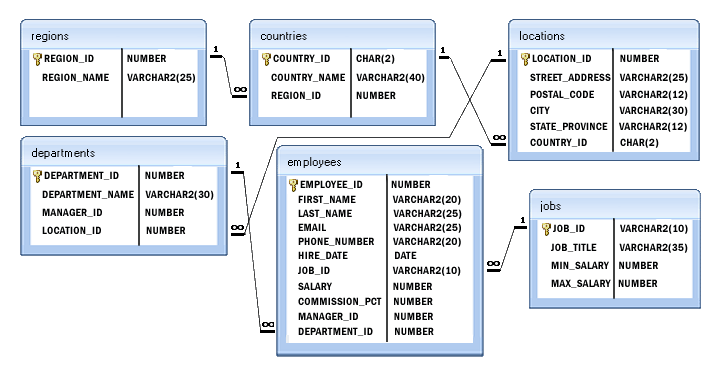
**SQL Lab**

**Database Schema:**



Tables on Schema :

SELECT \* FROM INFORMATION\_SCHEMA.TABLES;

Table Structure :

SELECT \* FROM INFORMATION\_SCHEMA.COLUMNS WHERE TABLE\_NAME='regions';

SELECT \* FROM INFORMATION\_SCHEMA.COLUMNS WHERE TABLE\_NAME='countriess';

SELECT \* FROM INFORMATION\_SCHEMA.COLUMNS WHERE TABLE\_NAME='locations';

SELECT \* FROM INFORMATION\_SCHEMA.COLUMNS WHERE TABLE\_NAME='departments';

SELECT \* FROM INFORMATION\_SCHEMA.COLUMNS WHERE TABLE\_NAME='employees';

SELECT \* FROM INFORMATION\_SCHEMA.COLUMNS WHERE TABLE\_NAME='jobs';

SELECT \* FROM INFORMATION\_SCHEMA.COLUMNS WHERE TABLE\_NAME='job\_history';

1. **Basic SELECT statements**

**`**

1. Write a query to display the names (first\_name, last\_name) using alias name “First Name", "Last Name".
   * SELECT FIRST\_NAME AS "First Name", LAST\_NAME AS "Last Name" FROM EMPLOYEES;
2. Write a query to get unique department ID from employee table.
   * SELECT DISTINCT DEPARTMENT\_ID FROM EMPLOYEES;

1. Write a query to get all employee details from the employee table order by first name, descending
   * SELECT \* FROM EMPLOYEES ORDER BY FIRST\_NAME DESC;
2. Write a query to get the names (first\_name, last\_name), salary, PF of all the employees (PF is calculated as 15% of salary)
   * SELECT CONCAT(FIRST\_NAME ,' ', LAST\_NAME) NAMES , SALARY, (SALARY\*0.15) AS "PF" FROM EMPLOYEES;
3. Write a query to get the employee ID, names (first\_name, last\_name), salary in ascending order of salary.
   * SELECT EMPLOYEE\_ID, FIRST\_NAME || ' ' || LAST\_NAME EMPLOYEE\_NAME FROM EMPLOYEES ORDER BY SALARY ASC;
4. Write a query to get the total salaries payable to employees
   * SELECT SUM(Salary) AS Total\_salary FROM Employees;
5. Write a query to get the maximum and minimum salary from employees table
   * Select MIN(Salary) Minimun\_salary, MAX(Salary) Maximam\_salary FROM Employees;
6. Write a query to get the average salary and number of employees in the employees table
   * SELECT AVG(SALARY) AS Average\_Salary, COUNT(EMPLOYEE\_ID) AS Total\_Employees FROM Employees;
7. Write a query to get the number of employees working with the company.
   * SELECT COUNT(EMPLOYEE\_ID)TOTAL\_EMPLOYEES FROM EMPLOYEES;
8. Write a query to get the number of jobs available in the employees table.
   * SELECT COUNT(DISTINCT JOB\_ID) Available\_Jobs FROM Employees;

1. Write a query get all first name from employees table in upper case
   * SELECT UPPER(FIRST\_NAME) AS First\_Name FROM Employees;
2. Write a query to get the first 3 characters of first name from employees table
   * SELECT SUBSTR(FIRST\_NAME,1,3) FIRST\_3\_CHARACTER\_OF\_NAME FROM EMPLOYEES;
   * SELECT LEFT(FIRST\_NAME, 3) FIRST\_3\_CHARACTER\_OF\_FIRST\_NAMR FROM EMPLOYEES;
3. Write a query to calculate 171\*214+625
   * SELECT (171\*214)+625 CALCULATION;
4. Write a query to get the names (for example Ellen Abel, Sundar Ande etc.) of all the employees from employees table
   * SELECT INITCAP(CONCAT(FIRST\_NAME ,' ', LAST\_NAME)) AS EMP\_NAME FROM EMPLOYEES;
5. Write a query to get first name from employees table after removing white spaces from both side
   * SELECT TRIM(FIRST\_NAME) "FIRST\_NAME" FROM EMPLOYEES;
   * SELECT TRIM(BOTH ' ' FROM FIRST\_NAME ) "FIRST\_NAME" FROM EMPLOYEES;
   * SELECT BTRIM(FIRST\_NAME,' ') "FIRST\_NAME" FROM EMPLOYEES;
   * SELECT BTRIM(FIRST\_NAME) "FIRST\_NAME" FROM EMPLOYEES;

1. Write a query to get the length of the employee names (first\_name, last\_name) from employees table.
   * SELECT FIRST\_NAME, LENGTH(FIRST\_NAME) FIRST\_NAME\_LENGTH, LAST\_NAME, LENGTH(LAST\_NAME) LAST\_NAME\_LENGTH FROM EMPLOYEES;
2. Write a query to check if the first\_name fields of the employees table contains numbers.
   * SELECT case when first\_name similar to '%[0-9]%' then 'Yes' else 'No' end as "First Name Contains Numbers", \* FROM employees ;
3. Write a query to select first 10 records from a table.
   * SELECT \* FROM DEPARTMENTS limit 10;
   * SELECT \* FROM DEPARTMENTS FETCH FIRST 10 ROWS ONLY;

--NOTE : Selecting first 10 records from the DEPARTMENTS table.

1. Write a query to get monthly salary (round 2 decimal places) of each and every employee (Note : Assume the salary field provides the 'annual salary' information)
   * SELECT employee\_id, ROUND(SALARY/12,2) MONTHLY\_SALARY FROM EMPLOYEES;
2. **Restricting and Sorting Data**
3. Write a query to display the name (first\_name, last\_name) and salary for all employees whose salary is not in the range $10,000 through $15,000.
   * SELECT CONCAT(FIRST\_NAME,' ',LAST\_NAME) EMPLOYEE\_NAME, SALARY FROM EMPLOYEES WHERE SALARY NOT BETWEEN 10000 AND 15000;
4. Write a query to display the name (first\_name, last\_name) and department ID of all employees in departments 30 or 100 in ascending order.
   * SELECT CONCAT(FIRST\_NAME,' ',LAST\_NAME) EMPLOYEE\_NAME, DEPARTMENT\_IDFROM EMPLOYEES WHERE DEPARTMENT\_ID IN (30,100);
   * SELECT CONCAT(FIRST\_NAME, ' ',LAST\_NAME) EMPLOYEE\_NAME, DEPARTMENT\_ID FROM EMPLOYEES WHERE DEPARTMENT\_ID =30 OR DEPARTMENT\_ID =100 ORDER BY DEPARTMENT\_ID;
5. Write a query to display the name (first\_name, last\_name) and salary for all employees whose salary is not in the range $10,000 through $15,000 and are in department 30 or 100.
   * SELECT CONCAT(FIRST\_NAME,' ',LAST\_NAME) EMPLOYEE\_NAME, SALARY FROM EMPLOYEES WHERE SALARY NOT BETWEEN 10000 AND 15000 AND DEPARTMENT\_ID IN (30,100);
6. Write a query to display the name (first\_name, last\_name) and hire date for all employees who were hired in 1987.
   * SELECT CONCAT(FIRST\_NAME,' ',LAST\_NAME) EMPLOYEE\_NAME, HIRE\_DATE FROM EMPLOYEES WHERE EXTRACT(YEAR FROM HIRE\_DATE)=1987;
   * SELECT CONCAT(FIRST\_NAME,' ',LAST\_NAME) EMPLOYEE\_NAME, HIRE\_DATE FROM EMPLOYEES WHERE TO\_CHAR(HIRE\_DATE,'YYYY')='1987';
7. Write a query to display the first\_name of all employees who have both "b" and "c" in their first name.
   * SELECT FIRST\_NAME FROM EMPLOYEES WHERE FIRST\_NAME LIKE '%c%' AND FIRST\_NAME LIKE '%b%';
8. Write a query to display the last name, job, and salary for all employees whose job is that of a Programmer or a Shipping Clerk, and whose salary is not equal to $4,500, $10,000, or $15,000
   * SELECT LAST\_NAME, JOB\_ID, SALARY FROM EMPLOYEES WHERE JOB\_ID IN ('IT\_PROG','SH\_CLERK') AND SALARY NOT IN (4500,10000,15000);
9. Write a query to display the last name of employees whose names have exactly 6 characters
   * SELECT LAST\_NAME FROM EMPLOYEES WHERE LENGTH(LAST\_NAME) =6;
   * SELECT LAST\_NAME FROM EMPLOYEES WHERE LAST\_NAME LIKE '\_\_\_\_\_\_';
10. Write a query to display the last name of employees having 'e' as the third character.
    * SELECT LAST\_NAME FROM EMPLOYEES WHERE LAST\_NAME LIKE '\_\_e%';
11. Write a query to display the jobs/designations available in the employees table.
    * SELECT DISTINCT JOB\_ID AVAILABLE\_JOBS FROM EMPLOYEES;
12. Write a query to display the name (first\_name, last\_name), salary and PF (15% of salary) of all employees.
    * SELECT CONCAT(FIRST\_NAME,' ',LAST\_NAME) EMPLOYEE\_NAME, SALARY, SALARY\*0.15 PF FROM EMPLOYEES;
13. Write a query to select all record from employees where last name in 'BLAKE', 'SCOTT', 'KING' and 'FORD'
    * SELECT \* FROM EMPLOYEES WHERE UPPER(LAST\_NAME) IN ('BLAKE','SCOTT','KING','FORD');
14. **Aggregate Functions and Group by**
15. Write a query to list the number of jobs available in the employees table .
    * SELECT COUNT(DISTINCT JOB\_ID) AVAILABLE\_JOBS FROM EMPLOYEES;
16. Write a query to get the total salaries payable to employees
    * SELECT SUM(SALARY) TOTAL\_PAYABLE\_SALARY FROM EMPLOYEES;
17. Write a query to get the minimum salary from employees table
    * SELECT MIN(SALARY) LOWEST\_SALARY FROM EMPLOYEES;

1. Write a query to get the maximum salary of an employee working as a Programmer
   * SELECT MAX(SALARY) HIGHEST\_SALARY FROM EMPLOYEES WHERE JOB\_ID='IT\_PROG';
2. Write a query to get the average salary and number of employees working the department 90
   * SELECT ROUND(AVG(SALARY),2) AVERAGE\_SALARY, COUNT(\*) TOTAL\_EMPLOYEES FROM EMPLOYEES GROUP BY DEPARTMENT\_ID HAVING DEPARTMENT\_ID=90;

1. Write a query to get the highest, lowest, sum, and average salary of all employees
   * SELECT MAX(SALARY) HIGHEST\_SALARY, MIN(SALARY) LOWEST\_SALARY, TRUNC(AVG(SALARY),2) AVERAGE\_SALARY, SUM(SALARY) TOTAL\_PAYABLE\_SALARY FROM EMPLOYEES;

1. Write a query to get the number of employees with the same job
   * SELECT JOB\_ID, COUNT(\*) TOTAL\_EMPLOYEE FROM EMPLOYEES GROUP BY JOB\_ID;
2. Write a query to get the difference between the highest and lowest salaries
   * SELECT MAX(SALARY)-MIN(SALARY) "HIGHEST-LOWEST" FROM EMPLOYEES ;
3. Write a query to find the manager ID and the salary of the lowest-paid employee for that manager.
   * SELECT MANAGER\_ID,MIN(SALARY) LOWEST\_SALARY\_OF\_EMPLOYEE\_UNDER\_MANAGER FROM EMPLOYEES GROUP BY MANAGER\_ID HAVING MANAGER\_ID IS NOT NULL ;
4. Write a query to get the department ID and the total salary payable in each department
   * SELECT DEPARTMENT\_ID, SUM(SALARY) TOTAL\_SALARY\_PAID FROM EMPLOYEES GROUP BY DEPARTMENT\_ID ;
5. Write a query to get the average salary for each job ID excluding programmer
   * SELECT JOB\_ID, AVG(SALARY) AVG\_SALARY FROM EMPLOYEES WHERE JOB\_ID<>'IT\_PROG' GROUP BY JOB\_ID;
   * SELECT JOB\_ID, AVG(SALARY) AVG\_SALARY FROM EMPLOYEES GROUP BY JOB\_ID HAVING JOB\_ID<>'IT\_PROG';
6. Write a query to get the total salary, maximum, minimum, average salary of employees (job ID wise), for department ID 90 only
   * SELECT JOB\_ID, MIN(SALARY) LOWEST\_SALARY, MAX(SALARY) HIGHEST\_SALARY, AVG(SALARY) AVERAGE\_SALARY, SUM(SALARY) TOTAL\_SALARY FROM EMPLOYEES WHERE DEPARTMENT\_ID=90 GROUP BY JOB\_ID;
   * SELECT JOB\_ID, MIN(SALARY) LOWEST\_SALARY, MAX(SALARY) HIGHEST\_SALARY, AVG(SALARY) AVERAGE\_SALARY, SUM(SALARY) TOTAL\_SALARY FROM EMPLOYEES GROUP BY JOB\_ID, DEPARTMENT\_ID HAVING DEPARTMENT\_ID=90 ;
7. Write a query to get the job ID and maximum salary of the employees where maximum salary is greater than or equal to $4000.
   * SELECT JOB\_ID, MAX(SALARY) HIGHEST\_SALARY FROM EMPLOYEES GROUP BY JOB\_ID HAVING MAX(SALARY)>=4000;
8. Write a query to get the average salary for all departments employing more than 10 employees.
   * SELECT DEPARTMENT\_ID, AVG(SALARY) AVERAGE\_SALARY FROM EMPLOYEES GROUP BY DEPARTMENT\_ID HAVING COUNT(\*) >10;
9. **Sub-Queries**
10. Write a query to find the name (first\_name, last\_name) and the salary of the employees who have a higher salary than the employee whose last\_name='Bull'

--Note : It may be possible that more than one person whose LAST\_NAME can be ‘BULL’

* + SELECT CONCAT(FIRST\_NAME,' ',LAST\_NAME) EMPLOYEE\_NAME, SALARY FROM EMPLOYEES WHERE SALARY > ALL (SELECT SALARY FROM EMPLOYEES WHERE UPPER(LAST\_NAME) ='BULL');
  + SELECT CONCAT(FIRST\_NAME,' ',LAST\_NAME) EMPLOYEE\_NAME, SALARY FROM EMPLOYEES WHERE SALARY > (SELECT MAX(SALARY) FROM EMPLOYEES WHERE UPPER(LAST\_NAME) ='BULL');

1. Write a query to find the name (first\_name, last\_name) of all employees who works in the IT department.
   * SELECT FIRST\_NAME || ' ' || LAST\_NAME FROM EMPLOYEES WHERE DEPARTMENT\_ID = ( SELECT DEPARTMENT\_ID FROM DEPARTMENTS WHERE UPPER(DEPARTMENT\_NAME) ='IT');
2. Write a query to find the name (first\_name, last\_name) of the employees who have a manager and worked in a USA based department.
   * SELECT CONCAT(FIRST\_NAME,' ',LAST\_NAME) EMPLOYEE\_NAME FROM EMPLOYEES E WHERE EXISTS (SELECT 'X' FROM EMPLOYEES E1 WHERE E.EMPLOYEE\_ID=E1.MANAGER\_ID) AND DEPARTMENT\_ID =ANY (SELECT DEPARTMENT\_ID FROM DEPARTMENTS WHERE LOCATION\_ID IN (SELECT LOCATION\_ID FROM LOCATIONS WHERE COUNTRY\_ID ='US')) ;
3. Write a query to find the name (first\_name, last\_name) of the employees who are managers.

--Using EXISTS Clause.

* + SELECT CONCAT(FIRST\_NAME,' ',LAST\_NAME) EMPLOYEE\_NAME FROM EMPLOYEES E WHERE EXISTS (SELECT 'X' FROM EMPLOYEES E1 WHERE E.EMPLOYEE\_ID=E1.MANAGER\_ID);

--Using IN Clause.

* + SELECT CONCAT(FIRST\_NAME,' ',LAST\_NAME) EMPLOYEE\_NAME FROM EMPLOYEES WHERE EMPLOYEE\_ID IN (SELECT DISTINCT MANAGER\_ID FROM EMPLOYEES);

1. Write a query to find the name (first\_name, last\_name), and salary of the employees whose salary is greater than the average salary.
   * SELECT CONCAT(FIRST\_NAME,' ',LAST\_NAME) EMPLOYEE\_NAME, SALARY FROM EMPLOYEES EMPLOYEES WHERE SALARY> (SELECT AVG(SALARY) FROM EMPLOYEES);
2. Write a query to find the name (first\_name, last\_name), and salary of the employees whose salary is equal to the minimum salary for their job grade.
   * SELECT CONCAT(FIRST\_NAME,' ',LAST\_NAME) EMPLOYEE\_NAME, SALARY FROM EMPLOYEES E WHERE SALARY =(SELECT MIN\_SALARY FROM JOBS J WHERE J.JOB\_ID=E.JOB\_ID);
3. Write a query to find the name (first\_name, last\_name), and salary of the employees who earns more than the average salary and works in any of the IT departments
   * SELECT CONCAT(FIRST\_NAME,' ',LAST\_NAME) EMPLOYEE\_NAME, SALARY FROM EMPLOYEES WHERE SALARY > (SELECT AVG(SALARY) FROM EMPLOYEES) AND DEPARTMENT\_ID IN (SELECT DEPARTMENT\_ID FROM DEPARTMENTS WHERE UPPER(DEPARTMENT\_NAME) ='IT' OR DEPARTMENT\_NAME LIKE 'IT %');
4. Write a query to find the name (first\_name, last\_name), and salary of the employees who earns more than the earning of Mr. Bell.
   * SELECT CONCAT(FIRST\_NAME,' ',LAST\_NAME) EMPLOYEE\_NAME, SALARY FROM EMPLOYEES WHERE SALARY > (SELECT SALARY FROM EMPLOYEES WHERE UPPER(LAST\_NAME) = 'BELL');

-- Note : If there is more then one person whose name is ‘Mr. Bell’ then in this case the query is quiet different.

1. Write a query to find the name (first\_name, last\_name), and salary of the employees who earn the same salary as the minimum salary for all departments.
   * SELECT CONCAT(FIRST\_NAME,' ',LAST\_NAME) EMPLOYEE\_NAME, SALARY FROM EMPLOYEES WHERE SALARY =(SELECT MIN(SALARY) FROM EMPLOYEES );
2. Write a query to find the name (first\_name, last\_name), and salary of the employees whose salary is greater than the average salary of all departments.
   * SELECT CONCAT(FIRST\_NAME,' ',LAST\_NAME) EMPLOYEE\_NAME, SALARY FROM EMPLOYEES WHERE SALARY > ALL (SELECT AVG(SALARY) FROM EMPLOYEES WHERE DEPARTMENT\_ID IS NOT NULL GROUP BY DEPARTMENT\_ID);
3. Write a query to find the name (first\_name, last\_name) and salary of the employees who earn a salary that is higher than the salary of all the Shipping Clerk (JOB\_ID = 'SH\_CLERK'). Sort the results of the salary of the lowest to highest.
   * SELECT FIRST\_NAME || ' ' || LAST\_NAME EMPLOYEE\_NAME, SALARY FROM EMPLOYEES WHERE SALARY> (SELECT MAX(SALARY) FROM EMPLOYEES WHERE JOB\_ID ='SH\_CLERK') ORDER BY SALARY;
   * SELECT FIRST\_NAME || ' ' || LAST\_NAME EMPLOYEE\_NAME, SALARY FROM EMPLOYEES WHERE SALARY> ALL (SELECT SALARY FROM EMPLOYEES WHERE JOB\_ID ='SH\_CLERK') ORDER BY SALARY ASC;
4. Write a query to find the name (first\_name, last\_name) of the employees who are not supervisors.
   * SELECT CONCAT(FIRST\_NAME,' ',LAST\_NAME) EMPLOYEE\_NAME FROM EMPLOYEES E1 WHERE NOT EXISTS (SELECT 'X' FROM EMPLOYEES E2 WHERE E2.MANAGER\_ID=E1.EMPLOYEE\_ID);
5. Write a query to display the employee ID, first name, last name, and department names of all employees.
   * SELECT EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME, (SELECT DEPARTMENT\_NAME FROM DEPARTMENTS WHERE DEPARTMENTS.DEPARTMENT\_ID = EMPLOYEES.DEPARTMENT\_ID) FROM EMPLOYEES;
6. Write a query to display the employee ID, first name, last name, salary of all employees whose salary is above average for their departments
   * SELECT EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME, SALARY FROM EMPLOYEES E1 WHERE SALARY > (SELECT AVG(SALARY) FROM EMPLOYEES E2 WHERE E1.DEPARTMENT\_ID=E2.DEPARTMENT\_ID);
7. Write a query to fetch even numbered records from employees table.
   * SELECT \* FROM EMPLOYEES WHERE MOD(EMPLOYEE\_ID,2)=0;
   * SELECT \* FROM (select \*, row\_number() over() AS records from employees) as Numbered\_records where mod(records,2)=0;
8. Write a query to find the 5th maximum salary in the employees table.
   * SELECT SALARY FROM (SELECT DISTINCT SALARY FROM EMPLOYEES ORDER BY SALARY DESC) EMP OFFSET 4 ROWS LIMIT 1;
   * SELECT SALARY FROM (SELECT DISTINCT SALARY, DENSE\_RANK() OVER(ORDER BY SALARY DESC) TOP\_SALARIES FROM EMPLOYEES ORDER BY TOP\_SALARIES ) EMP OFFSET 4 ROWS LIMIT 1;
   * SELECT salary FROM (select distinct salary , dense\_rank() over(order by salary desc) AS records from employees) as Numbered\_records where records=5;
9. Write a query to find the 4th minimum salary in the employees table.
   * SELECT SALARY FROM (SELECT DISTINCT SALARY FROM EMPLOYEES ORDER BY SALARY ASC) EMP OFFSET 3 ROWS LIMIT 1;
   * SELECT salary FROM (select distinct salary , dense\_rank() over(order by salary ) AS records from employees) as Numbered\_records where records=4;
10. Write a query to select last 10 records from a table
    * SELECT \* FROM EMPLOYEES ORDER BY EMPLOYEE\_ID DESC LIMIT 10;
    * SELECT \* FROM (select \*, row\_number() over( ) AS records from employees) as Numbered\_records order by records desc limit 10;
11. Write a query to list the department ID and name of all the departments where no employee is working.
    * SELECT DEPARTMENT\_ID, DEPARTMENT\_NAME FROM DEPARTMENTS D WHERE NOT EXISTS (SELECT 'X' FROM EMPLOYEES E WHERE E.DEPARTMENT\_ID=D.DEPARTMENT\_ID);
12. Write a query to get 3 maximum salaries
    * SELECT \* FROM (SELECT DISTINCT SALARY FROM EMPLOYEES ORDER BY SALARY DESC) EMP LIMIT 3;
    * SELECT SALARY FROM (SELECT DISTINCT SALARY, DENSE\_RANK() OVER(ORDER BY SALARY DESC)AS RECORDS FROM EMPLOYEES) AS SORTED\_RECORDS WHERE RECORDS<= 3;
    * SELECT DISTINCT SALARY FROM EMPLOYEES ORDER BY SALARY DESC LIMIT 3;
13. Write a query to get 3 minimum salaries.
    * SELECT \* FROM (SELECT DISTINCT SALARY FROM EMPLOYEES ORDER BY SALARY ) EMP LIMIT 3;
    * SELECT DISTINCT SALARY FROM EMPLOYEES ORDER BY SALARY ASC LIMIT 3;
14. Write a query to get nth max salaries of employees.
    * SELECT SALARY FROM (SELECT DISTINCT SALARY, DENSE\_RANK() OVER(ORDER BY SALARY DESC)AS RECORDS FROM EMPLOYEES) AS SORTED\_RECORDS WHERE RECORDS<= [n];
    * SELECT SALARY FROM (SELECT DISTINCT SALARY, DENSE\_RANK() OVER(ORDER BY SALARY DESC)AS RECORDS FROM EMPLOYEES ORDER BY RECORDS ) AS SORTED\_RECORDS LIMIT [n];
    * SELECT DISTINCT SALARY FROM EMPLOYEES ORDER BY SALARY DESC LIMIT [n];

NOTE : Where `n` is variable number.

**E. JOINS**

1. Write a query to find the addresses (location\_id, street\_address, city, state\_province, country\_name) of all the departments. (Hint : Use NATURAL JOIN)
   * SELECT DEPARTMENT\_ID, LOCATION\_ID, STREET\_ADDRESS, CITY, STATE\_PROVINCE, COUNTRY\_NAME FROM DEPARTMENTS NATURAL JOIN LOCATIONS NATURAL JOIN COUNTRIES ORDER BY DEPARTMENT\_ID;
2. Write a query to find the name (first\_name, last name), department ID and name of all the employees

-- I use Left Join because there is some employees who have not Department\_id or department\_id is null.

* + SELECT CONCAT(E.FIRST\_NAME,' ',E.LAST\_NAME) EMPLOYEE\_NAME, E.DEPARTMENT\_ID , D.DEPARTMENT\_NAME FROM EMPLOYEES E LEFT JOIN DEPARTMENTS D ON E.DEPARTMENT\_ID=D.DEPARTMENT\_ID;

1. Write a query to find the name (first\_name, last\_name), job, department ID and name of the employees who works in London
   * SELECT CONCAT(E.FIRST\_NAME,' ',E.LAST\_NAME) EMPLOYEE\_NAME, J.JOB\_TITLE, E.DEPARTMENT\_ID, D.DEPARTMENT\_NAME FROM EMPLOYEES E INNER JOIN DEPARTMENTS D ON E.DEPARTMENT\_ID=D.DEPARTMENT\_ID INNER JOIN LOCATIONS L ON L.LOCATION\_ID= D.LOCATION\_ID INNER JOIN JOBS J ON E.JOB\_ID = J.JOB\_ID WHERE UPPER(L.CITY) ='LONDON';
2. Write a query to find the employee id, name (last\_name) along with their manager\_id and name (last\_name)
   * SELECT E.EMPLOYEE\_ID, E.LAST\_NAME EMP\_NAME, E.MANAGER\_ID, M.LAST\_NAME MANAGER\_NAME FROM EMPLOYEES E INNER JOIN EMPLOYEES M ON E.MANAGER\_ID=M.EMPLOYEE\_ID;
   * SELECT E.EMPLOYEE\_ID, E.LAST\_NAME EMP\_NAME, E.MANAGER\_ID, M.LAST\_NAME MANAGER\_NAME FROM EMPLOYEES E, EMPLOYEES M WHERE E.MANAGER\_ID=M.EMPLOYEE\_ID;
3. Write a query to find the name (first\_name, last\_name) and hire date of the employees who was hired after 'Jones'.
   * SELECT CONCAT(E1.FIRST\_NAME,' ',E1.LAST\_NAME) EMPLOYEE\_NAME, E1.HIRE\_DATE HIRE\_ON FROM EMPLOYEES E1 , EMPLOYEES E2 WHERE E2.LAST\_NAME ='Jones' AND E1.HIRE\_DATE>E2.HIRE\_DATE;

* + SELECT CONCAT(E.FIRST\_NAME,' ',E.LAST\_NAME) EMPLOYEE\_NAME, E.HIRE\_DATE FROM EMPLOYEES E INNER JOIN EMPLOYEES E2 ON E2.LAST\_NAME='Jones' WHERE E2.HIRE\_DATE<E.HIRE\_DATE;

1. Write a query to get the department name and number of employees in the department.
   * SELECT D.DEPARTMENT\_NAME DEPT\_NAME, COALESCE(COUNT(E.EMPLOYEE\_ID),0) TOTAL\_EMPLOYEES FROM DEPARTMENTS D LEFT JOIN EMPLOYEES E ON D.DEPARTMENT\_ID=E.DEPARTMENT\_ID GROUP BY D.DEPARTMENT\_NAME ORDER BY COUNT(E.EMPLOYEE\_ID) DESC;
2. Write a query to find the employee ID, job title, number of days between ending date and starting date for all jobs in department 90.
   * SELECT JH.EMPLOYEE\_ID, J.JOB\_TITLE, DATE\_PART('DAYS',JH.END\_DATE-JH.START\_DATE ) DAYS\_WORKED FROM JOBS J JOIN JOB\_HISTORY JH ON J.JOB\_ID=JH.JOB\_ID AND JH.DEPARTMENT\_ID=90;
3. Write a query to display the department ID and name and first name of manager.
   * SELECT D.DEPARTMENT\_ID, D.DEPARTMENT\_NAME, E.FIRST\_NAME MANAGER\_NAME FROM DEPARTMENTS D JOIN EMPLOYEES E ON D.MANAGER\_ID=E.EMPLOYEE\_ID;
4. Write a query to display the department name, manager name, and city.
   * SELECT D.DEPARTMENT\_NAME, CONCAT(E.FIRST\_NAME, ' ',E.LAST\_NAME) MANAGER\_NAME, L.CITY "CITY" FROM DEPARTMENTS D LEFT OUTER JOIN EMPLOYEES E ON D.MANAGER\_ID=E.EMPLOYEE\_ID LEFT OUTER JOIN LOCATIONS L ON D.LOCATION\_ID=L.LOCATION\_ID;
5. Write a query to display the job title and average salary of employees.
   * SELECT J.JOB\_TITLE, ROUND(AVG(E.SALARY),2) AVG\_SALARY FROM JOBS J INNER JOIN EMPLOYEES E ON E.JOB\_ID=J.JOB\_ID GROUP BY J.JOB\_TITLE;
6. Write a query to display job title, employee name, and the difference between salary of the employee and minimum salary for the job
   * SELECT J.JOB\_TITLE, E.FIRST\_NAME || ' ' || E.LAST\_NAME EMPLOYEE\_NAME, E.SALARY-J.MIN\_SALARY AMOUNT\_MORE\_THAN\_MIN\_SALARY FROM EMPLOYEES E JOIN JOBS J ON E.JOB\_ID=J.JOB\_ID;
7. Write a query to display the job history that were done by any employee who is currently drawing more than 10000 of salary.
   * SELECT E.EMPLOYEE\_ID, JH.JOB\_ID PREV\_JOB, E.JOB\_ID CURRENT\_JOB, E.SALARY CURR\_SALARY, JH.DEPARTMENT\_ID PREV\_DEPT\_ID , E.DEPARTMENT\_ID CURR\_DEPT\_ID FROM EMPLOYEES E LEFT OUTER JOIN JOB\_HISTORY JH ON E.EMPLOYEE\_ID=JH.EMPLOYEE\_ID WHERE E.SALARY>10000 ORDER BY E.EMPLOYEE\_ID, E.JOB\_ID, E.DEPARTMENT\_ID,JH.START\_DATE;
8. Write a query to display department name, name (first\_name, last\_name), hire date, salary of the manager for all managers whose experience is more than 15 years.
   * SELECT D.DEPARTMENT\_NAME, CONCAT(E.FIRST\_NAME, ' ' ,E.LAST\_NAME) MANAGER\_NAME, E.HIRE\_DATE, E.SALARY FROM EMPLOYEES E JOIN DEPARTMENTS D ON D.DEPARTMENT\_ID=E.DEPARTMENT\_ID WHERE EXISTS (SELECT 'X' FROM EMPLOYEES E1 WHERE E1.MANAGER\_ID=E.EMPLOYEE\_ID AND (EXTRACT(YEAR FROM AGE(CURRENT\_DATE, E1.HIRE\_DATE)) \* 12 + EXTRACT(MONTH FROM AGE(CURRENT\_DATE, E1.HIRE\_DATE)))>15\*12);

**F. Date & Time**

1. Write a query to display the first day of the month (in datetime format) three months before the current month. (Sample current date : 2014-09-03 & Expected result : 2014-06-01).
   * SELECT DATE\_TRUNC('MONTH',CURRENT\_DATE)-INTERVAL '3 MONTHS' FIRST\_DAY\_OF\_DATE\_3\_MONTH\_BEFORE;
2. Write a query to display the last day of the month (in datetime format) three months before the current month.
   * SELECT DATE\_TRUNC('MONTH',CURRENT\_DATE+ INTERVAL '1 MONTH' )- INTERVAL '1 DAY' LASST\_DAY\_OF\_CURRENT\_MONTH;
3. Write a query to get the distinct Mondays from hire\_date in employees tables.
   * SELECT DISTINCT HIRE\_DATE MONDAYS\_IN\_HIRE\_DATE FROM EMPLOYEES WHERE EXTRACT(DOW FROM HIRE\_DATE)=1;
   * SELECT DISTINCT HIRE\_DATE MONDAYS FROM EMPLOYEES WHERE TO\_CHAR ( HIRE\_DATE,'FMDAY')='MONDAY';
4. Write a query to get the first day of the current year.
   * SELECT DATE\_TRUNC('YEAR',CURRENT\_DATE) FIRST\_DAY\_OF\_THE\_YEAR;
5. Write a query to get the last day of the current year.
   * SELECT DATE\_TRUNC('YEAR',CURRENT\_DATE) + INTERVAL '1 YEAR' - INTERVAL '1 DAY' LAST\_DAY\_OF\_THE\_YEAR;
6. Write a query to calculate the age in year.
   * SELECT EMPLOYEE\_ID, AGE(CURRENT\_DATE,HIRE\_DATE) EXPERIENCE FROM EMPLOYEES;
   * SELECT AGE(CURRENT\_DATE,TIMESTAMP '2000-06-22') "AGE";
7. Write a query to get the current date in the following format. (Sample date : 2014-09-04 & Output : September 4, 2014)
   * SELECT TO\_CHAR(CURRENT\_DATE, 'FMMONTH DD, YYYY') CURR\_DATE;
8. Write a query to get the current date in Thursday September 2014 format. (Thursday September 2014)
   * SELECT TO\_CHAR(CURRENT\_DATE, 'FMDAY FMMONTH YYYY') CURR\_DATE;
9. Write a query to extract the year from the current date
   * SELECT EXTRACT(YEAR FROM CURRENT\_DATE) CURRENT\_YEAR;
   * SELECT TO\_CHAR(CURRENT\_DATE,'YYYY') CURRENT\_YEAR;
   * SELECT EXTRACT(ISOYEAR FROM CURRENT\_DATE) CURRENT\_YEAR;
10. Write a query to get the DATE value from a given day (number in N). (Sample days: 730677 & Output : 2000-07-11)
    * SELECT TO\_TIMESTAMP( '0000-00-00 ', 'YYYY-MM-DD ')+ INTERVAL '730677 DAYS';
11. Write a query to get the first name and hire date from employees table where hire date between '1987-06-01' and '1987-07-30'
    * SELECT FIRST\_NAME, HIRE\_DATE FROM EMPLOYEES WHERE HIRE\_DATE BETWEEN '1987-06-01' AND '1987-07-30';
12. Write a query to display the current date in the following format. (Sample output: Thursday 4th September 2014 00:00:00)
    * SELECT TO\_CHAR(CURRENT\_DATE, 'FMDAY DDTH FMMONTH YYYY HH:MI:SS') CURR\_DATE;
13. Write a query to display the current date in the following format. (Sample output: 05/09/2014)
    * SELECT TO\_CHAR(CURRENT\_DATE, 'DD/MM/YYYY') CURR\_DATE;
14. Write a query to display the current date in the following format. (Sample output: 12:00 AM Sep 5, 2014)
    * SELECT TO\_CHAR(CURRENT\_DATE, 'HH12:MI AM MON DD,YYYY') CURR\_TIME\_DATE;
15. Write a query to get the firstname, lastname who joined in the month of June
    * SELECT FIRST\_NAME, LAST\_NAME FROM EMPLOYEES WHERE EXTRACT(MONTH FROM HIRE\_DATE) =6;
    * SELECT FIRST\_NAME, LAST\_NAME FROM EMPLOYEES WHERE TO\_CHAR(HIRE\_DATE,'FMMONTH')='JUNE';
16. Write a query to get the years in which more than 10 employees joined
    * SELECT EXTRACT(YEAR FROM HIRE\_DATE) "YEAR\_MORE\_THAN\_10\_EMP\_HIRED" FROM EMPLOYEES GROUP BY EXTRACT(YEAR FROM HIRE\_DATE) HAVING COUNT(\*)>10;
17. Write a query to get first name of employees who joined in 1987
    * SELECT FIRST\_NAME JOINED\_ON\_1987 FROM EMPLOYEES WHERE EXTRACT(YEAR FROM HIRE\_DATE)=1987;
18. Write a query to get department name, manager name, and salary of the manager for all managers whose experience is more than 5 years
    * SELECT D.DEPARTMENT\_NAME, E.FIRST\_NAME || ' ' || E.LAST\_NAME MANAGER\_NAME, E.SALARY FROM (SELECT EMPLOYEE\_ID, DEPARTMENT\_ID, FIRST\_NAME ,LAST\_NAME, SALARY, HIRE\_DATE FROM EMPLOYEES E1 WHERE EXISTS (SELECT 'X' FROM EMPLOYEES E2 WHERE E1.EMPLOYEE\_ID=E2.MANAGER\_ID)) E JOIN DEPARTMENTS D ON E.DEPARTMENT\_ID=D.DEPARTMENT\_ID WHERE EXTRACT(YEAR FROM AGE(CURRENT\_DATE, E.HIRE\_DATE) )>=5 AND EXTRACT(MONTH FROM INTERVAL '5 YEARS' -AGE(CURRENT\_DATE, E.HIRE\_DATE) )<=0 AND EXTRACT(DAY FROM INTERVAL '5 YEARS' -AGE(CURRENT\_DATE, E.HIRE\_DATE) )<0;
19. Write a query to get employee ID, last name, and date of first salary of the employees
    * SELECT EMPLOYEE\_ID, LAST\_NAME, DATE\_TRUNC('MONTH',HIRE\_DATE+INTERVAL '1 MONTH')-INTERVAL '1 DAY' FIRST\_SALARY\_DAY FROM EMPLOYEES;
20. Write a query to get first name, hire date and experience of the employees
    * SELECT FIRST\_NAME, HIRE\_DATE, AGE(CURRENT\_DATE, HIRE\_DATE) EXPERIENCE FROM EMPLOYEES;
21. Write a query to get the department ID, year, and number of employees joined
    * SELECT DEPARTMENT\_ID, EXTRACT (YEAR FROM HIRE\_DATE) YEARS, COUNT(EMPLOYEE\_ID) EMPLOYEES\_JOINED FROM EMPLOYEES GROUP BY DEPARTMENT\_ID, EXTRACT (YEAR FROM HIRE\_DATE) ORDER BY DEPARTMENT\_ID, YEARS;

**G. Strings**

1. Write a query to get the job\_id and related employee's id. Partial output of the query:
   * SELECT JOB\_ID, STRING\_AGG(EMPLOYEE\_ID ::VARCHAR(255), ','ORDER BY EMPLOYEE\_ID) EMPLOYEE\_IDS FROM EMPLOYEES GROUP BY JOB\_ID;
   * SELECT JOB\_ID, ARRAY\_AGG(EMPLOYEE\_ID)”Employees ID” FROM EMPLOYEES GROUP BY JOB\_ID;

|  |  |
| --- | --- |
| job\_id | Employees ID |
| AC\_ACCOUNT | 206 |
| AC\_MGR | 205 |
| AD\_ASST | 200 |
| AD\_PRES | 100 |
| AD\_VP | 101 ,102 |
| FI\_ACCOUNT | 110 ,113 ,111 ,109 ,112 |

1. Write a query to update the portion of the phone\_number in the employees table, within the phone number the substring '124' will be replaced by '999'
   * UPDATE EMPLOYEES SET PHONE\_NUMBER = REPLACE(PHONE\_NUMBER, '124', '999');
2. Write a query to get the details of the employees where the length of the first name greater than or equal to 8.
   * SELECT \* FROM EMPLOYEES WHERE LENGTH(FIRST\_NAME)>=8;
3. Write a query to display leading zeros before maximum and minimum salary
   * SELECT job\_id, LPAD(max\_salary::VARCHAR(255), 7, '0') "MAX\_SALARY", LPAD(min\_salary::VARCHAR(255), 7, '0') "MIN\_SALARY" FROM Jobs;
4. Write a query to append '@example.com' to email field.
   * SELECT EMAIL ||'@example.com' FROM EMPLOYEES;
   * UPDATE EMPLOYEES SET EMAIL = CONCAT(EMAIL, ['@example.com');](mailto:'@example.com');)
   * UPDATE EMPLOYEES SET EMAIL = EMAIL|| '@example.com';
5. Write a query to get the employee id, first name and hire month
   * SELECT EMPLOYEE\_ID, FIRST\_NAME, TO\_CHAR(HIRE\_DATE,'FMMONTH') HIRE\_MONTH FROM EMPLOYEES;
   * SELECT EMPLOYEE\_ID, FIRST\_NAME, EXTRACT(MONTH FROM HIRE\_DATE) HIRE\_MONTH FROM EMPLOYEES ORDER BY HIRE\_MONTH;
6. Write a query to get the employee id, email id (discard the last three characters)
   * SELECT EMPLOYEE\_ID, SUBSTRING(EMAIL,0,LENGTH(EMAIL)-3) EMAILS FROM EMPLOYEES;
   * SELECT EMPLOYEE\_ID, SUBSTR(EMAIL,0,LENGTH(EMAIL)-3) EMAILS FROM EMPLOYEES;
   * SELECT EMPLOYEE\_ID, EMAIL, RTRIM(EMAIL,SUBSTRING(EMAIL FROM '...$')) EMAILS FROM EMPLOYEES;
7. Write a query to find all employees where first names are in upper case
   * SELECT \* FROM EMPLOYEES WHERE FIRST\_NAME = UPPER(FIRST\_NAME);
8. Write a query to extract the last 4 character of phone numbers
   * SELECT EMPLOYEE\_ID, RIGHT(PHONE\_NUMBER,4) LAST\_4\_DIGIT\_OF\_PHONE\_NUMBER FROM EMPLOYEES;
   * SELECT SUBSTRING(PHONE\_NUMBER,LENGTH(PHONE\_NUMBER)-3) FROM EMPLOYEES;
9. Write a query to get the last word of the street address
   * SELECT LOCATION\_ID, STREET\_ADDRESS, SPLIT\_PART(TRIM(STREET\_ADDRESS), ' ',-1) LAST\_WORD\_OF\_STREET\_ADDRESS FROM LOCATIONS;
10. Write a query to get the locations that have minimum street length
    * SELECT \* FROM LOCATIONS WHERE LENGTH(STREET\_ADDRESS) = (SELECT MIN(LENGTH(STREET\_ADDRESS)) FROM LOCATIONS);
11. Write a query to display the first word from those job titles which contains more than one words
    * SELECT LEFT(TRIM(JOB\_TITLE),POSITION(' ' IN TRIM(JOB\_TITLE))) FROM JOBS WHERE POSITION(' ' IN TRIM(JOB\_TITLE))>0;
12. Write a query to display the length of first name for employees where last name contain character 'c' after 2nd position
    * SELECT EMPLOYEE\_ID, LENGTH(FIRST\_NAME) LENGTH\_OF\_FIRST\_NAME FROM EMPLOYEES WHERE LAST\_NAME SIMILAR TO '\_\_%c%';
13. Write a query that displays the first name and the length of the first name for all employees whose name starts with the letters 'A', 'J' or 'M'. Give each column an appropriate label. Sort the results by the employees' first names
    * SELECT FIRST\_NAME, CHAR\_LENGTH(FIRST\_NAME) FIRST\_NAME\_LENGTH FROM EMPLOYEES WHERE FIRST\_NAME SIMILAR TO '[AJM]%' ORDER BY FIRST\_NAME;
14. Write a query to display the first name and salary for all employees. Format the salary to be 10 characters long, left-padded with the $ symbol. Label the column SALARY
    * SELECT FIRST\_NAME, LPAD(SALARY::TEXT,10,'$') "SALARY" FROM EMPLOYEES;
15. Write a query to display the first eight characters of the employees' first names and indicates the amounts of their salaries with '$' sign. Each '$' sign signifies a thousand dollars. Sort the data in descending order of salary
    * SELECT SUBSTR(FIRST\_NAME,1,8) "FIRST\_NAME", TO\_CHAR(SALARY,'$99,999.00') "SALARY" FROM EMPLOYEES;
    * SELECT LEFT(FIRST\_NAME,8), TO\_CHAR(SALARY,'$999,999,999.00') "SALARY\_OF\_EMP" FROM EMPLOYEES ORDER BY SALARY;
16. Write a query to display the employees with their code, first name, last name and hire date who hired either on seventh day of any month or seventh month in any year
    * SELECT EMPLOYEE\_ID AS "EMPLOYEE CODE ", FIRST\_NAME, LAST\_NAME, HIRE\_DATE FROM EMPLOYEES WHERE EXTRACT(DAY FROM HIRE\_DATE) =7 OR EXTRACT(MONTH FROM HIRE\_DATE) =7;
    * SELECT EMPLOYEE\_ID AS "EMPLOYEE CODE ", FIRST\_NAME, LAST\_NAME, HIRE\_DATE FROM EMPLOYEES WHERE TO\_CHAR(HIRE\_DATE,'DD')='07' OR TO\_CHAR(HIRE\_DATE,'MM')='07';