 'blowout sale')
OR promo_id = (SELECT promo_id
FROM promotions
WHERE promo_name = 'everyday low price');, DELETE
FROM sales
WHERE promo_id IN (SELECT promo_id
FROM promotions
WHERE promo_name = 'blowout sale')
OR promo_name = 'everyday low price';, DELETE
FROM sales
WHERE promo_id IN (SELECT promo_id
FROM promotions
WHERE promo_name IN ('blowout sale','everyday low price'));

Mr. John is the president of a company. Five managers report to him. All other employees report to these managers.

Examine the code:
SELECT employee.ename FROM emp employee
WHERE employee.empno NOT IN (SELECT manager.mgr
FROM emp manager);

The above statement returns no rows selected. as the result. Why?

Select one:

- a. NOT IN operator is not allowed in subqueries.
- b. All employees have a manager.
- c. None of the employees has a manager.
- d. A NULL value is returned from the subquery. ✔

Your answer is correct.

The correct answer is: A NULL value is returned from the subquery.

Consider the below tables:

Customer Table

Column Name	Datatype	Constraint
custNo	Number	PK
custname	Varchar	
custaddress	varchar	
Cust_credit_limit	Number	

Grade Table

Column Name	Datatype	Constraint
Grade	Varchar	
Startval	Number	
Endval	Number	

To display names and grades of customers who have the highest credit limit.
Which SQL statements would accomplish the task?

Select one or more:

- a. SELECT custname, grade
  FROM customers, grades
  WHERE (SELECT MAX(cust_credit_limit)
  FROM customers) BETWEEN startval and endval;
- b. SELECT custname, grade ✔
  FROM customers, grades

WHERE (SELECT MAX(cust_credit_limit)
FROM customers) BETWEEN startval and endval
AND cust_credit_limit BETWEEN startval AND endval;

c. SELECT custname, grade ✔
FROM customers, grades
WHERE cust_credit_limit = (SELECT MAX(cust_credit_limit)
FROM customers)
AND cust_credit_limit BETWEEN startval AND endval;

d. SELECT custname, grade
FROM customers, grades
WHERE cust_credit_limit IN (SELECT MAX(cust_credit_limit)
FROM customers)
AND MAX(cust_credit_limit) BETWEEN startval AND endval;

Your answer is correct.

The correct answers are: SELECT custname, grade
FROM customers, grades
WHERE cust_credit_limit = (SELECT MAX(cust_credit_limit)
FROM customers)
AND cust_credit_limit BETWEEN startval AND endval;, SELECT custname, grade
FROM customers, grades
WHERE (SELECT MAX(cust_credit_limit)
FROM customers) BETWEEN startval and endval
AND cust_credit_limit BETWEEN startval AND endval;

The _____ join is based on all columns in the two tables that have the same column name.

Select one:

a. Left Outer

b. Full Outer

c. Natural ✔

d. Cross

Your answer is correct.

The correct answer is: Natural

Consider the below table:

Products Table

Column Name	Datatype	Constraint
Prod_id	Number	PK
Prod_name	Varchar	
Prod_list_price	varchar	
Cust_credit_limit	Number	

What would be the outcome of executing the below SQL statement?

```
select prod_name from products where prod_id in(
select prod_id from products where prod_list_price=(
select max(prod_list_price) from products where prod_list_price<(
select max(prod_list_price)from products)));
```

Select one:

- a. It produces an error
- b. It shown the names of all products whose list price is less than the maximum list price
- c. It shown the names of products whose list price is the second highest in the table ✔
- d. It shows the names of all products in the table.

Your answer is correct.

The correct answer is: It shown the names of products whose list price is the second highest in the table