EXERCISE-10

USING THE SET OPERATORS

Objectives
After the completion this exercise, the students 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

The set operators combine the results of two or more component queries into one result.

Queries containing set operators are called *compound queries*.

Operator	Returns
UNION	All distinct rows selected by either query
UNION ALL	All rows selected by either query, including all duplicates
INTERSECT	All distinct rows selected by both queries
MINUS	All distinct rows that are selected by the first SELECT statement and not selected in the second SELECT statement

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

UNION Operator

Guidelines

• The number of columns and the data types of the columns being selected must be identical in all the SELECT statements used in the query. The names of the columns need not be identical.

• UNION operates over all of the columns being selected.

• NULL values are not ignored during duplicate checking.

• The IN operator has a higher precedence than the UNION operator.

• By default, the output is sorted in ascending order of the first column of the SELECT clause.

Example:

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;

Example:
SELECT employee_id, job_id, department_id
FROM employees
UNION
SELECT employee_id, job_id, department_id
FROM job_history;

UNION ALL Operator

Guidelines

The guidelines for UNION and UNION ALL are the same, with the following two exceptions that pertain to UNION ALL:
• Unlike UNION, duplicate rows are not eliminated and the output is not sorted by default.
• The DISTINCT keyword cannot be used.
Example:
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;

INTERSECT Operator

Guidelines

• The number of columns and the data types of the columns being selected by the SELECT statements in the queries must be identical in all the SELECT statements used in the query. The names of the columns need not be identical.
• Reversing the order of the intersected tables does not alter the result.
• INTERSECT does not ignore NULL values.

Example:
Display the employee IDs and job IDs of those employees who currently have a job title that is the same as their job title when they were initially hired (that is, they changed jobs but have now gone back to
doing their original job).

SELECT employee_id, job_id FROM employees
INTERSECT
SELECT employee_id, job_id
FROM job_history;
Example
SELECT employee_id, job_id, department_id
FROM employees
INTERSECT
SELECT employee_id, job_id, department_id
FROM job_history;
MINUS Operator
Guidelines

• The number of columns and the data types of the columns being selected by the SELECT statements in the queries must be identical in all the SELECT statements used in the query. The names of the columns need not be identical.
• All of the columns in the WHERE clause must be in the SELECT clause for the MINUS operator to work.

Example:

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

SELECT employee_id,job_id
FROM employees
MINUS
SELECT employee_id,job_id
FROM job_history;

Find the Solution for the following:

1. The HR department needs a list of department IDs for departments that do not contain the job ID ST_CLERK. Use set operators to create this report.

```
SELECT dept_id FROM department
MINUS
SELECT department_id FROM employees
WHERE job_id = 'ST_CLERK';
```

	DEPT_ID
1	10
2	20
3	30
4	40

Download ▾ Execution time: 0.006 seconds

2. The HR department needs a list of countries that have no departments located in them. Display the country ID and the name of the countries. Use set operators to create this report.

```
SELECT country_id, country_name FROM countries
MINUS
SELECT l.country_id, c.country_name FROM locations l,
countries c WHERE l.country_id = c.country_id;
```

	COUNTRY_ID	COUNTRY_NAME
1	IN	India
2	UK	United Kingdom
3	AU	Australia
4	JP	Japan
5	DE	Germany
6	FR	France

Download ▾ Execution time: 0.134 seconds

3. Produce a list of jobs for departments 10, 50, and 20, in that order. Display job ID and department ID using set operators.

```
SELECT job_id, department_id, 'a' sort_order FROM employees
WHERE department_id = 10
UNION
SELECT job_id, department_id, 'b' FROM employees
WHERE department_id = 50
UNION
SELECT job_id, department_id, 'c' FROM employees
WHERE department_id = 20 ORDER BY sort_order;
```
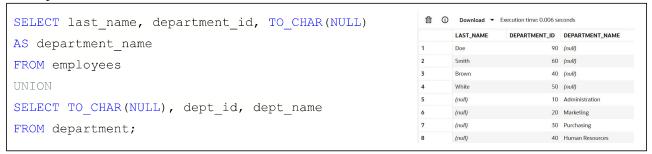
	JOB_ID	DEPARTMENT_ID	SORT_ORDER
1	SA_REP	50	b

Download ▾ Execution time: 0.007 seconds

4. Create a report that lists the employee IDs and job IDs of those employees who currently have a job title that is the same as their job title when they were initially hired by the company (that is, they changed jobs but have now gone back to doing their original job).

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

🗑 ⓘ	Download ▾	Execution time: 0.001 seconds

EMPLOYEE_ID	JOB_ID

No items to display.

5. The HR department needs a report with the following specifications:

- Last name and department ID of all the employees from the EMPLOYEES table, regardless of whether or not they belong to a department.

- Department ID and department name of all the departments from the DEPARTMENTS table, regardless of whether or not they have employees working in them Write a compound query to accomplish this.

```sql
SELECT last_name, department_id, TO_CHAR(NULL)
AS department_name
FROM employees
UNION
SELECT TO_CHAR(NULL), dept_id, dept_name
FROM department;
```

🗑 ⓘ	Download ▾	Execution time: 0.006 seconds

	LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
1	Doe	90	(null)
2	Smith	60	(null)
3	Brown	40	(null)
4	White	50	(null)
5	(null)	10	Administration
6	(null)	20	Marketing
7	(null)	30	Purchasing
8	(null)	40	Human Resources

Evaluation Procedure	Marks awarded
Query(5)	
Execution (5)	
Viva(5)	
Total (15)	
Faculty Signature	