Assignment - 2

DBMS

Name: Kamithkar Vinod
Course: PG DAC AUGUST 2025
PRN: 250850320040
Form No: 250500480
Date: 17-10-2025

Problem 1:

Task: Display each employee's name and hiredate from department 20.

Code:

```
SELECT ename, hiredate, deptno
FROM emp
WHERE deptno = 20;
```

Output:

```
mysql> SELECT ename, hiredate FROM emp WHERE deptno = 20;
+-------+------------+
| ename | hiredate   |
+-------+------------+
| SMITH | 1980-12-17 |
| JONES | 1981-04-02 |
| SCOTT | 1987-07-13 |
| ADAMS | 1987-07-12 |
| FORD  | 1981-12-03 |
+-------+------------+
```

Problem 2:

Task: Display each employee's name with hiredate and salary review date. Assume review date is one year after hiredate.

Code:

```
SELECT ename,
       hiredate,
```

```
        DATE_ADD(hiredate, INTERVAL 1 YEAR) AS "Salary Review Date
           "
FROM emp;
```

Output: —

```
mysql> SELECT ename, hiredate,
    -> REPLACE(hiredate, year(hiredate), 1 + year(hiredate)) "Salary Review Date"
    -> FROM emp;
+--------+------------+--------------------+
| ename  | hiredate   | Salary Review Date |
+--------+------------+--------------------+
| SMITH  | 1980-12-17 | 1981-12-17         |
| ALLEN  | 1981-02-20 | 1982-02-20         |
| WARD   | 1981-02-21 | 1982-02-21         |
| JONES  | 1981-04-02 | 1982-04-02         |
| MARTIN | 1981-09-08 | 1982-09-08         |
| BLAKE  | 1981-05-01 | 1982-05-01         |
| CLARK  | 1981-06-09 | 1982-06-09         |
| SCOTT  | 1987-07-13 | 1988-07-13         |
| KING   | 1981-11-17 | 1982-11-17         |
| TURNER | 1981-09-08 | 1982-09-08         |
| ADAMS  | 1987-07-12 | 1988-07-12         |
| JAMES  | 1981-12-03 | 1982-12-03         |
| FORD   | 1981-12-03 | 1982-12-03         |
| MILLER | 1982-01-23 | 1983-01-23         |
+--------+------------+--------------------+
```

Problem 3:

Task: Print a list of employees displaying just salary if more than 1500. If exactly 1500 then display 'On Target', if less than 1500 then display 'Below 1500'.

Code: —

```
SELECT ename,
       sal,
       CASE
         WHEN sal = 1500 THEN 'On Target'
         WHEN sal < 1500 THEN 'Below 1500'
         ELSE CAST(sal AS CHAR)
       END AS status
FROM emp;
```

Output: —

ename	sal	status
SMITH	800.00	Below 1500
ALLEN	1600.00	1600.00
WARD	1250.00	Below 1500
JONES	2975.00	2975.00
MARTIN	1250.00	Below 1500
BLAKE	2850.00	2850.00
CLARK	2450.00	2450.00
SCOTT	3000.00	3000.00
KING	5000.00	5000.00
TURNER	1500.00	On Target
ADAMS	1100.00	Below 1500
JAMES	950.00	Below 1500
FORD	3000.00	3000.00
MILLER	1300.00	Below 1500

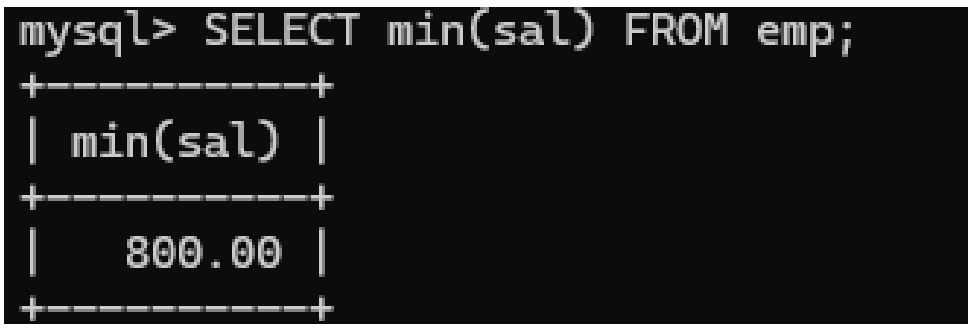
Problem 4:

Task: Find the minimum salary of all employees.

Code: —

```
SELECT MIN(sal) AS Min_Salary
FROM emp;
```

Output: —



```
mysql> SELECT min(sal) FROM emp;
+----------+
| min(sal) |
+----------+
|   800.00 |
+----------+
```

Problem 5:

Task: Find the minimum, maximum and average salaries of all employees.

Code:

```
SELECT MIN(sal) AS Min_Salary,
       MAX(sal) AS Max_Salary,
       AVG(sal) AS Avg_Salary
FROM emp;
```

Output:

```
mysql> SELECT min(sal), max(sal), avg(sal) FROM emp;
+----------+----------+-------------+
| min(sal) | max(sal) | avg(sal)    |
+----------+----------+-------------+
|   800.00 |  5000.00 | 2073.214286 |
+----------+----------+-------------+
```

Problem 6:

Task: List the minimum and maximum salary for each job type.

Code:

```
SELECT job,
       MIN(sal) AS Min_Salary,
       MAX(sal) AS Max_Salary
FROM emp
GROUP BY job;
```

Output:

```
mysql> SELECT job, min(sal) FROM emp GROUP BY job;
+-----------+----------+
| job       | min(sal) |
+-----------+----------+
| CLERK     |   800.00 |
| SALESMAN  |  1250.00 |
| MANAGER   |  2450.00 |
| ANALYST   |  3000.00 |
| PRESIDENT |  5000.00 |
+-----------+----------+
```

Problem 7:

Task: Find out the average salary and total remuneration for each job type.

Code: —

```
SELECT job,
       AVG(sal) AS Avg_Salary,
       SUM(sal + IFNULL(comm, 0)) AS Total_Remuneration
FROM emp
GROUP BY job;
```

Output: —

```
mysql> SELECT job, AVG(sal), SUM(sal + ifnull(comm, 0)) "Total Remuneration" FROM emp GROUP BY job;
+-----------+-------------+--------------------+
| job       | AVG(sal)    | Total Remuneration |
+-----------+-------------+--------------------+
| CLERK     | 1037.500000 |            4150.00 |
| SALESMAN  | 1400.000000 |            7800.00 |
| MANAGER   | 2758.333333 |            8275.00 |
| ANALYST   | 3000.000000 |            6000.00 |
| PRESIDENT | 5000.000000 |            5000.00 |
+-----------+-------------+--------------------+
```

Problem 8:

Task: Find out the difference between highest and lowest salaries.

Code: —

```
SELECT MIN(sal) AS Min_Salary,
       MAX(sal) AS Max_Salary,
       (MAX(sal) - MIN(sal)) AS Difference
FROM emp;
```

Output: —

```
mysql> SELECT MIN(sal), MAX(sal), (MAX(sal) - MIN(sal)) "Difference" FROM emp;
+----------+----------+------------+
| MIN(sal) | MAX(sal) | Difference |
+----------+----------+------------+
|   800.00 |  5000.00 |    4200.00 |
+----------+----------+------------+
```

Problem 9:

Task: Find all departments, which have more than 3 employees.

Code: —

```
SELECT deptno,
       COUNT(*) AS Employee_Count
FROM emp
GROUP BY deptno
HAVING COUNT(*) > 3;
```

Output: —

```
mysql> SELECT deptno, COUNT(*) FROM emp GROUP BY
 deptno HAVING COUNT(*) > 3;
+--------+----------+
| deptno | COUNT(*) |
+--------+----------+
|     20 |        5 |
|     30 |        6 |
+--------+----------+
```

Problem 10:

Task: Check whether all employee numbers are indeed unique.

Code: —

```sql
SELECT empno,
       COUNT(*) AS cnt
FROM emp
GROUP BY empno
HAVING COUNT(*) > 1;
```

Output: —

```
mysql> SELECT empno, COUNT(*) FROM emp GROUP BY empno;
+-------+----------+
| empno | COUNT(*) |
+-------+----------+
|  7369 |        1 |
|  7499 |        1 |
|  7521 |        1 |
|  7566 |        1 |
|  7654 |        1 |
|  7698 |        1 |
|  7782 |        1 |
|  7788 |        1 |
|  7839 |        1 |
|  7844 |        1 |
|  7876 |        1 |
|  7900 |        1 |
|  7902 |        1 |
|  7934 |        1 |
+-------+----------+
```

Problem 11:

Task: List the lowest paid employees working for each manager. Exclude any groups where the minimum salary is less than 1000. Sort the output by salary.

Code:

```
SELECT e.mgr,
       e.empno,
       e.ename,
       e.sal
FROM emp e
JOIN (
    SELECT mgr, MIN(sal) AS min_sal
    FROM emp
    GROUP BY mgr
    HAVING MIN(sal) >= 1000
) m ON e.mgr = m.mgr AND e.sal = m.min_sal
ORDER BY e.sal;
```

Output:

mgr	empno	ename	sal
7788	7876	ADAMS	1100.00
7782	7934	MILLER	1300.00
7839	7782	CLARK	2450.00
7566	7788	SCOTT	3000.00
7566	7902	FORD	3000.00

Problem 12:

Task: Display all employee names and their department names, in the order of department name.

Code:

```
SELECT e.ename,
       d.dname
FROM emp e
JOIN dept d ON e.deptno = d.deptno
ORDER BY d.dname;
```

Output:

```
mysql> SELECT ename, dname FROM emp, dept WHERE
emp.deptno = dept.deptno ORDER BY dept.dname;
+--------+------------+
| ename  | dname      |
+--------+------------+
| CLARK  | ACCOUNTING |
| KING   | ACCOUNTING |
| MILLER | ACCOUNTING |
| SMITH  | RESEARCH   |
| JONES  | RESEARCH   |
| SCOTT  | RESEARCH   |
| ADAMS  | RESEARCH   |
| FORD   | RESEARCH   |
| ALLEN  | SALES      |
| WARD   | SALES      |
| MARTIN | SALES      |
| BLAKE  | SALES      |
| TURNER | SALES      |
| JAMES  | SALES      |
+--------+------------+
```

Problem 13:

Task: Display all employee names, department number and department name.

Code: —

```
SELECT e.ename,
       d.deptno,
       d.dname
FROM emp e
JOIN dept d ON e.deptno = d.deptno;
```

Output: —

```
mysql> SELECT e.ename, d.deptno, d.dname FROM emp e, dept d
    -> WHERE e.deptno = d.deptno;
+--------+--------+------------+
| ename  | deptno | dname      |
+--------+--------+------------+
| CLARK  |     10 | ACCOUNTING |
| KING   |     10 | ACCOUNTING |
| MILLER |     10 | ACCOUNTING |
| SMITH  |     20 | RESEARCH   |
| JONES  |     20 | RESEARCH   |
| SCOTT  |     20 | RESEARCH   |
| ADAMS  |     20 | RESEARCH   |
| FORD   |     20 | RESEARCH   |
| ALLEN  |     30 | SALES      |
| WARD   |     30 | SALES      |
| MARTIN |     30 | SALES      |
| BLAKE  |     30 | SALES      |
| TURNER |     30 | SALES      |
| JAMES  |     30 | SALES      |
+--------+--------+------------+
```

Problem 14:

Task: Display the name, location and department of employees whose salary is more than 1500 a month.

Code: —

```sql
SELECT e.ename,
       d.loc,
       d.dname
FROM emp e
JOIN dept d ON e.deptno = d.deptno
WHERE e.sal > 1500;
```

Output: —

```
mysql> SELECT e.ename, d.loc, d.dname FROM emp e, dept d
    -> WHERE e.deptno = d.deptno AND e.sal > 1500;
+-------+----------+------------+
| ename | loc      | dname      |
+-------+----------+------------+
| ALLEN | CHICAGO  | SALES      |
| JONES | DALLAS   | RESEARCH   |
| BLAKE | CHICAGO  | SALES      |
| CLARK | NEW YORK | ACCOUNTING |
| SCOTT | DALLAS   | RESEARCH   |
| KING  | NEW YORK | ACCOUNTING |
| FORD  | DALLAS   | RESEARCH   |
+-------+----------+------------+
```

Problem 15:

Task: Show only employees on grade 3.

Code:

```
SELECT e.ename,
       e.sal,
       s.grade
FROM emp e
JOIN salgrade s ON e.sal BETWEEN s.losal AND s.hisal
WHERE s.grade = 3;
```

Output:

```
mysql> SELECT ename, sal, grade
    -> FROM emp e, salgrade s
    -> WHERE (e.sal BETWEEN s.losal and s.hisal) and s.grade=3;
+--------+---------+-------+
| ename  | sal     | grade |
+--------+---------+-------+
| ALLEN  | 1600.00 |     3 |
| TURNER | 1500.00 |     3 |
+--------+---------+-------+
```

Problem 16:

Task: Show all employees in 'Dallas'.

Code:

```
SELECT *
FROM emp
WHERE deptno = (
    SELECT deptno
    FROM dept
    WHERE loc = 'Dallas'
);
```

Output:

```
mysql> SELECT * FROM emp
    -> WHERE deptno = (SELECT deptno FROM dept WHERE loc = 'Dallas');
+-------+-------+---------+------+------------+---------+------+--------+
| empno | ename | job     | mgr  | hiredate   | sal     | comm | deptno |
+-------+-------+---------+------+------------+---------+------+--------+
|  7369 | SMITH | CLERK   | 7902 | 1980-12-17 |  800.00 | NULL |     20 |
|  7566 | JONES | MANAGER | 7839 | 1981-04-02 | 2975.00 | NULL |     20 |
|  7788 | SCOTT | ANALYST | 7566 | 1987-07-13 | 3000.00 | NULL |     20 |
|  7876 | ADAMS | CLERK   | 7788 | 1987-07-12 | 1100.00 | NULL |     20 |
|  7902 | FORD  | ANALYST | 7566 | 1981-12-03 | 3000.00 | NULL |     20 |
+-------+-------+---------+------+------------+---------+------+--------+
```

Problem 17:

Task: List the employee name, job, salary, and grade and department name for everyone in the company except clerks. Sort on salary, displaying the salary first.

Code:

```
SELECT e.sal  AS Salary,
       e.ename AS Employee_Name,
       e.job   AS Job,
       s.grade AS Grade,
       d.dname AS Department_Name
FROM emp e
JOIN dept d ON e.deptno = d.deptno
JOIN salgrade s ON e.sal BETWEEN s.losal AND s.hisal
WHERE UPPER(e.job) <> 'CLERK'
ORDER BY e.sal;
```

Output:

Salary	Employee_Name	Job	Grade	Department_Name
1250.00	WARD	SALESMAN	2	SALES
1250.00	MARTIN	SALESMAN	2	SALES
1500.00	TURNER	SALESMAN	3	SALES
1600.00	ALLEN	SALESMAN	3	SALES
2450.00	CLARK	MANAGER	4	ACCOUNTING
2850.00	BLAKE	MANAGER	4	SALES
2975.00	JONES	MANAGER	4	RESEARCH
3000.00	SCOTT	ANALYST	4	RESEARCH
3000.00	FORD	ANALYST	4	RESEARCH
5000.00	KING	PRESIDENT	5	ACCOUNTING

Problem 18:

Task: List the details of employees who earn 36000 a year or who are clerks.

Code:

```
SELECT *
FROM emp
WHERE (sal * 12) = 36000
   OR UPPER(job) = 'CLERK';
```

Output:

empno	ename	job	mgr	hiredate	sal	comm	deptno
7369	SMITH	CLERK	7902	1980-12-17	800.00	NULL	20
7788	SCOTT	ANALYST	7566	1987-07-13	3000.00	NULL	20
7876	ADAMS	CLERK	7788	1987-07-12	1100.00	NULL	20
7900	JAMES	CLERK	7698	1981-12-03	950.00	NULL	30
7902	FORD	ANALYST	7566	1981-12-03	3000.00	NULL	20
7934	MILLER	CLERK	7782	1982-01-23	1300.00	NULL	10

Problem 19:

Task: Display the department that has no employees.

Code:

```
SELECT dname
FROM dept
WHERE deptno NOT IN (SELECT DISTINCT deptno FROM emp);
```

Output:

	dname
▶	OPERATIONS

Problem 20:

Task: Find the employees who earn the highest salary in each job type. Sort in descending salary order.

Code:

```
SELECT e.job,
       e.ename,
       e.sal
FROM emp e
WHERE e.sal = (
    SELECT MAX(sal)
    FROM emp
    WHERE job = e.job
)
ORDER BY e.sal DESC;
```

Output: —

job	ename	sal
PRESIDENT	KING	5000.00
ANALYST	SCOTT	3000.00
ANALYST	FORD	3000.00
MANAGER	JONES	2975.00
SALESMAN	ALLEN	1600.00
CLERK	MILLER	1300.00

Problem 21:

Task: Find the most recently hired employees in each department ordered by hire date.

Code: —

```
SELECT e.deptno,
       e.ename,
       e.hiredate
FROM emp e
WHERE e.hiredate = (
    SELECT MAX(hiredate)
    FROM emp
    WHERE deptno = e.deptno
)
ORDER BY e.hiredate DESC;
```

Output: —

deptno	ename	hiredate
20	SCOTT	1987-07-13
10	MILLER	1982-01-23
30	JAMES	1981-12-03

Problem 22:

Task: Display the details of employees hired between Jan and June.

Code:

```
-- Works in Oracle / DBs supporting TO_CHAR; for MySQL use MONTH(
   hiredate) BETWEEN 1 AND 6
SELECT *
FROM emp
WHERE TO_CHAR(hiredate, 'MM') BETWEEN '01' AND '06';
```

Output:

empno	ename	job	mgr	hiredate	sal	comm	deptno
7499	ALLEN	SALESMAN	7698	1981-02-20	1600.00	300.00	30
7521	WARD	SALESMAN	7698	1981-02-21	1250.00	500.00	30
7566	JONES	MANAGER	7839	1981-04-02	2975.00	NULL	20
7698	BLAKE	MANAGER	7839	1981-05-01	2850.00	NULL	30
7782	CLARK	MANAGER	7839	1981-06-09	2450.00	NULL	10
7934	MILLER	CLERK	7782	1982-01-23	1300.00	NULL	10

Problem 23:

Task: Display the count, total salary and average salary of all employees in each department.

Code:

```
SELECT deptno,
       COUNT(*) AS Emp_Count,
       SUM(sal) AS Total_Salary,
       AVG(sal) AS Average_Salary
FROM emp
GROUP BY deptno;
```

Output:

deptno	Emp_Count	Total_Salary	Average_Salary
10	3	8750.00	2916.666667
20	5	10875.00	2175.000000
30	6	9400.00	1566.666667

Problem 24:

Task: Find a square root of the number 36.1111. The result should not contain any decimal spaces.

Code: —

```
-- Round the square root to nearest integer (no decimals)
SELECT ROUND(SQRT(36.1111)) AS Square_Root
FROM dual;
```

Output: —

	Square_Root
▶	6

Problem 25:

Task: Given a string 'HELLO$_T HERE_{'.Replaceall'_{with'!'marks.}}$

Code: —

```
SELECT REPLACE('HELLO_THERE_', '_', '!') AS NEW_STRING
FROM dual;
```

Output: —

	NEW_STRING
▶	HELLO!THERE!

Problem 26:

Task: Find the sum of the length of the strings. The String are CDAC, HYDERABAD.

Code: —

```
SELECT LENGTH('CDAC') + LENGTH('HYDERABAD') AS Total_Length
FROM dual;
```

Output: —

	Total_Length
▶	13

Problem 27:

Task: Find the job that was filled in the first half of the 1980 and the job that was filled during the same period in 1981.

Code: —

```
SELECT job, hiredate
FROM emp
WHERE (TO_CHAR(hiredate, 'YYYY') = '1980' AND TO_CHAR(hiredate, '
   MM') BETWEEN '01' AND '06')
   OR (TO_CHAR(hiredate, 'YYYY') = '1981' AND TO_CHAR(hiredate, '
      MM') BETWEEN '01' AND '06');
```

Output: —

	job	hiredate
▶	SALESMAN	1981-02-20
	SALESMAN	1981-02-21
	MANAGER	1981-04-02
	MANAGER	1981-05-01
	MANAGER	1981-06-09