Q1. **[9 Marks] (40 minutes)**

Create a package pkg\_sal with private function as **sal\_calc** and a public procedure **show\_sal** as follows:

**Private Function sal\_calc** will accept employee number and return 1 if

employee number is valid else 0 if not valid.

**Public Procedure show\_sal** will accept employee number of individual.

Procedure should call the function **sal\_calc** and if employee number is valid it should find the total and average salary paid to that department to which employee belongs.

In case function returns 0, procedure should display the appropriate error message by handling the error.

In case function returns 1, procedure should display the information in the following format:

**================================================**

Employee Number:\_\_\_\_\_\_\_\_\_\_Name of the employee:\_\_\_\_\_\_\_\_\_\_\_\_

Department Number: \_\_\_\_\_\_\_\_\_\_

**================================================**

Total Salary Paid in the department: \_\_\_\_\_\_\_\_

Average Salary Paid in the department: \_\_\_\_\_\_\_\_

Minimum Salary paid in the department:\_\_\_\_\_\_\_\_\_

Maximum Salary paid in the department : \_\_\_\_\_\_\_\_\_\_

**================================================**

**Marks Distribution:**

|  |  |  |
| --- | --- | --- |
| **1** | Checking Employee number in function | **2** |
| **2** | Calling function | **1** |
| **3** | Displaying the output | **2** |
| **4** | Handling error | **2** |
| **5** | Package Creation | **2** |

**Q2. [9 marks] (40 Minutes)**

Create a procedure **Show\_Details** which will take Mgr as input and do the following validations:

1. If mgr doesn’t exists then handle the Predefined named exception with appropriate error msg.
2. If mgr exists then display the details in the following way. If commission is NULL then it should be displayed as 0.

**Magaer Id:\_\_\_\_\_\_\_\_**

**Manager Name:\_\_\_\_\_\_\_\_\_**

**Depatment Name:\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Sr No Employee ID Employee Name Hiredate Commission**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**…. ….. …… ….**

**… ….. …… ….**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Marks Distribution**

|  |  |  |
| --- | --- | --- |
| **1** | Procedure definition | **1** |
| **2** | Validating Mgr | **2** |
| **3** | Displaying details in formatted Outpue | **4** |
| **4** | Exception Handling | **2** |

**Q3 [9 marks] (40 Minutes)**

Write a function GET\_Anual\_Salary which accepts Dept\_no as input & returns annual Salary for that dept.

There should exception handling the error with proper message if wrong department no has been entered by the user.

Write an anonymous block to call the function and display the output in the below format.

**Dept Id:\_\_\_\_\_\_**

**Dept Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Annual Salary:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Marks Distribution:**

|  |  |  |
| --- | --- | --- |
| 1. | Fetch Dept name | 1 |
| 2. | Function definition | 4 |
| 3. | Error Handling | 2 |
| 4 | Calling function from anonymous block and displaying the output | 2 |

**Q4.** **[10 Marks] (50 Minutes)**

Create a Package **pkg\_Show\_details** which has two overloaded procedures **prEmpDetails** as given below :

1. Public procedure **prEmpDetails** accepts **salary** as parameter.

The procedure should display the information (empno, ename, sal, deptno) for all

the employees who are getting salary less or equal to as entered via parameter.

1. Public procedure **prEmpDetails** accepts **Hire Date of employee** as parameter. The procedure should display the details (empno, ename, sal and deptno) of all the employees for those who have joining on or after the date entered by the user as parameter.

Invoke the first procedure using anonymous block. And the second procedure from host environment.

Note: Marks will be given only if the procedures are packaged procedures

**Marks Distribution:**

|  |  |  |
| --- | --- | --- |
| **1** | Package Creation | **2** |
| **2** | Procedure 1 | **3** |
| **3** | Procedure 2 | **3** |
| **4** | Invoking the procedures | **2** |

**Q5. Book Order Related details need to be stored as given below**

|  |  |  |
| --- | --- | --- |
| Book\_Code (P.K) | Order\_Id | Amount |
| **B101** | **1** | **2000** |
| **B102** | **4** | **2500** |
| **7** | **3070** |
| **B103** | **3** | **5850** |
| **2** | **7400** |

Define appropriate datatype, if required, and create the table capable of holding the above like values. Insert records for at least 2 parts. For each part provide details of at least 2 orders.

Write a procedure **updateOrder** which accepts Book\_Code, Order\_Id and Amount as parameter and update the same in the table

**Marks Distribution**

|  |  |
| --- | --- |
| **Table creation,defining appropriate data types** | **4** |
| **Inserting records** | **2** |
| **Procedure definition** | **3** |