# SQL LAB -II

NAME : YASH SHRIVASTAVA

EMPLOYEE ID: 2024-2403

1. Write a query to display the name (first name and last name) for those employees who gets more salary than the employee whose ID is 163.

SELECT CONCAT(FIRST\_NAME, ' ' , LAST\_NAME) EMPLOYEE\_NAME FROM EMPLOYEES WHERE SALARY>(SELECT SALARY FROM EMPLOYEES WHERE EMPLOYEE\_ID=163);

2. Write a query to display the name (first name and last name), salary, department id, job id for those employees who works in the same designation as the employee works whose id is 169.

SELECT CONCAT(FIRST\_NAME, ' ' , LAST\_NAME) EMPLOYEE\_NAME, SALARY, DEPARTMENT\_ID, JOB\_ID FROM EMPLOYEES WHERE JOB\_ID = (SELECT JOB\_ID FROM EMPLOYEES WHERE EMPLOYEE\_ID=169);

3. Write a query to display the name (first name and last name), salary, department id for those employees who earn such amount of salary which is the smallest salary of any of the departments.

SELECT CONCAT(FIRST\_NAME, ' ' , LAST\_NAME) EMPLOYEE\_NAME , SALARY, DEPARTMENT\_ID FROM EMPLOYEES WHERE SALARY =ANY(SELECT MIN(SALARY) FROM EMPLOYEES GROUP BY DEPARTMENT\_ID);

4. Write a query to display the employee id, employee name (first name and last name) for all employees who earn more than the average salary.

SELECT EMPLOYEE\_ID, CONCAT(FIRST\_NAME, ' ' , LAST\_NAME) EMPLOYEE\_NAME FROM EMPLOYEES WHERE SALARY> (SELECT AVG(SALARY) FROM EMPLOYEES);

5. Write a query to display the employee name (first name and last name), employee id and salary of all employees who report to Payam.

SELECT EMPLOYEE\_ID, CONCAT(FIRST\_NAME, ' ' , LAST\_NAME) EMPLOYEE\_NAME, SALARY FROM EMPLOYEES WHERE MANAGER\_ID =ANY (SELECT EMPLOYEE\_ID FROM EMPLOYEES WHERE UPPER(FIRST\_NAME) ='PAYAM' OR UPPER(LAST\_NAME) ='PAYAM');

6. Write a query to display the department number, name (first name and last name), job\_id and department name for all employees in the Finance department.

SELECT D.DEPARTMENT\_ID, D.DEPARTMENT\_NAME , CONCAT(E.FIRST\_NAME,' ',E.LAST\_NAME) EMPLOYEE\_NAME, E.JOB\_ID FROM EMPLOYEES E INNER JOIN DEPARTMENTS D ON E.DEPARTMENT\_ID = D.DEPARTMENT\_ID AND D.DEPARTMENT\_NAME ='Finance';

SELECT D.DEPARTMENT\_ID, D.DEPARTMENT\_NAME , CONCAT(E.FIRST\_NAME,' ',E.LAST\_NAME) EMPLOYEE\_NAME, E.JOB\_ID FROM EMPLOYEES E INNER JOIN DEPARTMENTS D ON E.DEPARTMENT\_ID = D.DEPARTMENT\_ID WHERE D.DEPARTMENT\_NAME='Finance';

7. Write a query to display all the information of an employee whose salary and reporting person id is 3000 and 121, respectively.

SELECT \* FROM EMPLOYEES WHERE SALARY =3000 AND MANAGER\_ID=121;

SELECT \* FROM EMPLOYEES WHERE (SALARY,MANAGER\_ID)=(3000,121);

8. Display all the information of an employee whose id is any of the number 134, 159 and 183.

SELECT \* FROM EMPLOYEES WHERE EMPLOYEE\_ID IN (134,159,183);

9. Write a query to display all the information of the employees whose salary is within the range 1000 and 3000.

SELECT \* FROM EMPLOYEES WHERE SALARY BETWEEN 1000 AND 3000;

10. Write a query to display all the information of the employees whose salary is within the range of smallest salary and 2500.

SELECT \* FROM EMPLOYEES WHERE SALARY BETWEEN (SELECT MIN(SALARY) FROM EMPLOYEES) AND 2500;

11. Write a query to display all the information of the employees who does not work in those departments where some employees works whose manager id within the range 100 and 200.

SELECT \* FROM EMPLOYEES WHERE DEPARTMENT\_ID NOT IN (SELECT DEPARTMENT\_ID FROM EMPLOYEES WHERE MANAGER\_ID BETWEEN 100 AND 120);

12. Write a query to display all the information for those employees whose id is any id who earn the second highest salary.

SELECT \* FROM EMPLOYEES WHERE SALARY = (SELECT DISTINCT SALARY FROM EMPLOYEES ORDER BY SALARY DESC OFFSET 1 ROW LIMIT 1);

13. Write a query to display the employee name (first name and last name) and hire date for all employees in the same department as Clara. Exclude Clara.

SELECT CONCAT(FIRST\_NAME,' ',LAST\_NAME) EMPLOYEE\_NAME, HIRE\_DATE FROM EMPLOYEES WHERE DEPARTMENT\_ID = ANY(SELECT DEPARTMENT\_ID FROM EMPLOYEES WHERE FIRST\_NAME='Clara') AND FIRST\_NAME<>'Clara';

14. Write a query to display the employee number and name (first name and last name) for all employees who work in a department with any employee whose name contains a T.

SELECT EMPLOYEE\_ID, CONCAT(FIRST\_NAME,' ',LAST\_NAME) EMPLOYEE\_NAME FROM EMPLOYEES WHERE DEPARTMENT\_ID = ANY (SELECT DEPARTMENT\_ID FROM EMPLOYEES WHERE FIRST\_NAME SIMILAR TO '%T%' OR LAST\_NAME SIMILAR TO '%T%');

15. Write a query to display the employee number, name (first name and last name), and salary for all employees who earn more than the average salary and who work in a department with any employee with a J in their name.

SELECT EMPLOYEE\_ID, CONCAT(FIRST\_NAME, ' ',LAST\_NAME) EMPLOYEE\_NAME, SALARY FROM EMPLOYEES WHERE SALARY >(SELECT AVG(SALARY) FROM EMPLOYEES) AND DEPARTMENT\_ID =ANY(SELECT DEPARTMENT\_ID FROM EMPLOYEES WHERE CONCAT(FIRST\_NAME,LAST\_NAME) SIMILAR TO '%J%');

16. Display the employee name (first name and last name), employee id, and job title for all employees whose department location is Toronto.

SELECT CONCAT(FIRST\_NAME,' ', LAST\_NAME) EMPLOYEE\_NAME, EMPLOYEE\_ID, (SELECT JOB\_TITLE FROM JOBS J WHERE J.JOB\_ID=E.JOB\_ID) FROM EMPLOYEES E WHERE DEPARTMENT\_ID IN (SELECT DEPARTMENT\_ID FROM DEPARTMENTS D WHERE LOCATION\_ID IN(SELECT LOCATION\_ID FROM LOCATIONS WHERE CITY='Toronto'));

SELECT E.EMPLOYEE\_ID, E.FIRST\_NAME, E.LAST\_NAME, J.JOB\_TITLE FROM EMPLOYEES E JOIN DEPARTMENTS D USING (DEPARTMENT\_ID) JOIN LOCATIONS L USING(LOCATION\_ID) JOIN JOBS J USING(JOB\_ID) WHERE L.CITY='Toronto';

17. Write a query to display the employee number, name (first name and last name) and job title for all employees whose salary is smaller than any salary of those employees whose job title is MK\_MAN.

SELECT E.EMPLOYEE\_ID, E.FIRST\_NAME, E.LAST\_NAME, J.JOB\_TITLE FROM EMPLOYEES E JOIN JOBS J USING(JOB\_ID) WHERE E.SALARY <ANY (SELECT SALARY FROM EMPLOYEES WHERE JOB\_ID='MK\_MAN');

SELECT EMPLOYEE\_ID, CONCAT(FIRST\_NAME,' ',LAST\_NAME) EMPLOYEE\_NAME, JOB\_TITLE FROM EMPLOYEES E JOIN JOBS J ON E.JOB\_ID=J.JOB\_ID WHERE SALARY <ANY (SELECT SALARY FROM EMPLOYEES WHERE JOB\_ID='MK\_MAN');

18. Write a query to display the employee number, name (first name and last name) and job title for all employees whose salary is smaller than any salary of those employees whose job title is MK\_MAN. Exclude Job title MK\_MAN.

SELECT EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME, JOB\_ID FROM EMPLOYEES WHERE SALARY <ANY (SELECT SALARY FROM EMPLOYEES WHERE JOB\_ID='MK\_MAN') AND JOB\_ID <>'MK\_MAN';

19. Write a query to display the employee number, name (first name and last name) and job title for all employees whose salary is more than any salary of those employees whose job title is PU\_MAN.

Exclude job title PU\_MAN.

SELECT EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME, JOB\_ID FROM EMPLOYEES WHERE SALARY > ANY (SELECT SALARY FROM EMPLOYEES WHERE JOB\_ID='PU\_MAN') AND JOB\_ID<>'PU\_MAN';

20. Write a query to display the employee number, name (first name and last name) and job title for all employees whose salary is more than any average salary of any department.

SELECT EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME, JOB\_ID FROM EMPLOYEES WHERE SALARY > ANY (SELECT AVG(SALARY) FROM EMPLOYEES GROUP BY DEPARTMENT\_ID) ;

21. Write a query to display the employee name( first name and last name ) and department for all employees for any existence of those employees whose salary is more than 3700.

SELECT FIRST\_NAME, LAST\_NAME, DEPARTMENT\_ID FROM EMPLOYEES E WHERE EXISTS (SELECT \* FROM EMPLOYEES WHERE SALARY>3700 AND DEPARTMENT\_ID=E.DEPARTMENT\_ID);

22. Write a query to display the department id and the total salary for those departments which contains at least one employee.

SELECT DEPARTMENT\_ID, SUM(SALARY) TOTAL\_SALARY FROM EMPLOYEES GROUP BY DEPARTMENT\_ID HAVING COUNT(\*)>0;

23. Write a query to display the employee id, name (first name and last name) and the job id column with a modified title SALESMAN for those employees whose job title is ST\_MAN and

DEVELOPER for whose job title is IT\_PROG.

SELECT EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME, CASE WHEN JOB\_ID='ST\_MAN' THEN 'SALESMAN' WHEN JOB\_ID='IT\_PROG' THEN 'DEVELOPER' ELSE JOB\_ID END "JOB" FROM EMPLOYEES ;

24. Write a query to display the employee id, name (first name and last name), salary and the SalaryStatus column with a title HIGH and LOW respectively for those employees whose salary is more than and less than the average salary of all employees.

SELECT EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME, SALARY, CASE WHEN SALARY> (SELECT AVG(SALARY) FROM EMPLOYEES) THEN 'HIGH' ELSE 'LOW' END "SalaryStatus" FROM EMPLOYEES;

25. Write a query to display the employee id, name (first name and last name), Salary, AvgCompare (salary - the average salary of all employees) and the SalaryStatus column with a title HIGH and LOW respectively for those employees whose salary is more than and less than the average salary of all employees.

SELECT EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME, SALARY, SALARY-(SELECT AVG(SALARY) FROM EMPLOYEES) "AvgCompare", CASE WHEN SALARY> (SELECT AVG(SALARY) FROM EMPLOYEES) THEN 'HIGH' ELSE 'LOW' END "SalaryStatus" FROM EMPLOYEES;

26. Write a subquery that returns a set of rows to find all departments that do actually have one or more employees assigned to them.

Subquery : SELECT DISTINCT DEPARTMENT\_ID FROM EMPLOYEES;

SELECT \* FROM DEPARTMENTS D WHERE EXISTS (SELECT 'X' FROM EMPLOYEES E WHERE E.DEPARTMENT\_ID=D.DEPARTMENT\_ID) ORDER BY DEPARTMENT\_ID;

27. Write a query that will identify all employees who work in departments located in the United Kingdom.

SELECT \* FROM EMPLOYEES WHERE DEPARTMENT\_ID IN ( SELECT DEPARTMENT\_ID FROM DEPARTMENTS WHERE LOCATION\_ID IN (SELECT LOCATION\_ID FROM LOCATIONS WHERE COUNTRY\_ID IN (SELECT COUNTRY\_ID FROM COUNTRIES WHERE COUNTRY\_NAME= 'United Kingdom')));

28. Write a query to identify all the employees who earn more than the average and who work in any of the IT departments.

SELECT \* FROM EMPLOYEES WHERE SALARY>(SELECT AVG(SALARY) FROM EMPLOYEES) AND DEPARTMENT\_ID IN (SELECT DEPARTMENT\_ID FROM DEPARTMENTS WHERE DEPARTMENT\_NAME LIKE 'IT %' OR DEPARTMENT\_NAME='IT');

1. Write a query to determine who earns more than Mr. Ozer.

--I use ‘ALL’ because it is possible to have more than a person name is Mr. Ozer

SELECT \* FROM EMPLOYEES WHERE SALARY > ALL ( SELECT SALARY FROM EMPLOYEES WHERE LAST\_NAME='Ozer');

30. Write a query to find out which employees have a manager who works for a department based in the US.

SELECT \* FROM EMPLOYEES WHERE MANAGER\_ID IN (SELECT EMPLOYEE\_ID FROM EMPLOYEES WHERE DEPARTMENT\_ID IN (SELECT DEPARTMENT\_ID FROM DEPARTMENTS WHERE LOCATION\_ID IN (SELECT LOCATION\_ID FROM LOCATIONS WHERE COUNTRY\_ID ='US')));

31. Write a query which is looking for the names of all employees whose salary is greater than 50% of their department’s total salary bill.

SELECT \* FROM EMPLOYEES E WHERE SALARY> (SELECT SUM(SALARY)/2 FROM EMPLOYEES E1 WHERE E.DEPARTMENT\_ID=E1.DEPARTMENT\_ID) ORDER BY DEPARTMENT\_ID;

32. Write a query to get the details of employees who are managers.

SELECT \* FROM EMPLOYEES E WHERE EXISTS ( SELECT 'X' FROM EMPLOYEES E1 WHERE E.EMPLOYEE\_ID = E1.MANAGER\_ID);

33. Write a query to get the details of employees who manage a department.

SELECT \* FROM EMPLOYEES E WHERE EXISTS( SELECT 'X' FROM DEPARTMENTS D WHERE E.EMPLOYEE\_ID=D.MANAGER\_ID);

SELECT \* FROM EMPLOYEES E WHERE EMPLOYEE\_ID IN (SELECT DISTINCT MANAGER\_ID FROM DEPARTMENTS);

34. Write a query to display the employee id, name (first name and last name), salary, department name and city for all the employees who gets the salary as the salary earn by the employee which is maximum within the joining person January 1st, 2002 and December 31st, 2003.

SELECT E.EMPLOYEE\_ID EMP\_ID, E.FIRST\_NAME "FIRST\_NAME" , E.LAST\_NAME "LAST\_NAME", E.SALARY "EMP\_SALARY", D.DEPARTMENT\_NAME "DEPT\_NAME", L.CITY "DEPT\_CITY" FROM EMPLOYEES E JOIN DEPARTMENTS D ON E.DEPARTMENT\_ID=D.DEPARTMENT\_ID JOIN LOCATIONS L ON D.LOCATION\_ID = L.LOCATION\_ID WHERE E.SALARY =(SELECT MAX(SALARY) FROM EMPLOYEES WHERE HIRE\_DATE BETWEEN '2002-01-01' AND '2003-12-31');

35. Write a query in SQL to display the department code and name for all departments which located in the city London.

SELECT DEPARTMENT\_ID, DEPARTMENT\_NAME FROM DEPARTMENTS WHERE LOCATION\_ID = (SELECT LOCATION\_ID FROM LOCATIONS WHERE CITY='London');

SELECT DEPARTMENT\_ID, DEPARTMENT\_NAME FROM DEPARTMENTS D WHERE EXISTS (SELECT 'X' FROM LOCATIONS L WHERE L.LOCATION\_ID=D.LOCATION\_ID AND L.CITY='London');

36. Write a query in SQL to display the first and last name, salary, and department ID for all those employees who earn more than the average salary and arrange the list in descending order on salary.

SELECT FIRST\_NAME, LAST\_NAME, SALARY, DEPARTMENT\_ID FROM EMPLOYEES WHERE SALARY>(SELECT AVG(SALARY) FROM EMPLOYEES) ORDER BY SALARY DESC;

37. Write a query in SQL to display the first and last name, salary, and department ID for those employees who earn more than the maximum salary of a department which ID is 40.

SELECT FIRST\_NAME, LAST\_NAME, SALARY, DEPARTMENT\_ID FROM EMPLOYEES WHERE SALARY> (SELECT MAX(SALARY) FROM EMPLOYEES WHERE DEPARTMENT\_ID = 40);

38. Write a query in SQL to display the department name and Id for all departments where they located, that Id is equal to the Id for the location where department number 30 is located.

SELECT DEPARTMENT\_ID, DEPARTMENT\_NAME FROM DEPARTMENTS WHERE LOCATION\_ID =(SELECT LOCATION\_ID FROM DEPARTMENTS WHERE DEPARTMENT\_ID =30);

39. Write a query in SQL to display the first and last name, salary, and department ID for all those employees who work in that department where the employee works who hold the ID 201.

SELECT FIRST\_NAME, LAST\_NAME, SALARY, DEPARTMENT\_ID FROM EMPLOYEES WHERE DEPARTMENT\_ID = (SELECT DEPARTMENT\_ID FROM EMPLOYEES WHERE EMPLOYEE\_ID=201);

40. Write a query in SQL to display the first and last name, salary, and department ID for those employees whose salary is equal to the salary of the employee who works in that department which ID is 40.

SELECT FIRST\_NAME, LAST\_NAME, SALARY, DEPARTMENT\_ID FROM EMPLOYEES WHERE SALARY = ANY (SELECT SALARY FROM EMPLOYEES WHERE DEPARTMENT\_ID =40);

41. Write a query in SQL to display the first and last name, and department code for all employees who work in the department Marketing.

SELECT FIRST\_NAME, LAST\_NAME, DEPARTMENT\_ID FROM EMPLOYEES WHERE DEPARTMENT\_ID = ( SELECT DEPARTMENT\_ID FROM DEPARTMENTS WHERE DEPARTMENT\_NAME = 'Marketing');

42. Write a query in SQL to display the first and last name, salary, and department ID for those employees who earn more than the minimum salary of a department which ID is 40.

SELECT FIRST\_NAME, LAST\_NAME, SALARY, DEPARTMENT\_ID FROM EMPLOYEES WHERE SALARY >(SELECT MIN(SALARY) FROM EMPLOYEES WHERE DEPARTMENT\_ID =40);

43. Write a query in SQL to display the full name, email, and hire date for all those employees who was hired after the employee whose ID is 165.

SELECT CONCAT(FIRST\_NAME,' ',LAST\_NAME) FULL\_NAME, EMAIL, HIRE\_DATE FROM EMPLOYEES WHERE HIRE\_DATE> (SELECT HIRE\_DATE FROM EMPLOYEES WHERE EMPLOYEE\_ID = 165);

44. Write a query in SQL to display the first and last name, salary, and department ID for those employees who earn less than the minimum salary of a department which ID is 70.

SELECT FIRST\_NAME, LAST\_NAME, SALARY, DEPARTMENT\_ID FROM EMPLOYEES WHERE SALARY< (SELECT MIN(SALARY) FROM EMPLOYEES WHERE DEPARTMENT\_ID=70);

1. Write a query in SQL to display the first and last name, salary, and department ID for those employees who earn less than the average salary, and also work at the department where the employee Laura is working as a first name holder.

-- I use ‘IN’ clause because it is possible then there is more than one person having name ‘Laura’

SELECT FIRST\_NAME, LAST\_NAME, SALARY, DEPARTMENT\_ID FROM EMPLOYEES WHERE SALARY < (SELECT AVG(SALARY) FROM EMPLOYEES) AND DEPARTMENT\_ID IN ( SELECT DEPARTMENT\_ID FROM EMPLOYEES WHERE FIRST\_NAME='Laura');

46. Write a query in SQL to display the first and last name, salary, and department ID for those employees whose department is located in the city London.

SELECT FIRST\_NAME, LAST\_NAME, SALARY, DEPARTMENT\_ID FROM EMPLOYEES WHERE DEPARTMENT\_ID IN ( SELECT DEPARTMENT\_ID FROM DEPARTMENTS WHERE LOCATION\_ID =(SELECT LOCATION\_ID FROM LOCATIONS WHERE CITY='London'));

47. Write a query in SQL to display the city of the employee whose ID 134 and works there.

SELECT CITY FROM LOCATIONS WHERE LOCATION\_ID = (SELECT LOCATION\_ID FROM DEPARTMENTS WHERE DEPARTMENT\_ID=(SELECT DEPARTMENT\_ID FROM EMPLOYEES WHERE EMPLOYEE\_ID =134) );

48. Write a query in SQL to display the the details of those departments which max salary is 7000 or above for those employees who already done one or more jobs.

SELECT \* FROM DEPARTMENTS D WHERE DEPARTMENT\_ID IN ( SELECT DEPARTMENT\_ID FROM EMPLOYEES WHERE EMPLOYEE\_ID IN (SELECT EMPLOYEE\_ID FROM JOB\_HISTORY GROUP BY EMPLOYEE\_ID HAVING COUNT(EMPLOYEE\_ID) > 1) GROUP BY DEPARTMENT\_ID HAVING MAX(SALARY)>=7000);

49. Write a query in SQL to display the detail information of those departments which starting salary is at least 8000.

SELECT \* FROM DEPARTMENTS WHERE DEPARTMENT\_ID IN (SELECT DEPARTMENT\_ID FROM EMPLOYEES GROUP BY DEPARTMENT\_ID HAVING MIN(SALARY)>=8000);

50. Write a query in SQL to display the full name (first and last name) of manager who is supervising 4 or more employees.

SELECT FIRST\_NAME || ' ' || LAST\_NAME MANAGER\_NAME FROM EMPLOYEES WHERE EMPLOYEE\_ID IN (SELECT MANAGER\_ID FROM EMPLOYEES GROUP BY MANAGER\_ID HAVING COUNT(EMPLOYEE\_ID)>=4);

51. Write a query in SQL to display the details of the current job for those employees who worked as a Sales Representative in the past.

SELECT E.FIRST\_NAME || ' ' || E.LAST\_NAME EMPLOYEE\_NAME, E.JOB\_ID, J.JOB\_TITLE, E.HIRE\_DATE, E.SALARY, E.DEPARTMENT\_ID FROM EMPLOYEES E NATURAL JOIN JOBS J WHERE EMPLOYEE\_ID IN (SELECT DISTINCT EMPLOYEE\_ID FROM JOB\_HISTORY WHERE JOB\_ID=(SELECT JOB\_ID FROM JOBS WHERE JOB\_TITLE='Sales Representative'));

52. Write a query in SQL to display all the information about those employees who earn second lowest salary of all the employees.

SELECT \* FROM EMPLOYEES WHERE SALARY = (SELECT DISTINCT SALARY FROM EMPLOYEES ORDER

BY SALARY OFFSET 1 ROW LIMIT 1);

53. Write a query in SQL to display the details of departments managed by Susan.

SELECT \* FROM DEPARTMENTS WHERE MANAGER\_ID IN (SELECT EMPLOYEE\_ID FROM EMPLOYEES WHERE FIRST\_NAME ='Susan');

54. Write a query in SQL to display the department ID, full name (first and last name), salary for those employees who is highest salary drawer in a department.

SELECT DEPARTMENT\_ID, FIRST\_NAME, LAST\_NAME, SALARY FROM EMPLOYEES WHERE (DEPARTMENT\_ID,SALARY) IN (SELECT DEPARTMENT\_ID,MAX(SALARY) FROM EMPLOYEES GROUP BY DEPARTMENT\_ID);

SELECT DEPARTMENT\_ID, FIRST\_NAME || ' ' || LAST\_NAME EMPLOYEE\_NAME, SALARY FROM EMPLOYEES E WHERE SALARY = (SELECT MAX(SALARY) FROM EMPLOYEES E1 WHERE E.DEPARTMENT\_ID =E1.DEPARTMENT\_ID);

55. Write a query in SQL to display all the information of those employees who did not have any job in the past.

SELECT \* FROM EMPLOYEES WHERE EMPLOYEE\_ID NOT IN (SELECT DISTINCT EMPLOYEE\_ID FROM JOB\_HISTORY);

1. Write a query to interchange the salaries of employee 120 and 122.

--NOTE : Using Concept Swapping without using 3rd Variable

--STEP 01:

UPDATE EMPLOYEES SET SALARY =SALARY+(SELECT SALARY FROM EMPLOYEES WHERE EMPLOYEE\_ID=120) WHERE EMPLOYEE\_ID =122;

--STEP 02:

UPDATE EMPLOYEES SET SALARY =(SELECT SALARY FROM EMPLOYEES WHERE EMPLOYEE\_ID=122)-SALARY WHERE EMPLOYEE\_ID =120;

--STEP 03:

UPDATE EMPLOYEES SET SALARY =SALARY-(SELECT SALARY FROM EMPLOYEES WHERE EMPLOYEE\_ID=120) WHERE EMPLOYEE\_ID =122;

57. Write a query to increase the salary of employee 115 based on the following conditions:

If experience is more than 10 years - increase salary by 20%,

If experience is greater than 5 years - increase salary by 10%,

Otherwise 5%.

UPDATE EMPLOYEES SET SALARY = CASE WHEN EXTRACT(YEAR FROM AGE(CURRENT\_DATE,HIRE\_DATE))>=10 THEN SALARY+SALARY\*0.20 WHEN EXTRACT(YEAR FROM AGE(CURRENT\_DATE,HIRE\_DATE))>=10 THEN SALARY+SALARY\*0.10 ELSE SALARY+SALARY\*0.10 END WHERE EMPLOYEE\_ID=115;

58. Write a query to change commission percentage as follows for employee with ID = 150.

If salary is more than 10000 then commission is 0.4%,

if Salary is less than 10000 but experience is more than 10 years then 0.35%,

if salary is less than 3000 then commission is 0.25%,

In the remaining cases commission is 0.15%.

UPDATE EMPLOYEES SET COMMISSION\_PCT = CASE WHEN SALARY>10000 THEN 0.4 WHEN SALARY<10000 AND EXTRACT(YEAR FROM AGE(CURRENT\_DATE,HIRE\_DATE))>=10 THEN 0.35 WHEN SALARY<3000 THEN 0.25 ELSE 0.15 END WHERE EMPLOYEE\_ID=115;

59. Write a query to find out the name of the employee and name of the department for the employee who is managing for employee 103.

SELECT FIRST\_NAME || ' ' || LAST\_NAME EMPLOYEE\_NAME, (SELECT DEPARTMENT\_NAME FROM DEPARTMENTS D WHERE D.DEPARTMENT\_ID=E.DEPARTMENT\_ID) FROM EMPLOYEES E WHERE EMPLOYEE\_ID = (SELECT MANAGER\_ID FROM EMPLOYEES WHERE EMPLOYEE\_ID=103);

1. Write a query to display the year in which maximum number of employees joined along with how many joined in each month in that year.

SELECT EXTRACT(YEAR FROM HIRE\_DATE) "YEARS", EXTRACT(MONTH FROM HIRE\_DATE) "MONTHS", COUNT(\*) EMPLOYEE\_JOINED FROM EMPLOYEES WHERE EXTRACT(YEAR FROM HIRE\_DATE) IN (SELECT EXTRACT(YEAR FROM HIRE\_DATE) FROM EMPLOYEES GROUP BY EXTRACT(YEAR FROM HIRE\_DATE) HAVING COUNT(\*)=(SELECT COUNT(\*) FROM EMPLOYEES GROUP BY EXTRACT(YEAR FROM HIRE\_DATE) ORDER BY COUNT(\*) DESC LIMIT 1) ) GROUP BY EXTRACT(YEAR FROM HIRE\_DATE), EXTRACT(MONTH FROM HIRE\_DATE);

61. Write a query to change salary of employee 130 to the salary of the employee with first name ‘Joe’. If Joe is not found then take average salary of all employees.

-- I use ‘MAX’ function because it is possible to have more than one person having name ‘Joe’

UPDATE EMPLOYEES SET SALARY =COALESCE((SELECT MAX(SALARY) FROM EMPLOYEES WHERE FIRST\_NAME ='Joe') ,(SELECT AVG(SALARY) FROM EMPLOYEES )) WHERE EMPLOYEE\_ID=130;

62. Write a query to display 5th and 10th employees in Employees table.

SELECT \* FROM( SELECT \*, ROW\_NUMBER() OVER(ORDER BY EMPLOYEE\_ID) AS RECORD\_NUMBER FROM EMPLOYEES ) AS EMP WHERE RECORD\_NUMBER IN (5,10);