

## **Exercise 1**

### **Creation of a database using DDL command**

- 1) Create a table Employee with attribute Emp\_No, Name, Designation, DOB. (Consider Emp\_No as primary key)
- 2) Display the description of Employee table
- 3) Add attributes HireDate, address
- 4) Delete the attribute Address
- 5) Create a table EmpBackup from the table Employee
- 6) Rename the EmpBackup table as EmpBkup
- 7) Delete the table EmpBkup

## **Exercise 2**

### **DML queries to retrieve information from the database**

- 1) Insert five records into the table
- 2) Select all records from the table
- 3) Add new attribute salary into Employee
- 4) Insert salary information to all employees in the table
- 5) List employees whose salary greater than 30000
- 6) Display annual salary of all employees
- 7) Delete the records of all employees in Dept\_No 6
- 8) Update the table by giving an increment of 5000 to manager
- 9) Display employee names in ascending order of salary

## **Exercise 3**

### **Creating a database to set various constraints**

- 1) Create table Department with fields Dept\_no (primary key), Dept\_Name (unique), location (Check)
- 2) Create table Employee with fields Emp\_no (primary key), Emp\_Name, Salary default 5000, Dept\_ID refer Dept\_No in Department table
- 3) Create table Dependent with fields Dep\_id, Dept\_Name and Emp\_id with references Emp\_no in Employee and on delete cascade
- 4) Insert and delete records by satisfying constraints

## **Exercise 4**

### **Implementation of various aggregate functions in SQL**

- a) Find the number of employees
- b. Find how many different salary packages are provided
- c. Find the average salary of all employees.
- d. Find the highest salary of all employees.
- e. Find the minimum salary.
- f. Display total salary of all employees
- g. Display recent hire date

## **Exercise 5**

### **Implementation of Order By, Group By& Having clause.**

- a. Find the number of employees in each department
- b. Display the total number of dependents for each employee
- c. Display average employee salary by department
- d. Display employee names in descending order
- e. Display average employee salary by department, but do not include departments with an average salary of less than 7500
- f. Find the dept\_no where the average salary of all employees is more than 1500.
- g. Display the total number of dependents for each employee for employees who have at least two dependents.

## Exercise 6

### Implementation of Build in type String function

1. Display the name of employees hired on Friday
2. List employees whose hire date is 2017-16-02
3. Select employees who have more than two years of experience
4. Select employees who have joined after 2000
5. Change employee names to uppercase in employee table
6. Write a query to update the portion of the Hiredate in the employees table, within the hiredate the substring '10' will be replaced by '12'.
7. Write a query to get the details of the employees where the length of the name greater than or equal to 5.
8. Write a query to append to \_CET to employee name field in employee
9. Write a query to extract the last 4 character of employee name
10. Write a query to find employees who have minimum length name
11. Find employees whose name start with the letter C
12. Write a query that displays the name and the length of the name for all employees whose name starts with the letters 'A', 'J'. Give each column an appropriate label. Sort the results by the employees' first names.
13. Add an attribute Status with values senior (having more than 5 years of experience) otherwise junior in employee table

## Exercise 7

### Implementation of SUB QUERIES

Create and insert values into the tables

Book(bid , title not null, author not null, status default available, price

Issued\_Book(bid, user\_id, issue\_date)

User(user\_id, name, branch, semester, no. of books )

1. Find out title of books issued before 2017/03/27
2. Find out author of books taken by Mizhi
3. Display the names of books issued on most recently

## Exercise 8

### Implementation of JOIN

Create and insert values into the tables

Book(bid , title not null, author not null, status default available, price

Issued\_Book(bid, user\_id, issue\_date)

User(user\_id, name, branch, semester, no. of books )

1. Display the name of books issued on 2017/03/27
2. Display the issue date of Navathe
3. Display the name of students who have taken book authored 'Navathe'