

1.(Exercise: retrieve the records from the table)

EMPLOYEES (Employee_Id, First_Name, Last_Name,
Email, Phone_Number, Hire_Date, Job_Id, Salary, Commission_Pct, Manager_Id, Department_Id)

1. create an employee's table with the following fields: (Emp_id, First_name, Last_name, Phone_No,Hire_date,Job_id,Emp_Salary,Comission_Pct,manager_id,Department_id)
2. Insert five records into the table employees
3. Display the table Employees
4. Find out the employee id, names, salaries of all the employees
5. Find the names of the employees who have a salary greater than or equal to 4800
6. List out the employees whose last name is 'AUSTIN'
7. Find the names of the employees who works in departments 60,70 and 80
8. Display the unique Manager_Id from employees table

PRACTICE 1

Microsoft Windows [Version 10.0.22631.3880]
(c) Microsoft Corporation. All rights reserved.

```
C:\Users\Utkarsha Gupta>mysql -u root -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.37 MySQL Community Server - GPL
```

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

```
mysql> create database WRC;
Query OK, 1 row affected (0.06 sec)
```

```
mysql> use WRC;
Database changed
mysql> CREATE TABLE Employees (
->   Emp_id INT PRIMARY KEY,
->   First_name VARCHAR(50),
```

```

-> Last_name VARCHAR(50),
-> Phone_No VARCHAR(20),
-> Hire_date DATE,
-> Job_id VARCHAR(10),
-> Emp_Salary DECIMAL(10, 2),
-> Commission_Pct DECIMAL(5, 2),
-> Manager_id INT,
-> Department_id INT
-> );

```

Query OK, 0 rows affected (0.07 sec)

```

mysql> INSERT INTO Employees (Emp_id, First_name, Last_name, Phone_No, Hire_date, Job_id,
Emp_Salary, Commission_Pct, Manager_id, Department_id)
-> VALUES

```

```

-> (1, 'John', 'Doe', '123-456-7890', '2020-01-15', 'J101', 5000, 0.10, 101, 60),
-> (2, 'Jane', 'Smith', '123-456-7891', '2019-02-20', 'J102', 4800, 0.12, 102, 70),
-> (3, 'Robert', 'Johnson', '123-456-7892', '2021-03-25', 'J103', 4600, 0.15, 103, 80),
-> (4, 'Michael', 'Austin', '123-456-7893', '2018-04-30', 'J104', 5100, 0.20, 104, 60),
-> (5, 'Emily', 'Davis', '123-456-7894', '2017-05-10', 'J105', 4700, 0.25, 105, 70);

```

Query OK, 5 rows affected (0.02 sec)

Records: 5 Duplicates: 0 Warnings: 0

```

mysql> SELECT * FROM Employees;

```

Emp_id	First_name	Last_name	Phone_No	Hire_date	Job_id	Emp_Salary	Commission_Pct	Manager_id	Department_id
1	John	Doe	123-456-7890	2020-01-15	J101	5000.00	0.10	101	60
2	Jane	Smith	123-456-7891	2019-02-20	J102	4800.00	0.12	102	70
3	Robert	Johnson	123-456-7892	2021-03-25	J103	4600.00	0.15	103	80
4	Michael	Austin	123-456-7893	2018-04-30	J104	5100.00	0.20	104	60
5	Emily	Davis	123-456-7894	2017-05-10	J105	4700.00	0.25	105	70

5 rows in set (0.01 sec)

```

mysql> SELECT Emp_id, First_name, Last_name, Emp_Salary FROM Employees;

```

Emp_id	First_name	Last_name	Emp_Salary
--------	------------	-----------	------------

	1		John		Doe		5000.00	
	2		Jane		Smith		4800.00	
	3		Robert		Johnson		4600.00	
	4		Michael		Austin		5100.00	
	5		Emily		Davis		4700.00	

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

5 rows in set (0.01 sec)

mysql> SELECT First_name, Last_name FROM Employees WHERE Emp_Salary >= 4800;

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

	First_name		Last_name	
--	------------	--	-----------	--

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

	John		Doe	
--	------	--	-----	--

	Jane		Smith	
--	------	--	-------	--

	Michael		Austin	
--	---------	--	--------	--

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

3 rows in set (0.01 sec)

mysql> SELECT * FROM Employees WHERE Last_name = 'Austin';

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

	Emp_id		First_name		Last_name		Phone_No		Hire_date		Job_id		Emp_Salary		Commission_Pct	
	Manager_id		Department_id													

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

	4		Michael		Austin		123-456-7893		2018-04-30		J104		5100.00		0.20		104	
	60																	

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

1 row in set (0.01 sec)

mysql> SELECT First_name, Last_name FROM Employees WHERE Department_id IN (60, 70, 80);

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

	First_name		Last_name	
--	------------	--	-----------	--

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

	John		Doe	
--	------	--	-----	--

	Jane		Smith	
--	------	--	-------	--

	Robert		Johnson	
--	--------	--	---------	--

	Michael		Austin	
--	---------	--	--------	--

	Emily		Davis	
--	-------	--	-------	--

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

5 rows in set (0.00 sec)

mysql> SELECT DISTINCT Manager_id FROM Employees;

+-----+

	Manager_id	
--	------------	--

+-----+

```
| 101 |
| 102 |
| 103 |
| 104 |
| 105 |
```

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

5 rows in set (0.01 sec)

2. (Exercise: update the records in the table)

Create Client_master with the following fields (ClientNO,

Name, Address, City, State, bal_due)

1. create a client master table with attributes
2. insert five records into the Client_Master
3. Display Client Master Table
4. Find the name of Clients whose balance_due >5000
5. Change the bal_due of ClientNO "C123" to Rs. 5100
6. Change the name of Client_master to Client12
7. Display the bal_due heading as "BALANCE" Client master table

PRACTICE 2

```
mysql> CREATE TABLE Client_Master (
-> ClientNO VARCHAR(10) PRIMARY KEY,
-> Name VARCHAR(100),
-> Address VARCHAR(255),
-> City VARCHAR(50),
-> State VARCHAR(50),
-> bal_due DECIMAL(10, 2)
-> );
```

Query OK, 0 rows affected (0.07 sec)

```
mysql> INSERT INTO Client_Master (ClientNO, Name, Address, City, State, bal_due)
-> VALUES
-> ('C101', 'Alice Johnson', '123 Elm St', 'Springfield', 'IL', 4500.00),
-> ('C102', 'Bob Smith', '456 Oak St', 'Centerville', 'OH', 6000.00),
-> ('C103', 'Charlie Brown', '789 Pine St', 'Shelbyville', 'IN', 3000.00),
-> ('C104', 'Diana Prince', '321 Maple St', 'Metropolis', 'NY', 7500.00),
-> ('C105', 'Evan Davis', '654 Birch St', 'Gotham', 'NJ', 4000.00);
```

Query OK, 5 rows affected (0.01 sec)

Records: 5 Duplicates: 0 Warnings: 0

```
mysql> SELECT * FROM Client_Master;
```

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

```
| ClientNO | Name      | Address    | City      | State | bal_due |
```

```

+-----+-----+-----+-----+-----+
| C101 | Alice Johnson | 123 Elm St | Springfield | IL | 4500.00 |
| C102 | Bob Smith | 456 Oak St | Centerville | OH | 6000.00 |
| C103 | Charlie Brown | 789 Pine St | Shelbyville | IN | 3000.00 |
| C104 | Diana Prince | 321 Maple St | Metropolis | NY | 7500.00 |
| C105 | Evan Davis | 654 Birch St | Gotham | NJ | 4000.00 |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

```

```
mysql> SELECT Name FROM Client_Master WHERE bal_due > 5000;
```

```

+-----+
| Name |
+-----+
| Bob Smith |
| Diana Prince |
+-----+
2 rows in set (0.00 sec)

```

```
mysql> UPDATE Client_Master
```

```
-> SET bal_due = 5100.00
```

```
-> WHERE ClientNO = 'C123';
```

```
Query OK, 0 rows affected (0.01 sec)
```

```
Rows matched: 0 Changed: 0 Warnings: 0
```

```
mysql> ALTER TABLE Client_Master RENAME TO Client12;
```

```
Query OK, 0 rows affected (0.05 sec)
```

```
mysql> SELECT ClientNO, Name, Address, City, State, bal_due AS BALANCE
-> FROM Client12;
```

```

+-----+-----+-----+-----+-----+
| ClientNO | Name | Address | City | State | BALANCE |
+-----+-----+-----+-----+-----+
| C101 | Alice Johnson | 123 Elm St | Springfield | IL | 4500.00 |
| C102 | Bob Smith | 456 Oak St | Centerville | OH | 6000.00 |
| C103 | Charlie Brown | 789 Pine St | Shelbyville | IN | 3000.00 |
| C104 | Diana Prince | 321 Maple St | Metropolis | NY | 7500.00 |
| C105 | Evan Davis | 654 Birch St | Gotham | NJ | 4000.00 |
+-----+-----+-----+-----+-----+
5 rows in set (0.01 sec)

```

3. Commands of Rollback and Commit :

Create Teacher table with the following fields (Name, DeptNo, Date of joining, DeptName, Location, Salary)

1. Create Teacher table with the following fields (Id,Name, DeptNo, Date of joining, DeptName, Location, Salary)
2. Insert five records
3. Give Increment of 25% salary for Mathematics Department.
4. Perform Rollback command
5. Give Increment of 15% salary for Commerce Department
6. Perform commit command

PRACTICE 3

```
mysql> CREATE TABLE Teacher (
-> Id INT PRIMARY KEY AUTO_INCREMENT,
-> Name VARCHAR(100),
-> DeptNo INT,
-> Date_of_joining DATE,
-> DeptName VARCHAR(50),
-> Location VARCHAR(100),
-> Salary DECIMAL(10, 2)
-> );
```

Query OK, 0 rows affected (0.04 sec)

```
mysql> INSERT INTO Teacher (Name, DeptNo, Date_of_joining, DeptName, Location, Salary)
-> VALUES
-> ('John Smith', 101, '2020-01-15', 'Mathematics', 'New York', 5000.00),
-> ('Jane Doe', 102, '2019-02-20', 'Science', 'Boston', 4500.00),
-> ('Emily Davis', 103, '2021-03-25', 'Commerce', 'Chicago', 6000.00),
-> ('Michael Brown', 104, '2018-04-30', 'Mathematics', 'Los Angeles', 5200.00),
-> ('Sarah Wilson', 105, '2017-05-10', 'Commerce', 'San Francisco', 5800.00);
```

Query OK, 5 rows affected (0.01 sec)

Records: 5 Duplicates: 0 Warnings: 0

```
mysql> select * from Teacher;
```

Id	Name	DeptNo	Date_of_joining	DeptName	Location	Salary
1	John Smith	101	2020-01-15	Mathematics	New York	5000.00
2	Jane Doe	102	2019-02-20	Science	Boston	4500.00
3	Emily Davis	103	2021-03-25	Commerce	Chicago	6000.00
4	Michael Brown	104	2018-04-30	Mathematics	Los Angeles	5200.00
5	Sarah Wilson	105	2017-05-10	Commerce	San Francisco	5800.00

5 rows in set (0.00 sec)

```
mysql> UPDATE Teacher
```

```
-> SET Salary = Salary * 1.25
-> WHERE DeptName = 'Mathematics';
Query OK, 2 rows affected (0.01 sec)
Rows matched: 2 Changed: 2 Warnings: 0
```

```
mysql> ROLLBACK;
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> UPDATE Teacher
-> SET Salary = Salary * 1.15
-> WHERE DeptName = 'Commerce';
Query OK, 2 rows affected (0.01 sec)
Rows matched: 2 Changed: 2 Warnings: 0
```

```
mysql> COMMIT;
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> select * from Teacher;
+-----+-----+-----+-----+-----+-----+
| Id | Name      | DeptNo | Date_of_joining | DeptName | Location | Salary |
+-----+-----+-----+-----+-----+-----+
| 1 | John Smith | 101 | 2020-01-15 | Mathematics | New York | 6250.00 |
| 2 | Jane Doe   | 102 | 2019-02-20 | Science    | Boston   | 4500.00 |
| 3 | Emily Davis | 103 | 2021-03-25 | Commerce   | Chicago  | 6900.00 |
| 4 | Michael Brown | 104 | 2018-04-30 | Mathematics | Los Angeles | 6500.00 |
| 5 | Sarah Wilson | 105 | 2017-05-10 | Commerce   | San Francisco | 6670.00 |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.01 sec)
```

4. (Exercise on the group by and order by clauses)

Create Sales table with the following fields (Sales No, Salesname, Branch, Salesamount, DOB)

1. Create a Sales Table with the following fields (Sales_No,Sales_Name,Branch,Sales_Amount,DOB)
2. Insert five records
3. Calculate total salesamount in each branch
4. Calculate average salesamount in each branch
5. Display all the salesmen, DOB who are born in the month of December as day in character format i.e. 21-Dec-09
6. Display the name and DOB of salesman in alphabetical order of the month.

PRACTICE 4

```
mysql> CREATE TABLE Sales (  
-> Sales_No INT PRIMARY KEY AUTO_INCREMENT,  
-> Sales_Name VARCHAR(100),  
-> Branch VARCHAR(50),  
-> Sales_Amount DECIMAL(10, 2),  
-> DOB DATE  
-> );
```

Query OK, 0 rows affected (0.04 sec)

```
mysql> INSERT INTO Sales (Sales_Name, Branch, Sales_Amount, DOB)  
-> VALUES  
-> ('Alice Johnson', 'North', 12000.00, '1985-12-15'),  
-> ('Bob Smith', 'South', 15000.00, '1990-06-22'),  
-> ('Charlie Brown', 'North', 8000.00, '1979-12-10'),  
-> ('Diana Prince', 'East', 20000.00, '1982-11-05'),  
-> ('Emily Davis', 'South', 18000.00, '1988-12-25');
```

Query OK, 5 rows affected (0.01 sec)

Records: 5 Duplicates: 0 Warnings: 0

```
mysql> select * from Sales;
```

Sales_No	Sales_Name	Branch	Sales_Amount	DOB
1	Alice Johnson	North	12000.00	1985-12-15
2	Bob Smith	South	15000.00	1990-06-22
3	Charlie Brown	North	8000.00	1979-12-10
4	Diana Prince	East	20000.00	1982-11-05
5	Emily Davis	South	18000.00	1988-12-25

5 rows in set (0.00 sec)

```
mysql> SELECT Branch, SUM(Sales_Amount) AS Total_Sales_Amount  
-> FROM Sales  
-> GROUP BY Branch;
```

Branch	Total_Sales_Amount
North	20000.00
South	33000.00
East	20000.00

3 rows in set (0.01 sec)

```
mysql> SELECT Branch, AVG(Sales_Amount) AS Average_Sales_Amount
-> FROM Sales
-> GROUP BY Branch;
```

Branch	Average_Sales_Amount
North	10000.000000
South	16500.000000
East	20000.000000

3 rows in set (0.01 sec)

```
mysql> SELECT Sales_Name, DATE_FORMAT(DOB, '%d-%b-%y') AS DOB
-> FROM Sales
-> WHERE MONTH(DOB) = 12;
```

Sales_Name	DOB
Alice Johnson	15-Dec-85
Charlie Brown	10-Dec-79
Emily Davis	25-Dec-88

3 rows in set (0.01 sec)

```
mysql> SELECT Sales_Name, DATE_FORMAT(DOB, '%d-%b-%y') AS DOB
-> FROM Sales
-> ORDER BY MONTH(DOB), DAY(DOB);
```

Sales_Name	DOB
Bob Smith	22-Jun-90
Diana Prince	05-Nov-82
Charlie Brown	10-Dec-79
Alice Johnson	15-Dec-85
Emily Davis	25-Dec-88

5 rows in set (0.01 sec)

```
mysql>SYSTEM CLS
```

5. Create an Emp table with the following fields:

(EmpNo, EmpName, Job,Basic, DA, HRA,PF, GrossPay, NetPay)

1. create an employee table with the following fields: (Emp_No,Emp_ Name, Designation, basic, DA, HRA, PF, Gross pay, Net pay)
2. Insert Five Records and calculate GrossPay and NetPay.
3. Adding column to table and Updating Attributes DA
4. Adding column to table and Updating Attributes HRA
5. Adding column to table and Updating Attributes PF
6. Adding column to table and Updating Attributes Gross Pay
7. Adding column to table and Updating Attributes Net Pay
8. Display the employee table
9. Display the employees whose Basic is lowest in each department.
10. If NetPay is less than Rs. 10,000 add Rs. 1200 as special allowance
11. Display the employees whose GrossPay lies between 10,000 & 20,000
12. Display all the employees who earn maximum salary.

PRACTICE 5

```
mysql> CREATE TABLE Emp (  
-> Emp_No INT PRIMARY KEY AUTO_INCREMENT,  
-> Emp_Name VARCHAR(100),  
-> Designation VARCHAR(50),  
-> Basic DECIMAL(10, 2),  
-> DA DECIMAL(10, 2),  
-> HRA DECIMAL(10, 2),  
-> PF DECIMAL(10, 2),  
-> GrossPay DECIMAL(10, 2),  
-> NetPay DECIMAL(10, 2)  
-> );
```

Query OK, 0 rows affected (0.03 sec)

```
mysql> INSERT INTO Emp (Emp_Name, Designation, Basic, DA, HRA, PF)  
-> VALUES  
-> ('John Doe', 'Manager', 8000.00, 2000.00, 1500.00, 800.00),  
-> ('Jane Smith', 'Developer', 6000.00, 1500.00, 1200.00, 600.00),  
-> ('Robert Brown', 'Tester', 5000.00, 1200.00, 1000.00, 500.00),  
-> ('Emily Davis', 'Analyst', 7000.00, 1800.00, 1300.00, 700.00),  
-> ('Michael Wilson', 'Support', 4000.00, 1000.00, 800.00, 400.00);
```

Query OK, 5 rows affected (0.01 sec)

Records: 5 Duplicates: 0 Warnings: 0

```
mysql>
```

```
mysql> -- Calculate GrossPay and NetPay
```

```
mysql> UPDATE Emp
```

```
-> SET GrossPay = Basic + DA + HRA,  
-> NetPay = GrossPay - PF;  
Query OK, 5 rows affected (0.01 sec)  
Rows matched: 5 Changed: 5 Warnings: 0
```

```
mysql> ALTER TABLE Emp ADD COLUMN DECIMAL(10, 2);  
Query OK, 5 rows affected (0.00 sec)
```

```
mysql> -- Assume updating DA as per new policy  
mysql> UPDATE Emp SET DA = CASE  
-> WHEN Designation = 'Manager' THEN 2500.00  
-> WHEN Designation = 'Developer' THEN 1800.00  
-> WHEN Designation = 'Tester' THEN 1500.00  
-> WHEN Designation = 'Analyst' THEN 2000.00  
-> WHEN Designation = 'Support' THEN 1200.00  
-> END;  
Query OK, 5 rows affected (0.00 sec)  
Rows matched: 5 Changed: 5 Warnings: 0
```

```
mysql> select * from Emp  
-> ;  
+-----+-----+-----+-----+-----+-----+-----+-----+  
| Emp_No | Emp_Name | Designation | Basic | DA | HRA | PF | GrossPay | NetPay |  
+-----+-----+-----+-----+-----+-----+-----+-----+  
| 1 | John Doe | Manager | 8000.00 | 2500.00 | 1500.00 | 800.00 | 11500.00 | 10700.00 |  
| 2 | Jane Smith | Developer | 6000.00 | 1800.00 | 1200.00 | 600.00 | 8700.00 | 8100.00 |  
| 3 | Robert Brown | Tester | 5000.00 | 1500.00 | 1000.00 | 500.00 | 7200.00 | 6700.00 |  
| 4 | Emily Davis | Analyst | 7000.00 | 2000.00 | 1300.00 | 700.00 | 10100.00 | 9400.00 |  
| 5 | Michael Wilson | Support | 4000.00 | 1200.00 | 800.00 | 400.00 | 5800.00 | 5400.00 |  
+-----+-----+-----+-----+-----+-----+-----+-----+  
5 rows in set (0.00 sec)
```

```
-- Assume updating HRA as per new policy  
mysql> UPDATE Emp SET HRA = CASE  
-> WHEN Designation = 'Manager' THEN 1600.00  
-> WHEN Designation = 'Developer' THEN 1400.00  
-> WHEN Designation = 'Tester' THEN 1200.00  
-> WHEN Designation = 'Analyst' THEN 1500.00  
-> WHEN Designation = 'Support' THEN 1000.00  
-> END;  
Query OK, 5 rows affected (0.01 sec)  
Rows matched: 5 Changed: 5 Warnings: 0
```

```
mysql> select * from Emp;
```

```

+-----+-----+-----+-----+-----+-----+-----+-----+
| Emp_No | Emp_Name   | Designation | Basic  | DA   | HRA   | PF   | GrossPay | NetPay |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | John Doe   | Manager    | 8000.00 | 2500.00 | 1600.00 | 800.00 | 11500.00 | 10700.00 |
| 2 | Jane Smith | Developer  | 6000.00 | 1800.00 | 1400.00 | 600.00 | 8700.00 | 8100.00 |
| 3 | Robert Brown | Tester    | 5000.00 | 1500.00 | 1200.00 | 500.00 | 7200.00 | 6700.00 |
| 4 | Emily Davis | Analyst    | 7000.00 | 2000.00 | 1500.00 | 700.00 | 10100.00 | 9400.00 |
| 5 | Michael Wilson | Support   | 4000.00 | 1200.00 | 1000.00 | 400.00 | 5800.00 | 5400.00 |
+-----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

```

```

mysql> -- Assume updating HRA as per new policy
mysql> UPDATE Emp SET HRA = CASE
-> WHEN Designation = 'Manager' THEN 1600.00
-> WHEN Designation = 'Developer' THEN 1400.00
-> WHEN Designation = 'Tester' THEN 1200.00
-> WHEN Designation = 'Analyst' THEN 1500.00
-> WHEN Designation = 'Support' THEN 1000.00
-> END;

```

Query OK, 0 rows affected (0.00 sec)
Rows matched: 5 Changed: 0 Warnings: 0

```

mysql> select * from Emp;
+-----+-----+-----+-----+-----+-----+-----+-----+
| Emp_No | Emp_Name   | Designation | Basic  | DA   | HRA   | PF   | GrossPay | NetPay |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | John Doe   | Manager    | 8000.00 | 2500.00 | 1600.00 | 800.00 | 11500.00 | 10700.00 |
| 2 | Jane Smith | Developer  | 6000.00 | 1800.00 | 1400.00 | 600.00 | 8700.00 | 8100.00 |
| 3 | Robert Brown | Tester    | 5000.00 | 1500.00 | 1200.00 | 500.00 | 7200.00 | 6700.00 |
| 4 | Emily Davis | Analyst    | 7000.00 | 2000.00 | 1500.00 | 700.00 | 10100.00 | 9400.00 |
| 5 | Michael Wilson | Support   | 4000.00 | 1200.00 | 1000.00 | 400.00 | 5800.00 | 5400.00 |
+-----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

```

```

mysql> ALTER TABLE Emp ADD COLUMN PF DECIMAL(10, 2);
ERROR 1060 (42S21): Duplicate column name 'PF'
mysql>

```

```

mysql> -- Assume updating PF as per new policy
mysql> UPDATE Emp SET PF = CASE
-> WHEN Designation = 'Manager' THEN 900.00
-> WHEN Designation = 'Developer' THEN 700.00
-> WHEN Designation = 'Tester' THEN 600.00
-> WHEN Designation = 'Analyst' THEN 800.00

```

-> WHEN Designation = 'Support' THEN 500.00

-> END;

Query OK, 5 rows affected (0.01 sec)

Rows matched: 5 Changed: 5 Warnings: 0

mysql> select * from Emp;

Emp_No	Emp_Name	Designation	Basic	DA	HRA	PF	GrossPay	NetPay
1	John Doe	Manager	8000.00	2500.00	1600.00	900.00	11500.00	10700.00
2	Jane Smith	Developer	6000.00	1800.00	1400.00	700.00	8700.00	8100.00
3	Robert Brown	Tester	5000.00	1500.00	1200.00	600.00	7200.00	6700.00
4	Emily Davis	Analyst	7000.00	2000.00	1500.00	800.00	10100.00	9400.00
5	Michael Wilson	Support	4000.00	1200.00	1000.00	500.00	5800.00	5400.00

5 rows in set (0.00 sec)

mysql> -- No need to add this column again; we calculated it in step 2. Ensure to recalculate if needed.

mysql> -- No need to add this column again; we calculated it in step 2. Ensure to recalculate if needed.

mysql> SELECT * FROM Emp;

Emp_No	Emp_Name	Designation	Basic	DA	HRA	PF	GrossPay	NetPay
1	John Doe	Manager	8000.00	2500.00	1600.00	900.00	11500.00	10700.00
2	Jane Smith	Developer	6000.00	1800.00	1400.00	700.00	8700.00	8100.00
3	Robert Brown	Tester	5000.00	1500.00	1200.00	600.00	7200.00	6700.00
4	Emily Davis	Analyst	7000.00	2000.00	1500.00	800.00	10100.00	9400.00
5	Michael Wilson	Support	4000.00	1200.00	1000.00	500.00	5800.00	5400.00

5 rows in set (0.00 sec)

mysql> select * from Emp where Basic in (select min(Basic) from Emp group by Designation);

Emp_No	Emp_Name	Designation	Basic	DA	HRA	PF	GrossPay	NetPay	SpecialAllowance
1	John Doe	Manager	8000.00	2500.00	1600.00	900.00	11500.00	10700.00	0.00
2	Jane Smith	Developer	6000.00	1800.00	1400.00	700.00	8700.00	8100.00	1200.00
3	Robert Brown	Tester	5000.00	1500.00	1200.00	600.00	7200.00	6700.00	1200.00
4	Emily Davis	Analyst	7000.00	2000.00	1500.00	800.00	10100.00	9400.00	1200.00

5	Michael Wilson	Support	4000.00	1200.00	1000.00	500.00	5800.00	5400.00	1200.00
---	----------------	---------	---------	---------	---------	--------	---------	---------	---------

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

5 rows in set (0.01 sec)

```
mysql> ALTER TABLE Emp ADD COLUMN SpecialAllowance DECIMAL(10, 2);
ERROR 1060 (42S21): Duplicate column name 'SpecialAllowance'
```

```
mysql>
```

```
mysql> UPDATE Emp
```

```
-> SET SpecialAllowance = CASE
```

```
->   WHEN NetPay < 10000 THEN 1200.00
```

```
->   ELSE 0.00
```

```
-> END;
```

Query OK, 0 rows affected (0.00 sec)

Rows matched: 5 Changed: 0 Warnings: 0

```
mysql> UPDATE Emp
```

```
-> SET SpecialAllowance = CASE
```

```
->   WHEN NetPay < 10000 THEN 1200.00
```

```
->   ELSE 0.00
```

```
-> END;
```

Query OK, 0 rows affected (0.00 sec)

Rows matched: 5 Changed: 0 Warnings: 0

```
mysql> SELECT *
```

```
-> FROM Emp
```

```
-> WHERE GrossPay BETWEEN 10000 AND 20000;
```

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

Emp_No	Emp_Name	Designation	Basic	DA	HRA	PF	GrossPay	NetPay	SpecialAllowance
--------	----------	-------------	-------	----	-----	----	----------	--------	------------------

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

1	John Doe	Manager	8000.00	2500.00	1600.00	900.00	11500.00	10700.00	0.00
---	----------	---------	---------	---------	---------	--------	----------	----------	------

4	Emily Davis	Analyst	7000.00	2000.00	1500.00	800.00	10100.00	9400.00	1200.00
---	-------------	---------	---------	---------	---------	--------	----------	---------	---------

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

2 rows in set (0.01 sec)

```
mysql> SELECT *
```

```
-> FROM Emp
```

```
-> WHERE GrossPay = (SELECT MAX(GrossPay) FROM Emp);
```

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

Emp_No	Emp_Name	Designation	Basic	DA	HRA	PF	GrossPay	NetPay	SpecialAllowance
--------	----------	-------------	-------	----	-----	----	----------	--------	------------------

```

+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | John Doe | Manager | 8000.00 | 2500.00 | 1600.00 | 900.00 | 11500.00 | 10700.00 |
| 0.00 |

```

1 row in set (0.00 sec)

mysql> system cls;

6. Employee Database an Enterprise wishes to maintain a database to automate its operations.

Enterprise is divided into certain departments and each department consists of employees. The

following two tables describes the automation schemas

Dept (deptno, dname, loc) Emp (empno, ename, job, mgr, hiredate, sal, comm, deptno)

1. Create Dept table: Dept (deptno, dname, loc)
2. Create Dept table: Emp (empno, ename, job, mgr, hiredate, sal, comm, deptno)
3. Insert data int Dept and Emp tables
4. Update the employee salary by 15%, whose experience is greater than 30 years
5. Delete the employees, who completed 30 years of service.
6. Display the manager who is having maximum number of employees working under him?
7. Create a view, which contain employee names and their manager

PRACTICE 6

```

mysql> CREATE TABLE Dept (
-> deptno INT PRIMARY KEY,
-> dname VARCHAR(100),
-> loc VARCHAR(100)
-> );

```

Query OK, 0 rows affected (0.03 sec)

```

mysql> CREATE TABLE Emp1 (
-> empno INT PRIMARY KEY AUTO_INCREMENT,
-> ename VARCHAR(100),
-> job VARCHAR(50),
-> mgr INT,
-> hiredate DATE,
-> sal DECIMAL(10, 2),

```

```

-> comm DECIMAL(10, 2),
-> deptno INT,
-> FOREIGN KEY (deptno) REFERENCES Dept(deptno),
-> FOREIGN KEY (mgr) REFERENCES Emp1(empno)
-> );

```

Query OK, 0 rows affected (0.04 sec)

mysql> -- Insert data into Dept table

mysql> INSERT INTO Dept (deptno, dname, loc)

-> VALUES

-> (10, 'Sales', 'New York'),

-> (20, 'Marketing', 'Los Angeles'),

-> (30, 'HR', 'Chicago');

Query OK, 3 rows affected (0.01 sec)

Records: 3 Duplicates: 0 Warnings: 0

mysql>

mysql> -- Insert data into Emp1 table

mysql> INSERT INTO Emp1 (ename, job, mgr, hiredate, sal, comm, deptno)

-> VALUES

-> ('John Doe', 'Manager', NULL, '1990-06-15', 8000.00, 500.00, 10),

-> ('Jane Smith', 'Salesperson', 1, '1995-04-20', 6000.00, 400.00, 10),

-> ('Robert Brown', 'Clerk', 1, '1992-08-25', 4000.00, 200.00, 20),

-> ('Emily Davis', 'Analyst', 1, '1988-11-30', 7000.00, 300.00, 20),

-> ('Michael Wilson', 'Salesperson', 1, '1985-02-15', 5000.00, 250.00, 30);

Query OK, 5 rows affected (0.01 sec)

Records: 5 Duplicates: 0 Warnings: 0

mysql> select * from Dept;

```

+-----+-----+-----+
| deptno | dname   | loc      |
+-----+-----+-----+
| 10     | Sales   | New York |
| 20     | Marketing | Los Angeles |
| 30     | HR      | Chicago  |
+-----+-----+-----+

```

3 rows in set (0.00 sec)

mysql> select * from Emp1;

```

+-----+-----+-----+-----+-----+-----+-----+
| empno | ename    | job      | mgr | hiredate | sal   | comm  | deptno |
+-----+-----+-----+-----+-----+-----+-----+
| 1     | John Doe | Manager  | NULL | 1990-06-15 | 8000.00 | 500.00 | 10     |
| 2     | Jane Smith | Salesperson | 1 | 1995-04-20 | 6000.00 | 400.00 | 10     |

```


3	Robert Brown	Clerk	1	1992-08-25	4000.00	200.00	20
4	Emily Davis	Analyst	1	1988-11-30	7000.00	300.00	20
5	Michael Wilson	Salesperson	1	1985-02-15	5000.00	250.00	30

-----+

5 rows in set (0.00 sec)

mysql> UPDATE Emp1

-> SET sal = sal * 1.15

-> WHERE DATEDIFF(CURDATE(), hiredate) / 365 > 30;

Query OK, 4 rows affected (0.01 sec)

Rows matched: 4 Changed: 4 Warnings: 0

mysql> select * from Emp1;

-----+

empno	ename	job	mgr	hiredate	sal	comm	deptno
1	John Doe	Manager	NULL	1990-06-15	9200.00	500.00	10
2	Jane Smith	Salesperson	1	1995-04-20	6000.00	400.00	10
3	Robert Brown	Clerk	1	1992-08-25	4600.00	200.00	20
4	Emily Davis	Analyst	1	1988-11-30	8050.00	300.00	20
5	Michael Wilson	Salesperson	1	1985-02-15	5750.00	250.00	30

-----+

5 rows in set (0.00 sec)

mysql> DELETE FROM Emp1

-> WHERE DATEDIFF(CURDATE(), hiredate) / 365 >= 30;

ERROR 1451 (23000): Cannot delete or update a parent row: a foreign key constraint fails (`wrc`.`emp1`, CONSTRAINT `emp1_ibfk_2` FOREIGN KEY (`mgr`) REFERENCES `emp1` (`empno`))

mysql> SELECT mgr, COUNT(*) AS num_employees

-> FROM Emp1

-> WHERE mgr IS NOT NULL

-> GROUP BY mgr

-> ORDER BY num_employees DESC

-> LIMIT 1;

-----+

mgr	num_employees
-----	---------------

-----+

1	4
---	---

-----+

1 row in set (0.01 sec)

mysql> CREATE VIEW EmployeeManager AS

-> SELECT e1.ename AS Employee_Name, e2.ename AS Manager_Name

-> FROM Emp1 e1

```
-> LEFT JOIN Emp1 e2 ON e1.mgr = e2.empno;  
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> SET FOREIGN_KEY_CHECKS = 0;  
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> DELETE FROM Emp1  
-> WHERE DATEDIFF(CURDATE(), hiredate) / 365 >= 30;  
Query OK, 4 rows affected (0.01 sec)
```

```
mysql> select * from Emp1;  
+-----+-----+-----+-----+-----+-----+-----+  
| empno | ename   | job      | mgr | hiredate | sal   | comm  | deptno |  
+-----+-----+-----+-----+-----+-----+-----+  
| 2 | Jane Smith | Salesperson | 1 | 1995-04-20 | 6000.00 | 400.00 | 10 |  
+-----+-----+-----+-----+-----+-----+-----+  
1 row in set (0.00 sec)
```

7. Using Employee Database above perform the following queries

- a) Determine the names of employee, who earn more than their managers.
- b) Determine the names of employees, who take highest salary in their departments.
- c) Determine the employees, who are located at the same place.
- d) Determine the employees, whose total salary is like the minimum Salary of any department.
- e) Determine the department which does not contain any employees.

PRACTICE 7

```
mysql> SELECT e1.ename AS Employee_Name, e1.sal AS Employee_Salary, e2.ename AS  
Manager_Name, e2.sal AS Manager_Salary  
-> FROM Emp1 e1  
-> JOIN Emp1 e2 ON e1.mgr = e2.empno  
-> WHERE e1.sal > e2.sal;  
Empty set (0.00 sec)
```

```
mysql> INSERT INTO Emp1 (ename, job, mgr, hiredate, sal, comm, deptno)  
-> VALUES  
-> ('John Doe', 'Manager', NULL, '1990-06-15', 8000.00, 500.00, 10),  
-> ('Jane Smith', 'Salesperson', 1, '1995-04-20', 6000.00, 400.00, 10),  
-> ('Robert Brown', 'Clerk', 1, '1992-08-25', 4000.00, 200.00, 20),
```

```
-> ('Emily Davis', 'Analyst', 1, '1988-11-30', 7000.00, 300.00, 20),
-> ('Michael Wilson', 'Salesperson', 1, '1985-02-15', 5000.00, 250.00, 30);
Query OK, 5 rows affected (0.01 sec)
Records: 5 Duplicates: 0 Warnings: 0
```

```
mysql> SELECT e1.ename AS Employee_Name, e1.sal AS Employee_Salary, e2.ename AS
Manager_Name, e2.sal AS Manager_Salary
-> FROM Emp1 e1
-> JOIN Emp1 e2 ON e1.mgr = e2.empno
-> WHERE e1.sal > e2.sal;
Empty set (0.00 sec)
```

```
mysql> SELECT e1.ename AS Employee_Name, e1.deptno, e1.sal
-> FROM Emp1 e1
-> JOIN (
->   SELECT deptno, MAX(sal) AS max_sal
->   FROM Emp1
->   GROUP BY deptno
-> ) e2 ON e1.deptno = e2.deptno AND e1.sal = e2.max_sal;
+-----+-----+-----+
| Employee_Name | deptno | sal   |
+-----+-----+-----+
| John Doe      | 10     | 8000.00 |
| Emily Davis   | 20     | 7000.00 |
| Michael Wilson | 30     | 5000.00 |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

```
mysql> SELECT e1.ename AS Employee_Name, e1.deptno, e1.sal
-> FROM Emp1 e1
-> JOIN (
->   SELECT deptno, MAX(sal) AS max_sal
->   FROM Emp1
->   GROUP BY deptno
-> ) e2 ON e1.deptno = e2.deptno AND e1.sal = e2.max_sal;
+-----+-----+-----+
| Employee_Name | deptno | sal   |
+-----+-----+-----+
| John Doe      | 10     | 8000.00 |
| Emily Davis   | 20     | 7000.00 |
| Michael Wilson | 30     | 5000.00 |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

```
mysql> SELECT e1.ename, e1.sal
```

```
-> FROM Emp1 e1
```

```
-> WHERE e1.sal = (SELECT MIN(sal) FROM Emp1 GROUP BY deptno);
```

```
ERROR 1242 (21000): Subquery returns more than 1 row
```

```
mysql> SELECT e1.ename, e1.sal from Emp1 e1 where e1.sal = (select min(sal) from Emp1 group by deptno);
```

```
ERROR 1242 (21000): Subquery returns more than 1 row
```

```
mysql> SELECT e1.ename, e1.sal from Emp1 e1 where e1.sal in (select min(sal) from Emp1 group by deptno);
```

```
+-----+-----+
| ename      | sal    |
+-----+-----+
| Jane Smith  | 6000.00 |
| Jane Smith  | 6000.00 |
| Robert Brown | 4000.00 |
| Michael Wilson | 5000.00 |
+-----+-----+
```

```
4 rows in set (0.00 sec)
```

```
mysql> SELECT e1.ename, e1.sal
```

```
-> FROM Emp1 e1
```

```
-> WHERE e1.sal = (SELECT MIN(sal) FROM Emp1 GROUP BY deptno);
```

```
ERROR 1242 (21000): Subquery returns more than 1 row
```

```
mysql> SELECT d.deptno, d.dname
```

```
-> FROM Dept d
```

```
-> LEFT JOIN Emp1 e ON d.deptno = e.deptno
```

```
-> WHERE e.empno IS NULL;
```

```
Empty set (0.00 sec)
```

```
mysql> select * from Emp1;
```

```
+-----+-----+-----+-----+-----+-----+-----+-----+
| empno | ename      | job      | mgr | hiredate | sal    | comm    | deptno |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 2 | Jane Smith | Salesperson | 1 | 1995-04-20 | 6000.00 | 400.00 | 10 |
| 6 | John Doe   | Manager    | NULL | 1990-06-15 | 8000.00 | 500.00 | 10 |
| 7 | Jane Smith | Salesperson | 1 | 1995-04-20 | 6000.00 | 400.00 | 10 |
| 8 | Robert Brown | Clerk      | 1 | 1992-08-25 | 4000.00 | 200.00 | 20 |
| 9 | Emily Davis | Analyst     | 1 | 1988-11-30 | 7000.00 | 300.00 | 20 |
| 10 | Michael Wilson | Salesperson | 1 | 1985-02-15 | 5000.00 | 250.00 | 30 |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

```
6 rows in set (0.00 sec)
```

8. Using the tables “DEPARTMENTS” and “EMPLOYEES” above perform the following queries

- Display the employee details, departments that the departments are same in both the emp and dept.
- Display the employee name and Department name by implementing a left outer join.
- Display the employee name and Department name by implementing a right outer join.
- Display the details of those who draw the salary greater than the average salary.

PRACTICE 8

```
mysql> SELECT e.*
```

```
-> FROM Emp1 e
```

```
-> JOIN Dept d ON e.deptno = d.deptno;
```

empno	ename	job	mgr	hiredate	sal	comm	deptno
2	Jane Smith	Salesperson	1	1995-04-20	6000.00	400.00	10
6	John Doe	Manager	NULL	1990-06-15	8000.00	500.00	10
7	Jane Smith	Salesperson	1	1995-04-20	6000.00	400.00	10
8	Robert Brown	Clerk	1	1992-08-25	4000.00	200.00	20
9	Emily Davis	Analyst	1	1988-11-30	7000.00	300.00	20
10	Michael Wilson	Salesperson	1	1985-02-15	5000.00	250.00	30

```
6 rows in set (0.00 sec)
```

```
mysql> SELECT e.ename AS Employee_Name, d.dname AS Department_Name
```

```
-> FROM Emp1 e
```

```
-> LEFT JOIN Dept d ON e.deptno = d.deptno;
```

Employee_Name	Department_Name
Jane Smith	Sales
John Doe	Sales
Jane Smith	Sales
Robert Brown	Marketing
Emily Davis	Marketing
Michael Wilson	HR

```
6 rows in set (0.01 sec)
```

```
mysql> SELECT e.ename AS Employee_Name, d.dname AS Department_Name
```

```
-> FROM Emp1 e
-> RIGHT JOIN Dept d ON e.deptno = d.deptno;
```

```
+-----+-----+
| Employee_Name | Department_Name |
+-----+-----+
| Jane Smith    | Sales          |
| John Doe      | Sales          |
| Jane Smith    | Sales          |
| Robert Brown  | Marketing      |
| Emily Davis   | Marketing      |
| Michael Wilson| HR             |
+-----+-----+
6 rows in set (0.00 sec)
```

```
mysql> SELECT e.*
-> FROM Emp1 e
-> WHERE e.sal > (SELECT AVG(sal) FROM Emp1);
```

```
+-----+-----+-----+-----+-----+-----+-----+-----+
| empno | ename   | job   | mgr | hiredate | sal   | comm  | deptno |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 6 | John Doe | Manager | NULL | 1990-06-15 | 8000.00 | 500.00 | 10 |
| 9 | Emily Davis | Analyst | 1 | 1988-11-30 | 7000.00 | 300.00 | 20 |
+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.01 sec)
```

```
mysql>
```

