Code the Following questions using the Antiques Dataset

*****For full credit, submit both your **coding answers** along with a **snipped image of your code output**. Also, some questions require explanations, in addition to code and code output.

On a PC, you can use the Snipping tool, and on MAC, Command/SHIFT/4 allows you to snip. Please don't take entire screen captures. Google can provide additional information for both snipping and the MAC capture.

1. Explain the difference between the UNION ALL and UNION operators
    -
    - In what cases are they equivalent?
    - When they are equivalent, which one should you use?

2. Write a query that generates a virtual auxiliary table of 10 numbers in the range 1 through 10

```
SELECT 1 AS N

UNION ALL SELECT 2

UNION ALL SELECT 3

UNION ALL SELECT 4

UNION ALL SELECT 5

UNION ALL SELECT 6

UNION ALL SELECT 7

UNION ALL SELECT 8

UNION ALL SELECT 9

UNION ALL SELECT 10;
```

-
  - Tables involved: no table
  - HINT: The following code gets you started on this program.

SELECT 1 as N

UNION ALL SELECT 2;

--Desired output

n

1

2

3

4

5

6

7

8

9

10

(10 row(s) affected)

3. Would question 2 works if you used UNION instead of UNION all?

Yes, it would work if used union instead of union all but it might not be most efficient because union removes duplicates from the result set, which in this case won't make a difference since we are generating distinct numbers. However, it requires additional processing to identify and remove duplicates, which could impact performance for larger datasets.

4. Write a query (using set operators) that returns customer and employee pairs

    -
    - that were part of Sales, but don't include any related to customerID 1, as these don't count toward promotions
    - Tables involved: Antiques SALE Table
    - Solve this problem using the EXCEPT operators
    - *Note: This problem is easily solved with a simple WHERE statement. However, we use it to illustrate how the EXCEPT operator functions.

Answer

```
SELECT CustomerID, EmployeeID

FROM SALE

EXCEPT

SELECT CustomerID, EmployeeID

FROM SALE

WHERE CustomerID = 1;
```

33 %

Results | Messages

	CustomerID	EmployeeID
1	4	5
2	7	5
3	7	7
4	12	5
5	13	7
6	17	25
7	18	11
8	20	17
9	21	17
10	21	25
11	21	28
12	23	16
13	24	29
14	27	19
15	27	25
16	28	30
17	30	10
18	32	12
19	34	22
20	36	30
21	38	30
22	42	12
23	42	21
24	43	20
25	45	30
26	49	18
27	50	21
28	50	30
29	52	20
30	54	21
31	56	20
32	60	20
33	60	21
34	60	24

Query executed successfully.

5. Run the following Code. If you receive an error, describe it.

```
SELECT  *

FROM CUSTOMER

INTERSECT

SELECT *

FROM SALE
```

ERROR: This code will have an error while running the code is

All queries combined using a UNION, INTERSECT or EXCEPT operator must have an equal number of expressions in their target lists. The error message indicate that the number of columns returned by both select statement before and after the intersecpt operator must be the same. In this case, number of columns returned by both select statement must match.

Explanation : Based upon the rule of the error number of column must be the same and data type should be the same.

6. Using an INTERSECT Operator, fix the code above to show the CustomerIDs that are represented in both tables. What does it mean if a customer shows ups in the output (see below)?

```
SELECT CustomerID

FROM CUSTOMER

INTERSECT

SELECT CustomerID

FROM SALE;
```

133 %

Results | Messages

	CustomerID
1	1
2	4
3	7
4	12
5	13
6	17
7	18
8	20
9	21
10	23
11	24
12	27
13	28
14	30
15	32
16	34
17	36
18	38
19	42
20	43
21	45
22	49
23	50
24	52
25	54
26	56
27	60

7. What is the difference between using the INTERSECT operator and INNER JOIN? Look up a VENN Diagram for SQL INTERSECT and SQL INNER JOIN. Are they different?

The INTERSECT operator and INNER JOIN serve similar purposes in SQL, but they have some key differences:

INTERSECT Operator:

- The INTERSECT operator is used to retrieve the common records between two SELECT statements.
- It returns only the rows that appear in both result sets of the SELECT statements.
- The number and order of columns in both SELECT statements must be the same.
- It does not require a join condition or explicit relationship between tables.

INNER JOIN:

- The INNER JOIN clause is used to combine rows from two or more tables based on a related column between them.
- It returns only the rows where there is a match between the columns specified in the join condition.
- It allows joining tables based on different criteria, not just common values.
- It can involve multiple tables and complex join conditions.

Comparison:

- The primary difference lies in their usage and functionality:
- INTERSECT focuses on finding common records between two result sets based on matching values in the same column(s).
- INNER JOIN combines rows from different tables based on a specified relationship or join condition, which may involve multiple columns.

Venn Diagram Comparison:

- In a Venn diagram representing SQL INTERSECT, the overlapping region between two circles represents the common records shared by both result sets.
- In a Venn diagram representing SQL INNER JOIN, the overlapping region between two circles represents the matching rows obtained by joining tables based on the specified condition.

Overall, while both INTERSECT and INNER JOIN can be used to retrieve common records, they have distinct usage scenarios and mechanisms for achieving the desired results.

8. Are the names of the columns in a result set determined by the first or second query (i.e., query before a UNION or query after a UNION)? What are the implications of this for assigning aliases?

IN Sql operation in sql, the names of the column in the result set are determined by the column names from the first query. The column names in the subsequent queries don't affect the name of the column.

The implication of this assigning aliases are as follows

1. Consistency of the data types
2. Order of the columns.

With this implication we can effectively use aliases and structure queries to produce the desired results set when using UNION operation in SQL.

9. ORDER BY isn't allowed in many SQL OPERATORS such as CTE and Derived Tables. What are the implications for the ORDER BY clause with regards to Set Operators?

Order by clause is not allowed directly within set operator such as UNION, INTERSECT, and EXCEPT in SQL. This restriction is because the set operator combines the result of multiple queries and applying order by directly to them would not make logical sense the order of rows is not preserved after the set operation.

However, there are workaround to achieve ordering using set operator.

1. Order before set operation.
2. Ordering after set operation
3. Use of Row Numbering .

That's it! You did it!