LAB NO.: 3 **Date:**

INTERMEDIATE SQL

Objectives:

In this lab, student will be able to:

- Understand the set operations and intermediate level queries.

SET Operations in SQL:

Multiple queries using the set operators **UNION**, **UNION ALL**, **INTERSECT**, and **MINUS**. All set operators have equal precedence. If a SQL statement contains multiple set operators, then Oracle Database evaluates them from the left to right unless parentheses explicitly specify another order.

The corresponding expressions in the select lists of the component queries of a compound query must match in number and must be in the same data type group (such as numeric or character).

The **UNION** operator returns only distinct rows that appear in either result.

```
SELECT product_id FROM order_items
UNION
SELECT product_id FROM inventories;
```

The following statement combines the results with the **INTERSECT** operator, which returns only those rows returned by both queries:

```
SELECT product_id FROM inventories
INTERSECT
SELECT product_id FROM order_items;
```

The following statement combines results with the **MINUS** operator, which returns only unique rows returned by the first query but not by the second:

```
SELECT product_id FROM inventories
MINUS
SELECT product_id FROM order_items;
```

CREATE VIEW Statement

In SQL, a view is a virtual table based on the result-set of an SQL statement. A view contains rows and columns, just like a real table. The fields in a view are fields from one or more real tables in the database.

SQL CREATE VIEW Syntax

```
CREATE VIEW view_name AS
SELECT column_name(s)
FROM table_name
WHERE condition;
```

LAB EXERCISE:

Implement the following Queries on UNIVERSITY Database:

Set Operations

UNION (Use union all to retain duplicates):

1. Find courses that ran in Fall 2009 or in Spring 2010

INTERSECT (Use intersect all to retain duplicates):

2. Find courses that ran in Fall 2009 and in spring 2010

MINUS:

3. Find courses that ran in Fall 2009 but not in Spring 2010

Null values

4. Find the name of the course for which none of the students registered.

Nested Subqueries

<u>Set Membership (in / not in):</u>

5. Find courses offered in Fall 2009 and in Spring 2010.
6. Find the total number of students who have taken course taught by the instructor with ID 10101.
7. Find courses offered in Fall 2009 but not in Spring 2010.
8. Find the names of all students whose name is same as the instructor's name.

<u>Set Comparison (>=some/all)</u>

1. Find names of instructors with salary greater than that of some (at least one) instructor in the Biology department.
10. Find the names of all instructors whose salary is greater than the salary of all instructors in the Biology department.
11. Find the departments that have the highest average salary.
12. Find the names of those departments whose budget is lesser than the average salary of all instructors.

<u>Test for Empty Relations (exists/ not exists)</u>

13. Find all courses taught in both the Fall 2009 semester and in the Spring 2010 semester.
14. Find all students who have taken all courses offered in the Biology department.

<u>Test for Absence of Duplicate Tuples</u>

15. Find all courses that were offered at most once in 2009.
16. Find all the students who have opted at least two courses offered by CSE department.

<u>Subqueries in the From Clause</u>

17. Find the average instructors salary of those departments where the average salary is greater than 42000

Views

18. Create a view all_courses consisting of course sections offered by Physics department in the Fall 2009, with the building and room number of each section.
19. Select all the courses from all_courses view.
20. Create a view department_total_salary consisting of department name and total salary of that department.