Title: MySQL

README file

Description of the solutions

The solution provides SQL scripts for each challenge, based on the following logic:

CH01: List all Employees whose salary is greater than 1,000 but not 2,000. Show the Employee Name, Department, and Salary.

- Use the SELECT statement to retrieve the employee name, department and salary from the EMP and DEPT tables, and alias the selected column as a readable name using the AS clause.
- Use the INNER JOIN clause to link the EMP and DEPT tables based on the DEPTNO column.
- Apply the WHERE clause to filter the results and include only those with salaries greater than 1,000 and less than 2,000.

```
MariaDB [COMPANY1]> /*Get the employee name, department and salary from the EMP and
   /*>  DEPT tables and aliased as "Name", "Department", "Salary".
   /*> Join the EMP table with DEPT table using the DEPTNO as the foreign key.
   /*> Filter the results to include only salaries greater than 1,000 and less than 2000.*/
MariaDB [COMPANY1]> SELECT
    ->     ENAME AS Name, DNAME AS Department, SAL AS Salary FROM EMP
    ->     INNER JOIN DEPT ON EMP.DEPTNO = DEPT.DEPTNO
    ->     WHERE SAL > 1000 AND SAL < 2000;
+--------+------------+---------+
| Name   | Department | Salary  |
+--------+------------+---------+
| ALLEN  | SALES      | 1600.00 |
| WARD   | SALES      | 1250.00 |
| MARTIN | SALES      | 1250.00 |
| TURNER | SALES      | 1500.00 |
| ADAMS  | RESEARCH   | 1100.00 |
| MILLER | ACCOUNTING | 1300.00 |
+--------+------------+---------+
6 rows in set (0.000 sec)
```

CH02: Count the number of people in department 30 who receive a salary and a commission.

- Use the SELECT statement along with the COUNT() function to count the number of rows. The result column is aliased using the AS clause.
- Apply the WHERE clause to filter the results and include only those in department 30 who receive both salary and commission.

```
MariaDB [COMPANY1]> /*Count the number of employee and alias the result column as “Count”.
   /*> Filter the results to include only employee who receive both salary and
   /*>  commission in department 30.*/
MariaDB [COMPANY1]> SELECT
    ->      COUNT(*) AS Count FROM EMP
    ->      WHERE DEPTNO = 30 AND SAL > 0 AND COMM > 0;
+-------+
| Count |
+-------+
|     3 |
+-------+
1 row in set (0.000 sec)
```

CH03: Find the name and salary of employees having a salary greater or equal to 1,000 and live in Dallas.

- Use the SELECT statement to retrieve the employee name and salary from the EMP table and alias the selected column as a readable name using the AS clause.
- Use the INNER JOIN clause to link the EMP and DEPT tables based on the DEPTNO column.
- Apply the WHERE clause to filter the results and include only those with salaries greater than or equal to 1,000 and located in Dallas.

```
MariaDB [COMPANY1]> /*Get the employee name and salary from the EMP table and aliased as
   /*>  "Name" and "Salary".
   /*> Join the EMP table with the DEPT table using the DEPTNO as the foreign key.
   /*> Filter the results to include only employee with salary greater than or equal
   /*>  to 1,000 and located in “DALLAS”.*/
MariaDB [COMPANY1]> SELECT
    ->     ENAME AS Name, SAL AS Salary FROM EMP
    ->     INNER JOIN DEPT ON EMP.DEPTNO = DEPT.DEPTNO
    ->     WHERE SAL >= 1000 AND LOC = 'DALLAS';
+-------+---------+
| Name  | Salary  |
+-------+---------+
| JONES | 2975.00 |
| SCOTT | 3000.00 |
| ADAMS | 1100.00 |
| FORD  | 3000.00 |
+-------+---------+
4 rows in set (0.000 sec)
```

CH04: Find all departments that do not have any current employees.

- Use the SELECT statement to retrieve the department number and name from the DEPT table and alias the selected column as a readable name using the AS clause.
- Use the LEFT JOIN clause to link the DEPT and EMP tables based on the DEPTNO column, retaining all department numbers present in the DEPT table.
- Use the NULL condition to filter the results and include only those with null values in the DEPTNO column.

```
MariaDB [COMPANY1]> /*Get the department number and department name from the DEPT table and aliased
   /*>  as "DeptNumber"and "Department".
   /*> Join the DEPT table with the EMP table using the DEPTNO as the foreign key,
   /*>  retaining all department numbers present in the DEPT table.
   /*> Filter the results to include only department number with null values.*/
MariaDB [COMPANY1]> SELECT
    ->     DEPT.DEPTNO AS DeptNumber, DNAME AS Department FROM DEPT
    ->     LEFT JOIN EMP ON DEPT.DEPTNO = EMP.DEPTNO
    ->     WHERE EMP.DEPTNO IS NULL;
+------------+------------+
| DeptNumber | Department |
+------------+------------+
|         40 | OPERATIONS |
+------------+------------+
1 row in set (0.000 sec)
```

CH05: List the department number, average salary, and count of employees of each department.

- Use the SELECT statement to retrieve the required columns from the EMP table, including:
    - the department number and alias as readable name using the AS clause,
    - the average salary calculated using the AVG() function and aliased using the AS clause,
    - the count of employee calculated using the COUNT() function and aliased using the AS clause.
- Apply the GROUP BY clause to group the results by department number in order to calculate the average salary and count of employee for each department.

```
MariaDB [COMPANY1]> /*Get the department number, average salary and number of employee from
   /*>  EMP table and aliased as "DeptNumber", "AvgSalary"and "Count".
   /*> Group the results by department number to calculate average salary and
   /*>  count of employee for each department.*/
MariaDB [COMPANY1]> SELECT
    ->     DEPTNO AS DeptNumber, AVG(SAL) AS AvgSalary, COUNT(*) AS Count FROM EMP
    ->     GROUP BY DEPTNO;
+------------+-------------+-------+
| DeptNumber | AvgSalary   | Count |
+------------+-------------+-------+
|         10 | 2916.666667 |     3 |
|         20 | 2175.000000 |     5 |
|         30 | 1566.666667 |     6 |
+------------+-------------+-------+
3 rows in set (0.000 sec)
```

Alternative approaches

Below are alternative approaches using different MySQL scripts to achieve the same or similar outcomes as the original solutions.

CH01: This alternative approach uses the BETWEEN operator to filter salaries greater than 1,000 and less than 2,000, which achieves the same result as the original script.

```
MariaDB [COMPANY1]> SELECT
    ->     ENAME AS Name, DNAME AS Department, SAL AS Salary FROM EMP
    ->     INNER JOIN DEPT ON EMP.DEPTNO = DEPT.DEPTNO
    ->     WHERE SAL BETWEEN 1001 AND 1999;
+--------+------------+---------+
| Name   | Department | Salary  |
+--------+------------+---------+
| ALLEN  | SALES      | 1600.00 |
| WARD   | SALES      | 1250.00 |
| MARTIN | SALES      | 1250.00 |
| TURNER | SALES      | 1500.00 |
| ADAMS  | RESEARCH   | 1100.00 |
| MILLER | ACCOUNTING | 1300.00 |
+--------+------------+---------+
6 rows in set (0.000 sec)
```

CH02: This alternative approach uses the condition SAL IS NOT NULL AND COMM IS NOT NULL instead of SAL > 0 AND COMM > 0 to filter out non-null values. This achieves a similar result as the original script, but zero value will also be counted. Therefore, the original script outputs '3' and the alternative outputs '4'. Which approach is better depends on whether the user wants to count employee who are entitled to commission but have not received any at the moment.

```
MariaDB [COMPANY1]> SELECT
    ->     COUNT(*) AS Count FROM EMP
    ->     WHERE DEPTNO = 30 AND SAL IS NOT NULL AND COMM IS NOT NULL;
+-------+
| Count |
+-------+
|     4 |
+-------+
1 row in set (0.000 sec)
```

CH03: This alternative approach uses a subquery and the IN operator instead of using the JOIN clause to achieve the same result.

```
MariaDB [COMPANY1]> SELECT
    -> ENAME AS Name, SAL AS Salary FROM EMP
    -> WHERE SAL >= 1000 AND EMP.DEPTNO IN
    -> (SELECT DEPT.DEPTNO FROM DEPT WHERE LOC = 'DALLAS');
+-------+---------+
| Name  | Salary  |
+-------+---------+
| JONES | 2975.00 |
| SCOTT | 3000.00 |
| ADAMS | 1100.00 |
| FORD  | 3000.00 |
+-------+---------+
4 rows in set (0.000 sec)
```

CH04: This alternative approach uses a subquery to retrieve the DEPTNO from the EMP table and uses the NOT IN operator to identify departments without any employees by checking for department numbers that do not exist in the subquery result.

```
MariaDB [COMPANY1]> SELECT
    ->     DEPT.DEPTNO AS DeptNumber, DNAME AS Department FROM DEPT
    ->     WHERE DEPT.DEPTNO NOT IN (SELECT EMP.DEPTNO FROM EMP);
+------------+------------+
| DeptNumber | Department |
+------------+------------+
|         40 | OPERATIONS |
+------------+------------+
1 row in set (0.001 sec)
```

CH05: This alternative approach, utilizing a LEFT JOIN between the DEPT and EMP tables based on the DEPTNO column, achieves a similar outcome to the original script. It retains all department numbers from the DEPT table, including those that do not exist in the EMP table, which means departments without any current employees will also be included in the output. Which approach is better depends on whether the user wants to consider all departments, including those without any current employees.

```
MariaDB [COMPANY1]> SELECT
    -> DEPT.DEPTNO AS DeptNumber, AVG(SAL) AS AvgSalary,
    -> COUNT(EMP.EMPNO) AS Count FROM DEPT
    -> LEFT JOIN EMP ON DEPT.DEPTNO = EMP.DEPTNO
    -> GROUP BY DEPT.DEPTNO;
+------------+-------------+-------+
| DeptNumber | AvgSalary   | Count |
+------------+-------------+-------+
|         10 | 2916.666667 |     3 |
|         20 | 2175.000000 |     5 |
|         30 | 1566.666667 |     6 |
|         40 |        NULL |     0 |
+------------+-------------+-------+
4 rows in set (0.001 sec)
```