

8

Using the Set Operators

Objectives

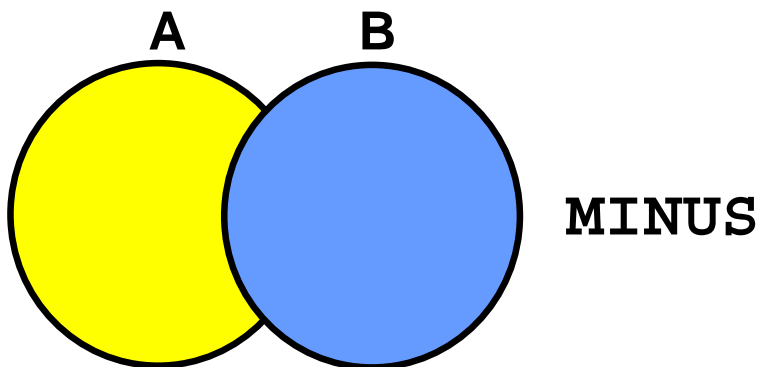
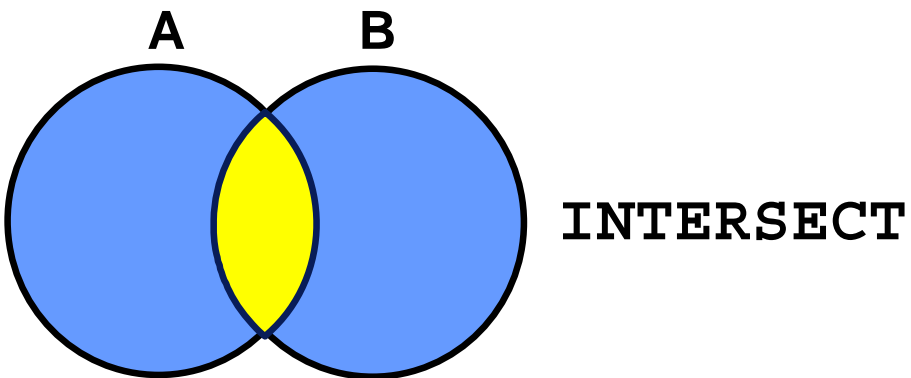
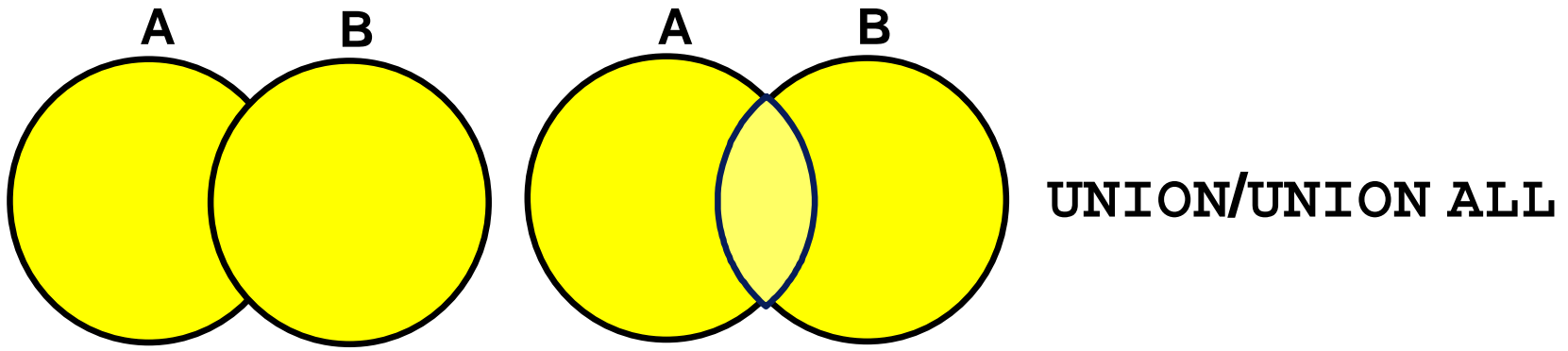
After completing this lesson, you should be able to do the following:

- 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 the SELECT statements
- Using the ORDER BY clause in set operations

Set Operators



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.
- 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 the SELECT statements
- Using the ORDER BY clause in set operations

Tables Used in This Lesson

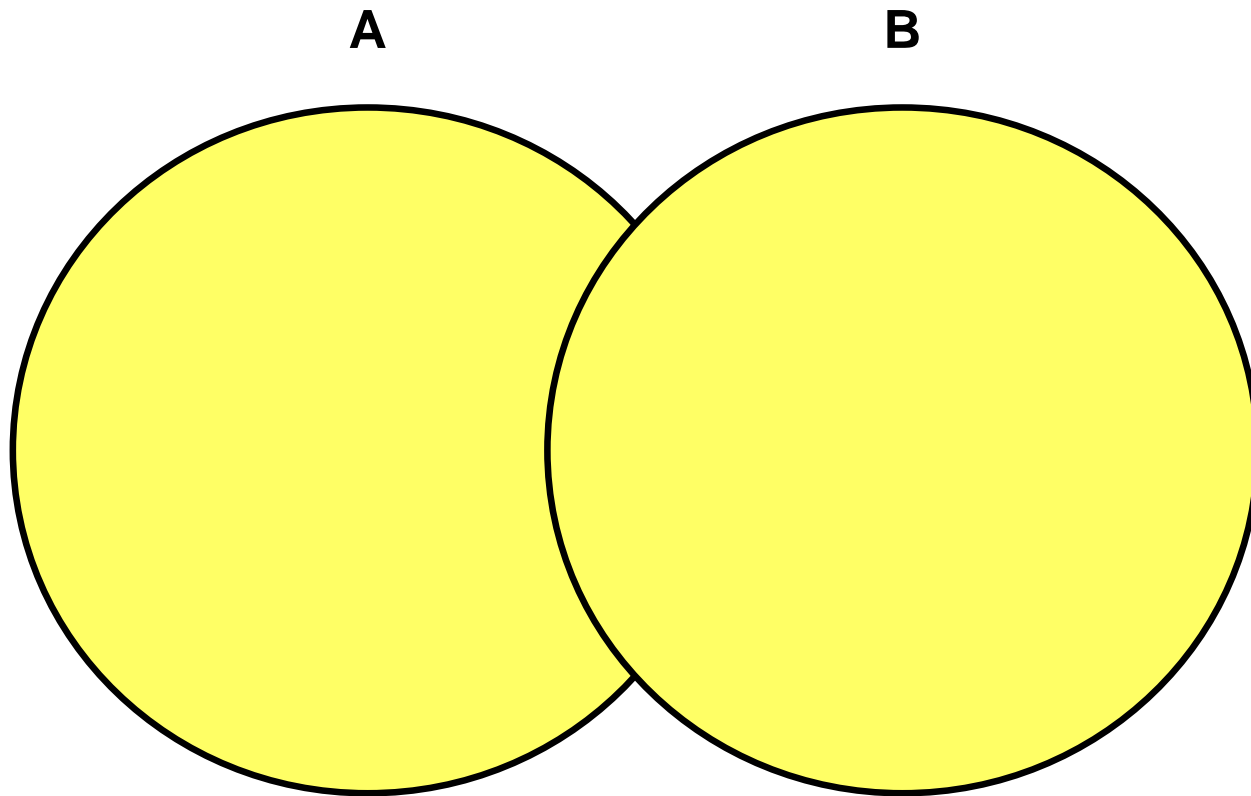
The tables used in this lesson are:

- **EMPLOYEES:** Provides details regarding all current employees
- **JOB_HISTORY:** Records the details of the start date and end date of the former job, and the job identification number and department when an employee switches jobs

Lesson Agenda

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

UNION Operator



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

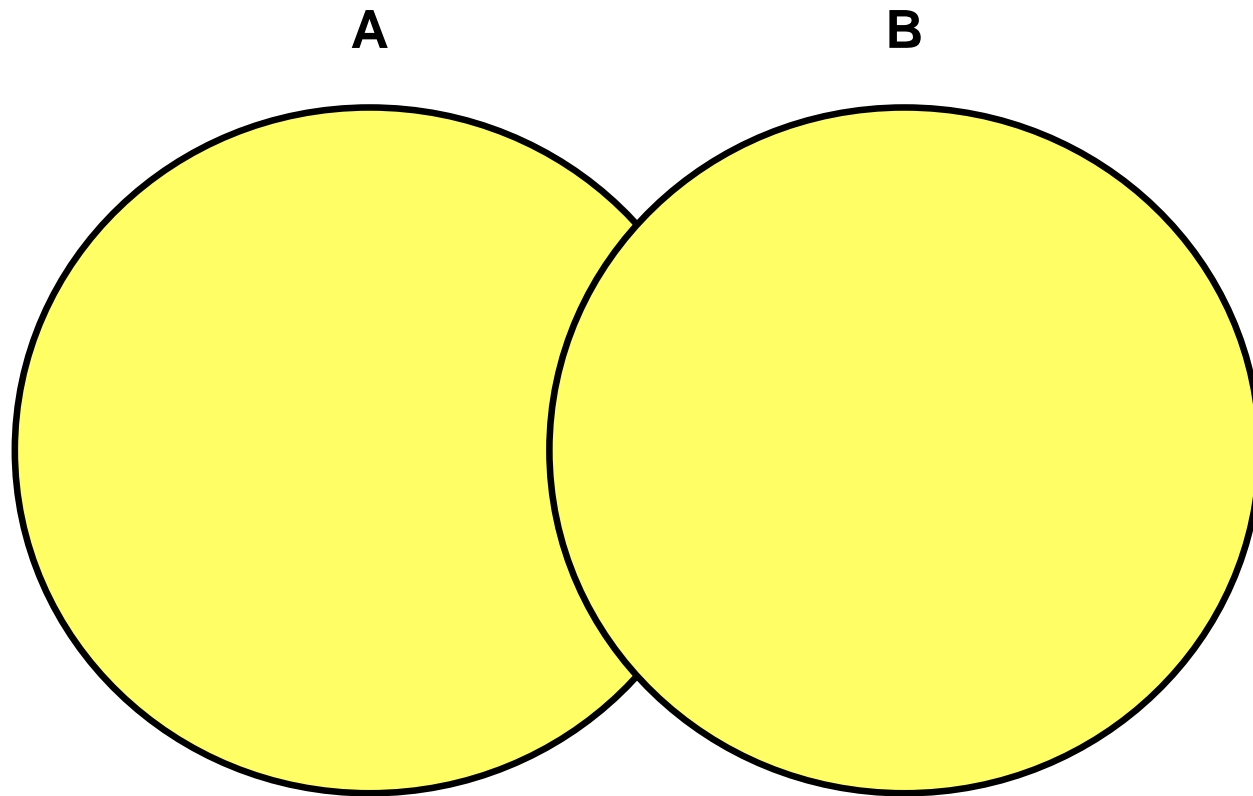
Using the UNION Operator

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

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

| | EMPLOYEE_ID | JOB_ID |
|-----|-------------|------------|
| 1 | 100 | AD_PRES |
| 2 | 101 | AC_ACCOUNT |
| ... | | |
| 22 | 200 | AC_ACCOUNT |
| 23 | 200 | AD_ASST |
| ... | | |
| 27 | 205 | AC_MGR |
| 28 | 206 | AC_ACCOUNT |

UNION ALL Operator



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.

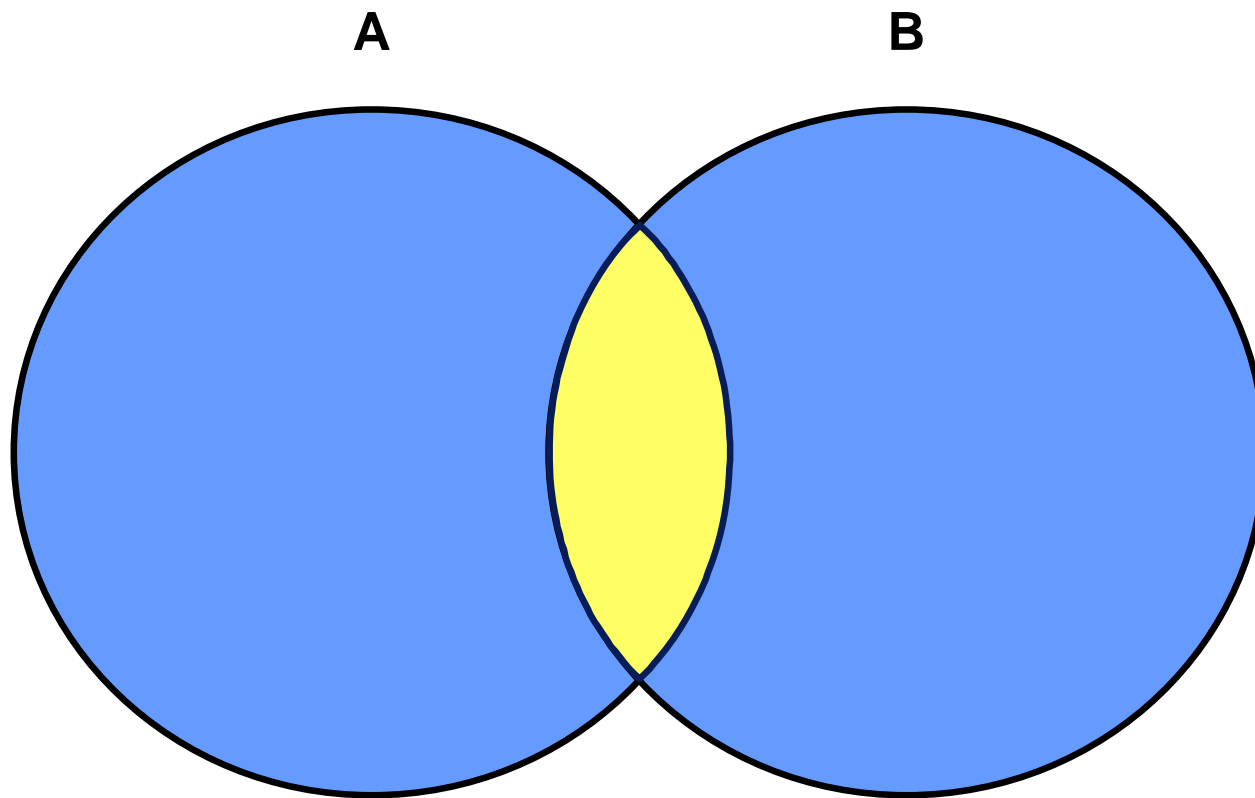
```
SELECT employee_id, job_id, department_id
FROM employees
UNION ALL
SELECT employee_id, job_id, department_id
FROM job_history
ORDER BY employee_id;
```

| | EMPLOYEE_ID | JOB_ID | DEPARTMENT_ID |
|-----|-------------|------------|---------------|
| 1 | 100 | AD_PRES | 90 |
| ... | | | |
| 17 | 149 | SA_MAN | 80 |
| 18 | 174 | SA_REP | 80 |
| 19 | 176 | SA_REP | 80 |
| 20 | 176 | SA_MAN | 80 |
| 21 | 176 | SA_REP | 80 |
| 22 | 178 | SA_REP | (null) |
| 23 | 200 | AD_ASST | 10 |
| ... | | | |
| 30 | 206 | AC_ACCOUNT | 110 |

Lesson Agenda

- Set Operators: Types and guidelines
- Tables used in this lesson
- UNION and UNION ALL operator
- **INTERSECT operator**
- MINUS operator
- Matching the 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 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).

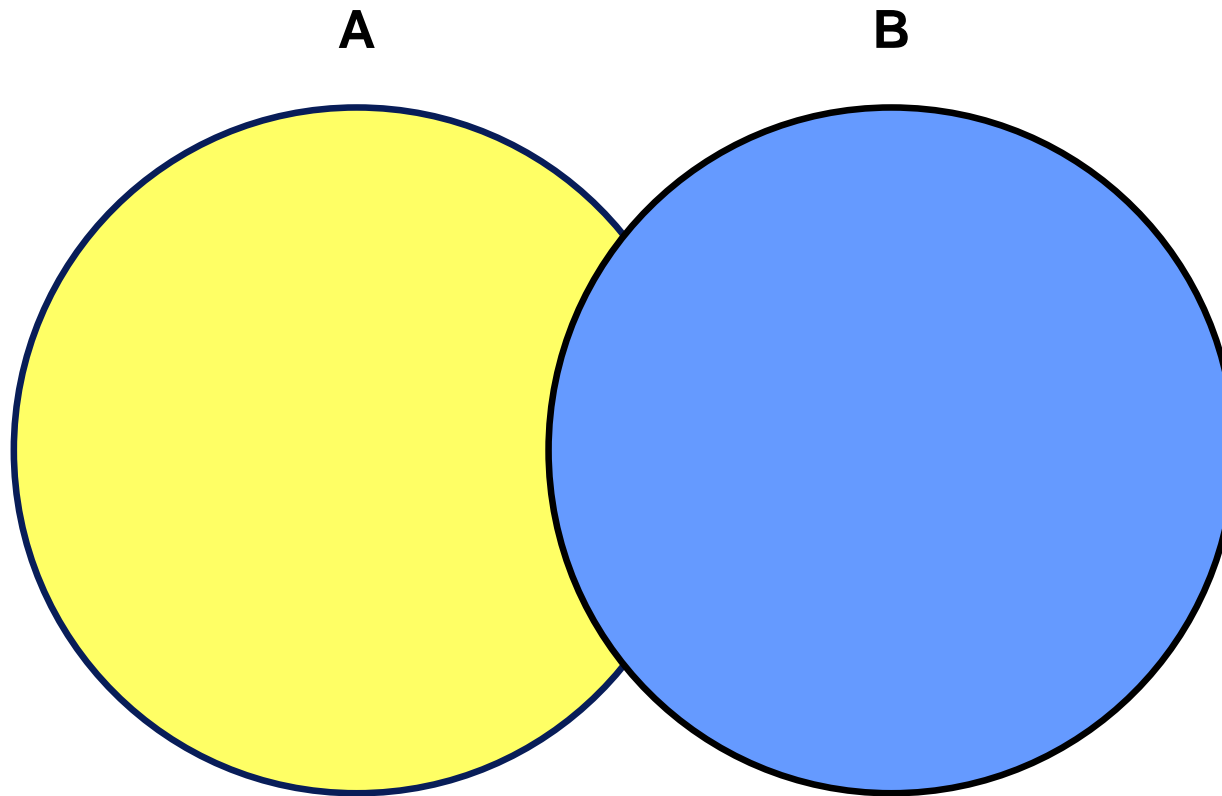
```
SELECT employee_id, job_id
FROM   employees
INTERSECT
SELECT employee_id, job_id
FROM   job_history;
```

| |  EMPLOYEE_ID |  JOB_ID |
|---|---|--|
| 1 | 176 | SA_REP |
| 2 | 200 | AD_ASST |

Lesson Agenda

- Set Operators: Types and guidelines
- Tables used in this lesson
- UNION and UNION ALL operator
- INTERSECT operator
- **MINUS operator**
- Matching the 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 of those employees who have not changed their jobs even once.

```
SELECT employee_id
FROM   employees
MINUS
SELECT employee_id
FROM   job_history;
```

| | EMPLOYEE_ID |
|---|-------------|
| 1 | 100 |
| 2 | 103 |
| 3 | 104 |

...

| | |
|----|-----|
| 13 | 202 |
| 14 | 205 |
| 15 | 206 |

Lesson Agenda

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

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 employee_id, job_id, salary
FROM   employees
UNION
SELECT employee_id, job_id, 0
FROM   job_history;
```

| | EMPLOYEE_ID | JOB_ID | SALARY |
|-----|-------------|------------|--------|
| 1 | 100 | AD_PRES | 24000 |
| 2 | 101 | AC_ACCOUNT | 0 |
| 3 | 101 | AC_MGR | 0 |
| 4 | 101 | AD_VP | 17000 |
| 5 | 102 | AD_VP | 17000 |
| ... | | | |
| 29 | 205 | AC_MGR | 12000 |
| 30 | 206 | AC_ACCOUNT | 8300 |

Lesson Agenda

- Set Operators: Types and guidelines
- Tables used in this lesson
- UNION and UNION ALL operator
- INTERSECT operator
- MINUS operator
- Matching the 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 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.

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 8: Overview

In this practice, you create reports by using:

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