

## Assignment –II

-Yash Yadav

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

```
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 |
+-----+
5 rows in set (0.07 sec)
```

2. Display each employee's name with hiredate and salary review date.

Assume review date is one year after hiredate.

```
mysql> SELECT ename, hiredate, DATE_ADD(hiredate, INTERVAL 1 YEAR) AS review_date
-> FROM emp;
+-----+-----+-----+
| ename | hiredate | 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 |
+-----+-----+-----+
14 rows in set (0.01 sec)
```

3. 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'.

```
mysql> SELECT ename,
->           CASE
->             WHEN sal > 1500 THEN sal
->             WHEN sal = 1500 THEN 'On Target'
->             ELSE 'Below 1500'
->           END AS salary_status
->   FROM emp;
+-----+-----+
| ename | salary_status |
+-----+-----+
| SMITH | Below 1500
| ALLEN | 1600.00
| WARD  | Below 1500
| JONES | 2975.00
| MARTIN | Below 1500
| BLAKE | 2850.00
| CLARK | 2450.00
| SCOTT | 3000.00
| KING  | 5000.00
| TURNER | On Target
| ADAMS | Below 1500
| JAMES | Below 1500
| FORD  | 3000.00
| MILLER | Below 1500
+-----+-----+
14 rows in set (0.01 sec)
```

4. Find the minimum salary of all employees.

```
mysql> SELECT MIN(sal) AS min_salary
->   FROM emp;
+-----+
| min_salary |
+-----+
|     800.00 |
+-----+
1 row in set (0.01 sec)
```

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

```

mysql> SELECT MIN(sal) AS min_salary,
      ->          MAX(sal) AS max_salary,
      ->          AVG(sal) AS avg_salary
      -> FROM emp;
+-----+-----+-----+
| min_salary | max_salary | avg_salary |
+-----+-----+-----+
|     800.00 |    5000.00 | 2073.214286 |
+-----+-----+-----+
1 row in set (0.00 sec)

```

6. List the minimum and maximum salary for each job type.

```

mysql> SELECT job, MIN(sal) AS min_salary, MAX(sal) AS max_salary
      -> FROM emp
      -> GROUP BY job;
+-----+-----+-----+
| job   | min_salary | max_salary |
+-----+-----+-----+
| CLERK |     800.00 |    1300.00 |
| SALESMAN |    1250.00 |    1600.00 |
| MANAGER |    2450.00 |    2975.00 |
| ANALYST |    3000.00 |    3000.00 |
| PRESIDENT |    5000.00 |    5000.00 |
+-----+-----+-----+
5 rows in set (0.02 sec)

```

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

```

mysql> SELECT job, AVG(sal) AS avg_salary, SUM(sal) AS total_salary
      -> FROM emp
      -> GROUP BY job;
+-----+-----+-----+
| job   | avg_salary | total_salary |
+-----+-----+-----+
| CLERK | 1037.500000 |    4150.00 |
| SALESMAN | 1400.000000 |    5600.00 |
| MANAGER | 2758.333333 |    8275.00 |
| ANALYST | 3000.000000 |    6000.00 |
| PRESIDENT | 5000.000000 |    5000.00 |
+-----+-----+-----+
5 rows in set (0.00 sec)

```

8. Find out the difference between highest and lowest salaries.

```
mysql> SELECT MAX(sal) - MIN(sal) AS salary_difference
-> FROM emp;
+-----+
| salary_difference |
+-----+
|        4200.00 |
+-----+
1 row in set (0.01 sec)
```

9. Find all departments, which have more than 3 employees.

```
mysql> SELECT deptno, COUNT(*) AS emp_count
-> FROM emp
-> GROUP BY deptno
-> HAVING COUNT(*) > 3;
+-----+
| deptno | emp_count |
+-----+
|      20 |          5 |
|      30 |          6 |
+-----+
2 rows in set (0.00 sec)
```

10. Check whether all employee numbers are indeed unique.

```
mysql> SELECT empno, COUNT(*) AS count_occurrences
-> FROM emp
-> GROUP BY empno
-> HAVING COUNT(*) > 1;
Empty set (0.00 sec)
```

11. 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.

```

mysql> -- Get manager-wise minimum salary (>=1000), then join back to get employees
mysql> WITH mgr_min AS (
    >     SELECT mgr, MIN(sal) AS min_sal
    >     FROM emp
    >     GROUP BY mgr
    >     HAVING MIN(sal) >= 1000
    > )
    >     SELECT e.empno, e.ename, e.job, e.mgr, e.sal
    >     FROM emp e
    >     JOIN mgr_min m ON e.mgr = m.mgr AND e.sal = m.min_sal
    >     ORDER BY e.sal;
+-----+-----+-----+-----+-----+
| empno | ename | job   | mgr   | sal   |
+-----+-----+-----+-----+-----+
| 7876  | ADAMS | CLERK | 7788  | 1100.00 |
| 7934  | MILLER | CLERK | 7782  | 1300.00 |
| 7782  | CLARK  | MANAGER | 7839  | 2450.00 |
| 7788  | SCOTT  | ANALYST | 7566  | 3000.00 |
| 7902  | FORD   | ANALYST | 7566  | 3000.00 |
+-----+-----+-----+-----+-----+
5 rows in set (0.22 sec)

```

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

```

mysql> SELECT e.ename, d.dname
    >     FROM emp e
    >     JOIN dept d ON e.deptno = d.deptno
    >     ORDER BY d.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 |
+-----+-----+
14 rows in set (0.01 sec)

mysql> |

```

13. Display all employee names, department number and department name.

```

mysql> SELECT e.ename, e.deptno, d.dname
   -> FROM emp e
   -> LEFT JOIN dept d ON e.deptno = d.deptno;
+-----+-----+-----+
| ename | deptno | dname  |
+-----+-----+-----+
| SMITH |      20 | RESEARCH
| ALLEN |      30 | SALES
| WARD  |      30 | SALES
| JONES |      20 | RESEARCH
| MARTIN|      30 | SALES
| BLAKE |      30 | SALES
| CLARK |      10 | ACCOUNTING
| SCOTT |      20 | RESEARCH
| KING  |      10 | ACCOUNTING
| TURNER|      30 | SALES
| ADAMS |      20 | RESEARCH
| JAMES |      30 | SALES
| FORD  |      20 | RESEARCH
| MILLER|      10 | ACCOUNTING
+-----+-----+-----+
14 rows in set (0.01 sec)

```

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

```

mysql> SELECT e.ename, d.loc AS location, d.dname AS department
   -> FROM emp e
   -> JOIN dept d ON e.deptno = d.deptno
   -> WHERE e.sal > 1500;
+-----+-----+-----+
| ename | location | department |
+-----+-----+-----+
| ALLEN | CHICAGO | SALES
| JONES | DALLAS  | RESEARCH
| BLAKE | CHICAGO | SALES
| CLARK | NEW YORK | ACCOUNTING
| SCOTT | DALLAS  | RESEARCH
| KING  | NEW YORK | ACCOUNTING
| FORD  | DALLAS  | RESEARCH
+-----+-----+-----+
7 rows in set (0.00 sec)

```

15. Show only employees on grade 3.

```

mysql> SELECT *
   -> FROM emp
   -> WHERE deptno = 30;
+-----+-----+-----+-----+-----+-----+-----+-----+
| 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   |
| 7654  | MARTIN| SALESMAN | 7698  | 1981-09-08 | 1250.00 | 1400.00 | 30   |
| 7698  | BLAKE  | MANAGER | 7839  | 1981-05-01 | 2850.00 | NULL   | 30   |
| 7844  | TURNER| SALESMAN | 7698  | 1981-09-08 | 1500.00 | 0.00   | 30   |
| 7900  | JAMES  | CLERK   | 7698  | 1981-12-03 | 950.00  | NULL   | 30   |
+-----+-----+-----+-----+-----+-----+-----+-----+
6 rows in set (0.02 sec)

```

16.Show all employees in 'Dallas'

```
mysql> SELECT e.*  
-> FROM emp e  
-> JOIN dept d ON e.deptno = d.deptno  
-> WHERE d.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   |  
+-----+-----+-----+-----+-----+-----+-----+-----+  
5 rows in set (0.00 sec)
```

17.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.

```
mysql> SELECT e.sal AS Salary, e.ename, e.job, d.dname AS Department  
-> FROM emp e  
-> LEFT JOIN dept d ON e.deptno = d.deptno  
-> WHERE e.job <> 'CLERK'  
-> ORDER BY e.sal DESC;  
+-----+-----+-----+-----+  
| Salary | ename | job   | Department |  
+-----+-----+-----+-----+  
| 5000.00 | KING  | PRESIDENT | ACCOUNTING |  
| 3000.00 | SCOTT | ANALYST  | RESEARCH   |  
| 3000.00 | FORD  | ANALYST  | RESEARCH   |  
| 2975.00 | JONES | MANAGER  | RESEARCH   |  
| 2850.00 | BLAKE | MANAGER  | SALES      |  
| 2450.00 | CLARK | MANAGER  | ACCOUNTING |  
| 1600.00 | ALLEN | SALESMAN | SALES      |  
| 1500.00 | TURNER | SALESMAN | SALES      |  
| 1250.00 | WARD  | SALESMAN | SALES      |  
| 1250.00 | MARTIN | SALESMAN | SALES      |  
+-----+-----+-----+-----+
```

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

```
mysql> SELECT *  
-> FROM emp  
-> WHERE sal * 12 = 36000 OR job = 'CLERK';  
+-----+-----+-----+-----+-----+-----+-----+-----+  
| 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   |  
+-----+-----+-----+-----+-----+-----+-----+-----+  
6 rows in set (0.00 sec)
```

19.Display the department that has no employees.

```

mysql> SELECT d.deptno, d.dname
-> FROM dept d
-> LEFT JOIN emp e ON d.deptno = e.deptno
-> WHERE e.deptno IS NULL;
+-----+-----+
| deptno | dname      |
+-----+-----+
|      40 | OPERATIONS |
+-----+
1 row in set (0.00 sec)

```

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

```

mysql> WITH job_max AS (
->   SELECT job, MAX(sal) AS max_sal
->   FROM emp
->   GROUP BY job
-> )
->   SELECT e.empno, e.ename, e.job, e.sal
->   FROM emp e
->   JOIN job_max jm ON e.job = jm.job AND e.sal = jm.max_sal
->   ORDER BY e.sal DESC;
+-----+-----+-----+-----+
| empno | ename  | job       | sal    |
+-----+-----+-----+-----+
| 7839  | KING    | PRESIDENT | 5000.00 |
| 7788  | SCOTT   | ANALYST   | 3000.00 |
| 7902  | FORD    | ANALYST   | 3000.00 |
| 7566  | JONES   | MANAGER   | 2975.00 |
| 7499  | ALLEN   | SALESMAN  | 1600.00 |
| 7934  | MILLER  | CLERK     | 1300.00 |
+-----+-----+-----+-----+
6 rows in set (0.02 sec)

```

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

```

mysql> WITH dept_max_hire AS (
    ->     SELECT deptno, MAX(hiredate) AS max_hire
    ->     FROM emp
    ->     GROUP BY deptno
    -> )
    ->     SELECT e.empno, e.ename, e.deptno, e.hiredate
    ->     FROM emp e
    ->     JOIN dept_max_hire dmh ON e.deptno = dmh.deptno AND e.hiredate = dmh.max_hire
    ->     ORDER BY e.hiredate DESC;
+-----+-----+-----+-----+
| empno | ename | deptno | hiredate |
+-----+-----+-----+-----+
| 7788  | SCOTT |      20 | 1987-07-13 |
| 7934  | MILLER |     10 | 1982-01-23 |
| 7900  | JAMES |     30 | 1981-12-03 |
+-----+-----+-----+-----+
3 rows in set (0.01 sec)

```

22. Display the details of employees hired between Jan and June.

```

mysql> SELECT *
    ->     FROM emp
    ->     WHERE MONTH(hiredate) BETWEEN 1 AND 6;
+-----+-----+-----+-----+-----+-----+-----+-----+
| 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    |
+-----+-----+-----+-----+-----+-----+-----+-----+
6 rows in set (0.01 sec)

```

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

```

mysql> SELECT deptno, COUNT(*) AS emp_count, SUM(sal) AS total_salary, AVG(sal) AS avg_salary
    ->     FROM emp
    ->     GROUP BY deptno;
+-----+-----+-----+-----+
| deptno | emp_count | total_salary | avg_salary |
+-----+-----+-----+-----+
| 10    |      3    |    8750.00  | 2916.666667 |
| 20    |      5    |   10875.00  | 2175.000000 |
| 30    |      6    |    9400.00  | 1566.666667 |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

```

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

```

mysql> SELECT TRUNCATE(SQRT(36.1111), 0) AS sqrt_trunc;
+-----+
| sqrt_trunc |
+-----+
|       6 |
+-----+
1 row in set (0.00 sec)

mysql>
mysql> -- or round to nearest integer
mysql> SELECT ROUND(SQRT(36.1111), 0) AS sqrt_round;
+-----+
| sqrt_round |
+-----+
|       6 |
+-----+
1 row in set (0.00 sec)

mysql>
mysql> -- or cast to integer (floor)
mysql> SELECT FLOOR(SQRT(36.1111)) AS sqrt_floor;
+-----+
| sqrt_floor |
+-----+
|       6 |
+-----+
1 row in set (0.00 sec)

```

25. Given a string 'HELLO\_THERE\_'. Replace all '\_' with '!' marks.

```

mysql> SELECT REPLACE('HELLO_THERE_', '_', '!') AS changed_string;
+-----+
| changed_string |
+-----+
| HELLO!THERE!   |
+-----+
1 row in set (0.00 sec)

mysql> -- Result: 'HELLO!THERE!'

```

26. Find the sum of the length of the strings. The String are CDAC,

HYDERABAD.

```

mysql> -- Result: 'HELLO!THERE!'
mysql> SELECT (CHAR_LENGTH('CDAC') + CHAR_LENGTH('HYDERABAD')) AS total_length;
+-----+
| total_length |
+-----+
|       13 |
+-----+
1 row in set (0.01 sec)

```

27. 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.

```
mysql> SELECT DISTINCT job FROM emp WHERE hiredate BETWEEN '1981-01-01' AND '1981-06-30';
+-----+
| job   |
+-----+
| SALESMAN |
| MANAGER  |
+-----+
2 rows in set (0.00 sec)
```