1. **Select Queries**
   1. Display all departments from department table.

Ans: SELECT \* from departments;

* 1. Display all employees from employee table.

Ans: SELECT \* from employees;

* 1. Select the employee in department 30.

Ans: SELECT \* from employees WHERE department\_id=30;

* 1. List the names, numbers and departmentno of all clerks.

Ans: SELECT first\_name,last\_name,phone\_number,department\_id FROM employees

-> WHERE job\_id LIKE '%CLERK%';

* 1. Find the depart numbers and the name of employee of all dept with Deptno greater or equal to 20.

Ans: SELECT first\_name,department\_id FROM employees

-> WHERE department\_id >=20;

* 1. Find the employees whose commission is greater than their salary.
  2. Find the employees whose commission is greater than 60 percent of their salary.
  3. Find the employee whose commission is greater than 50 percent of their salary.

Ans: SELECT \* FROM employees

-> WHERE commission\_pct>salary \*0.5;

* 1. List the name, job and salary of all employees in dept 20 who earn more than 2000.
  2. Find all salesmen in dept 30 whose salary is greater than or equal to Rs. 1500.
  3. Find all the employees whose job is either a president or manager.

Ans: SELECT \* FROM employees

-> WHERE job\_id LIKE '%MGR%' OR '%MAN%' OR '%PRES%';

* 1. Find all managers who are not in dept 30.

Ans: SELECT \* FROM employees

-> WHERE job\_id LIKE '%MGR%' OR '%MAN%'

-> AND department\_id<>30;

* 1. Find the details of all managers and clerks in dept 10.
  2. Find the details of all manager (in any dept) and all clerks in dept 10
  3. Find the details of all managers in dept 10 and all clerks in dept 20.
  4. Find all employees who are neither clerks nor manager but whose salary is greater than or equal to Rs. 2000.

Ans: SELECT \* FROM employees

-> WHERE NOT (job\_id LIKE '%MGR%' OR job\_id LIKE '%CLERK%')

-> AND salary>=2000;

* 1. Find the employees who earns between Rs. 1200 and Rs.1400.

Ans: SELECT \* FROM employees

-> WHERE salary BETWEEN 1200 AND 1400;

* 1. Find the employees who are clerks, analysts or salesman.

Ans: SELECT \* FROM employees WHERE job\_id LIKE '%SA%'

-> UNION

-> SELECT \* FROM employees WHERE job\_id LIKE '%CLERK%';

* 1. Find the employees who are not clerks, analyst or salesman.

Ans: SELECT \* FROM employees WHERE NOT job\_id LIKE '%SA%'

-> UNION

-> SELECT \* FROM employees WHERE NOT job\_id LIKE '%CLERK%';

* 1. Find the employees who do not receive a commission i.e. commission is NULL.

Ans: SELECT \* FROM employees

-> WHERE commission\_pct IS NULL;

* 1. Find the employee whose commission is Rs. 0.

Ans: SELECT \* FROM employees

-> WHERE commission\_pct=0;

* 1. Find the different jobs of the employees receiving commission.

Ans: SELECT job\_id FROM employees

-> WHERE commission\_pct IS NOT NULL;

* 1. Find all employees who do not receive a commission or whose Commission is less than Rs. 1000.

Ans: SELECT \* FROM employees WHERE commission\_pct IS NULL

-> UNION

-> SELECT \* FROM employees WHERE salary\*commission\_pct<1000;

* 1. The employees who not receiving commission are entailed to Rs. 250, Show the net earnings of all employees. (find about **nvl() )**
  2. Find all employees whose total earnings are greater than Rs. 2000.

Ans: SELECT \* FROM employees

-> WHERE salary>2000;

* 1. Find all employees whose names begin with m.

Ans: SELECT \* FROM employees

-> WHERE first\_name LIKE 'm%';

* 1. Find all employees whose names end with m.

Ans: SELECT \* FROM employees

-> WHERE first\_name LIKE '%m';

* 1. Find all employees whose names contain the letter m.

Ans: SELECT \* FROM employees

-> WHERE first\_name LIKE '%m%';

* 1. Find the employees whose names are 5 characters long and end with n.

Ans: SELECT \* FROM employees

-> WHERE length(first\_name)=5

-> AND first\_name LIKE '%n';

* 1. Find the employees who have the letter r as the third letter in their name.

Ans: SELECT \* FROM employees

-> WHERE first\_name LIKE '\_\_r%';

1. **Numeric, Character & Date Function** 
   1. Find all employees hired in month of February (of any year).

Ans: SELECT \* FROM employees

-> WHERE month(hire\_date)=2;

* 1. Find all employees who were hired on the last day of the month.

Ans: SELECT \* FROM employees

-> WHERE Last\_Day(hire\_date)=hire\_date;

* 1. Find the employees who were hired more than 12 years ago.

Ans: SELECT \* FROM employees

-> WHERE datediff(NOW(),hire\_date)>12;

* 1. Find the managers hired in the year 2007.

Ans: SELECT \* FROM employees

-> WHERE YEAR(hire\_date)=2007

-> AND job\_id LIKE '%MAN%' OR '%MGR%';

* 1. Display the names and the jobs of all employees, separated by ','(comma). For example (smith, clerk).

Ans: SELECT concat(first\_name,',',job\_id) FROM employees;

* 1. Display the names of all employees with the initial letter only in capitals.

Ans: SELECT CONCAT(UCASE(LEFT(first\_name, 1)),SUBSTRING(first\_name, 2)) FROM employees;

* 1. Display the names of all employees, right aligning them to 15 characters.

Ans: SELECT RIGHT(first\_name,15) FROM employees;

* 1. Display the names of all employees, padding them to right up-to 15 characters with '-'.

Ans: SELECT RPAD(first\_name,15,'-') FROM employees;

* 1. Display the length of the name of all employees.

Ans: SELECT length(first\_name) FROM employees;

* 1. Display the names of all employees centering them with 20 characters.

Ans:

* 1. Display the names of all employees without any leading 'a'.

Ans: SELECT TRIM(LEADING 'A' FROM first\_name) AS Lead FROM employees;

* 1. Display the names of all employees without any trailing 'r'.

Ans: SELECT TRIM(TRAILING 'R' FROM first\_name) AS Lead FROM employees;

* 1. Show the first three characters of the names of all employees.

Ans: SELECT SUBSTRING(first\_name,1,3) FROM employees;

* 1. Show the last three characters of the names of all employees.

Ans: SELECT SUBSTRING(first\_name,-3) FROM employees;

* 1. Display the names of all employees replacing any 'a' with 'e'.

Ans: SELECT REPLACE(first\_name,'a','e') FROM employees;

* 1. Display the names of all employees and the position at which the string 'ar' occurs in the name.

Ans: SELECT first\_name,POSITION('ar' IN first\_name) AS POST FROM employees;

* 1. Show the salary of all employees rounding it to the nearest Rs. 1000. For example (3790 will be 4000)

Ans: SELECT ROUND(salary,-3) AS Rounded FROM employees;

* 1. Show the daily salary of all employees assuming a month has 30 days.

Ans: SELECT first\_name,(salary/12)/30 AS DAILY\_SALARY FROM employees;

* 1. Display the name of all employees, and their bonus. Assume each Employee gets a bonus of 20 percent of his salary subject to the Maximum of Rs. 500.

Ans: SELECT first\_name,(salary+(salary\*0.2)) AS BONUS

-> FROM employees WHERE (salary\*0.2)<=500;

* 1. Display the name of all employees, and their bonus. Assume each employee gets a bonus of 20 percent of his salary subject to the Maximum of Rs. 200

Ans: SELECT first\_name,(salary+(salary\*0.2)) AS BONUS

-> FROM employees WHERE (salary\*0.2)<=200;

* 1. For each employee display the number of days passed since the employee joined the company.

Ans: SELECT datediff(NOW(), hire\_date) AS DAYS FROM employees;

* 1. For each employee display the number of months passed since the Employee joined the company.

Ans: SELECT TIMESTAMPDIFF( MONTH, hire\_date, now() ) as Month FROM employees;

* 1. Display the tenure of service in the years, months and days for all Employees in character format. Assume every month has 30 days.

Ans: SELECT CONCAT(TIMESTAMPDIFF(YEAR, hire\_date, CURDATE()), ' Years ',

-> TIMESTAMPDIFF(MONTH, hire\_date, CURDATE())%12, ' Month ',

-> TIMESTAMPDIFF(DAY, (hire\_date + INTERVAL TIMESTAMPDIFF(YEAR, hire\_date,CURDATE()) YEAR

-> + INTERVAL TIMESTAMPDIFF(MONTH, hire\_date, CURDATE())%12 MONTH), CURDATE()), ' Days')

-> AS `Duration` FROM employees;

* 1. Display the employee details in the following manner. 'Miler joined on the twenty-third of January of the year nineteen hundred and eighty Two'.

Ans: SELECT concat

-> (first\_name, ' joined on the ', DATE\_FORMAT(hire\_date,'%D'),' of ', DATE\_FORMAT(hire\_date,'%M'),' of the year ', CONVERT(DATE\_FORMAT(hire\_date,'%Y'),CHAR)) AS DISPLAY

-> FROM employees;

1. **Ordering by Queries**
   1. Display the details of all employees, sorted on the names.

Ans: SELECT first\_name

-> FROM employees

-> ORDER BY first\_name;

* 1. Display the name of all employees, based on their tenure, with the oldest employee coming first.

Ans: SELECT first\_name,hire\_date

-> FROM employees

-> ORDER BY hire\_date;

* 1. Display the names, job and salary of all employees sorted on jobs and Salary.

Ans: SELECT first\_name,job\_id,salary

-> FROM employees

-> ORDER BY job\_id,salary;

* 1. Display the names, job and salary of all employees, sorted on jobs and within job, sorted on the descending order of salary.

Ans: SELECT first\_name,job\_id,salary

-> FROM employees

-> ORDER BY job\_id IN

-> (SELECT salary

-> FROM employees

-> ORDER BY salary DESC);

* 1. Display the names, job and salary of all employees, sorted on Descending order of job and within job, sorted on the descending order of salary.

Ans: SELECT first\_name,job\_id,salary

-> FROM employees

-> ORDER BY job\_id DESC,salary DESC;

* 1. Display the name, month and year of all employees, sorted on the month of their hire date irrespective of the year.

Ans: SELECT first\_name,MONTH(hire\_date) AS MONTH,YEAR(hire\_date) AS YEAR

-> FROM employees

-> ORDER BY MONTH(hire\_date);

* 1. Display the name, month and year of joining of all employees, sorted on the month of their hire date, and within that on the year, with the earliest year appearing first.

Ans: SELECT first\_name,MONTH(hire\_date) AS MONTH,YEAR(hire\_date) AS YEAR

-> FROM employees

-> ORDER BY MONTH(hire\_date) IN

-> (SELECT YEAR(hire\_date) AS YEAR

-> FROM employees

-> ORDER BY YEAR(hire\_date));