**Database Systems**

**Fall 2020**

**LAB – 10**

Hope you are fine and feeling yourself comfortable and exciting to play with Structured Query Language (SQL) in this first lab. You have been taught in the lecture about SQL and query writing. Why not to get all what you have learned with a hands-on experience? So, let’s Start!

**The objective of this lab is to:**

1. PL/SQL, Cursor, Procedure, Functions.

**Course & Lab Instructor:** Sir Asif Sohail

Instructions:

* Gossips are not allowed. So be gentle and do what you know. The lab is not to deduct your sessional marks but to prepare you to achieve good marks in quizzes, mids and finals and finally have good grades. So, try to perform all your tasks in time and at your own.
* Teacher assistants are for your help, so be nice with them. Respect them as they are teaching you. Raise your hands if you have some problem and need help from TA.
* Avoid calling them by raising your voice and disturbing the environment of Lab.
* You must revise the content of the past lectures before starting the lab, it will help you resolve most of your general queries and give you the confidence that you can do it.
* Evaluation will be considered final and you cannot debate for the marks. So, focus on performing the tasks when the time is given to you.
* TA may deduct your marks for any kind of ill-discipline or misconduct from your side.
* Evaluation of tasks will be conducted in lab.
* Anyone caught being indulged in the act of plagiarism would be awarded an “F” grade in lab.
* Finally, pray before you start. And, Best of Luck!
* **Max Time: 50mins.**

**Instructions:**

1. This is an individual lab, you are strictly **NOT** allowed to discuss your solution with your fellow colleagues, even not allowed asking how is he/she is doing, it may result in negative marking. You can **ONLY** discuss with your TAs or with me.
2. Your TAs will be available from 1:00pm to 4:00pm for help. Alternatively, you can send

**Task 01: ­­Loop, Record, Exception [2 Marks]**

1. **Display a message *“Avengers Assemble”* 6 times using a loop. (0.5)**
2. **Using records display the name and job of employee with emp no 7536. Your output should be *“King works as a President”.* (0.5)**
3. **Implement exception handling for multiple rows and no data by displaying message as *“No data found”* or *“Multiple rows returned”* using if conditions. (1)**

**Task 02: Cursor [5 Marks]**

1. **Using cursor display employee data who work as CLERKS in the form *“SMITH is a CLERK”*. (1)**
2. **Using cursor display DEPT table data. Remember that DEPT table has three columns DEPTNO, DNAME and LOC. Data must be displayed in format “Deptno: 10, Dname: ACCOUNTING, Loc: NEW YORK”. (2)**
3. **Using cursor display SALGRADE table data. Remember that SALEGRADE table has three columns GRADE, HISAL and LOSAL. Data must be displayed in format “1->1200->700”. (2)**

**Task 03: Procedure, Function [13 Marks]**

1. **Create a procedure named ADD which takes two numbers and displays their sum. (2)**
2. **The above procedure displays the sum. Alter the above procedure in a way that it returns the sum as output in a variable. Use the output variable and display it yourself. (2)**
3. **Create a PL/SQL procedure SALARY\_SHEET that receives *empno* as a parameter and prepares the salary sheet of an employee as per the following rules:**

* **Medical Allowance: 10% of the salary**
* **House Rent: 25% of the salary**
* **Conveyance Allowance: 15% of the salary**
* **Tax Deductions: 12% of the salary**

**The procedure should output the employee’s name, job, computed allowances, tax deductions, and net salary(salary after adding all allowances and subtracting Tax) all values in different variables.**

***You need to make variables to hold all these values* (5)**

1. **Create a PL/SQL procedure FIND\_MAX () that displays the highest salary and the name of the employee receiving the highest salary. (2)**
2. **Create a function named ANNUAL\_SALARY that takes salary from table EMP and returns it after multiplying salary with 12. After creating this function, test it on the EMP table. (1)**
3. **Create a function named NAME\_CHANGER that takes ename from table emp and returns it after changing the ‘E’ in all names with ‘I’. After creating this function, test it on the EMP table. (1)**

**Good Luck 😉**