**DBMSL Problem Statement for Mock Test 1**

**1. Design and Develop SQL DDL statements which demonstrate the use of SQL objects such as Table, View, Index, Sequence.**

1 Create & display database Called EMP\_DETAILS .

**2** Create table EMP( empId, empName, empSal, EmpDeptId) &

DEPT(deptId , deptName varchar(10) ,LOC varchar(10)) &

Set primary key to deptId from DEPT relation & forigian key to EmpDeptId)

3 Describe table EMP & DEPT.

4 Insert few records. & display records of EMP AND DEPT table.

5 Add column emp\_joiningdate to EMP relation & describe EMP table.

6 Change the datatype and size of column empsal from EMP relation.

7 Create view on DEPT table AS DEPT\_VIEW having all records and fields.

8 Create view on EMP table as emp\_view having empId,empName and emp\_joiningdate with all records.

9 Update the empId to 100 having name 'ABC' of emp\_view.

10 Create index IND1 on DEPT table on field deptName. And show Index.

11 Create Index on empName and empsal in EMP table and display index.

12 Drop view emp\_view.

13 Rename table EMP to Employee.

14Truncate table Employee.

**2. SQL queries using SQL DML statements: Insert, Select, Update, Delete with operators, functions, and set operator.**

Consider two relations.

**Emp( empId,empName,empSal,empDeptId)**

**Dept(deptId int, deptName varchar(10))**

1 .Create database ASS\_B1

2 Create table Emp and Dept

3 Set appropriate primary key and Foreign key.

4. Insert few Record.

5.Display information of those emp whose salary is greater than 20000.

6.List the employee details in the ascending order of their basic salary

7.List employees belonging to department 30, 40, or 10

8.List the employee details whose salary is between 10000 to 30000.

9.List the employee details whose name starts with A and salary is greater than 20000.

10.List the names of emp whose are working in 'Computer' department.

11.Change deptName from 'Computer' to 'Computer Engineering'.

12.List the total salary, maximum and minimum salary and average salary of the employees, for department 30.

13.List average sal of each deptID.

**3. SQL queries using SQL DML statements: all types of Join, Sub-Query and View.**

1.Create following Tables

Emp(Emp\_name,City,EmpId)

Emp\_salary(Emp\_name,Deptname ,Salary)

Emp\_Details(EMPNO,ENAME,JOB,HIREDATE,SAL,DEPTNO)

Display all records.

1.List the employee details along with Deptname and salary ,city using left outer join & right outer join and inner Join. In Emp and Emp\_salary relation.

2.Create view EMP\_SALARY1 on Emp\_salary table containing Emp\_name,Deptname.

3 Display records of employess working in smith's dept using EMP\_Details relation.

4.Disply salary of employee whose salary is more than salary of FORD using EMP\_Details relation.

5.Display records of employee who are senior to JONES using Emp\_Details relation.

6.Create view Emp\_View on Emp relation contaianing Emp\_name and City.

7.Update city in Emp\_View whose EmpId id 10.

8.Display view Emp\_View.

**4. PLSQL stored procedure:**

Consider the following schema and write a stored procedure to find those customers who have taken a loan. Categorize these loan borrowers as critical, moderate and nominal based on their loan amount. Insert all the details of borrowers along with the category in borrower\_category table.

Customer(Cust\_name, AccNo, Balance, city)

Loan(Loan\_no, branch\_name, Amount)

Borrower(Cust\_name, Loan\_no)

borrowr\_category(Cust\_name,AccNo,Loan\_no,branch\_name,amount,category)

**5. PLSQL Cursors:**

Consider the following schema and write a stored procedure to find details of customers who have not taken a loan. Use cursor to find which of these customers have balance more than 10,000 and insert their details in loan\_eligibility table.

Account (Acc\_no, branch\_name, balance)

Customer (Cust\_name, Acc\_no)

Borrower(Cust\_name, Loan\_no)

**6. PLSQL stored procedure & functions:**

Consider the following schema and write a PLSQL function that accepts customer name and returns the credit score for that customer. Let the credit score be computed on a scale of 0 to 5 by a stored procedure based on the account balance and loan amount of the customers. (hint: if loan\_amount is 50% of account balance then, credit\_score: 3)

Customer(Cust\_name, AccNo, Balance, city)

Loan(Loan\_no, branch\_name, Amount)

Borrower(Cust\_name, Loan\_no)

**7. Triggers:**

Consider a Employee database and write row-level triggers for the same. Implement both before & after triggers for the relevant database tables at the time of insertion, update and deletion. Implement using following schema:

**EMPLOYEE**(Emp\_Id, First\_Name, Last\_Name, Email, Phone\_No, Hire\_Date, Job\_Profile, Salary, HRA)

**COMPANY\_INFO**(Emp\_Count, Salary\_Expenses)

**EMP\_LOG**(Emp\_Id, Old\_Salary, New\_Salary, Edit\_Time, Job\_Status)

* **Before insert:** Check the column value of FIRST\_NAME, LAST\_NAME, JOB\_ID for following criteria:
* If there are any spaces before or after the FIRST\_NAME, LAST\_NAME, use TRIM() function to remove them.
* The value of the JOB\_PROFILE will be converted to upper cases by UPPER() function.
* **After insert:** Every time an INSERT happens into EMPLOYEE table, insert relevant information into the EMP\_LOG table. Also update the COMPANY\_INFO table
* **Before update:** Each time the HRA is updated for the EMPLOYEE table, convert it into decimal value (i.e. for 10%, store 0.1)
* **After Update:** Each time the HRA is updated, accordingly update the salary in EMPLOYEE table & keep track of updated salary in EMP\_LOG table.
* **Before Delete:** Every time a DELETE happens on EMPLOYEE table, accordingly change the JOB\_STATUS in EMP\_LOG table from ACTIVE to DELETED & keep track of EDIT\_TIME
* **After Delete:** Keep the COMPANY\_INFO table updated