

# DATABASE SYSTEM

Name: Laiba Wazir  
Roll No: 23-AI-60  
Section: A2  
Dept: Artificial Intelligence  
Submitted To: Ms. Noor Ul Huda

## Table of Contents

Lab: 01 (Structured Query Language) .....	3
Lab: 02(Structured Query Language) .....	3
Lab: 03(SINGLE AND MULTIPLE ROW FUNCTION) .....	12
Lab: 04(EXAMPLES OF SINGLE AND MULTIPLE ROW FUNCTIONS) .....	18
Lab: 05(JOINS).....	24
Lab: 06(JOINS IN SQL EXAMPLE) .....	27
Lab: 07(ENTITY RELATIONSHIP DIAGRAM) .....	32
Lab: 08(SUBQUERY IN SQL) .....	36
Lab: 09(CREATE TABLES) .....	40
Lab: 10(DATA MANIPULATION OPERATION IN SQL) .....	45
Lab: 11(OPEN ENDED LAB 1) .....	46
Lab: 12(CREATE AND INSERT TABLE) .....	47
Lab: 13(REATING VIEW) .....	52

## LAB # 1 & 2

### (Structured Query Language)

1. Retrieve all records from the EMP table:

```
SELECT *FROM EMP;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7499	ALLEN	SALESMAN	7698	2/20/1981	1840	300	20
7521	WARD	SALESMAN	7698	2/22/1981	1250	500	30
7566	JONES	MANAGER	7839	4/2/1981	2975	-	20
7654	MARTIN	SALESMAN	7698	9/28/1981	1250	1400	30
7698	BLAKE	MANAGER	7839	5/1/1981	2850	-	30
7782	CLARK	MANAGER	7839	6/9/1981	2450	-	10
7788	SCOTT	ANALYST	7566	12/9/1982	3000	-	20
7839	KING	PRESIDENT	-	11/17/1981	5000	-	10
7844	TURNER	SALESMAN	7698	9/8/1981	1500	0	30
7876	ADAMS	CLERK	7698	1/12/1983	1100	-	20

2. Retrieve all records from the DEPT table:

```
SELECT *FROM DEPT;
```

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

rows returned in 0.01 seconds    [Download](#)

3. Retrieve the ENAME and DEPTNO columns from the EMP table:

```
SELECT ENAME, DEPTNO FROM EMP;
```

ENAME	DEPTNO
ALLEN	20
WARD	30
JONES	20
MARTIN	30
BLAKE	30
CLARK	10
SCOTT	20
KING	10
TURNER	30

4. Retrieve the ENAME and EMPNO columns from the EMP table

SELECT ENAME, EMPNO FROM EMP;

ENAME	EMPNO
ALLEN	7499
WARD	7521
JONES	7566
MARTIN	7654
BLAKE	7698
CLARK	7782
SCOTT	7788
KING	7839

5. Retrieve records from EMP where the ENAME is 'JONES':

SELECT \*FROM EMP WHERE ENAME='JONES';

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7566	JONES	MANAGER	7839	4/2/1981	2975	-	20

1 rows returned in 0.01 seconds    [Download](#)

6. Retrieve records from EMP where EMPNO is 7499:

SELECT \*FROM EMP WHERE EMPNO=7499;

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7499	ALLEN	SALESMAN	7698	2/20/1981	1840	300	20

1 rows returned in 0.01 seconds [Download](#)

7. Retrieve EMPNO and ENAME from EMP where SAL is greater than 2500:

SELECT ENAME, EMPNO FROM EMP WHERE SAL>=2500;

ENAME	EMPNO
JONES	7566
BLAKE	7698
SCOTT	7788
KING	7839
FORD	7902

5 rows returned in 0.00 seconds [Download](#)

8. Retrieve EMPNO and JOB from EMP where SAL is between 1500 and 2500:

SELECT EMPNO, JOB, SAL FROM EMP WHERE SAL>=1500 AND SAL<=2500;

EMPNO	JOB	SAL
7499	SALESMAN	1840
7521	SALESMAN	1250
7654	SALESMAN	1250
7782	MANAGER	2450
7844	SALESMAN	1500
7876	CLERK	1100
7900	CLERK	950
7934	CLERK	1300

8 rows returned in 0.01 seconds [Download](#)

9. Retrieve ENAME from EMP where HIREDATE is before May 5, 1982:

SELECT ENAME FROM EMP WHERE HIREDATE<TO\_DATE('15/05/1982', 'DD/MM/YYYY');

ENAME
ALLEN
WARD
JONES
MARTIN
BLAKE
CLARK
KING
TURNER

10. Retrieve records from EMP where DEPTNO in 10 - 30

SELECT ENAME FROM EMP WHERE DEPTNO=10 OR DEPTNO=30;

ENAME
WARD
MARTIN
BLAKE
CLARK
KING
TURNER
JAMES

11. Retrieve ENAME from EMP where SAL in 10 - 30:

SELECT ENAME FROM EMP WHERE DEPTNO IN (10,30);

ENAME
WARD
MARTIN
BLAKE
CLARK
KING

12. Retrieve ENAME from EMP where SAL is any 10 - 30:

SELECT ENAME FROM EMP WHERE DEPTNO =ANY (10,30);

ENAME
WARD
MARTIN
BLAKE
CLARK
KING
TURNER
JAMES

13. Retrieve ENAME from EMP where SAL is all 10 - 30:

```
SELECT ENAME FROM EMP WHERE DEPTNO =ALL (10,30);
```

Results	Explain	Describe	Saved SQL
no data found			

14. Retrieve ENAME from EMP where SAL is not between 10 and 30:

```
SELECT ENAME FROM EMP WHERE SAL NOT BETWEEN 1500 AND 2500;
```

ENAME
WARD
JONES
MARTIN
BLAKE
SCOTT

15. Retrieve records from EMP where the ENAME is 'ALLEN':

```
SELECT *FROM EMP WHERE ENAME='ALLEN';
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7499	ALLEN	SALESMAN	7698	2/20/1981	1840	300	20

16. Retrieve records from EMP where ENAME have not this Alphabet

```
SELECT FROM EMP WHERE ENAME LIKE '%S%';
```

17. Retrieve ENAME from EMP where ENAME start with A

```
SELECT ENAME FROM EMP WHERE ENAME LIKE 'A%';
```

ENAME
ALLEN
ADAMS

2 rows returned in 0.00 seconds    [Download](#)

18. Retrieve ENAME from EMP where ENAME ends with K

```
SELECT ENAME FROM EMP WHERE ENAME LIKE '%K';
```

ENAME
CLARK

1 rows returned in 0.01 seconds    [Download](#)

19. Retrieve ENAME from EMP where ENAME have last second Alphabet "E"

```
SELECT ENAME FROM EMP WHERE ENAME LIKE '%E_';
```

ENAME
ALLEN
JONES
TURNER
JAMES
MILLER

5 rows returned in 0.01 seconds    [Download](#)

20. Retrieve ENAME from EMP where ENAME have 3rd last Alphabet IS "A"

```
SELECT ENAME FROM EMP WHERE ENAME LIKE '%A__';
```

ENAME
WARD
BLAKE
CLARK
ADAMS

4 rows returned in 0.00 seconds [Download](#)

21. Retrieve the employee names, department numbers, and locations for employees working in the 'D. ALLAS' location.

```
SELECT EMP.ENAME, EMP.DEPTNO, DEPT.LOC FROM EMP, DEPT WHERE EMP.DEPTNO =
DEPT.DEPTNO AND DEPT.LOC = 'DALLAS';
```

ENAME	DEPTNO	LOC
ALLEN	20	DALLAS
JONES	20	DALLAS
SCOTT	20	DALLAS
ADAMS	20	DALLAS
FORD	20	DALLAS

5 rows returned in 0.02 seconds [Download](#)

22. Find all the unique job titles in the employee table.

```
SELECT UNIQUE JOB FROM EMP;
```

JOB
SALESMAN
ANALYST
CLERK
MANAGER
PRESIDENT

5 rows returned in 0.00 seconds [Download](#)

23. Retrieve the current system date and the database user

```
SELECT SYSDATE FROM DUAL;
```

SYSDATE	
1/21/2025	
1 rows returned in 0.01 seconds	<a href="#">Download</a>

24. Show the employee names, department numbers, and their salary tripled.

```
SELECT EMP.ENAME,DEPT.DEPTNO,EMP.SAL*3 FROM EMP ,DEPT;
```

ENAME	DEPTNO	EMP.SAL*3
ALLEN	10	5520
WARD	10	3750
JONES	10	8925
MARTIN	10	3750
BLAKE	10	8550
CLARK	10	7350
SCOTT	10	9000
KING	10	15000
TURNER	10	4500
ADAMS	10	3300

25. List all employees who are not managers.

```
SELECT ENAME FROM EMP WHERE JOB !='MANAGER';
```

ENAME
ALLEN
WARD
MARTIN
SCOTT
KING
TURNER

26. List the names and jobs of employees whose job title is neither clerk, manager, nor analyst:

```
SELECT ENAME,JOB FROM EMP WHERE JOB NOT IN('CLERK','MANAGER','ANALYST');
```

ENAME	JOB
ALLEN	SALESMAN
WARD	SALESMAN
MARTIN	SALESMAN
KING	PRESIDENT
TURNER	SALESMAN

5 rows returned in 0.01 seconds    [Download](#)

27. Retrieve employee names and their annual salaries (monthly salary \* 12), ordered by annual salary:

```
SELECT ENAME, SAL * 12 AS ANNUAL_SALARY FROM EMP ORDER BY ANNUAL_SALARY;
```

ENAME	ANNUAL_SALARY
JAMES	11400
ADAMS	13200
WARD	15000
MARTIN	15000
MILLER	15600
TURNER	18000
ALLEN	22080

## Lab # 03

### (Single-row and multiple-row functions in SQL)

- How can you display employee names with job titles, ensuring the names have the first letter capitalized and the job titles are in lowercase?

SELECT INITCAP(ENAME) AS EMPLOYEE\_NAME, LOWER(JOB) AS JOB FROM EMP;

EMPLOYEE_NAME	JOB
Allen	salesman
Ward	salesman
Jones	manager
Martin	salesman
Blake	manager
Clark	manager
Scott	analyst
King	president
Turner	salesman

- How can you retrieve the employee number, name in uppercase, and department number for the employee named 'Blake' (case-insensitive)?

SELECT EMPNO, UPPER(ENAME) AS EMPLOYEE\_NAME, DEPTNO FROM EMP WHERE UPPER(ENAME)='BLAKE';

EMPNO	EMPLOYEE_NAME	DEPTNO
7698	BLAKE	30

1 rows returned in 0.01 seconds    [Download](#)

- How can you display the names of all employees along with the length of their names?

SELECT ENAME, LENGTH(ENAME) AS LENGTH\_OF\_NAME FROM EMP;

ENAME	LEHGTH_OF_NAME
ALLEN	5
WARD	4
JONES	5
MARTIN	6
BLAKE	5
CLARK	5
SCOTT	5
KING	4

4. How can you find the position of the letter 'R' in the names of all employees?

```
SELECT ENAME, INSTR(ENAME, 'R') AS POSITION_OF_R FROM EMP;
```

ENAME	POSITION_OF_R
ALLEN	0
WARD	3
JONES	0
MARTIN	3
BLAKE	0
CLARK	4
SCOTT	0

5. How can you concatenate the employee names with their manager numbers?

```
SELECT ENAME || " " || DEPTNO AS EMPLOYEE_NAME_WITH_DEPTNO FROM EMP;
```

EMPLOYEE_NAME_WITH_DEPTNO
ALLEN20
WARD30
JONES20
MARTIN30
BLAKE30
CLARK10

6. How can you concatenate the employee names with their job titles?

```
SELECT ENAME || ' - ' || JOB AS EMPLOYEE_NAME_WITH_JOB FROM EMP;
```

EMPLOYEE_NAME_WITH_JOB	
ALLEN	- SALESMAN
WARD	- SALESMAN
JONES	- MANAGER
MARTIN	- SALESMAN
BLAKE	- MANAGER
CLARK	- MANAGER

7. How can you extract a substring from the string ‘string’, starting at the second character and of length five?

```
SELECT SUBSTR('STRING',4,2)AS SUBSTRING_COLUMN FROM DUAL;
```

SUBSTRING_COLUMN	
IN	
1 rows returned in 0.00 seconds <a href="#">Download</a>	

8. How can you extract a substring from each employee’s name, starting from the second character and up to ten characters long?

```
SELECT ENAME, SUBSTR(ENAME, 2, 10) AS SUBSTRING_NAME FROM EMP;
```

ENAME	SUBSTRING_NAME
ALLEN	LLEN
WARD	ARD
JONES	ONES
MARTIN	RTIN
BLAKE	LAKF

9. How can you extract a substring of two characters starting from the second character of each employee’s name?

```
SELECT ENAME, SUBSTR(ENAME, 2, 2) AS SUBSTRING_TWO_CHAR FROM EMP;
```

ENAME	SUBSTRING_TWO_CHAR
ALLEN	LL
WARD	AR
JONES	ON
MARTIN	AR
BLAKE	LA
CLARK	LA
SCOTT	CO
KING	IN

10. How can you extract the first four characters from the string ‘noor ul huda’?

SELECT SUBSTR('NOOR UL HUDA',1,4) AS SUNSTRING\_COLUMN FROM DUAL;

SUNSTRING_COLUMN
NOOR
1 rows returned in 0.01 seconds <a href="#">Download</a>

11. How can you extract a substring from the string ‘noor ul huda’ starting from position 0 and of length four?

SELECT SUBSTR('noor ul huda', 1, 4) AS SUBSTRING\_START\_ZERO FROM DUAL;

SUBSTRING_START_ZERO
noor
1 rows returned in 0.01 seconds <a href="#">Download</a>

12. How can you extract a substring starting from the 8th character from the end of the string ‘noor ul huda’, with a length of 10 characters?

SELECT SUBSTR('noor ul huda', -8, 10) AS SUBSTRING\_FROM\_END FROM DUAL;

SUBSTRING_FROM_END
ul huda
1 rows returned in 0.00 seconds <a href="#">Download</a>

13. How can you retrieve the employee number, the concatenation of employee name and job title, the length of the employee name, and the position of the letter ‘A’ in the name, for employees whose job titles start with ‘SALES’?

```
SELECT EMPNO, ENAME || '' || JOB AS EMP_JOB, LENGTH(ENAME) AS NAME_LENGTH,
INSTR(ENAME, 'A') AS POSITION_OF_A FROM EMP WHERE JOB LIKE 'SALES%';
```

Results	Explain	Describe	Saved SQL	History
EMPNO	EMP_JOB	NAME_LENGTH	POSITION_OF_A	
7499	ALLEN SALESMAN	5	1	
7521	WARD SALESMAN	4	2	
7654	MARTIN SALESMAN	6	2	
7844	TURNER SALESMAN	6	0	

4 rows returned in 0.01 seconds    [Download](#)

14. How can you round the number 45.923 to 2 decimal places, 0 decimal places, and to the nearest 10?

```
SELECT ROUND(45.923, 2) AS ROUND_TWO_DECIMALS, ROUND(45.923, 0) AS
ROUND_ZERO_DECIMALS, ROUND(45.923, -1) AS ROUND_NEAREST_TEN FROM DUAL;
```

ROUND_TWO_DECIMALS	ROUND_ZERO_DECIMALS	ROUND_NEAREST_TEN
45.92	46	50

1 rows returned in 0.00 seconds    [Download](#)

15. Calculate the average salary of employees

```
SELECT AVG(SAL) AS AVERAGE_OF_EMPLOYEES FROM EMP;
```

AVERAGE_OF_EMPLOYEES
2189.61538461538461538461538461538461539

1 rows returned in 0.00 seconds    [Download](#)

16. SELECT AVG(SAL) AS AVERAGE\_OF\_EMPLOYEES FROM EMP WHERE DNAME='SALES';

```
SELECT MAX(SAL) AS MAXIMUM_SALARY FROM EMP;
```

ASMAXIMUM_SALARY	
5000	
1 rows returned in 0.01 seconds	<a href="#">Download</a>

17. Calculate the total number of employees grouped by job title

SELECT JOB,COUNT(\*) FROM EMP GROUP BY JOB;

JOB	COUNT(*)
SALESMAN	4
ANALYST	2
CLERK	3
MANAGER	3
PRESIDENT	1

5 rows returned in 0.00 seconds [Download](#)

**LAB: 04****(EXAMPLES OF SINGLE AND MULTIPLE ROW FUNCTIONS)**

1. SELECT \* FROM EMP ORDER BY SAL ASC;

OUTPUT:

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	12/17/1980	800	-	20
7900	JAMES	CLERK	7698	12/03/1981	950	-	30
7876	ADAMS	CLERK	7788	01/12/1983	1100	-	20
7521	WARD	SALESMAN	7698	02/22/1981	1250	500	30
7654	MARTIN	SALESMAN	7698	09/28/1981	1250	1400	30
7934	MILLER	CLERK	7782	01/23/1982	1300	-	10
7844	TURNER	SALESMAN	7698	09/08/1981	1500	0	30
7499	ALLEN	SALESMAN	7698	02/20/1981	1600	300	30

2. SELECT ENAME, HIREDATE FROM EMP ORDER BY HIREDATE DESC;

OUTPUT:

ENAME	HIREDATE
ADAMS	01/12/1983
SCOTT	12/09/1982
MILLER	01/23/1982
FORD	12/05/1981
JAMES	12/03/1981
KING	11/17/1981
MARTIN	09/28/1981
TURNER	09/08/1981

3. SELECT \* FROM EMP ORDER BY DEPTNO, SAL ASC;

OUTPUT:

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7934	MILLER	CLERK	7782	01/23/1982	1300	-	10
7782	CLARK	MANAGER	7839	06/09/1981	2450	-	10
7839	KING	PRESIDENT	-	11/17/1981	5000	-	10
7369	SMITH	CLERK	7902	12/17/1980	800	-	20
7876	ADAMS	CLERK	7788	01/12/1983	1100	-	20
7566	JONES	MANAGER	7839	04/02/1981	2975	-	20
7788	SCOTT	ANALYST	7566	12/09/1982	3000	-	20
7902	FORD	ANALYST	7566	12/03/1981	3000	-	20

laiba.wazir0@gmail.com laiba-wazir #en

Copyright © 1999-2024, Oracle and/or its affiliates.

Oracle APD12415

4. SELECT ENAME FROM EMP ORDER BY JOB ASC, SAL DESC;

OUTPUT:

ENAME
FORD
SCOTT
MILLER
ADAMS
JAMES
SMITH
JONES
BLAKE

5. SELECT JOB, SUM(SAL) AS TOTAL\_SAL FROM EMP GROUP BY JOB ORDER BY JOB ASC;  
 OUTPUT:

JOB	TOTAL_SAL
ANALYST	6000
CLERK	4150
MANAGER	8275
PRESIDENT	5000
SALESMAN	5600

6. SELECT DEPTNO, ROUND(AVG(SAL), 2) AS AVG\_SAL FROM EMP GROUP BY DEPTNO;  
 OUTPUT:

DEPTNO	AVG_SAL
30	1566.67
10	2916.67
20	2175

7. SELECT DNAME, LOC, COUNT(EMPNO), ROUND(AVG(SAL), 2) FROM DEPT D, EMP E WHERE D.DEPTNO = E.DEPTNO GROUP BY DNAME, LOC;  
 OUTPUT:

DNAME	LOC	COUNT(EMPNO)	ROUND(AVG(SAL),2)
RESEARCH	DALLAS	5	2175
ACCOUNTING	NEW YORK	3	2916.67
SALES	CHICAGO	6	1566.67

8. SELECT LOC, COUNT(EMPNO) FROM EMP, DEPT WHERE EMP.DEPTNO = DEPT.DEPTNO GROUP BY LOC;  
 OUTPUT:

LOC	COUNT(EMPNO)
NEW YORK	3
DALLAS	5
CHICAGO	6

9. SELECT JOB, SUM(COMM) FROM EMP GROUP BY JOB;

OUTPUT:

JOB	SUM(COMM)
CLERK	-
SALESMAN	2200
ANALYST	-
MANAGER	-
PRESIDENT	-

10. SELECT DEPTNO, MIN(HIREDATE) FROM EMP GROUP BY DEPTNO;

OUTPUT:

DEPTNO	MIN(HIREDATE)
30	02/20/1981
10	06/09/1981
20	12/17/1980

11. SELECT ENAME, NVL(TO\_CHAR(COMM), 'NO COMMISION') AS "COMM/NON-COMM" FROM EMP;

OUTPUT:

ENAME	COMM/NON-COMM
SMITH	NO COMMISION
ALLEN	300
WARD	500
JONES	NO COMMISION
MARTIN	1400
BLAKE	NO COMMISION
CLARK	NO COMMISION
SCOTT	NO COMMISION

12. SELECT ENAME, NVL2(COMM, 'HAS COMMISSION', 'N/A') AS "COMM/NON-COMM EMPLOYEES" FROM EMP;

OUTPUT:

ENAME	COMM/NON-COMM EMPLOYEES
SMITH	N/A
ALLEN	HAS COMMISSION
WARD	HAS COMMISSION
JONES	N/A
MARTIN	HAS COMMISSION
BLAKE	N/A
CLARK	N/A
SCOTT	N/A

13. SELECT ENAME FROM EMP WHERE COMM IS NULL;

OUTPUT:

ENAME
SMITH
JONES
BLAKE
CLARK
SCOTT
KING
ADAMS

14. SELECT ENAME, COMM FROM EMP WHERE COMM IS NOT NULL;

OUTPUT:

ENAME	COMM
ALLEN	300
WARD	500
MARTIN	1400
TURNER	0

15. SELECT JOB FROM EMP WHERE COMM IS NULL;

OUTPUT:

JOB
CLERK
MANAGER
MANAGER
MANAGER
ANALYST
PRESIDENT
CLERK
CLERK
ANALYST
CLERK

16. SELECT ENAME, MGR, COMM FROM EMP WHERE COMM IS NOT NULL AND COMM != 0;

OUTPUT:

ENAME	MGR	COMM
ALLEN	7698	300
WARD	7698	500
MARTIN	7698	1400

17. SELECT E.DEPTNO, SUM(SAL) AS "TOTAL\_SALARY" FROM EMP E, DEPT D WHERE E.DEPTNO =

D.DEPTNO GROUP BY E.DEPTNO ORDER BY SUM(SAL) DESC;

OUTPUT:

DEPTNO	TOTAL_SALARY
20	10875
30	9400
10	8750

18. SELECT JOB, AVG(SAL) AS "AVERAGE\_SALARY" FROM EMP GROUP BY JOB ORDER BY JOB ASC,

AVG(SAI) ASC;

## OUTPUT:

19. SELECT E.DEPTNO, ROUND(AVG(SAL), 2) AS "AVERAGE\_SALARY" FROM EMP E, DEPT D WHERE E.DEPTNO = D.DEPTNO GROUP BY E.DEPTNO HAVING ROUND(AVG(SAL), 2)>=2000 ORDER BY

AVG(SAL) A

OUTPUT:	
DEPTNO	AVERAGE_SALARY
20	2175
10	2916.67

20. SELECT DEPTNO, COUNT(\*) AS "EMPLOYEE\_COUNT" FROM EMP GROUP BY DEPTNO HAVING COUNT(\*) > 3;

## OUTPUT:

DEPTNO	EMPLOYEE_COUNT
30	6
20	5

21. SELECT MAX(AVG(SAL)) AS "MAX\_IN\_AVG\_SALARIES" FROM EMP GROUP BY SAL;  
OUTPUT:

MAX_IN_AVG_SALARIES
5000

22. SELECT MIN(SUM(SAL)) AS MIN\_TOTAL\_SALARY FROM EMP GROUP BY JOB;  
OUTPUT:

MIN_TOTAL_SALARY
4150

1 rows returned in 0.01 seconds    [Download](#)

**LAB: 05****(JOINS)****EXERCISES**

- i. To display the employee name, department name, and location of all employees who earn a commission.

```
SELECT E.ENAME, D.DNAME, D.LOC FROM EMP E JOIN DEPT D ON E.DEPTNO = D.DEPTNO
WHERE E.COMM IS NOT NULL;
```

ENAME	DNAME	LOC
ALLEN	RESEARCH	DALLAS
WARD	SALES	CHICAGO
MARTIN	SALES	CHICAGO
TURNER	SALES	CHICAGO

4 rows returned in 0.02 seconds    [Download](#)

- ii. To display all the employee's name (including KING who has no manager) and their manager name.

```
SELECT W.ENAME AS Employee, M.ENAME AS Manager FROM EMP W LEFT JOIN EMP M ON
W.MGR = M.EMPNO;
```

EMPLOYEE	MANAGER
SCOTT	JONES
FORD	JONES
ALLEN	BLAKE
WARD	BLAKE
MARTIN	BLAKE

- iii. To display the name of all employees whose manager is **KING**.

```
SELECT E.ENAME FROM EMP E JOIN EMP M ON E.MGR = M.EMPNO WHERE M.ENAME = 'KING';
```

ENAME
JONES
BLAKE
CLARK

3 rows returned in 0.01 seconds    [Download](#)

- iv. Write a query to display the name, job, department number and department name for all employees who work in DALLAS.

```
SELECT E.ENAME, E.JOB, E.DEPTNO, D.DNAME FROM EMP E JOIN DEPT D ON E.DEPTNO = D.DEPTNO WHERE D.LOC = 'DALLAS';
```

ENAME	JOB	DEPTNO	DNAME
ALLEN	SALESMAN	20	RESEARCH
JONES	MANAGER	20	RESEARCH
SCOTT	ANALYST	20	RESEARCH
ADAMS	CLERK	20	RESEARCH
FORD	ANALYST	20	RESEARCH

5 rows returned in 0.01 seconds    [Download](#)

- v. Display the employee name and employee number along with their manager's name Manager Number. Label the columns Employee, Emp#, Manager, and Manager#, respectively.

```
SELECT W.ENAME AS Employee, W.EMPNO AS Emp#, M.ENAME AS Manager, M.EMPNO AS Manager# FROM EMP W LEFT JOIN EMP M ON W.MGR = M.EMPNO;
```

EMPLOYEE	EMP#	MANAGER	MANAGER#
SCOTT	7788	JONES	7566
FORD	7902	JONES	7566
ALLEN	7499	BLAKE	7698
WARD	7521	BLAKE	7698
MARTIN	7654	BLAKE	7698
TURNER	7844	BLAKE	7698

 laibawazir60@gmail.com  
 Laiba wazir 

Copyright © 1999, 2024, Oracle and/or its  
affiliates.

Oracle APEX 24.2.0

## LAB: 06

### (JOINS IN SQL EXAMPLE)

**EXERCISE:**

- Fetch following details

Result: Order No, Order Date, Product Name

```
SELECT O.OrderID, E.FirstName || ' ' || E.LastName AS EmployeeName FROM Orders O JOIN
Employees E ON O.EmployeeID = E.EmployeeID WHERE O.ShippedDate > O.RequiredDate;
```

ORDERID	EMPLOYEE NAME
10451	Margaret Peacock
10483	Robert King
10515	Andrew Fuller
10523	Robert King
10545	Laura Callahan
10264	Michael Suvama

Q laibawazir60@gmail.com Copyright © 1999-2024, Oracle and/or its affiliates. All rights reserved. Oracle APEX 24.2.0

- Fetch following details

Result: Order No, Order Date, Product Name, Customer Name

```
SELECT ORDERID, ORDERDATE, PRODUCTNAME, CONTACTNAME FROM ORDERDETAILS
NATURAL JOIN ORDERS NATURAL JOIN PRODUCTS NATURAL JOIN CUSTOMERS ORDER
BY ORDERDATE
```

ORDERID	ORDERDATE	PRODUCTNAME	CONTACTNAME
10248	7/4/1996	Mozzarella di Giovanni	Paul Henriot
10498	4/7/1997	Singaporean Hokkien Fried Mee	Carlos Hernández
10498	4/7/1997	Boston Crab Meat	Carlos Hernández
10498	4/7/1997	Guaraná Fantástica	Carlos Hernández
10499	4/8/1997	Rössle Sauerkraut	Carlos González

Q laibawazir60@gmail.com Copyright © 1999, 2024, Oracle and/or its affiliates. All rights reserved. Oracle APEX 24.2

## 3. Fetch following details

Result: Order No, Order Date, Product Name, Category Name, Customer Name

```
SELECT o.ORDERID AS "Order No", o.ORDERDATE AS "Order Date", p.PRODUCTNAME AS "Product Name", c.CATEGORYNAME AS "Category Name", cu.CONTACTNAME AS "Customer Name" FROM ORDERS o JOIN ORDERDETAILS od ON o.ORDERID = od.ORDERID JOIN PRODUCTS p ON od.PRODUCTID = p.PRODUCTID JOIN CATEGORIES c ON p.CATEGORYID = c.CATEGORYID JOIN CUSTOMERS cu ON o.CUSTOMERID = cu.CUSTOMERID ORDER BY o.ORDERDATE;
```

<b>Order No</b>	<b>Order Date</b>	<b>Product Name</b>	<b>Category Name</b>	<b>Customer Name</b>
10248	7/4/1996	Queso Cabrales	Dairy Products	Paul Henriot
10248	7/4/1996	Singaporean Hokkien Fried Mee	Grains/Cereals	Paul Henriot
10248	7/4/1996	Mozzarella di Giovanni	Dairy Products	Paul Henriot
10249	7/5/1996	Tofu	Produce	Karin Josephs
10249	7/5/1996	Manjimup Dried Apples	Produce	Karin Josephs
10250	7/8/1996	Jack's New England Clam Chowder	Seafood	Mario Pontes

 laibawazir60@gmail.com

Copyright © 1999, 2024, Oracle and/or its

Oracle APEX 24.2.0

## 4. Select all orders having products belonging to 'Sea Food' category

Result: OrderNo, OrderDate, Product Name

```
SELECT ORDERS.ORDERID, ORDERS.ORDERDATE, PRODUCTS.PRODUCTNAME
FROM CATEGORIES
JOIN PRODUCTS ON CATEGORIES.CATEGORYID = PRODUCTS.CATEGORYID
JOIN ORDERDETAILS ON PRODUCTS.PRODUCTID = ORDERDETAILS.PRODUCTID
JOIN ORDERS ON ORDERDETAILS.ORDERID = ORDERS.ORDERID
WHERE CATEGORIES.CATEGORYNAME = 'Seafood';
```

ORDERID	ORDERDATE	PRODUCTNAME
10436	2/5/1997	Spegesild
10444	2/12/1997	Jack's New England Clam Chowder
10448	2/17/1997	Boston Crab Meat
10449	2/18/1997	Ikura
10450	2/19/1997	Ikura
10454	2/21/1997	Spegesild

5. List suppliers in the order of no. of products supplied (Supplier Name, No Of Products).

Result: Supplier Name, No. of Products

```

SELECT SUPPLIERS.COMPANYNAME AS SUPPLIERNAME, COUNT(PRODUCTS.PRODUCTID)
AS
NO_OF_PRODUCTS
FROM SUPPLIERS
JOIN PRODUCTS
ON SUPPLIERS.SUPPLIERID = PRODUCTS.SUPPLIERID
GROUP BY SUPPLIERS.COMPANYNAME

```

SUPPLIERNAME	NO_OF_PRODUCTS
Grandma Kelly's Homestead	3
Svensk Sjöföda AB	3
Mayumi's	3
Pavlova, Ltd.	5
Nord-Ost-Fisch Handelsgesellschaft mbH	1
Pasta Buttini s.r.l.	2

6. Select Suppliers supplying more than 4 products.

Result: Supplier Name

```
SELECT SUPPLIERS.COMPANYNAME AS SUPPLIERNAME
FROM SUPPLIERS
JOIN PRODUCTS ON SUPPLIERS.SUPPLIERID = PRODUCTS.SUPPLIERID
GROUP BY SUPPLIERS.COMPANYNAME
HAVING COUNT(PRODUCTS.PRODUCTID) > 4;
```

SUPPLIERNAME
Pavlova, Ltd.
Plutzer Lebensmittelgrossmärkte AG

2 rows returned in 0.01 seconds     [Download](#)

7. Fetch no. of employees working in each region.

(RegionName, No. of employees)

```
SELECT e.Region, COUNT(e.EmployeeID) AS No_of_Employees FROM Employees e GROUP BY
e.Region
```

REGION	NO_OF_EMPLOYEES
WA	5
-	4

2 rows returned in 0.02 seconds     [Download](#)

8. Fetch no. of employees in each region. If there is no employee in any region, even then region name should appear in the list with employee count of 0.

(RegionName, No. of employees)

```
SELECT C.CompanyName AS CustomerName
FROM Customers C
LEFT JOIN Orders O ON C.CustomerID = O.CustomerID
```

WHERE O.OrderID IS NULL;

CUSTOMERNAME
Paris spcialit�s
FISSA Fabrica Inter. Salchichas S.A.

2 rows returned in 0.01 seconds    [Download](#)

9. Fetch Customers who have not placed any order.  
(Customer Name)

SELECT c.CompanyName AS CustomerName FROM Customers c LEFT JOIN Orders o ON c.CustomerID = o.CustomerID WHERE o.OrderID IS NULL

CUSTOMERNAME
Paris spcialit�s
FISSA Fabrica Inter. Salchichas S.A.

2 rows returned in 0.01 seconds    [Download](#)

[Oracle Database Home Page](#)    Copyright © 1999-2024, Oracle Corporation. All rights reserved.    [Oracle Database 24c](#)

1. Select Top 3 employees of company. Employees are ranked on the basis of no. of orders they have processed.  
(*ROWNUM <= 3, ORDER BY number of orders processed*)

SELECT e.EmployeeID, e.FirstName, e.LastName, COUNT(o.OrderID) AS NumberOfOrders FROM Employees e JOIN Orders o ON e.EmployeeID = o.EmployeeID GROUP BY e.EmployeeID, e.FirstName, e.LastName ORDER BY NumberOfOrders DESC FETCH FIRST 3 ROWS ONLY;

EMPLOYEEID	FIRSTNAME	LASTNAME	NUMBEROFORDERS
4	Margaret	Peacock	156
3	Janet	Leverling	127
1	Nancy	Davolio	123

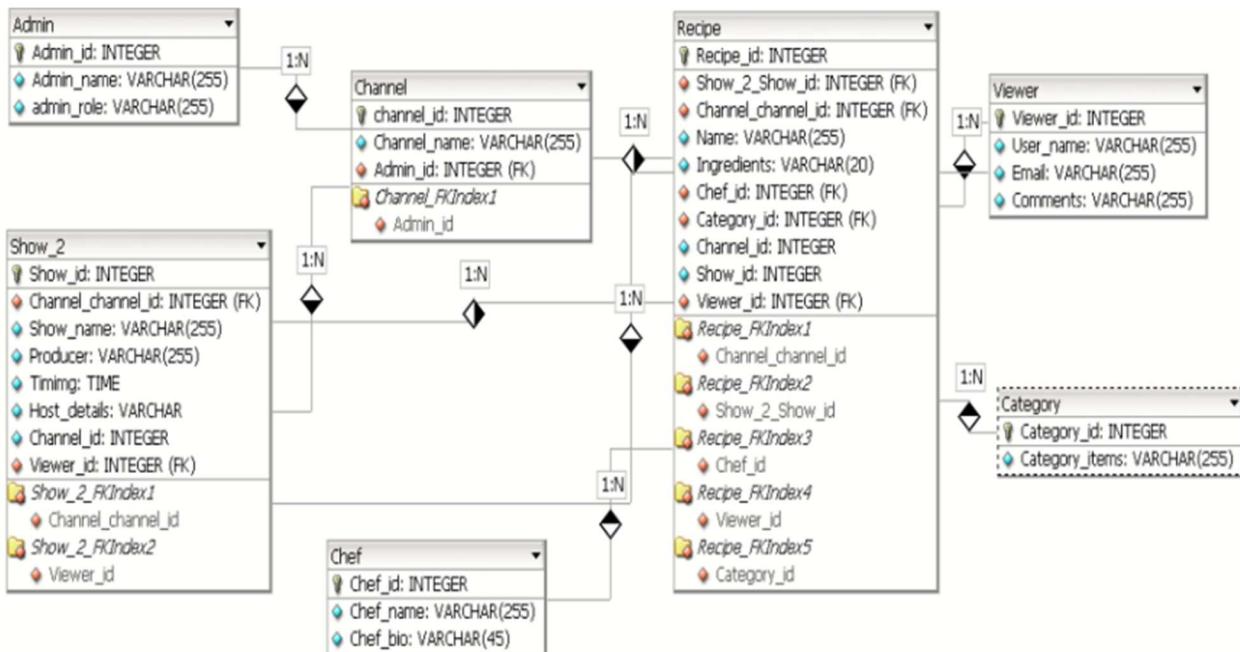
3 rows returned in 0.02 seconds    [Download](#)

## LAB: 07

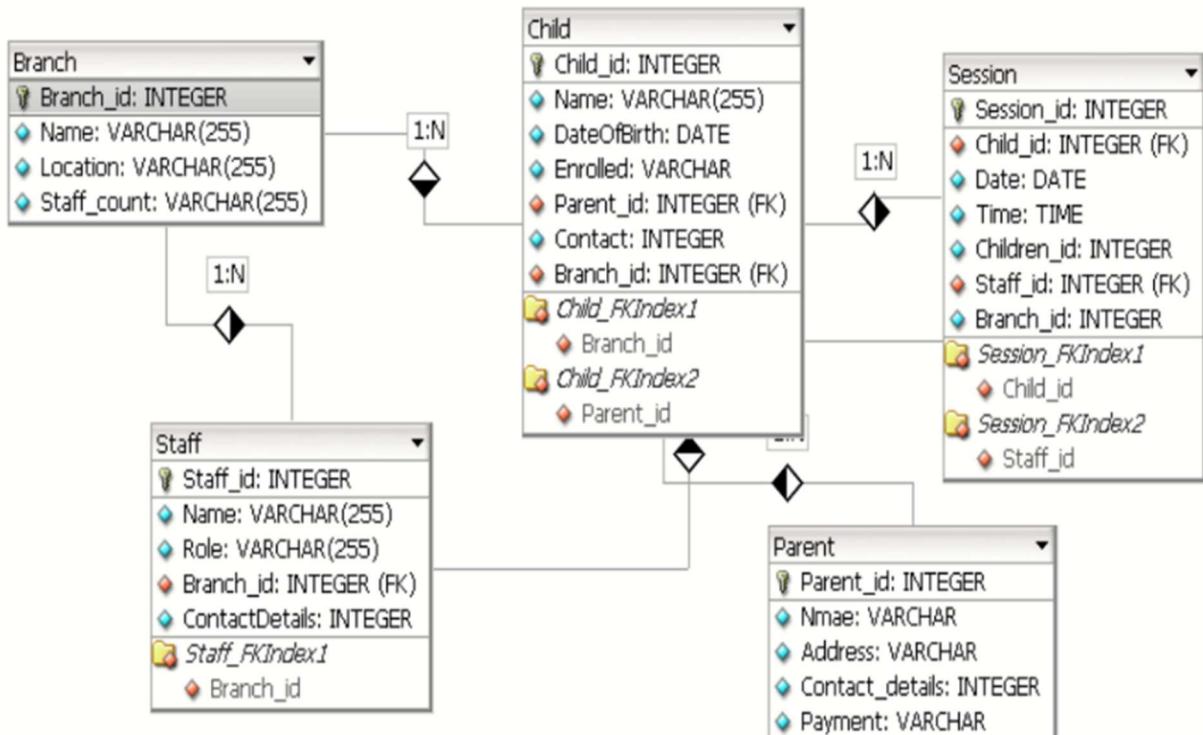
### (ENTITY RELATIONSHIP DIAGRAM)

**Question 1:**

Food is one of human beings' favorite obsessions. Most people spend a great deal of time physically and mentally preoccupied with food: we organize and prepare meals, we daydream about what's for lunch or dinner; we eat. "We are What We Eat" As food is a major part of every one's life so cooking channels are getting so much popularity nowadays. Cooking Channels are television specialty channels that only on air programs about food and cooking. A number of cooking shows have run for many seasons on channels. Many of the more popular cooking shows have had flamboyant hosts whose unique personalities have made them into celebrities. From recent past, famous chefs are hosting the shows. The organization keeps the records of these shows, name, producer, timing etc along with the detailed information about the hosts. Administrator has full rights on this data as he/she is responsible for maintaining the system. The channels also store the cooking recipes categories wise. There are categories like Breakfast, Salad Entrees, Soups, Fish, Chicken, Meat, Vegetarian Entrees, Side Salad/Dressings, Side Vegetables, and Desserts. Chefs can enter/edit/delete the recipes. Viewers can view these recipes any time through website. Viewers can also enter their own recipes along with their own record. They can also comment on the recipes of chefs and other people and on shows of the channel. All the users have to register themselves to access the system. Mr. X wants to open a new cooking channel and has approached you to create a system to store the whole data related to cooking shows and recipes.

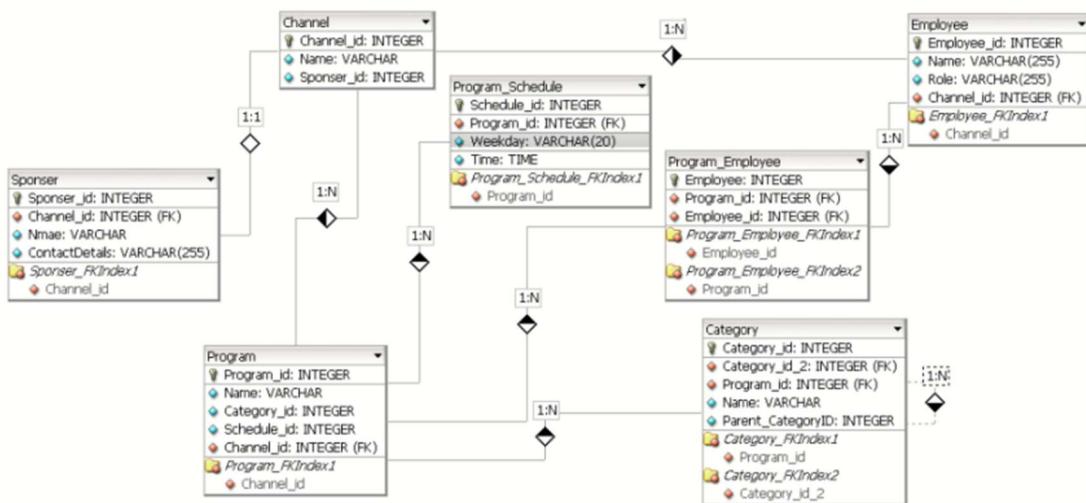
**Question 2:**

Nightingale's is a private nursery organization which takes in children from the age of one year old up to the age of five years. Nightingale's has 5 branches in the North Wales. The Nursery previously had a paper based system for all of its administration and would now like to part-computerize the administrative and personnel side of the business. (For the moment the payroll (holidays, absences and salaries) for staff will not be dealt with). Each nursery can employ up to 50 staff and will need to have a minimum of 10 staff at each nursery:- these are either Manager, Nursery Room Teachers, administrative staff or auxiliary staff (e.g. cleaners and caretakers). At any one time each member of staff will only work at one branch. Children enroll on sessions at the nursery. Most children will be enrolled for a whole term at a time. (They may attend either morning or afternoon sessions or both. Also if a child is enrolled in all the morning sessions they may occasionally stay for afternoon sessions.) A register has to be kept for each session listing the children attending the session and the nursery room teachers taking the session. Only the nursery room teachers will take sessions with the children. There may be more than one nursery room teacher in each session. There are different numbers of children in each session as some children come for a few mornings or afternoons per week and other children are full-time. (Please note that there is no time-tabling system detailing the activities; children arrive for sessions and are then taken for various activities by the nursery room teachers) For the purposes of the database one of the child's parents will be regarded as having enrolled the child and will be responsible for the payment of fees. The fees for the nursery are calculated at a session rate. A session is regarded as a morning or an afternoon. Children can attend for any number of sessions in a week, with the maximum being 10 sessions (the nursery is not open on Saturdays and Sundays). At the moment the rate for each session for each child is £20. There are no reductions for having two children from the same family enrolled. Parents pay monthly. The staff and children will need to have the names, addresses and phone numbers of up to two emergency contact people.



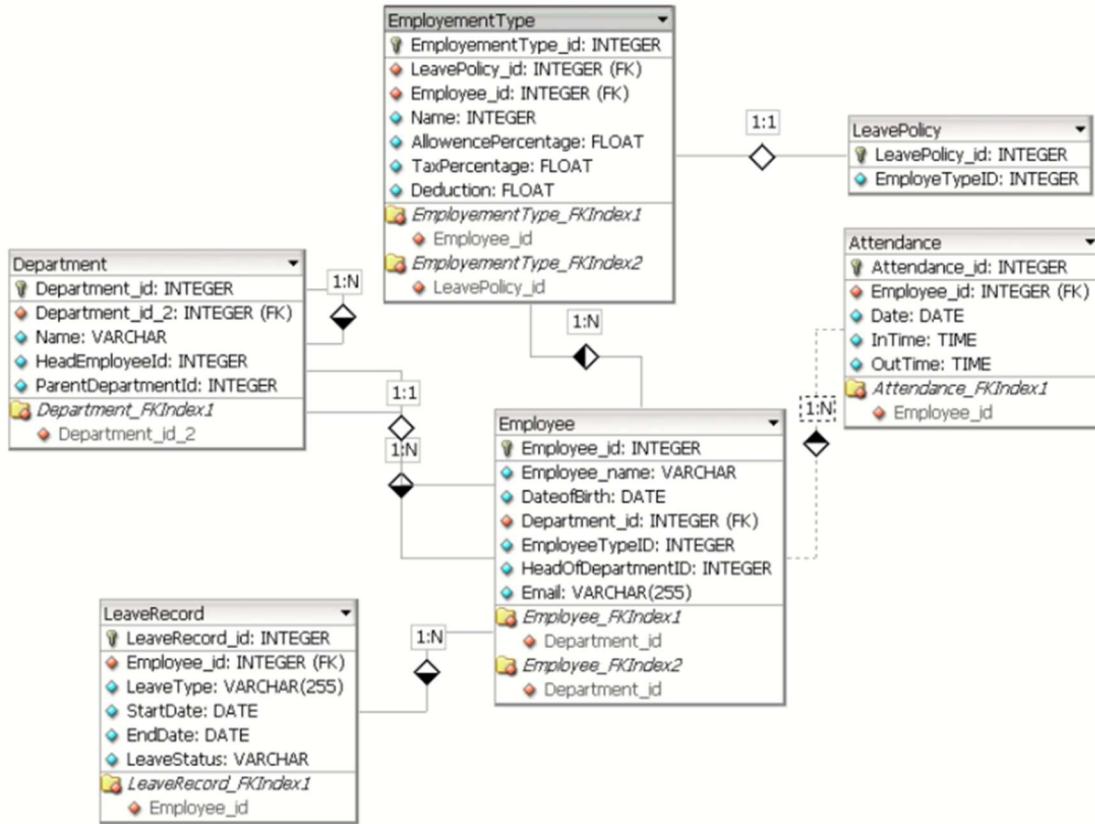
### Question 3:

A software system is meant to maintain information about all TV programs telecasted on different channels and their schedules. Here are particulars of that system: There are several channels. Each channel is sponsored by an organization. Different programs are telecasted on different channels. Each Program belongs to a certain category. Categories can further be subdivided into sub-categories, hence forming a hierarchy (E.g. Programs can be categorized as Entertainment, News, Sports, Religious or Informative. ‘Entertainment’ can further be sub-divided into Drama, Stage Shows, Movies) Every channel has its own list of employees. Many Employees are involved in the preparation of programs. Each employee can work on many programs. Each employee has its defined role in every program e.g. Actor, Producer, Director, Technician, Anchor etc System needs to keep information about program schedule i.e. weekday and time a program is telecasted. One program can be telecasted more than once in a week.



## Question 4:

A company has several employees working in different departments. Departments can further be grouped into sub-departments. Each department has an employee as head of the department. Each employee is assigned certain employment type (e.g. Permanent/Contractual/Daily wages) and on the basis of this type, salary structure is applied. Salary structure defines % allowance, % deductions and % tax. Attendance register is maintained for each employee on daily basis recording daily attendance of employees and their sign-in and sign-out timings. Allowed leaves track is also maintained so that if employee has availed 90% of his leaves, he should be given a warning notice.



**LAB: 08****(SUBQUERY IN SQL)**

1. Write down SQL queries to perform the following functions: -

- i. To display the employee number and name for all employees who earn more than the average salary.  
Sort the results in descending order of salary.

**QUERY:**

```
SELECT EMPNO, ENAME, SAL FROM EMP WHERE SAL > (SELECT AVG(SAL) FROM EMP)
ORDER BY SAL DESC;
```

**OUTPUT:**

EMPNO	ENAME	SAL
7839	KING	5000
7902	FORD	3000
7788	SCOTT	3000
7566	JONES	2975
7698	BLAKE	2850
7782	CLARK	2450

- ii. To display the employee name and salary of all employees who report to *king*.

**QUERY:**

```
SELECT ENAME, SAL FROM EMP WHERE MGR = (SELECT EMPNO FROM EMP WHERE
ENAME = 'KING');
```

**OUTPUT:**

ENAME	SAL
JONES	2975
BLAKE	2850
CLARK	2450

3 rows returned in 0.02 seconds    [Download](#)

iii. To display the department number, name and job for all employees in the *Sales* department.

**QUERY:**

```
SELECT DEPTNO, ENAME, JOB FROM EMP WHERE DEPTNO = (SELECT DEPTNO FROM DEPT WHERE DNAME = 'SALES');
```

**OUTPUT:**

DEPTNO	ENAME	JOB
30	WARD	SALESMAN
30	MARTIN	SALESMAN
30	BLAKE	MANAGER
30	TURNER	SALESMAN
30	JAMES	CLERK

6 rows returned in 0.02 seconds    [Download](#)

iv. To display the name, hiredate and salary for all employees who have both the same salary and commission as *scott*.

**QUERY:**

```
SELECT ENAME, HIREDATE, SAL FROM EMP WHERE SAL = (SELECT SAL FROM EMP WHERE ENAME = 'SCOTT') AND COMM = (SELECT COMM FROM EMP WHERE ENAME = 'SCOTT');
```

**OUTPUT:**

Results	Explain	Des
<b>no data found</b>		

v. To display the employee name, department number, and job title for all employees whose location is *Dallas*.

**QUERY:**

```
SELECT ENAME, DEPTNO, JOB FROM EMP WHERE DEPTNO = (SELECT DEPTNO FROM DEPT WHERE LOC = 'DALLAS');
```

**OUTPUT:**

ENAME	DEPTNO	JOB
SMITH	20	CLERK
JONES	20	MANAGER
SCOTT	20	ANALYST
ADAMS	20	CLERK
FORD	20	ANALYST

5 rows returned in 0.01 seconds [Download](#)

vi. Write a query to display the employee name and hiredate for all employees in the same department as Blake. Exclude Blake.

**QUERY:**

```
SELECT ENAME, HIREDATE FROM EMP WHERE DEPTNO = (SELECT DEPTNO FROM EMP WHERE ENAME = 'BLAKE') AND ENAME <> 'BLAKE';
```

**OUTPUT:**

ENAME	HIREDATE
ALLEN	2/20/1981
WARD	2/22/1981
MARTIN	9/28/1981
TURNER	9/8/1981
JAMES	12/3/1981

5 rows returned in 0.01 seconds [Download](#)

vii. Display the employee number, name and salary for all employees who earn more than the average salary and who work in department with any employee with a T in their name.

**QUERY:**

```
SELECT EMPNO, ENAME, SAL FROM EMP WHERE SAL > (SELECT AVG(SAL) FROM EMP) AND DEPTNO IN (SELECT DEPTNO FROM EMP WHERE ENAME LIKE '%T%');
```

**OUTPUT:**

EMPNO	ENAME	SAL
7566	JONES	2975
7788	SCOTT	3000
7902	FORD	3000
7698	BLAKE	2850

4 rows returned in 0.01 seconds [Download](#)

**LAB: 9****(Creating and managing tables)****1. EMPLOYEE\_TABLE:**

```
CREATE TABLE EMPLOYEE_TABLE (
    EmplNo NUMBER PRIMARY KEY,
    Employee_Name VARCHAR2(100),
    Designation VARCHAR2(100),
    Qualification VARCHAR2(100),
    JoinDate DATE
);
```

Column Name	Data Type	Nullable	Default	Primary Key	Comment	Identity
EMPLNO	NUMBER	N		1		
EMPLOYEE_NAME	VARCHAR2(100 BYTE)	Y				
DESIGNATION	VARCHAR2(100 BYTE)	Y				
QUALIFICATION	VARCHAR2(100 BYTE)	Y				
JOINDATE	DATE	Y				

**2. GRADE\_TABLE:**

```
CREATE TABLE GRADE_TABLE (
    Designation VARCHAR2(100) PRIMARY KEY,
    Grade NUMBER CHECK (Grade BETWEEN 1 AND 20),
    TotalPosts NUMBER,
    PostsAvailable NUMBER (20),
    CONSTRAINTS CHK_POST CHECK (PostsAvailable <= TotalPosts)
);
```

Column Name	Data Type	Nullable	Default	Primary Key	Comment	Identity
DESIGNATION	VARCHAR2(100 BYTE)	N		1		
GRADE	NUMBER	Y				
TOTALPOSTS	NUMBER	Y				
POSTSAVAILABLE	NUMBER(20,0)	Y				

### 3. PROJECT\_TABLE:

```
CREATE TABLE PROJECT_TABLE (
    PID NUMBER(4) PRIMARY KEY,
    Title VARCHAR2(200),
    Client VARCHAR2(100),
    DurationInWeeks NUMBER,
    Status VARCHAR2(20) CHECK (Status IN ('New', 'In Progress', 'Complete'))
);
```

Column Name	Data Type	Nullable	Default	Primary Key	Comment	Identity
PID	NUMBER(4,0)	N		1		
TITLE	VARCHAR2(200 BYTE)	Y				
CLIENT	VARCHAR2(100 BYTE)	Y				
DURATIONINWEEKS	NUMBER	Y				
STATUS	VARCHAR2(20 BYTE)	Y				

### 4. TRAINING\_TABLE:

```
CREATE TABLE TRAINING_TABLE (
    Tcode VARCHAR2(10) PRIMARY KEY,
    Title VARCHAR2(200),
    StartDate DATE,
    EndDate DATE
);
```

Column Name	Data Type	Nullable	Default	Primary Key	Comment	Identity
TCODE	VARCHAR2(10 BYTE)	N		1		
TITLE	VARCHAR2(200 BYTE)	Y				
STARTDATE	DATE	Y				
ENDDATE	DATE	Y				

## 5. EMP\_PROJECT\_TABLE:

```
CREATE TABLE EMP_PROJECT_TABLE (
    EmplNo NUMBER,
    PID NUMBER,
    Performance VARCHAR2(20) CHECK (Performance IN ('Excellent', 'Good', 'Fair', 'Bad', 'Poor')),
    PRIMARY KEY (EmplNo, PID),
    CONSTRAINT FK_EMP_PROJECT_EMP FOREIGN KEY (EmplNo) REFERENCES
EMPLOYEE_TABLE (EmplNo),
    CONSTRAINT FK_EMP_PROJECT_PROJ FOREIGN KEY (PID) REFERENCES
PROJECT_TABLE (PID)
);
```

Column Name	Data Type	Nullable	Default	Primary Key	Comment	Identity
EMPLNO	NUMBER	N		1		
PID	NUMBER	N		2		
PERFORMANCE	VARCHAR2(20 BYTE)	Y				

## 6. EMP\_TRAINING\_TABLE:

```
CREATE TABLE EMP_TRAINING_TABLE (
    EmplNo NUMBER,
    Tcode VARCHAR2(10),
    Attendance NUMBER CHECK (Attendance >= 0 AND Attendance <= 100),
    PRIMARY KEY (EmplNo, Tcode),
```

CONSTRAINT FK\_EMP\_TRAINING\_EMP FOREIGN KEY (EmplNo) REFERENCES EMPLOYEE\_TABLE (EmplNo),

CONSTRAINT FK\_EMP\_TRAINING\_TRAINING FOREIGN KEY (Tcode) REFERENCES TRAINING\_TABLE (Tcode)

);

EMP_TRAINING_TABLE							
Columns	Data	Indexes	Constraints	Grants	Statistics	Triggers	Dependencies
Column Name	Data Type	Nullable	Default		Primary Key	Comment	Identity
EMPLNO	NUMBER	N			1		
TCODE	VARCHAR2(10 BYTE)	N			2		
ATTENDANCE	NUMBER	Y					

1. Write SQL statements to add

A. *Gender* column to **EMP** table. The only possible values are *Male* and *Female*.

#### QUERY:

ALTER TABLE EMPLOYEE\_TABLE

ADD GENDER VARCHAR2(10) CHECK (GENDER IN ('MALE','FEMALE'));

EMPLOYEE_TABLE							
Columns	Data	Indexes	Constraints	Grants	Statistics	Triggers	Dependencies
Column Name	Data Type	Nullable	Default		Primary Key	Comment	Identity
EMPLNO	NUMBER	N			1		
EMPLOYEE_NAME	VARCHAR2(100 BYTE)	Y					
DESIGNATION	VARCHAR2(100 BYTE)	Y					
QUALIFICATION	VARCHAR2(100 BYTE)	Y					
JOINDATE	DATE	Y					
GENDER	VARCHAR2(10 BYTE)	Y					

B. *Instructor\_Name* column to **TRAINING** table.

#### QUERY:

ALTER TABLE TRAINING\_TABLE ADD INSTRUCTOR\_NAME VARCHAR2(20);

TRAINING_TABLE							
Columns	Data	Indexes	Constraints	Grants	Statistics	Triggers	Dependencies
+ Add Column		Modify Column		AI Rename Column		Drop Column	
Column Name	Data Type	Nullable	Default			Primary Key	Comment
TCODE	VARCHAR2(10 BYTE)	N				1	
TITLE	VARCHAR2(200 BYTE)	Y					
STARTDATE	DATE	Y					
ENDDATE	DATE	Y					
INSTRUCTOR_NAME	VARCHAR2(20 BYTE)	Y					

C. *Salary* column to **GRADE** table.

#### QUERY:

```
ALTER TABLE GRADE_TABLE ADD SALARY NUMBER (10);
```

GRADE_TABLE							
Columns	Data	Indexes	Constraints	Grants	Statistics	Triggers	Dependencies
+ Add Column		Modify Column		AI Rename Column		Drop Column	
Column Name	Data Type	Nullable	Default			Primary Key	Comment
DESIGNATION	VARCHAR2(100 BYTE)	N				1	
GRADE	NUMBER	Y					
TOTALPOSTS	NUMBER	Y					
POSTSAVAILABLE	NUMBER(20,0)	Y					
SALARY	NUMBER(10,0)	Y					

2. What is *database schema*? What are the different objects included in it?

#### ANSWER:

A database schema is the formal structure that defines how data is organized within a database. It acts as a blueprint that specifies the organization of data elements and their relationships.

The key objects in a database schema include tables (which store data in rows and columns), columns (defining data attributes), primary and foreign keys (establishing relationships between tables), views (virtual tables based on queries), indexes (improving data retrieval speed), stored procedures (saved SQL code), triggers (automated actions), and constraints (rules governing data values).

## LAB: 10

### DATA MANIPULATION OPERATION IN SQL

1. Define Transaction. How is it terminated? Describe the different operations included in a transaction.

**Answer:**

A transaction is a logical work unit containing one or more SQL statements executed to perform a specific task. Transactions ensure data consistency and integrity in a database.

**Termination of Transactions:**

- A transaction is terminated when:
  1. A COMMIT statement is issued, making all changes permanent.
  2. A ROLLBACK statement is issued, discarding all pending changes.
  3. A DDL (e.g., CREATE, DROP) or DCL (e.g., GRANT, REVOKE) statement is executed.
  4. The user exits the SQL environment.
  5. A system failure occurs.

**Operations Included in a Transaction:**

1. INSERT: Add new rows to a table.
2. UPDATE: Modify existing rows in a table.
3. DELETE: Remove rows from a table.

**2. Write down SQL statements to perform following functions:-**

- i. Increase the salary by 250 of all clerks with a salary less than 900 .

**query:**

```
UPDATE EMP SET SAL = SAL + 250 WHERE JOB = 'CLERK' AND SAL < 900;
```

- ii. Transfer the employee with number 7890 to department 20 and increase his salary by 15%.

**query:**

```
UPDATE EMP SET DEPTNO = 20, SAL = SAL * 1.15 WHERE EMPNO = 7499
```

- iii. Increase the salary of employee with number 7369 by 10% of the salary of employee with number 7499.

**query:**

```
UPDATE EMP SET SAL = SAL + (SELECT SAL * 0.10 FROM EMP WHERE EMPNO = 7499) WHERE EMPNO = 7369;
```

iv. Assign to employee 7876 the same manager as the employee 7900.

**query:**

```
UPDATE EMP
```

```
SET MGR = (SELECT MGR FROM EMP WHERE EMPNO = 7900) WHERE EMPNO = 7876;
```

v. Remove all employees who were hired before 1981.

**query:**

```
DELETE FROM EMP WHERE HIREDATE < TO_DATE('01-JAN-1981', 'DD-MON-YYYY');
```

#### UPDATED EMP TABLE:

Results	Explain	Describe	Saved SQL	History				
EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	
7499	ALLEN	SALESMAN	7698	2/20/1981	1840	300	20	
7521	WARD	SALESMAN	7698	2/22/1981	1250	500	30	
7566	JONES	MANAGER	7839	4/2/1981	2975	-	20	
7654	MARTIN	SALESMAN	7698	9/28/1981	1250	1400	30	
7698	BLAKE	MANAGER	7839	5/1/1981	2850	-	30	
7782	CLARK	MANAGER	7839	6/9/1981	2450	-	10	
7788	SCOTT	ANALYST	7566	12/9/1982	3000	-	20	
7839	KING	PRESIDENT	-	11/17/1981	5000	-	10	
7844	TURNER	SALESMAN	7698	9/8/1981	1500	0	30	
7876	ADAMS	CLERK	7698	1/12/1983	1100	-	20	

## LAB: 12

### CREATE AND INSERT TABLES

**1. Write the SQL Code that will create the table of above structure (table name should be like TABLENAME\_-23XXX-CE**

Table Name	
<b>Employee</b>	
EMP_NUM	NUMBER
EMP_NAME	VARCHAR2
EMP_FNAME	VARCHAR2
EMP_INITIAL	VARCHAR2(1)
EMP_HIREDATE	DATE
JOB_CODE	NUMBER
EMP_YEAR	NUMBER

```
CREATE TABLE EMPLOYEE_23XXX_CE ( EMP_NUM NUMBER PRIMARY KEY, EMP_NAME VARCHAR2(100),
EMP_FNAME VARCHAR2(100), EMP_INITIAL VARCHAR2(1), EMP_HIREDATE DATE, JOB_CODE NUMBER,
EMP_YEAR NUMBER );
```

EMPLOYEE_23XXX_CE							
Columns	Data	Indexes	Constraints	Grants	Statistics	Triggers	Dependencies
Column Name	Data Type	Nullable	Default	Primary Key	Comment	Identity	DDL
EMP_NUM	NUMBER	N		1			
EMP_NAME	VARCHAR2(100 BYTE)	Y					
EMP_FNAME	VARCHAR2(100 BYTE)	Y					
EMP_INITIAL	VARCHAR2(1 BYTE)	Y					
EMP_HIREDATE	DATE	Y					
JOB_CODE	NUMBER	Y					

Table Name	
<b>Assignment</b>	
ASSIGN_NUM	NUMBER
ASSIGN_DATE	DATE
PRO_NUM	NUMBER
EMP_NUM	NUMBER
ASSIGN_JOB	NUMBER
ASSIGN_CHG_HR	NUMBER
ASSIGN_HOURS	NUMBER
ASSIGN_CHARGE	NUMBER

```
CREATE TABLE ASSIGNMENT_23XXX_CE ( ASSIGN_NUM NUMBER PRIMARY KEY, ASSIGN_DATE DATE,
PRO_NUM NUMBER, EMP_NUM NUMBER, ASSIGN_JOB NUMBER, ASSIGN_CHG_HR NUMBER,
ASSIGN_HOURS NUMBER, ASSIGN_CHARGE NUMBER, FOREIGN KEY (EMP_NUM) REFERENCES
EMPLOYEE_23XXX_CE(EMP_NUM), FOREIGN KEY (PRO_NUM) REFERENCES
PROJECT_23XXX_CE(PROJ_NUM), FOREIGN KEY (ASSIGN_JOB) REFERENCES JOB_23XXX_CE(JOB_CODE)
);
```

ASSIGNMENT_23XXX_CE									
Columns	Data	Indexes	Constraints	Grants	Statistics	Triggers	Dependencies	DDL	Sample Queries
+ Add Column	Modify Column	Rename Column	Drop Column	UI Defaults	Refresh	More			
Column Name	Data Type	Nullable	Default		Primary Key	Comment	Identity		
ASSIGN_NUM	NUMBER	Y							
PRO_NUM	NUMBER	Y							
EMP_NUM	NUMBER	Y							
ASSIGN_JOB	NUMBER	Y							
ASSIGN_CHG_HR	NUMBER	Y							
ASSIGN_HOURS	NUMBER	Y							
ASSIGN_CHARGE	NUMBER	Y							

---

## Job

JOB_CODE	NUMBER
JOB_DESCRIPTION	VARCHAR2
JOB_CHG_HR	NUMBER
JOB_LAST-UPDATE	DATE

CREATE TABLE JOB\_23XXX\_CE ( JOB\_CODE NUMBER PRIMARY KEY, JOB\_DESCRIPTION VARCHAR2(100),  
JOB\_CHG\_HR NUMBER, JOB\_LAST\_UPDATE DATE );

Column Name	Data Type	Nullable	Default	Primary Key	Comment	Identity
JOB_CODE	NUMBER	N		1		
JOB_DESCRIPTION	VARCHAR2(100 BYTE)	Y				
JOB_CHG_HR	NUMBER	Y				
JOB_LAST_UPDATE	DATE	Y				

## Project

PROJ_NUM	NUMBER
PROJ_NAME	VARCHAR2
PROJ_VALUE	NUMBER
PROJ_BALANCE	NUMBER
EMP_NUM	NUMBER

---

CREATE TABLE PROJECT\_23XXX\_CE ( PROJ\_NUM NUMBER PRIMARY KEY, PROJ\_NAME VARCHAR2(100),  
PROJ\_VALUE NUMBER, PROJ\_BALANCE NUMBER, EMP\_NUM NUMBER, FOREIGN KEY (EMP\_NUM)  
REFERENCES EMPLOYEE\_23XXX\_CE(EMP\_NUM) );

**PROJECT\_23XXX\_CE**

Columns	Data	Indexes	Constraints	Grants	Statistics	Triggers	Dependencies	DDL	Sample Queries
+ Add Column	Modify Column	Rename Column	Drop Column	UI Defaults	Refresh	More			
Column Name	Data Type	Nullable	Default		Primary Key	Comment			Identity
PROJ_NUM	NUMBER	N			1				
PROJ_NAME	VARCHAR2(100 BYTE)	Y							
PROJ_VALUE	NUMBER	Y							
PROJ_BALANCE	NUMBER	Y							
EMP_NUM	NUMBER	Y							

1. Perform INSERT Command in the above tables of database ConstructCo (please find the data on page 3-4)

INSERT INTO EMPLOYEE:

```
INSERT INTO EMPLOYEE_23XXX_CE (EMP_NUM, EMP_NAME, EMP_FNAME, EMP_INITIAL, EMP_HIREDATE, JOB_CODE, EMP_YEAR) VALUES (1001, 'John Doe', 'John', 'D', TO_DATE('01-JAN-2020', 'DD-MON-YYYY'), 101, 2020);
```

Results Explain Describe Saved SQL History

EMP_NUM	EMP_NAME	EMP_FNAME	EMP_INITIAL	EMP_HIREDATE	JOB_CODE	EMP_YEAR
1001	John Doe	John	D	1/1/2020	101	2020

INSERT INTO JOB:

```
INSERT INTO JOB_23XXX_CE (JOB_CODE, JOB_DESCRIPTION, JOB_CHG_HR, JOB_LAST_UPDATE) VALUES (101, 'Consultant', 50, TO_DATE('01-JAN-2021', 'DD-MON-YYYY'))
```

Results Explain Describe Saved SQL History

JOB_CODE	JOB_DESCRIPTION	JOB_CHG_HR	JOB_LAST_UPDATE
101	Consultant	50	1/1/2021

1 rows returned in 0.02 seconds [Download](#)

INSERT INTO ASSIGNMENT:

```
INSERT INTO ASSIGNMENT_23XXX_CE (ASSIGN_NUM, ASSIGN_DATE, PRO_NUM, EMP_NUM, ASSIGN_JOB, ASSIGN_CHG_HR, ASSIGN_HOURS, ASSIGN_CHARGE) VALUES (3001, TO_DATE('10-JAN-2021', 'DD-MON-YYYY'), 2001, 1001, 101, 50, 100, 5000);
```

Results Explain Describe Saved SQL History

ASSIGN_NUM	ASSIGN_DATE	PRO_NUM	EMP_NUM	ASSIGN_JOB	ASSIGN_CHG_HR	ASSIGN_HOURS	ASSIGN_CHARGE
3001	1/10/2021	2001	1001	101	50	100	5000

1 rows returned in 0.02 seconds [Download](#)

INSERT INTO PROJECT:

```
INSERT INTO PROJECT_23XXX_CE (PROJ_NUM, PROJ_NAME, PROJ_VALUE, PROJ_BALANCE, EMP_NUM)
VALUES (2001, 'Building Construction', 500000, 100000, 1001);
```

PROJ_NUM	PROJ_NAME	PROJ_VALUE	PROJ_BALANCE	EMP_NUM
2001	Building Construction	500000	100000	1001

rows returned in 0.01 seconds [Download](#)

3.Using the EMPLOYEE, JOB, and PROJECT tables in the ConstructCo database, write the SQL code that will produce the results shown as in below figure.

PROJ_NAME	PROJ_VALUE	PROJ_BALANCE	EMP_LNAME	EMP_FNAME	EMP_INITIAL	JOB_CODE	JOB_DESCRIPTION	JOB_CHG_HOUR
Rolling Tide	805000.00	500345.20	Senior	David	H	501	Systems Analyst	96.75
Evergreen	1453500.00	1002350.00	Arbough	June	E	500	Programmer	35.75
Starflight	2650500.00	2309880.00	Alonzo	Maria	D	500	Programmer	35.75
Amber Wave	3500500.00	2110346.00	Washington	Ralph	B	501	Systems Analyst	96.75

```
SELECT E.EMP_FNAME, E.EMP_NAME, E.EMP_INITIAL, J.JOB_DESCRIPTION, J.JOB_CHG_HR,
P.PROJ_NAME, P.PROJ_VALUE FROM EMPLOYEE_23XXX_CE E LEFT JOIN JOB_23XXX_CE J ON
E.JOB_CODE = J.JOB_CODE LEFT JOIN PROJECT_23XXX_CE P ON E.EMP_NUM = P.EMP_NUM;
```

EMP_FNAME	EMP_NAME	EMP_INITIAL	JOB_DESCRIPTION	JOB_CHG_HR	PROJ_NAME	PROJ_VALUE
John	John Doe	D	Consultant	50	Building Construction	500000

rows returned in 0.02 seconds [Download](#)

4.Write the SQL code to calculate the ASSIGN\_CHARGE values in the ASSIGNMENT table in the ConstructCo database. Note that ASSIGN\_CHARGE is a derived attribute that is calculated by multiplying ASSIGN\_CHG\_HR by ASSIGN\_HOURS.

```
UPDATE ASSIGNMENT_23XXX_CE SET ASSIGN_CHARGE = ASSIGN_CHG_HR * ASSIGN_HOURS;
```

ASSIGN_NUM	ASSIGN_DATE	PRO_NUM	EMP_NUM	ASSIGN_JOB	ASSIGN_CHG_HR	ASSIGN_HOURS	ASSIGN_CHARGE
3001	1/10/2021	2001	1001	101	50	100	5000

1 rows returned in 0.00 seconds [Download](#)

## LAB: 13

### (CREATING VIEWS)

#### **EXERCISES:**

i. Create views for following purposes: i. To display each designation and number of employees with that particular designation.

```
CREATE VIEW emp_designation_count AS SELECT job AS designation, COUNT(empno) AS num_of_employees FROM emp GROUP BY job;
```

DESIGNATION	NUM_OF_EMPLOYEES
SALESMAN	4
ANALYST	2
CLERK	3
MANAGER	3
PRESIDENT	1

5 rows returned in 0.02 seconds [Download](#)

ii. To display employee number, employee name, project title and employee performance in that project.

```
CREATE VIEW emp_project_performance AS SELECT e.empno, e.ename, p.project_title, ep.performance
FROM emp e JOIN emp_project ep ON e.empno = ep.empno JOIN project p ON ep.project_id =
p.project_id;
```

iii. To display employee number, employee name and number of projects in which employee performance is excellent.

```
CREATE VIEW emp_excellent_projects AS SELECT e.empno, e.ename, COUNT(ep.project_id) AS num_of_excellent_projects
FROM emp e JOIN emp_project ep ON e.empno = ep.empno WHERE
ep.performance = 'Excellent' GROUP BY e.empno, e.ename
```