**Introduction Oracle SQL and PL/SQL**

**Login to SCOTT schema and write queries for the following:**



**Module 1/Day 1**

**SELECT Statement:**

1. List tables in your schema and check for existence of DEPT, EMP and SALGERADE tables
2. If these tables do NOT exists – execute the script in the embedded DemoBld.SQL file to create and populate the tables .
3. List all columns and all rows from DEPT
4. List all columns and all rows from EMP
5. List all columns and all rows from SALGRADE
6. List employee number, name and salary from employee table
7. List employee number, name and salary from employee table where salary is > 3000
8. List employees joined after year 1981
9. List all clerks (JOB = ‘CLERK’)
10. List employees in the ascending order of salary
11. List employees in ascending order of job within descending order of deptno
12. List distinct departments from employee table
13. List distinct jobs in each department from employee table
14. List name, salary and annual salary in the descending order of annual salary – annual salary is a computed column – SAL \* 12
15. List employees whose salary is not in the range of 2000 and 3000
16. List name and the deptno for all employees who are NOT members of departments 10 and 20
17. List employees for whom COMM is not applicable
18. List employees for whom COMM is applicable
19. List employees in ascending order of COMM and note how NULLs are sorted
20. List employees whose names start with ‘’SMITH’
21. List employees whose name contain the ‘MI’
22. List employees whose name start with an \_ (underscore) char.
23. List all employees joined between two given dates.
24. List all clerks in deptno 10
25. List total/sum, maximum, minimum, average of salary from employee table
26. List average and count of commission of all employees in department 10
27. List department wise no of employees and total salary
28. List total salary Job wise within each department
29. List department wise total salary for deptno 10 and 20 only
30. List department wise total salary where total salary is > 6000
31. SELECT COUNT(\*), COUNT(COMM) FROM EMP; - explain why the two counts are different

**Sub Queries:**

1. List employees whose job is same as that of ‘SMITH’
2. List employees who have joined after ‘ADAM’
3. List employees who salary is greater than ‘SCOTT’s salary
4. List employees getting the maximum salary
5. List employees show salary is > the max salary of all employees in deptno 30
6. List all employees whose deptno and Job are same as that of employee with empno 7788.
7. List employee who are not managers
8. List all managers
9. List all employees who earn(salary) more than the average salary in their own department
10. List employees whose salary is greater than their manager’s salary
11. List details of departments from DEPT table for which there are no employees in EMP table

**Module 2/Day 2**

**Joins:**

1. List employee name, department number and their corresponding department name by joining EMP and DEPT tables
2. List employee name and their manager name by joining EMP table to itself
3. List employee name, department name and their grade by joining EMP, DEPT and SALGRADE tables
4. List employees who work in ‘Research’ department by joining EMP and DEPT tables
5. List all rows from EMP table and only the matching rows from DEPT table – LEFT OUTER JOIN
6. List only matching rows from EMP table and all rows from DEPT table – RIGTH OUTER JOIN
7. Write a query to perform full outer join between EMP and DEPT tables
8. List employee name, their manager name and their manager’s manager name

**DDL**

1. Create DEPARTMENT table with the following columns with appropriate data type and width
   1. Deptno PK
   2. Danme
   3. Location
2. Create EMPLOYEE table with the following columns with appropriate data type and width
   1. empno PK
   2. ename not null
   3. designation
   4. sex
   5. basic\_salary (> 0 and < 500000)
   6. Date of joining
   7. Deptno reference deptno of DEPARTMENT table

3. Alter table EMPLOYEE add column commission

4. Alter table EMPLOYEE add constraint SEX in (‘M’, ‘F’)

4. Create Index on ename column of EMPLOYEE table

5. Create exact replica of EMPLOYEE table with no data

6. Create new table called EX\_EMP with columns empno, ename, basic\_salary and populate the data from EMPLOYEE table

7. Drop the Index created on ename column.

**DML**

1. Insert at least 5 valid rows into DEPARTMENT table and commit the changes
2. Insert at least 15 valid rows in EMPLOYEE table and commit the changes
3. Update basic\_salary by 10% for employees in deptno 10 and 20 and commit the changes
4. Update basic\_salary and commission by 10% and 2% for all employees for whom commission is currently applicable and commit the changes
5. Update the designation of given employee to MANAGER based of given employee number and commit the changes.
6. Delete employees joined before a given year and commit changes
7. Delete all rows from employee tables.
8. Query employee table
9. ROLLBACK
10. Query employee table
11. Delete all rows from employee tables permanently using appropriate DDL command

**Procedure and Functions:**

1. Create a procedure named DISP\_EMP\_DETAILS with 3 parameters 🡪 one [ie. iEmpNo ] is an “IN” mode and other two are [ie. sGrade and sSalary] “OUT” mode parameters. The procedure should retrieve the Grade (this is column of SALGRADE table) and Salary for the specified employee number [ie. iEmpNo ] by joining EMP and SALGRADE table and assign the retrieved values to the “OUT” mode parameters.

Note.:

It should display an appropriate error message if the specified employee number

does not exist in “Employee” table.

Call the created procedure using Bind variable and print the details.

1. Create a procedure named DISPLAY\_RECORDS which accepts the P\_JOB as a parameter and display all the employees (empno, sal, deptno, job) from the “EMP” table matching the given P\_JOB in the following format :-

EmployeeNumber Salary DepartmentNumber Job

XXXXXXXX 99,999 99 CLERK

XXXXXXXX 9,999 12 CLERK

Note.:

It should display an appropriate error message if there are no employees with the given JOB

1. Create a function named GET\_EMP\_ANNSAL which accepts employee Number as

a parameter and Returns Annual Salary of the given employee from EMP table if the record exist otherwise returns -1 ( formula to computer ANNUAL\_SALARY = SAL \* 12)