**Lab Cycle – 1**

Problem 1

Aim: To define, manipulate and control data using suitable Structured Query Language (SQL) statements for the given database schema.

Problem 2

Aim: To implement SQL-DDL statements for adding and removing various database integrity constraints during and after database creation.

Problem 3

Aim: To write SQL Select Statements for the following simple queries that retrieve data from a single table.

Problem 4

Aim: To write SQL queries for retrieving data from multiple tables using Joins

Problem 5

Aim: To write SQL Select Statements using Aggregate Functions, Group By and Having clauses for the following queries that retrieve data from university database.

Problem 6

Aim: To write nested queries for answering the following queries that retrieve data from university database:

**Lab Cycle – 2**

Aim: To write Select statements for answering the following queries using SQL single row functions

**Lab Cycle – 3**

Problem 1

Aim: To write a PL/SQL block to check whether a given number is a palindrome or not.

Problem 2

Aim: To write a PL/SQL Program that prints the first n fibonacci numbers.

Problem 3

Aim: To write a PL/SQL block which will find out LCM and HCF of two numbers and store the result in a database table.

Problem 4

Aim: Consider the following relation schemas

Inventory

|  |  |  |
| --- | --- | --- |
| Product\_ID | Product\_name | Quantity |

Purchase\_Record

|  |  |  |
| --- | --- | --- |
| Product\_ID | Status | Date |

Write a PL/SQL block to read the quantity of a product from inventory and if it is > 0 reduce the quantity by 1 and record the status of purchase of that product as ‘PURCHASED’. Otherwise record the status of purchase of that product as ‘OUT OF STOCK’. While recording the status of a purchase, record the date of transaction.

Problem 5

Aim:

Create a table employee with eno, ename, and basic\_pay attributes, insert 3 to 4 records and write a PL/SQL block to calculate the Gross salary & Net salary for an employee for the following conditions:

* + - HRA is 15% of basic.
    - DA is 62% of basic.
    - PF is 780/- if gross salary exceeds 8000, otherwise 600/-.
    - Professional tax is 2% of basic.

and then print the employee no, name, hra, da, pf, ptax, gross salary & net salary for that employee.

Problem 6

Aim:

Consider the following relation schemas

Emp

|  |  |  |  |
| --- | --- | --- | --- |
| empid | name | salary | dno |

Del\_History

|  |  |  |
| --- | --- | --- |
| dno | Rows\_deleted | date |

Write a PL/SQL block to delete records of all employees who belong to a particular department and then record the dno, no of rows deleted and date on which deletion occurred in the Del\_History table using implicit cursors.

Problem 7

Aim:

Given the table EMPLOYEE (EmpNo, Name, Salary, Designation, DeptID) write a cursor to select the five highest and lowest paid employees from the table.

Problem 8

Aim:

Write a PL/SQL block to handle the following built-in exceptions

***no\_data\_found***

***too\_many\_rows***

***zero\_divide***

Problem 9

Aim:

Write a PL/SQL block to check whether the quantity of any product in Inventory table is < 0. If so, using an exception display relevant message and update quantity to 0.

Inventory

|  |  |  |
| --- | --- | --- |
| Product\_ID | Product\_name | Quantity |

**Lab Cycle-4**

Problem 1

Aim:

Write a stored procedure, raise\_salary which accepts an employee number, increment and modifies salary of that employee in employee table. Modified salary = salary increase amount+ current salary. If employee number is not found or if the current salary is null, it should raise an exception. Otherwise, updates the salary.

Problem 2

Aim:

Write a PL/SQL function that accepts department number and returns the total salary of the department.

Problem 3

Aim:

Write a PL/SQL block that computes increment of an employee in employee table by using incr function which takes employee number as argument, calculates increment and returns the same based on the following criteria:

If salary <= 3000 – increment = 30% of salary

If salary > 3000 and <= 6000– increment = 20% of salary

Else increment = 10% of salary.

Problem 4

Aim:

Write a stored procedure that displays the employee names and their total earnings from the Emp Table. *Hint: Total earning of an employee = 12\*(gross\_salary+commission)*

Problem 5

Aim:

Create a database trigger that checks whether the new salary of employee is less than existing salary. If so, raise an appropriate exception and avoid that updation.

Problem 6

Aim:

Consider the following tables

PERSINFO

|  |  |  |
| --- | --- | --- |
| EMPNO | NAME | AGE |

AUDITPERSINFO

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| EMPNO | NAME | AGE | OPERATION | ODATE |

PERSINFO is the table for which the auditing must be performed and AUDITPERSINFO is the table which keeps track of the records deleted or modified. Create a database trigger audit\_trial. This trigger is forced when an UPDATE or a DELETE is performed on the table PERSINFO. It first checks for the operation being performed on the table. Then depending on the operation, a variable (that corresponds to operation) is assigned the value ‘UPDATE’ or ‘DELETE’ and then inserts the updated/deleted record into AUDITPERSINFO.