

Lab #1

(DDL statements)

DBMS [4th Semester, BIM]

Write SQL statement in Oracle/MS SQL/MySQL for the following operations and display output of the query.

1. Create **Department** table with the following structure.

Name	Type
DEPTNO	INT
DNAME	VARCHAR(20)
LOCATION	VARCHAR(10)

Where **DEPTNO** as the primary key

2. Create a table called **EMPLOYEE** with the following structure.

Name	Type
EMPNO	INT
ENAME	VARCHAR(20)
POST	VARCHAR(10)
GENDER	CHAR(1)
CITIZENSHIPNO	VARCHAR(10)
SALARY	NUMERIC(10,6)
DEPTNO	INT

Define following constraints in the table.

- i. **EMPNO** is primary key
- ii. **DEPTNO** is foreign key from **Department** table.
- iii. Allow NULL for all columns except **ENAME** and **POST**.
- iv. **GENDER** column allows only M or F (Check Constraint)
- v. **CITIZENSHIPNO** should be unique
- vi. Make default value for **SALARY** is '10000'

3. Add a column **Email** to the **Employee** table. The field **Email** should be VARCHAR (40) type and NULL allowed.
4. Modify the column width of the **POST** field of **Employee** table. [Hint: Increase size from 10 to 15]
5. Add a check constraint in **Employee** table such that value in **EmpNo** field should be between 100 and 500 only.
6. Drop a column **Email** to the **Employee** table.
7. Create a table **CopyOfEmployee** of same structure as **Employee** table. [Use syntax: *create table tablename as select * from existing_tablename;*]
8. Drop the **CopyOfEmployee** table.

Lab #2

(Insert and Select Query)

DBMS [4th Semester, BIM]

Write SQL statement in Oracle/MS SQL/MySQL for the following operations and display output of the query.

1. Insert three records into **Department** table.
2. Insert any five records into **Employee** table using a single insert command.
3. Insert a record into Employee table with values for EMPNO, ENAME, POST, GENDER, CITIZENSHIPNO, DEPTNO only (i.e. no value for SALARY).
4. Update values in **Employee** table to set the **salary** of all employees to Rs15000/- who are working as **Manager** POST
5. Retrieve name of employee, post from the **Employee** table
6. Delete the records of only those who are working as **lecturer**
7. List all the records in the **Employee** table order by **salary** in ascending order.
8. List all the records in the Employee table order by **salary** in descending order.
9. Display records of only those employees whose **deptno** is 30.
10. Display **deptno** from the table **Employee** avoiding the duplicated values (i.e distinct values).
11. Display records of employee whose CITIZENSHIPNO is NULL.
12. Display records of employee whose salary is greater than 20000.
13. Display records of employee whose salary is greater than 20000 and post is '**Lecturer**'.
14. Display the records of employee with concatenation of EMPNO and ENAME fields separated with ‘---’
15. Display records of all employee with aliasing original name of columns in the table (i.e. Rename name of columns while displaying the records).

Lab #3

(Patterns, Aggregate functions and group by clauses)

DBMS [4th Semester , BIM]

Write SQL statement in Oracle/MS SQL/MySQL for the following operations and display output of the query.

1. Display all the details of the records of employee whose name starts with ‘A’.
2. Display all the details of the records of employee whose name does not starts with ‘A’
3. Display records of employee whose name starts with ‘A’ and ends with ‘n’.
4. Display records of employee whose name starts with ‘R’ and follows 2 characters.
5. Display the rows from Employee table whose **salary** ranges from 15000 to 30000.
6. Calculate the total and average **salary** amount of the **Employee** table.
7. Count the total records in the **Employee** table.
8. Determine the maximum and minimum **salary** and rename the column as **max_salary** and **min_salary**.
9. Find how many **posts** are available in **Employee** table.
10. Find the difference between maximum and minimum **salaries** of employees in the organization.
11. Find number of employees working in each department.
12. Find minimum salary for each department.
13. Find maximum and minimum salary amount for each post.
14. Find department wise average salary of employee in each post.
15. Find the name of department whose average salary is greater than 20000.

Lab #4

(Nested Queries, Joins)

DBMS [4th Semester, BIM]

Write SQL statement in Oracle/MS SQL/MySQL for the following operations and display output of the query.

1. Display all **employee names** and **salary** whose **salary** is greater than minimum salary of the company and post title starts with '**M**'.
2. Issue a query to find all the employees who work in the same post as **Arjun**.
3. Issue a query to display information about employees who earn more than any employee in '**Admin**' department.
4. Write query to find Cartesian Product of **Employee** and **Department** table.
5. Display employee name and their department name where they work.
6. Display the employee and department's detail that the departments are same in both the **Employee** and **Department** table.
7. Display the employee details, departments that the departments are not same in both the **Employee** and **Department**.
8. Display the Employee name and Department Name by implementing a **left outer join**.
9. Display the Employee name and Department Name by implementing a **right outer join**.
10. Display the Employee name, Post, Department Name and Location by implementing a **full outer join**.
11. Display the details of those who draw the salary greater than the **average** salary.
12. Display all the department numbers available with the **Department** and **Employee** tables avoiding duplicates.
13. Display all the department numbers available with the both **Department** and **Employee** tables.
14. Display all the dept numbers available in **Department** and not in **Employee** table.
15. Find records of employee working in '**Admin**' and '**Technical**' Department using set operation (i.e. union).

Lab #5
(Update, Delete, View and Trigger)

DBMS [4th Semester, BIM]

Write SQL statement in Oracle/MS SQL/MySQL for the following operations and display output of the query.

1. Increase **salary** of each employee by 10%.
2. Increase **salary** of employees working as '**Manager**' by 5%
3. Change name of an employee whose **EmpNo=101** such that new name will be '**Sita Devi**' .
4. Delete all employees working in department number '**200**'.
5. Delete all employees working in '**Admin**' department.
6. Create a view named **EmpView1** to display only the details of the employees [Horizontal partitioning] working in manager post and write a query to show records from the view.
7. Create a view named **EmpView2** to display **name, post** and **salary** of all employees [Vertical partitioning] and write a query to show records from the view.
8. Modify above view **EmpView2** to display only name and salary of employees.
9. Drop view **EmpView1**.
10. Create a trigger called **emp_update** which is executed when record in **Employee** table is updated. It copies data into another table say **Emp_log**.
11. Write query to add an user '**ramesh**' with password '**P@ssw0rd**'
12. Grant select, insert privileges in Employee table to '**ramesh**' user.
13. Write query to implement Transaction Control Language.