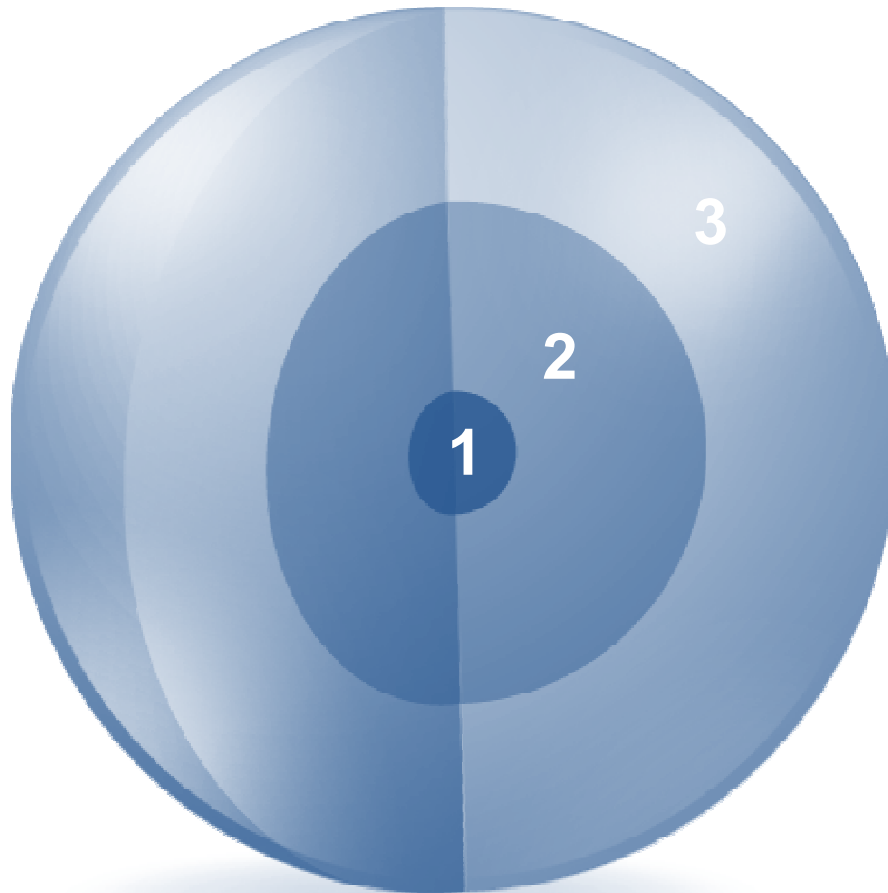




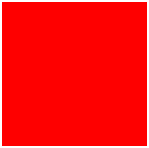
Lesson 8

Using the Set Operators

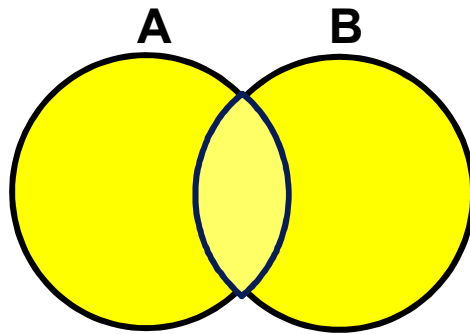
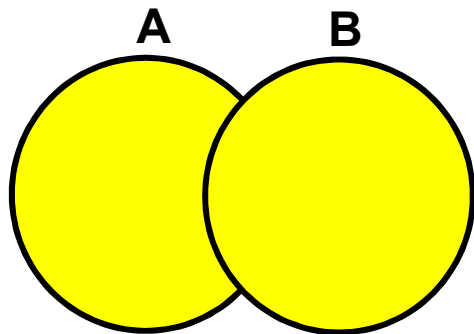
What You will learn at the end of this Session?



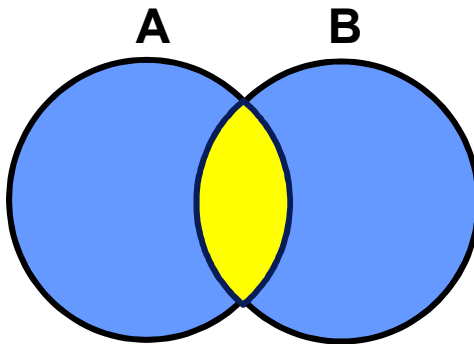
1. Describe set operators
2. Use a set operator to combine multiple queries into a single query
3. Control the order of rows returned



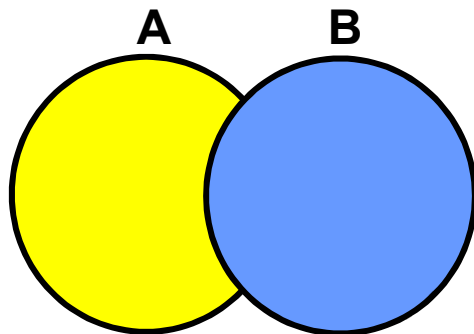
Set Operators



UNION/UNION ALL



INTERSECT

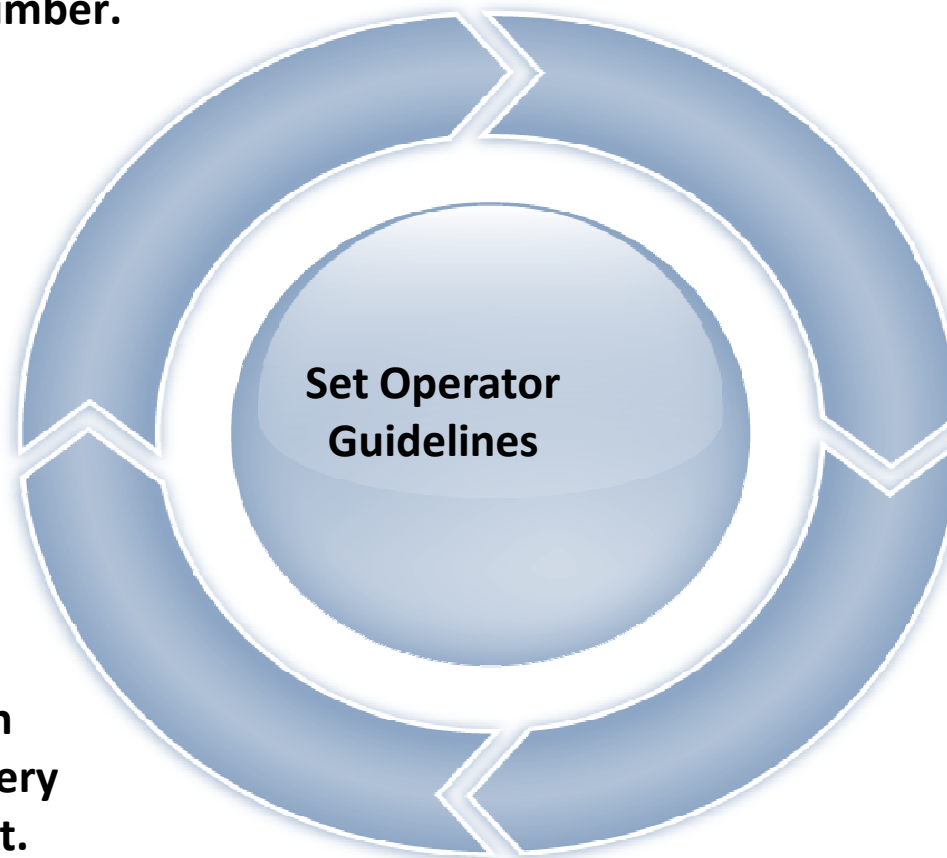


MINUS

Set Operator Guidelines

The expressions in the SELECT lists must match in number.

The data type of each column in the second query must match the data type of its corresponding column in the first query.

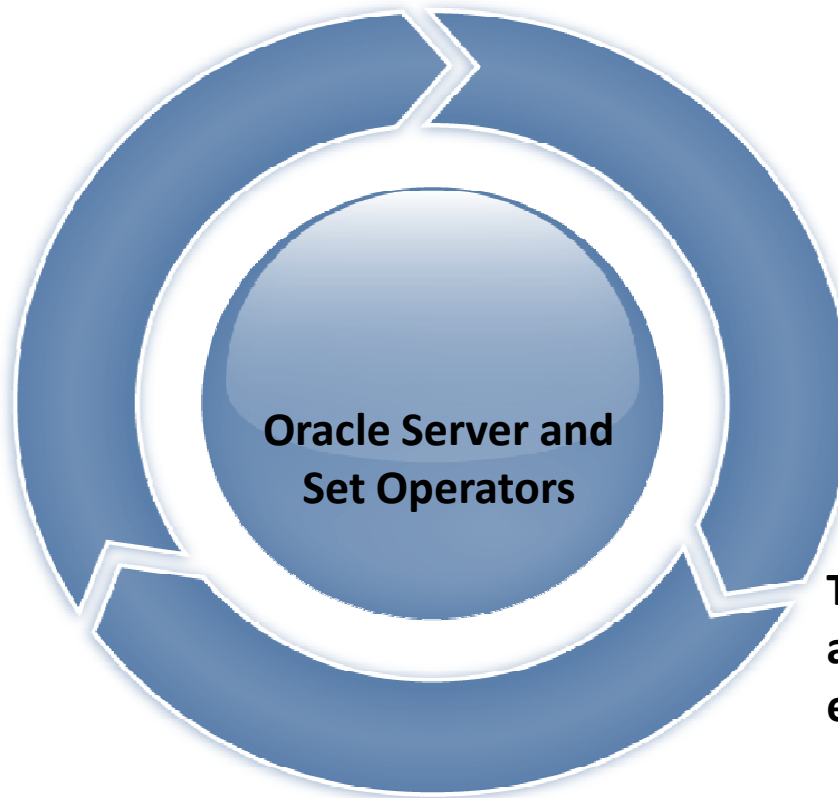


ORDER BY clause can appear only at the very end of the statement.

Parentheses can be used to alter the sequence of execution.

Oracle Server and Set Operators

Column names from the first query appear in the result.



Duplicate rows are automatically eliminated except in UNION ALL.

The output is sorted in ascending order by default except in UNION ALL.

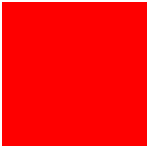


Tables Used in This Lesson

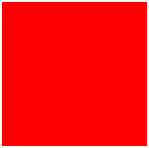
•The tables used in this lesson are:

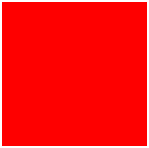
ORDERS: Provides details regarding all current orders.

ORDER_ITEMS: Records the details of the order in the form of product ID, price per unit, quantity ordered etc.

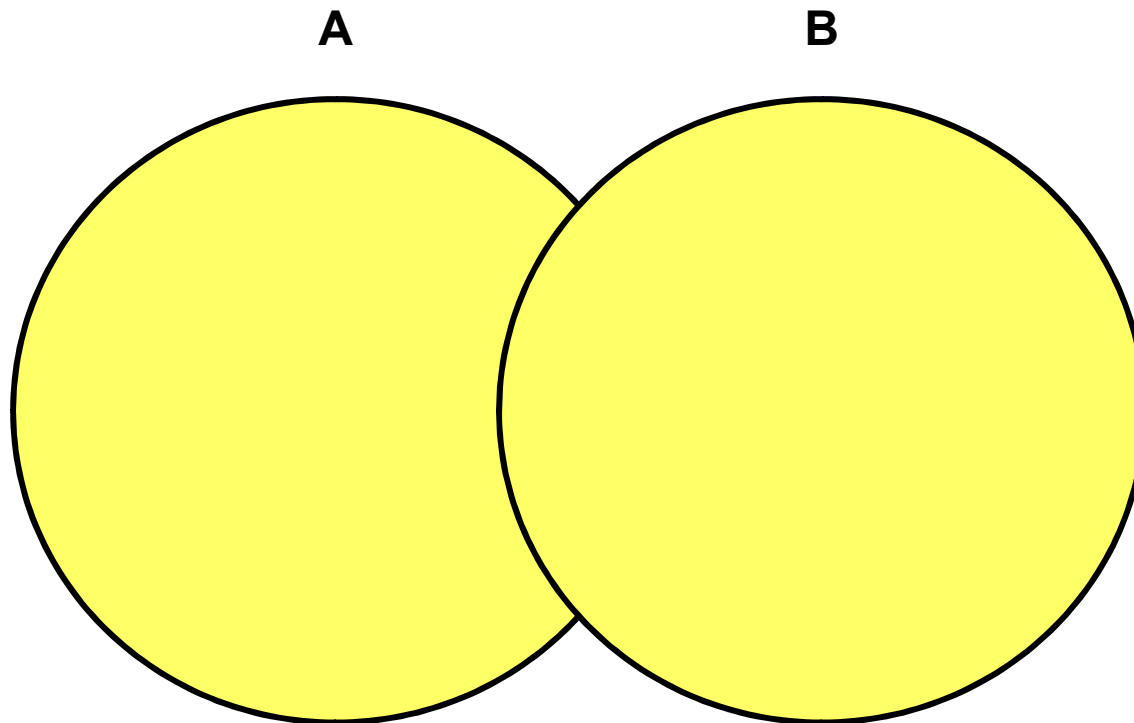








UNION Operator



The UNION operator returns rows from both queries after eliminating duplications.

Using the UNION Operator

```
SELECT order_id
FROM orders
UNION
SELECT order_id
FROM order_items ORDER BY order_id;
```

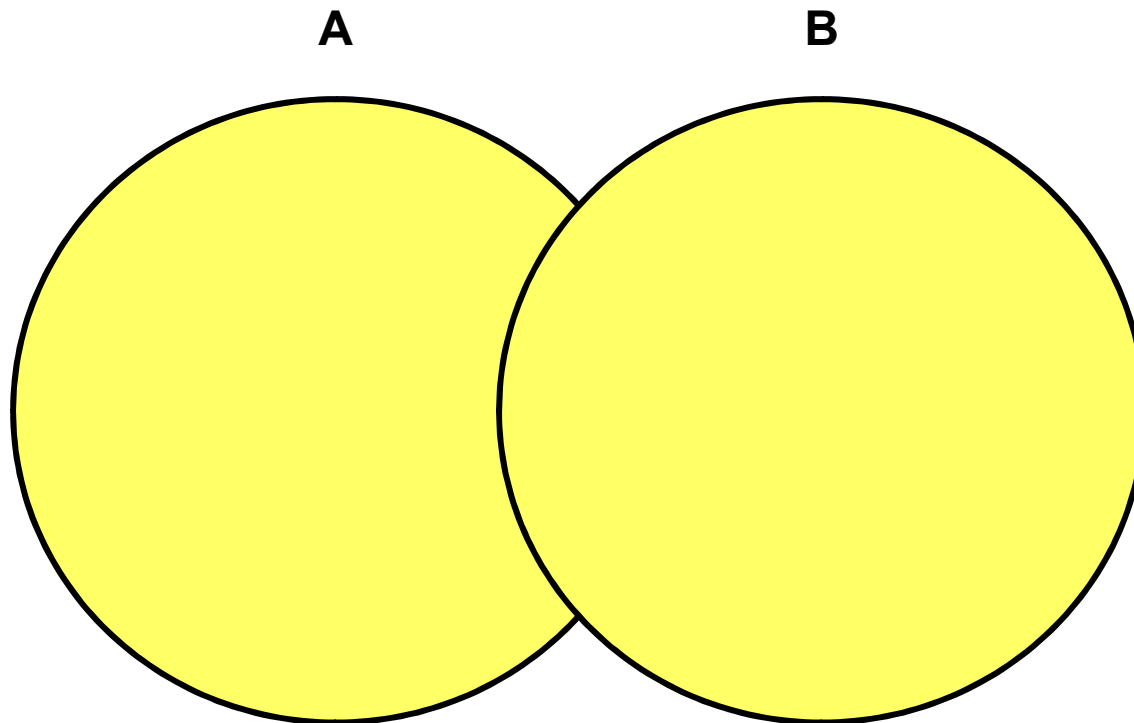
	ORDER_ID
1	2354
2	2355
3	2356
4	2357
5	2358

...

73	2426
74	2427

...





The UNION ALL operator returns rows from both queries, including all duplications.

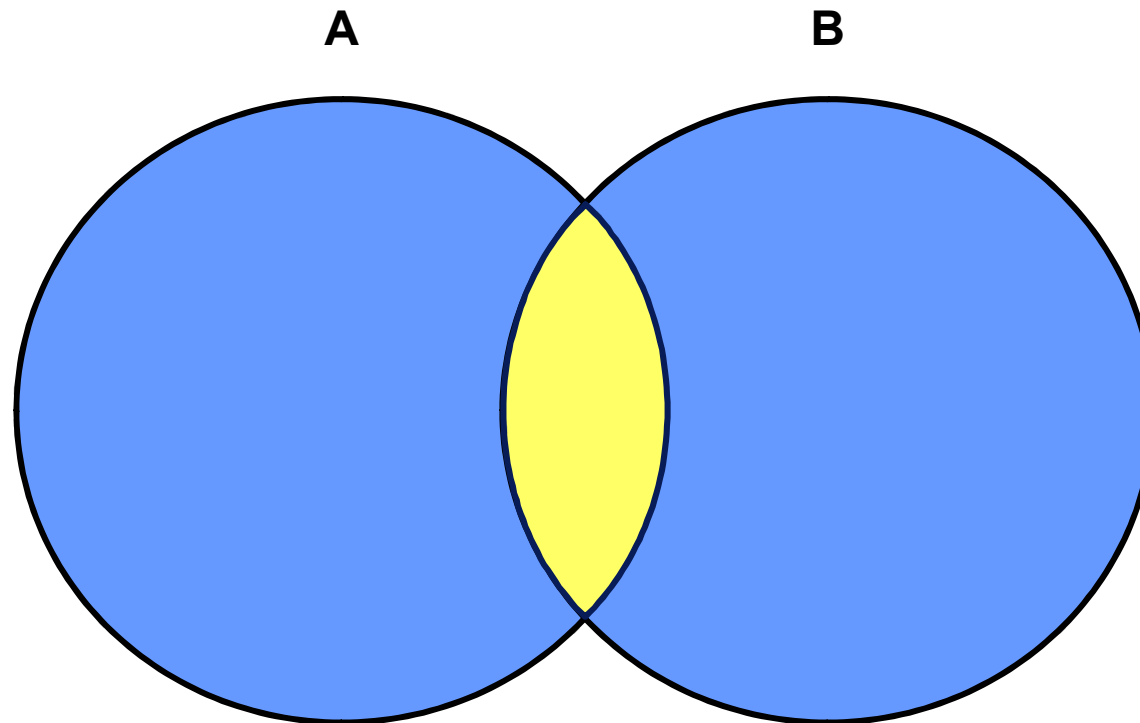
Using the UNION ALL Operator

- Display the current and previous departments of all employees.

```
SELECT order_id
FROM orders
UNION ALL
SELECT order_id
FROM order_items
ORDER BY order_id
```

	ORDER_ID
1	2354
2	2354
3	2354
4	2354
5	2354
...	
15	2355
16	2355
17	2355
18	2355
19	2355
20	2355

INTERSECT Operator



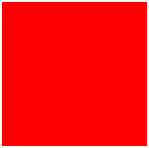
The `INTERSECT` operator returns rows that are common to both queries.

Using the INTERSECT Operator

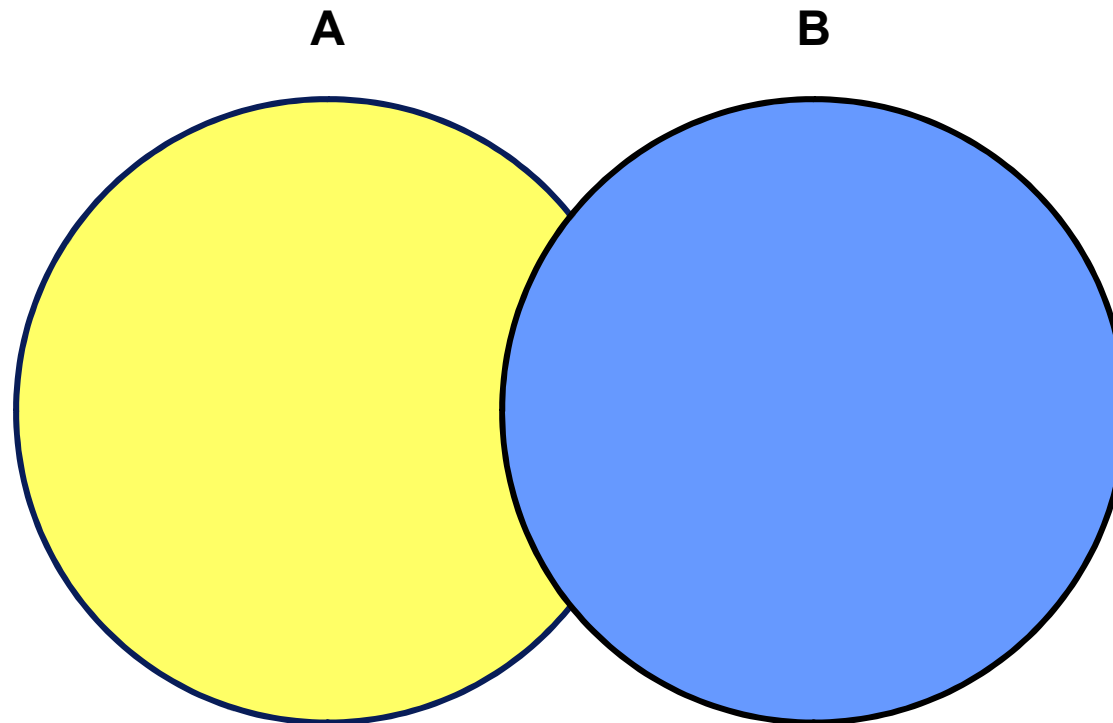
- Display the employee IDs and job IDs of those employees who currently have a job title that is the same as their previous one (that is, they changed jobs but have now gone back to doing the same job they did previously).

```
SELECT employee_id, job_id
FROM   employees
INTERSECT
SELECT employee_id, job_id
FROM   job_history;
```

	EMPLOYEE_ID	JOB_ID
1	176	SA_REP
2	200	AD_ASST



MINUS Operator



The MINUS operator returns all the distinct rows selected by the first query, but not present in the second query result set.

Using the MINUS Operator

- Display the employee IDs of those employees who have not changed their jobs even once.

```
SELECT order_id  
FROM orders  
MINUS  
SELECT order_id  
FROM order_items  
ORDER BY order_id;
```

	ORDER_ID
1	2360
2	2370
3	2373
4	2374
...	
16	2417
17	2425
18	2433
19	2441



Matching the SELECT Statements

- Using the UNION operator, display the location ID, department name, and the state where it is located.
- You must match the data type (using the TO_CHAR function or any other conversion functions) when columns do not exist in one or the other table.

```
SELECT location_id, department_name "Department",  
       TO_CHAR(NULL) "Warehouse location"  
FROM departments  
UNION  
SELECT location_id, TO_CHAR(NULL) "Department",  
       state_province  
FROM locations;
```

Matching the SELECT Statement: Example

- Using the UNION operator, display the employee ID, job ID, and salary of all employees.

```
SELECT order_id, product_id
FROM order_items
UNION
SELECT order_id, 0
FROM orders
ORDER BY order_id;
```

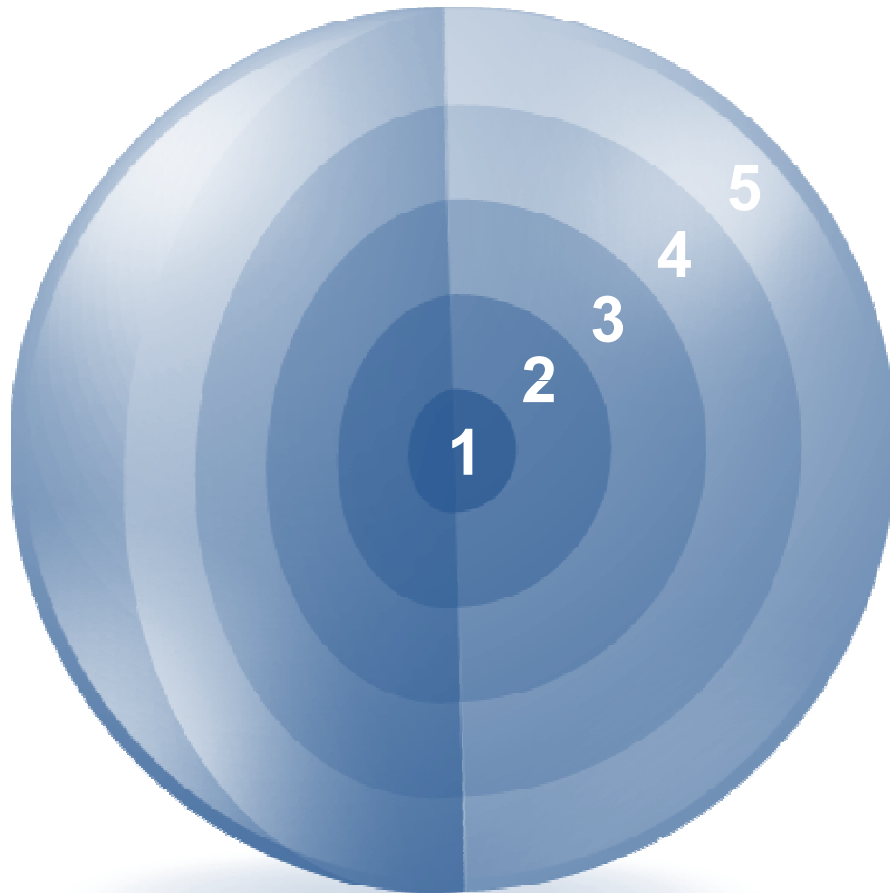
	ORDER_ID	PRODUCT_ID
1	2354	0
2	2354	3106
3	2354	3114
4	2354	3123
5	2354	3129
...		
56	2359	2381
57	2360	0
58	2361	0



Using the ORDER BY Clause in Set Operations

- The ORDER BY clause can appear only once at the end of the compound query.
- Component queries cannot have individual ORDER BY clauses.
- The ORDER BY clause recognizes only the columns of the first SELECT query.
- By default, the first column of the first SELECT query is used to sort the output in an ascending order.

- **Identify the set operator guidelines.**
 - 1.The expressions in the SELECT lists must match in number.**
 - 2.Parentheses may not be used to alter the sequence of execution.**
 - 3.The data type of each column in the second query must match the data type of its corresponding column in the first query.**
 - 4.The ORDER BY clause can be used only once in a compound query, unless a UNION ALL operator is used.**



1. **UNION** to return all distinct rows
2. **UNION ALL** to return all rows, including duplicates
3. **INTERSECT** to return all rows that are shared by both queries
4. **MINUS** to return all distinct rows that are selected by the first query, but not by the second
5. **ORDER BY** only at the very end of the statement

Practice 8: Overview

In this practice, you create reports by using:

