9

Using the Set Operators

Objectives

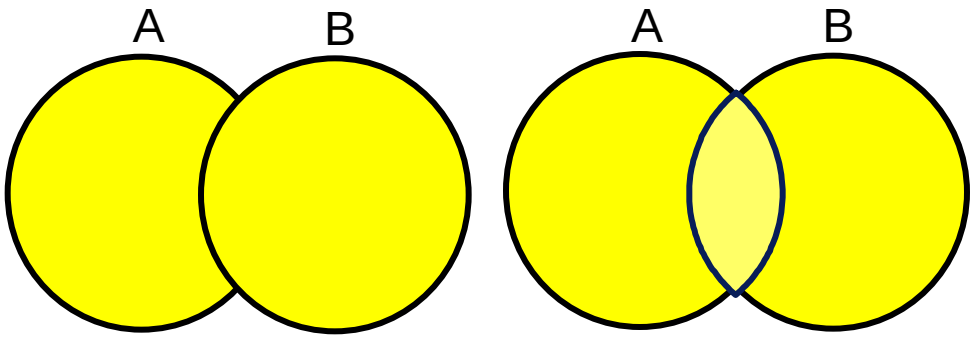
After completing this lesson, you should be able to do the following:

- Describe set operators
- Use a set operator to combine multiple queries into a single query
- Control the order of rows returned

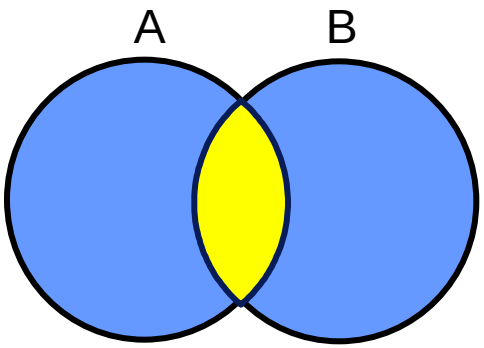
Lesson Agenda

- Set operators: Types and guidelines
- Tables used in this lesson
- `UNION` and `UNION ALL` operator
- `INTERSECT` operator
- `MINUS` operator
- Matching `SELECT` statements
- Using the `ORDER BY` clause in set operations

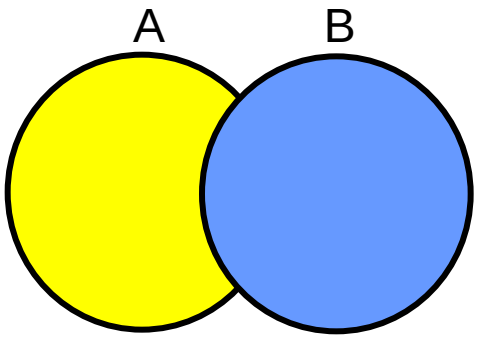
Set Operators



UNION/UNION ALL



INTERSECT



MINUS

Set Operator Rules

- The expressions in the SELECT lists must match in number.
- The data type of each column in the subsequent query must match the data type of its corresponding column in the first query.
- Parentheses can be used to alter the sequence of execution.
- ORDER BY clause can appear only at the very end of the statement.

Oracle Server and Set Operators

- Duplicate rows are automatically eliminated except in `UNION ALL`.
- Column names from the first query appear in the result.
- The output is sorted in ascending order by default except in `UNION ALL`.

Lesson Agenda

- Set operators: Types and guidelines
- Tables used in this lesson
- `UNION` and `UNION ALL` operator
- `INTERSECT` operator
- `MINUS` operator
- Matching `SELECT` statements
- Using the `ORDER BY` clause in set operations

Tables Used in This Lesson

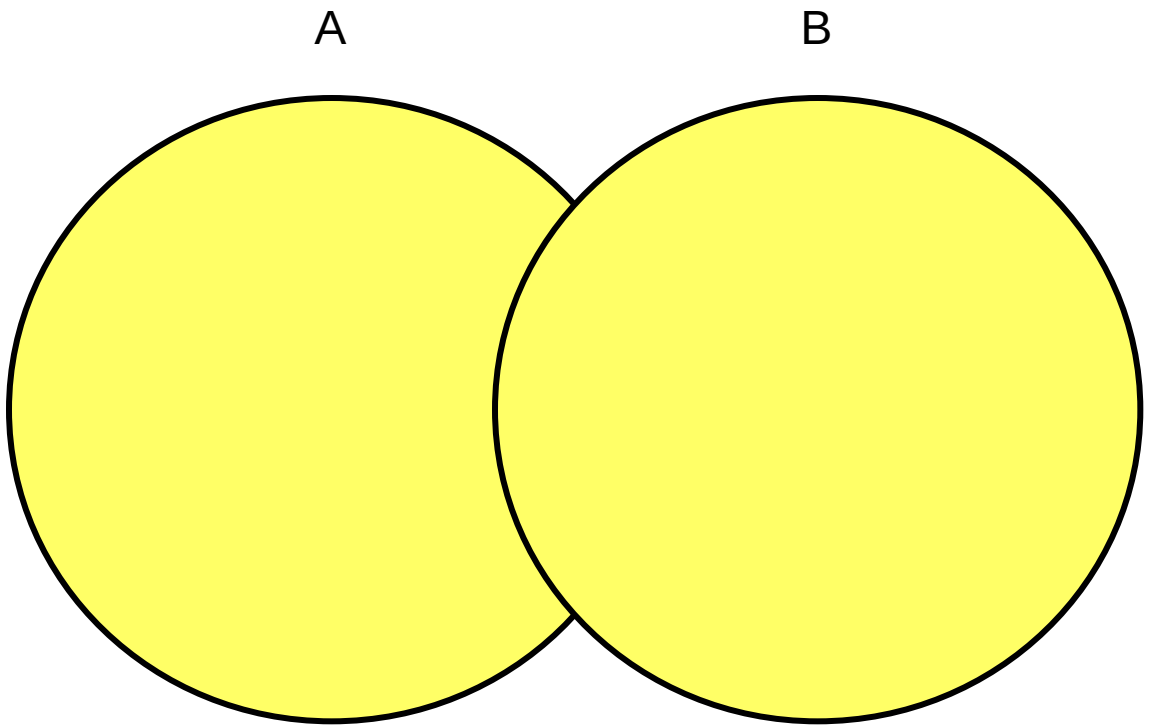
The tables used in this lesson are:

- `EMPLOYEES`: Provides details regarding all current employees
- `RETIRED_EMPLOYEES`: Provides details regarding all past employees

Lesson Agenda

- Set operators: Types and guidelines
- Tables used in this lesson
- `UNION` and `UNION ALL` operator
- `INTERSECT` operator
- `MINUS` operator
- Matching `SELECT` statements
- Using the `ORDER BY` clause in set operations

UNION Operator



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

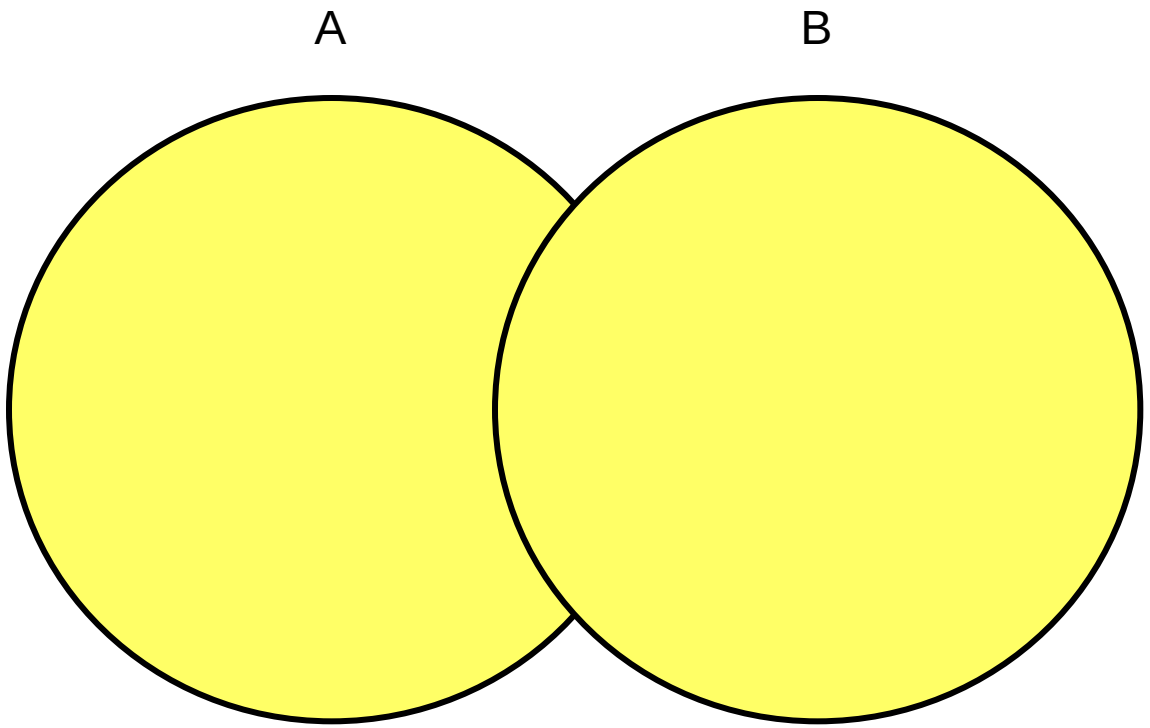
Using the UNION Operator

Display the job details of all the current and retired employees. Display each job only once.

```
SELECT job_id
FROM   employees
UNION
SELECT job_id
FROM   retired_employees
```

	JOB_ID
1	AC_ACCOUNT
2	AC_MGR
3	AD_ASST
4	AD_PRES
5	AD_VP
6	FI_ACCOUNT
7	FI_MGR
8	IT_PROG
9	MK_MAN
10	MK_REP
11	PU_CLERK
12	PU_MAN
13	SA_MAN
14	SA_REP
15	ST_CLERK
16	ST_MAN

UNION ALL Operator



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

Using the UNION ALL Operator

Display the jobs and departments of all current and previous employees.

```
SELECT job_id, department_id
FROM   employees
UNION ALL
SELECT job_id, department_id
FROM   retired_employees
ORDER BY  job_id;
```

	JOB_ID	DEPARTMENT_ID
1	AC_ACCOUNT	110
2	AC_MGR	110
3	AD_ASST	10
4	AD_PRES	90
5	AD_PRES	90
6	AD_VP	90
7	AD_VP	80
8	AD_VP	90
9	AD_VP	90

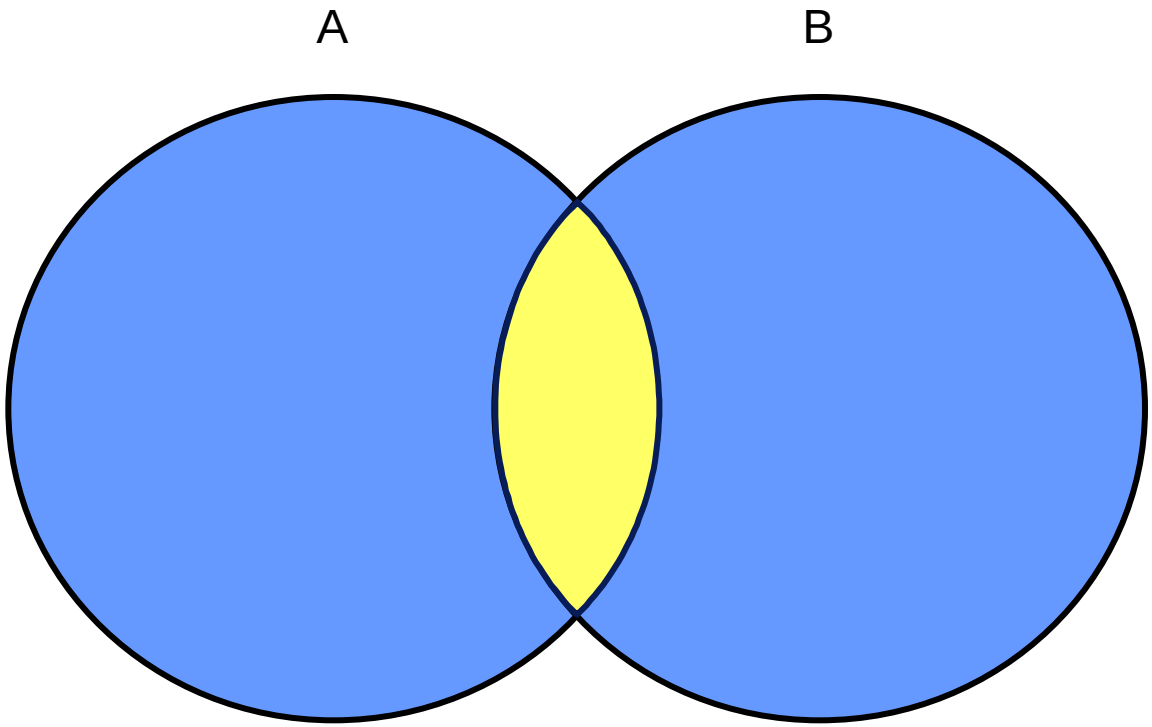
...

28	SA_REP	80
29	SA_REP	80
30	SA_REP	(null)
31	ST_CLERK	50
32	ST_CLERK	50
33	ST_CLERK	50
34	ST_CLERK	50
35	ST_MAN	50

Lesson Agenda

- Set operators: Types and guidelines
- Tables used in this lesson
- `UNION` and `UNION ALL` operator
- `INTERSECT` operator
- `MINUS` operator
- Matching `SELECT` statements
- Using `ORDER BY` clause in set operations

INTERSECT Operator



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

Using the INTERSECT Operator

Display the common manager IDs and department IDs of current and previous employees.

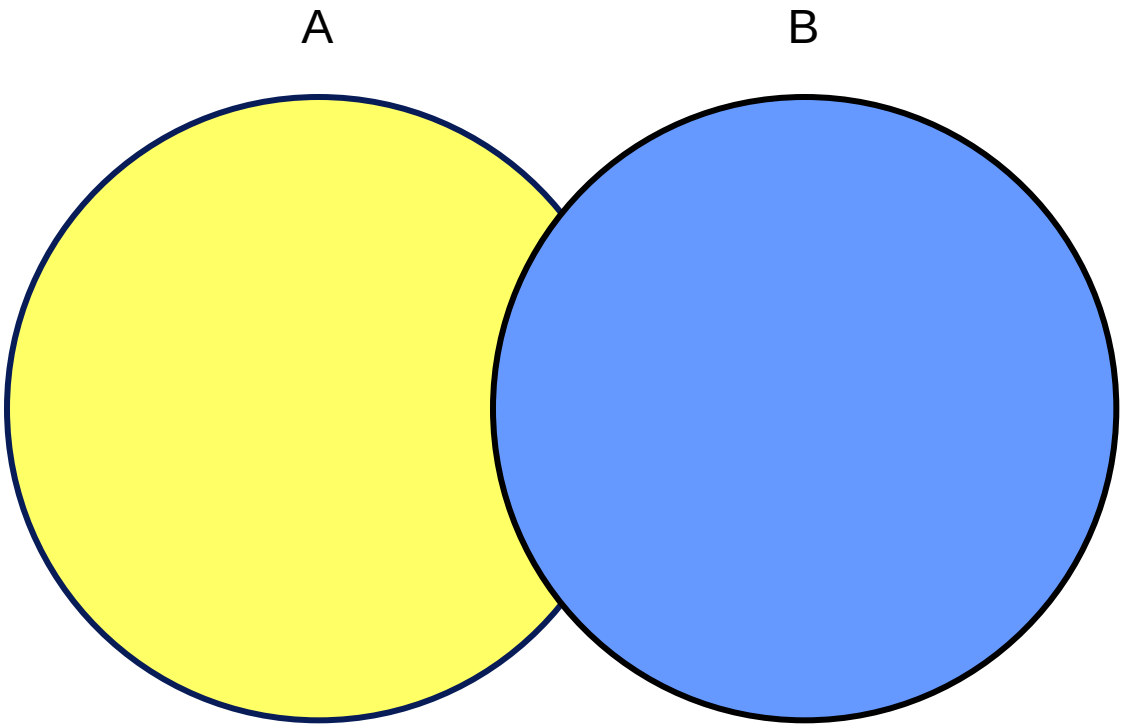
```
SELECT  manager_id,department_id
FROM   employees
INTERSECT
SELECT manager_id,department_id
FROM   retired_employees
```

	MANAGER_ID	DEPARTMENT_ID
1	149	80

Lesson Agenda

- Set operators: Types and guidelines
- Tables used in this lesson
- `UNION` and `UNION ALL` operator
- `INTERSECT` operator
- `MINUS` operator
- Matching `SELECT` statements
- Using the `ORDER BY` clause in set operations

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 and Job IDs of those employees who works in the sales department.

```
SELECT employee_id, job_id
FROM employees
WHERE department_id = 80
MINUS
SELECT employee_id, job_id
FROM retired_employees
WHERE department_id = 90;
```

	EMPLOYEE_ID	JOB_ID
1	149	SA_MAN
2	174	SA_REP
3	176	SA_REP

Lesson Agenda

- Set operators: Types and guidelines
- Tables used in this lesson
- `UNION` and `UNION ALL` operator
- `INTERSECT` operator
- `MINUS` operator
- Matching `SELECT` statements
- Using `ORDER BY` clause in set operations

Matching SELECT Statements

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 name, `department_id`, and `location_id` of all employees.

```
SELECT  FIRST_NAME, JOB_ID, TO_DATE(hire_date)"HIRE_DATE"
FROM employees
UNION
SELECT FIRST_NAME, JOB_ID, TO_DATE(NULL)"HIRE_DATE"
FROM retired_employees;
```

	FIRST_NAME	JOB_ID	HIRE_DATE
1	Alex	PU_CLERK	(null)
2	Alexander	IT_PROG	03-JAN-06
3	Alexandera	IT_PROG	(null)
4	Bruce	IT_PROG	21-MAY-07
5	Bruk	IT_PROG	(null)
6	Curtis	ST_CLERK	29-JAN-05
7	Dany	FI_ACCOUNT	(null)
8	Del	PU_MAN	(null)
9	Diana	IT_PROG	07-FEB-07

...

Lesson Agenda

- Set operators: Types and guidelines
- Tables used in this lesson
- `UNION` and `UNION ALL` operator
- `INTERSECT` operator
- `MINUS` operator
- Matching `SELECT` statements
- Using the `ORDER BY` clause in set operations

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.

Quiz

Identify two set operator guidelines.

a. The expressions in the SELECT lists must match in number.
b. Parentheses may not be used to alter the sequence of execution.
c. The data type of each column in the second query must match the data type of its corresponding column in the first query.
d. The ORDER BY clause can be used only once in a compound query, unless a UNION ALL operator is used.

Summary

In this lesson, you should have learned how to use:

- `UNION` to return all distinct rows
- `UNION ALL` to return all rows, including duplicates
- `INTERSECT` to return all rows that are shared by both queries
- `MINUS` to return all distinct rows that are selected by the first query, but not by the second
- `ORDER BY` only at the very end of the statement

Practice 9: Overview

In this practice, you create reports by using:

- The `UNION` operator
- The `INTERSECT` operator
- The `MINUS` operator