

Assignment: SQL Tasks for Salesforce Private Limited - Database Operations (MySQL) :

Answer

Creating Database Table.

```
mysql> use assignment;
```

Database changed

```
mysql> create table departments(
```

```
    -> department_id INT PRIMARY KEY,
```

```
    -> DEPARTMENT_NAME VARCHAR(50) NOT NULL UNIQUE);
```

Query OK, 0 rows affected (0.06 sec)

```
mysql> DESC DEPARTMENTS;
```

Field	Type	Null	Key	Default	Extra
department_id	int	NO	PRI	NULL	
DEPARTMENT_NAME	varchar(50)	NO	UNI	NULL	

2 rows in set (0.01 sec)

Creating Employee Table.

```
mysql> desc employees;
```

Field	Type	Null	Key	Default	Extra
employee_id	int	NO	PRI	NULL	
first_name	varchar(50)	NO		NULL	
last_name	varchar(50)	NO		NULL	
email	varchar(100)	NO	UNI	NULL	
hire_date	date	NO		NULL	
salary	int	NO		NULL	
department_id	int	NO	MUL	NULL	

7 rows in set (0.00 sec)

Creating Project Table

```
mysql> CREATE TABLE Projects (  
    ->  project_id INT PRIMARY KEY,  
    ->  project_name VARCHAR(100) NOT NULL UNIQUE,  
    ->  start_date DATE,  
    ->  end_date DATE,  
    ->  department_id INT NOT NULL,  
    ->  FOREIGN KEY (department_id) REFERENCES Departments(department_id)  
    -> );
```

Query OK, 0 rows affected (0.05 sec)

```
mysql> desc projects;
```

Field	Type	Null	Key	Default	Extra
project_id	int	NO	PRI	NULL	
project_name	varchar(100)	NO	UNI	NULL	
start_date	date	YES		NULL	
end_date	date	YES		NULL	
department_id	int	NO	MUL	NULL	

5 rows in set (0.00 sec)

Inserting Data into Departments Table and display it .

```
mysql> INSERT INTO Departments(department_id, department_name) values(1,'IT'),
```

```
-> (2,'HR'),
```

```
-> (3,'Sales'),
```

```
-> (4,'Finance');
```

Query OK, 4 rows affected (0.01 sec)

Records: 4 Duplicates: 0 Warnings: 0

```
mysql> insert into departments(department_id, department_name)values(5, 'Marketing');
```

Query OK, 1 row affected (0.00 sec)

```
mysql> select * from Departments;
```

```
+-----+-----+
```

```
| department_id | DEPARTMENT_NAME |
```

```
+-----+-----+
```

```
|      4 | Finance      |
```

```
|      2 | HR           |
```

```
|      1 | IT           |
```

```
|      5 | Marketing    |
```

```
|      3 | Sales        |
```

```
+-----+-----+
```

5 rows in set (0.00 sec)

Inserting Data into Employees Table and display it .

```
mysql> INSERT INTO Employees (employee_id, first_name, last_name, email,
hire_date, salary, department_id) VALUES

-> (101, 'Ravi', 'Sharma', 'ravi.sharma@specialforce.com', '2017-05-15', 55000, 1),
-> (102, 'Neha', 'Kapoor', 'neha.kapoor@specialforce.com', '2019-03-23', 48000, 2),
-> (103, 'Jyoti', 'Verma', 'jyoti.verma@specialforce.com', '2020-11-02', 60000, 1),
-> (104, 'Anil', 'Patil', 'anil.patil@specialforce.com', '2018-09-18', 70000, 3),
-> (105, 'Pooja', 'Singh', 'pooja.singh@specialforce.com', '2021-06-10', 40000, 4),
-> (106, 'Sanjay', 'Iyer', 'sanjay.iyer@specialforce.com', '2018-01-22', 75000, 3),
-> (107, 'Jatin', 'Reddy', 'jatin.reddy@specialforce.com', '2021-12-12', 85000, 2),
-> (108, 'Shreya', 'Mehta', 'shreya.mehta@specialforce.com', '2022-04-19', 30000, 5),
-> (109, 'Rajesh', 'Gupta', 'rajesh.gupta@specialforce.com', '2020-08-11', 90000, 1),
-> (110, 'Kavita', 'Nair', 'kavita.nair@specialforce.com', '2021-02-07', 50000, 2);
```

Query OK, 10 rows affected (0.01 sec)

Records: 10 Duplicates: 0 Warnings: 0

```
mysql> select * from Employees;
```

```
+-----+-----+-----+-----+-----+-----+-----+
-----+

| employee_id | first_name | last_name | email                                | hire_date | salary |
department_id |
+-----+-----+-----+-----+-----+-----+-----+
-----+

|      101 | Ravi      | Sharma   | ravi.sharma@specialforce.com | 2017-05-15 | 55000 |
1 |
|      102 | Neha      | Kapoor   | neha.kapoor@specialforce.com | 2019-03-23 | 48000 |
2 |
```

103	Jyoti	Verma	jyoti.verma@specialforce.com	2020-11-02	60000
104	Anil	Patil	anil.patil@specialforce.com	2018-09-18	70000
105	Pooja	Singh	pooja.singh@specialforce.com	2021-06-10	40000
106	Sanjay	Iyer	sanjay.iyer@specialforce.com	2018-01-22	75000
107	Jatin	Reddy	jatin.reddy@specialforce.com	2021-12-12	85000
108	Shreya	Mehta	shreya.mehta@specialforce.com	2022-04-19	30000
109	Rajesh	Gupta	rajesh.gupta@specialforce.com	2020-08-11	90000
110	Kavita	Nair	kavita.nair@specialforce.com	2021-02-07	50000

```

+-----+-----+-----+-----+-----+-----+-----+
-----+

```

10 rows in set (0.00 sec)

Inserting Data into Projects Table and display it .

```
mysql> INSERT INTO Projects (project_id, project_name, start_date, end_date,
department_id) VALUES
```

```

-> (201, 'Project Phoenix', '2021-01-15', '2022-07-30', 1),
-> (202, 'Client Onboarding', '2020-06-20', NULL, 3),
-> (203, 'Financial Overhaul', '2019-03-10', '2021-12-15', 4),
-> (204, 'Marketing Revamp', '2022-03-01', NULL, 5),
-> (205, 'Internal System Audit', '2023-02-15', NULL, 2);
```

Query OK, 5 rows affected (0.01 sec)


Records: 5 Duplicates: 0 Warnings: 0

```
mysql> select * from projects;
```

project_id	project_name	start_date	end_date	department_id
201	Project Phoenix	2021-01-15	2022-07-30	1
202	Client Onboarding	2020-06-20	NULL	3

	203	Financial Overhaul		2019-03-10		2021-12-15		4	
	204	Marketing Revamp		2022-03-01		NULL		5	
	205	Internal System Audit		2023-02-15		NULL		2	
+-----+-----+-----+-----+-----+									

Queries to Perform:

 **Query 1:** Write a query to retrieve the first name, last name, and department name of all employees. If an employee does not belong to any department, the department name should be NULL.



```
mysql> SELECT
-> Employees.first_name,
-> Employees.last_name,
-> Departments.department_name
-> FROM
-> EMPLOYEES
-> LEFT JOIN DEPARMENTS ON
-> Employees.department_id = Departments.department_id;
ERROR 1146 (42S02): Table 'assignment.deparments' doesn't exist
```

```
mysql> SELECT
-> E.first_name,
-> E.last_name,
-> D.department_name
-> FROM
-> Employees E
-> LEFT JOIN
-> Departments D
-> ON
-> E.department_id = D.department_id;
+-----+-----+-----+
| first_name | last_name | department_name |
+-----+-----+-----+
| Ravi      | Sharma   | IT              |
| Neha      | Kapoor  | HR              |
| Jyoti     | Verma    | IT              |
| Anil      | Patil    | Sales           |
| Pooja     | Singh    | Finance         |
| Sanjay    | Iyer     | Sales           |
| Jatin     | Reddy    | HR              |
| Shreya    | Mehta    | Marketing       |
| Rajesh    | Gupta    | IT              |
| Kavita    | Nair     | HR              |
+-----+-----+-----+
```

🚦 **Query 2: Write a query to find all employees in the IT department who earn more than ₹50,000.**

➔

```
mysql> SELECT
```

```
-> Employees.first_name,
```

```
-> Employees.last_name,
```

```
-> Employees.salary,
```

```
-> Departments.department_name
```

```
-> FROM
```

```
-> Employees
```

```
-> JOIN
```

```
-> Departments
```

```
-> ON
```

```
-> Employees.department_id = Departments.department_id
```

```
-> WHERE
```

```
-> Departments.department_name = 'IT'
```

```
-> AND Employees.salary > 50000;
```

```
+-----+-----+-----+-----+
```

```
| first_name | last_name | salary | department_name |
```


```
+-----+-----+-----+-----+
```

```
| Ravi      | Sharma   | 55000 | IT              |
```

```
| Jyoti     | Verma    | 60000 | IT              |
```

```
| Rajesh    | Gupta    | 90000 | IT              |
```

```
+-----+-----+-----+-----+
```

 **Query 3: Write a query to list the first name, last name, and email of all employees whose first name starts with 'J' and whose email contains specialforce.com.**

➔

```
mysql> SELECT
```

```
-> Employees.first_name,
```

```
-> Employees.last_name,
```

```
-> Employees.email
```

```
-> FROM
```

```
-> Employees
```


```
-> WHERE
```

```
-> Employees.first_name LIKE 'J%'
```

```
-> AND Employees.email LIKE '%specialforce.com%';
```

```
+-----+-----+-----+
| first_name | last_name | email                |
+-----+-----+-----+
| Jyoti      | Verma     | jyoti.verma@specialforce.com |
| Jatin      | Reddy     | jatin.reddy@specialforce.com |
+-----+-----+-----+
```

2 rows in set (0.00 sec)

 **Query 4: Write a query to find all the distinct department names in the Departments table.**

➔

```
mysql> SELECT DISTINCT Departments.department_name FROM Departments;
```

```
+-----+
| department_name |
```




```

+-----+
| Finance |
| HR      |
| IT      |
| Marketing |
| Sales   |
+-----+

```

5 rows in set (0.00 sec)

 **Query 5: Write a query to calculate the total salary expenditure of each department.**



```

mysql> SELECT
-> Departments.department_name,
-> SUM(Employees.salary)
-> FROM
-> Employees
-> JOIN
-> Departments
-> ON
-> Employees.department_id = Departments.department_id
-> GROUP BY
-> Departments.department_name;

```

```

+-----+-----+
| department_name | SUM(Employees.salary) |
+-----+-----+
| Finance        |          40000 |
| HR             |          183000 |
| IT             |          205000 |
| Marketing      |           30000 |
| Sales          |          145000 |
+-----+-----+

```

Query 6: Write a query to find the average salary of employees in the Finance department.



```

mysql> SELECT AVG(Employees.salary)

```

-> FROM Employees

-> join Departments ON

-> Employees.department_id = Departments.department_id

-> WHERE

-> Departments.department_name = 'Finance';


+-----+

| AVG(Employees.salary) |

+-----+

| 40000.0000 |

+-----+

 **Query 7: Write a query to find the minimum and maximum salaries of employees in the Sales department.**

➔

mysql> SELECT

-> MIN(Employees.salary),

-> MAX(Employees.salary)

-> FROM

-> Employees

-> JOIN

-> Departments

-> ON

-> Employees.department_id = Departments.department_id


-> WHERE

-> Departments.department_name = 'Sales';

+-----+-----+

| MIN(Employees.salary) | MAX(Employees.salary) |

70000	75000

 **Query 8: Write a query to count the number of employees in each department.**

➔

```
mysql> SELECT Department_name,COUNT(Employees.employee_id) FROM
Employees JOIN Departments ON Employees.department_id =
Departments.department_id GROUP BY Departments.department_name;
```

Department_name	COUNT(Employees.employee_id)
Finance	1
HR	3
IT	3
Marketing	1
Sales	2

 **Query 9: Write a query to find all employees who were hired between January 1, 2018, and December 31, 2020. Sort the result by hire date in ascending order.**

➔

```
mysql> SELECT
-> Employees.first_name,
-> Employees.last_name,
-> Employees.hire_date
-> FROM
-> Employees
-> WHERE
-> Employees.hire_date BETWEEN '2018-01-01' AND '2020-12-31'
-> ORDER BY
-> Employees.hire_date ASC;
```

first_name	last_name	hire_date
Sanjay	Iyer	2018-01-22
Anil	Patil	2018-09-18
Neha	Kapoor	2019-03-23
Rajesh	Gupta	2020-08-11
Jyoti	Verma	2020-11-02

 **Query 10: Write a query to list all employees who do not have an email address.**

➔

```
mysql> SELECT
```

```
-> Employees.first_name,
-> Employees.last_name
-> FROM
-> Employees
-> WHERE
-> Employees.email IS NULL;
```

Empty set (0.00 sec)

 **Query 11: Write a query to find all employees who work in HR, Finance, or IT departments.**

➔

```
mysql> SELECT
```

```
-> Employees.first_name,
-> Employees.last_name,
-> Departments.department_name
-> FROM
-> Employees
-> JOIN
-> Departments
-> ON
-> Employees.department_id = Departments.department_id
-> WHERE
-> Departments.department_name IN ('HR', 'Finance', 'IT');
```

first_name	last_name	department_name
Pooja	Singh	Finance
Neha	Kapoor	HR
Jatin	Reddy	HR
Kavita	Nair	HR
Ravi	Sharma	IT
Jyoti	Verma	IT
Rajesh	Gupta	IT

🚩 **Query 12:** Write a query to list the first name, last name, and salary of employees earning between ₹30,000 and ₹70,000. Sort the results by salary in descending order.

→

mysql> SELECT

-> Employees.first_name,

-> Employees.last_name,

-> Employees.salary

-> FROM

-> Employees

-> WHERE

-> Employees.salary BETWEEN 30000 AND 70000

-> ORDER BY

-> Employees.salary DESC;

first_name	last_name	salary
Anil	Patil	70000
Jyoti	Verma	60000

Ravi	Sharma	55000	
Kavita	Nair	50000	
Neha	Kapoor	48000	
Pooja	Singh	40000	
Shreya	Mehta	30000	
+-----+	+-----+	+-----+	+-----+

Transaction Management Tasks:

Use transaction control statements to manage the salary updates as follows:

Task 1: Increase HR Salaries:

Write a query to increase the salaries of all employees in the HR department by 5%.
Start a transaction before applying the changes.



```
mysql> UPDATE Employees
-> SET salary = salary * 1.05
-> WHERE department_id = (SELECT
-> department_id FROM Departments WHERE department_name = 'HR');
```

Query OK, 3 rows affected (0.01 sec)

Rows matched: 3 Changed: 3 Warnings: 0

```
mysql> select * from employees;
```

+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+
-----+						
employee_id	first_name	last_name	email		hire_date	salary
department_id						
+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+
-----+						
101	Ravi	Sharma	ravi.sharma@specialforce.com	2017-05-15	55000	
1						
102	Neha	Kapoor	neha.kapoor@specialforce.com	2019-03-23		
50400	2					
103	Jyoti	Verma	jyoti.verma@specialforce.com	2020-11-02	60000	
1						
104	Anil	Patil	anil.patil@specialforce.com	2018-09-18	70000	
3						
105	Pooja	Singh	pooja.singh@specialforce.com	2021-06-10	40000	
4						

106	Sanjay	Iyer	sanjay.iyer@specialforce.com	2018-01-22	75000
107	Jatin	Reddy	jatin.reddy@specialforce.com	2021-12-12	89250
108	Shreya	Mehta	shreya.mehta@specialforce.com	2022-04-19	30000
109	Rajesh	Gupta	rajesh.gupta@specialforce.com	2020-08-11	90000
110	Kavita	Nair	kavita.nair@specialforce.com	2021-02-07	52500

Task 2: Savepoint Before Sales Increase:

Set a savepoint before increasing the salaries of employees in the Sales department by 3%.



```
mysql>
```

```
mysql> UPDATE Employees
```

```
-> SET salary = salary * 1.03
```

```
-> WHERE department_id = (
```

```
-> SELECT department_id FROM Departments WHERE department_name = 'Sales' );
```

Query OK, 2 rows affected (0.00 sec)

Rows matched: 2 Changed: 2 Warnings: 0

Task 3: Rollback Sales Salary Increase:

Rollback to the savepoint created before the Sales salary increase.



```
mysql> ROLLBACK TO SAVEPOINT before_sales_update;
```

Query OK, 0 rows affected (0.01 sec)

```
mysql> select * from Employees;
```

--	--	--	--	--	--

employee_id	first_name	last_name	email	hire_date	salary	department_id
101	Ravi	Sharma	ravi.sharma@specialforce.com	2017-05-15	55000	1
102	Neha	Kapoor	neha.kapoor@specialforce.com	2019-03-23	50400	2
103	Jyoti	Verma	jyoti.verma@specialforce.com	2020-11-02	60000	1
104	Anil	Patil	anil.patil@specialforce.com	2018-09-18	72100	3
105	Pooja	Singh	pooja.singh@specialforce.com	2021-06-10	40000	4
106	Sanjay	Iyer	sanjay.iyer@specialforce.com	2018-01-22	77250	3
107	Jatin	Reddy	jatin.reddy@specialforce.com	2021-12-12	89250	2
108	Shreya	Mehta	shreya.mehta@specialforce.com	2022-04-19	30000	5
109	Rajesh	Gupta	rajesh.gupta@specialforce.com	2020-08-11	90000	1
110	Kavita	Nair	kavita.nair@specialforce.com	2021-02-07	52500	2



Task 4: Commit the Transaction:

After rolling back the Sales increase, commit the changes made to the HR department salaries.



```
mysql> COMMIT;
```

Query OK, 0 rows affected (0.01 sec)

 **Query 13: Write a query to join the Employees and Departments tables to list employees and their department names. Make sure all employees are included, even if they don't belong to any department.**



```
mysql> SELECT Employees.first_name, Employees.last_name,  
Departments.department_name  
-> FROM Employees  
-> LEFT JOIN Departments  
-> ON Employees.department_id = Departments.department_id;
```

```
+-----+-----+-----+  
| first_name | last_name | department_name |  
+-----+-----+-----+  
| Ravi      | Sharma   | IT              |  
| Neha      | Kapoor  | HR              |  
| Jyoti     | Verma    | IT              |  
| Anil      | Patil    | Sales           |  
| Pooja     | Singh    | Finance         |  
| Sanjay    | Iyer     | Sales           |  
| Jatin     | Reddy    | HR              |  
| Shreya    | Mehta    | Marketing       |  
| Rajesh    | Gupta    | IT              |  
| Kavita    | Nair     | HR              |  
+-----+-----+-----+
```

 **Query 14: Write a query to list employees who are working on projects that started after January 1, 2023.**



```
mysql> SELECT Employees.first_name, Employees.last_name, Projects.project_name,  
Projects.start_date
```

```
-> FROM Employees
```

```
-> JOIN Departments
```

```
-> ON Employees.department_id = Departments.department_id
```


```
-> JOIN Projects
```

```
-> ON Departments.department_id = Projects.department_id
```

```
-> WHERE Projects.start_date > '2023-01-01';
```

```
+-----+-----+-----+-----+
```

first_name	last_name	project_name	start_date
Neha	Kapoor	Internal System Audit	2023-02-15
Jatin	Reddy	Internal System Audit	2023-02-15
Kavita	Nair	Internal System Audit	2023-02-15

 **Query 15: Write a query to list all departments, even those without any employees assigned.**



```
mysql> SELECT Departments.department_id, Departments.department_name,
Employees.employee_id
```

```
-> FROM Departments
```


```
-> LEFT JOIN Employees
```

```
-> ON Departments.department_id = Employees.department_id;
```

department_id	department_name	employee_id
4	Finance	105
2	HR	102
2	HR	107
2	HR	110
1	IT	101
1	IT	103
1	IT	109
5	Marketing	108
3	Sales	104

	3	Sales		106	
--	---	-------	--	-----	--

+-----+	+-----+	+-----+	+-----+
---------	---------	---------	---------

 **Query 16: Write a query to find the employee with the highest salary in each department.**

➔

```
mysql> SELECT employee_id, first_name, last_name, department_id, salary
```

```
-> FROM Employees
```

```
-> WHERE(department_id, salary) IN
```

```
-> (SELECT department_id, MAX(salary)
```

```
-> FROM Employees
```

```
-> GROUP BY department_id);
```

+-----+	+-----+	+-----+	+-----+	+-----+
---------	---------	---------	---------	---------

	employee_id		first_name		last_name		department_id		salary	
--	-------------	--	------------	--	-----------	--	---------------	--	--------	--

+-----+	+-----+	+-----+	+-----+	+-----+
---------	---------	---------	---------	---------

	105		Pooja		Singh		4		40000	
--	-----	--	-------	--	-------	--	---	--	-------	--


	106		Sanjay		Iyer		3		77250	
--	-----	--	--------	--	------	--	---	--	-------	--

	107		Jatin		Reddy		2		89250	
--	-----	--	-------	--	-------	--	---	--	-------	--

	108		Shreya		Mehta		5		30000	
--	-----	--	--------	--	-------	--	---	--	-------	--

	109		Rajesh		Gupta		1		90000	
--	-----	--	--------	--	-------	--	---	--	-------	--

+-----+	+-----+	+-----+	+-----+	+-----+
---------	---------	---------	---------	---------

 **Query 17: Write a query to remove all data from the Employees table but keep the structure intact.**

➔

```
mysql> TRUNCATE TABLE Employees;
```

Query OK, 0 rows affected (0.07 sec)

```
mysql> select * from Employees;
```

Empty set (0.01 sec)

```
mysql> desc Employees;
```

Field	Type	Null	Key	Default	Extra
employee_id	int	NO	PRI	NULL	
first_name	varchar(50)	NO		NULL	
last_name	varchar(50)	NO		NULL	
email	varchar(100)	NO	UNI	NULL	
hire_date	date	NO		NULL	
salary	int	NO		NULL	
department_id	int	NO	MUL	NULL	

 **Query 18: Write a query to drop the Projects table from the database.**


➔

```
mysql> DROP TABLE projects;
```

Query OK, 0 rows affected (0.04 sec)

```
mysql> desc projects;
```

ERROR 1146 (42S02): Table 'assignment.projects' doesn't exist

 **Query 19: SpecialForce Private Limited realized they need to store the phone numbers of employees. Write a query to add a new column phone_number (VARCHAR(15)) to the Employees table using the ALTER statement.**

➔

```
mysql> ALTER TABLE Employees ADD phone_number VARCHAR(15);
```


Query OK, 0 rows affected (0.03 sec)

Records: 0 Duplicates: 0 Warnings: 0

 **Query 20: The company also decided to track the budget for each project. Write a query to add a column budget (DECIMAL(10,2)) to the Projects table.**

→

```
mysql> ALTER TABLE Projects ADD budget DECIMAL(10,2);  
ERROR 1146 (42S02): Table 'assignment.projects' doesn't exist
```

 **Query 21: Write a query to find the 2nd largest salary from the Employees table using:**

- A subquery.
- The LIMIT clause.

→

```
mysql> SELECT MAX(salary)
```

```
-> FROM Employees
```

```
-> WHERE salary <
```

```
-> (SELECT MAX(salary)
```

```
-> FROM Employees);
```

```
+-----+
```

```
| MAX(salary) |
```

```
+-----+
```

```
|    NULL    |
```

```
+-----+
```

 **Query 22: Write a query to find the 3rd largest salary from the Employees table using:**

- A subquery.
- The LIMIT clause.

→

```
mysql> SELECT salary
```

-> FROM Employees


-> ORDER BY salary DESC

-> LIMIT 1 OFFSET 1;

Empty set (0.00 sec)

 **Query 23: Write a query to drop the Projects table.**

➔ mysql> DROP TABLE Projects;

 **Query 24: Write a query to truncate the Employees table.**

➔ mysql> TRUNCATE TABLE Employees;

Query OK, 0 rows affected (0.06 sec)