

LAB 4 - SET OPERATORS

1. Explain the difference between the UNION ALL and UNION operators?

Union means joining two or more data sets into a single set.

A union is used for extracting rows using the conditions specified in the query while Union All is used for extracting all the rows from a set of two tables.

The only difference between Union and Union All is that Union extracts the rows that are being specified in the query while Union All extracts all the rows including the duplicates (repeated values) from both the queries.

- In what cases are they equivalent?

If we don't have duplicates in our dataset, Union ALL and Union would be equivalent.

- When they are equivalent, which one should you use?

Union ALL runs faster than Union because it includes duplicates. We need to use Union ALL because it's faster.

According to the Ben-Gan book,

The UNION ALL operator unifies the two input query result sets and doesn't remove duplicates from the result. The UNION operator (implied DISTINCT) also unifies the two input query result sets, but it does remove duplicates from the result.

The two have different meanings when the result can potentially have duplicates. They have an equivalent meaning when the result can't have duplicates, such as when you're unifying disjoint sets (for example, sales 2015 with sales 2016).

When they do have the same meaning, you need to use UNION ALL by default. That's to avoid paying unnecessary performance penalties for the work involved in removing duplicates when they don't exist.

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

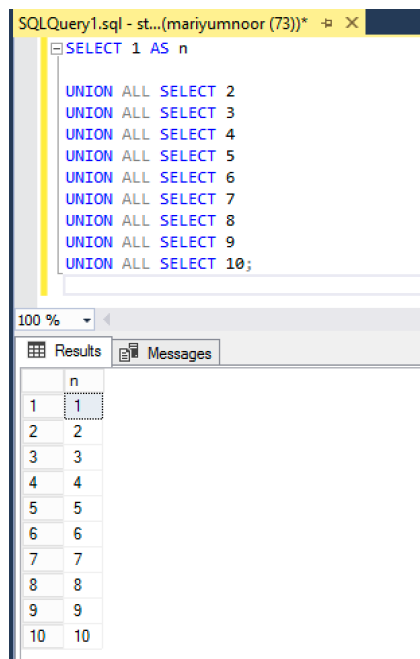
7

8

9

10

(10 row(s) affected)



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

Yes, it works!

```
--Question 2
SELECT 1 AS n
UNION SELECT 2
UNION SELECT 3
UNION SELECT 4
UNION SELECT 5
UNION SELECT 6
UNION SELECT 7
UNION SELECT 8
UNION SELECT 9
UNION SELECT 10;
```

00 %

Results Messages

	n
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10

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

```
-- 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  
  
SELECT CustomerID, EmployeeID  
FROM Sale  
  
EXCEPT  
  
SELECT CustomerID, EmployeeID  
FROM Sale  
WHERE CustomerID = 1
```

100 %

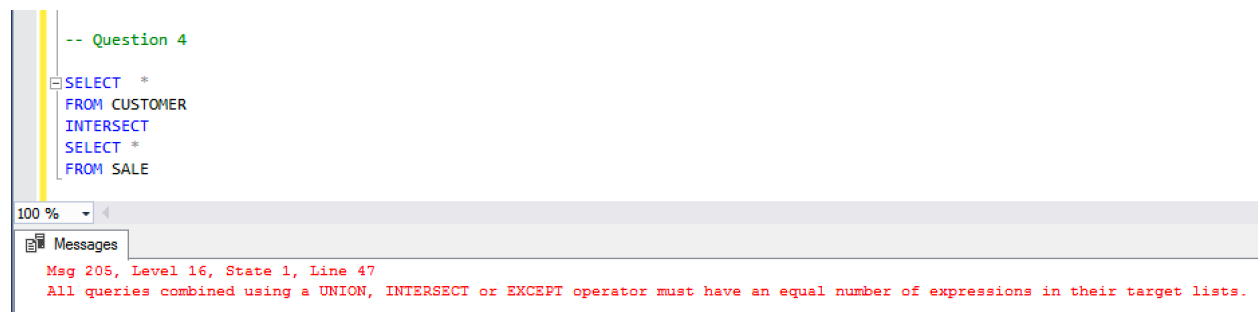
Results

Messages

	CustomerID	EmployeeID
1	2	3
2	4	5
3	7	5
4	7	7
5	12	5
6	13	7

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

```
SELECT *  
FROM CUSTOMER  
INTERSECT  
SELECT *  
FROM SALE
```



The screenshot shows a SQL query editor with the following code:

```
-- Question 4  
SELECT *  
FROM CUSTOMER  
INTERSECT  
SELECT *  
FROM SALE
```

Below the editor, a 'Messages' pane displays the following error:

```
Msg 208, Level 16, State 1, Line 47  
All queries combined using a UNION, INTERSECT or EXCEPT operator must have an equal number of expressions in their target lists.
```

In order to use union or union all numbers of columns and their types should be identical across all result sets!

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
```

100 %

Results Messages

	CustomerID
1	1
2	2
3	4
4	7
5	12
6	13

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?

In general, joins are used to combine related rows in two tables, while intersect/except are used to compare two tables and show the data that is the same or different between two tables.

The book says:

“If instead of using the INTERSECT operator you use an alternative tool like an inner join or a correlated subquery, you need to add special treatment for NULLs—for example, assuming the alias E for Employees and C for Customers, using the predicate E.region = C.region OR (E.region IS NULL AND C.region IS NULL).”

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?

The output column names are referred from the first query i.e. when we run the SELECT statements with any of the Set operators and result set of each of the queries may have different column names, so the result of the SELECT statement refers the column names from the first query in the operation.

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?

Individual result set ordering is not allowed with Set operators.

ORDER BY can appear once at the end of the query. For example, UNION and INTERSECT operators are commutative, i.e. the order of queries is not important; it doesn't change the final result.