



Atma Ram Sanatan Dharma Collage  
University of Delhi



# DATABASE MANAGMENT SYSTEM

## PRACTICAL FILE

Submitted By -  
Suraj Rai  
Roll no. 21/18102  
BSc (Hons) Computer Science

Submitted To -  
Manisha Bargi Ma'am  
Department of Computer science

# INDEX

| S No. | Objective                                                                                                                          |
|-------|------------------------------------------------------------------------------------------------------------------------------------|
| 31    | Query to display the total number of supervisors without listing their names.                                                      |
| 32    | Query to display the Department Name, Location Name, No. of Employees and the average salary for all employees in that department. |
| 33    | Query to display Name and Hire Date for all employees in the same dept. as Blake.                                                  |
| 34    | Query to display the Employee No. And Name for all employees who earn more than the average salary.                                |
| 35    | Query to display Employee Number and Name for all employees who work in a department with any employee whose name contains a 'T'.  |
| 36    | Query to display the names and salaries of all employees who report to supervisor named 'King'.                                    |
| 37    | Query to display the department no, name and job for all employees in the Sales department.                                        |
| 38    | Display names of employees along with their department name who have more than 20 years' experience.                               |
| 39    | Display total number of departments at each location.                                                                              |
| 40    | Find the department name in which at least 20 employees work in.                                                                   |
| 41    | Query to find the employee' name who is not supervisor and name of supervisor supervising more than 5 employees.                   |
| 42    | Query to display the job type with maximum and minimum employees.                                                                  |

## DATABASE SCHEMA

### RELATION: EMPLOYEE

```
mysql> DESC EMPLOYEE;
```

| Field         | Type          | Null | Key | Default | Extra |
|---------------|---------------|------|-----|---------|-------|
| Eno           | char(3)       | NO   | PRI | NULL    |       |
| Ename         | varchar(50)   | NO   |     | NULL    |       |
| Job_type      | varchar(50)   | NO   |     | NULL    |       |
| SupervisorEno | char(3)       | YES  | MUL | NULL    |       |
| Hire_date     | date          | NO   |     | NULL    |       |
| Dno           | int           | YES  | MUL | NULL    |       |
| Commission    | decimal(10,2) | YES  |     | NULL    |       |
| Salary        | decimal(7,2)  | NO   |     | NULL    |       |

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

### RELATION: DEPARTMENT

```
mysql> DESC DEPARTMENT;
```

| Field    | Type        | Null | Key | Default   | Extra |
|----------|-------------|------|-----|-----------|-------|
| Dno      | int         | NO   | PRI | NULL      |       |
| Dname    | varchar(50) | YES  |     | NULL      |       |
| Location | varchar(50) | YES  |     | New Delhi |       |

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

## DDL

```
CREATE TABLE EMPLOYEE
```

```
(  
    Eno          CHAR(3)    NOT NULL,  
    Ename        VARCHAR(50) NOT NULL,  
    Job_type     VARCHAR(50) NOT NULL,  
    SupervisorEno CHAR(3),  
    Hire_date    DATE       NOT NULL,  
    Dno          INT,  
    Commission   DEC(10, 2),  
    Salary       DEC(7, 2)  NOT NULL,  
    PRIMARY KEY (Eno)  
);
```

```
CREATE TABLE DEPARTMENT
```

```
(  
    Dno      INT NOT NULL,  
    Dname    VARCHAR(50),  
    Location VARCHAR(50) DEFAULT 'New Delhi',  
    PRIMARY KEY (Dno)  
);
```

```
ALTER TABLE EMPLOYEE
```

```
ADD CONSTRAINT SupervisorConstraint  
FOREIGN KEY (SupervisorEno) REFERENCES EMPLOYEE (Eno);
```

```
ALTER TABLE EMPLOYEE
```

```
ADD CONSTRAINT DepartmentConstraint  
FOREIGN KEY (Dno) REFERENCES DEPARTMENT (Dno);
```

## DML

```
INSERT INTO DEPARTMENT
VALUES (10, 'Headquarters', 'Mumbai'),
       (45, 'Legal', 'Mumbai'),
       (30, 'IT', 'Bengaluru'),
       (20, 'Outreach', 'Dallas'),
       (95, 'Purchase', 'Dallas');
```

```
INSERT INTO DEPARTMENT (Dno, Dname)
VALUES (90, 'Sales'),
       (40, 'Administration'),
       (50, 'Research');
```

```
INSERT INTO EMPLOYEE
VALUES ('778', 'John Smith', 'Managing Director', NULL, '1980-06-12', 10, NULL, 5800.23),
       ('153', 'Andy Beaford', 'Manager', '778', '2000-09-08', 40, NULL, 502.30),
       ('23', 'Sudipto Ghosh', 'Manager', '778', '1981-11-21', 30, NULL, 2465.92),
       ('25', 'Anmol Gupta', 'Manager', '778', '1981-09-11', 40, NULL, 2865.92),
       ('86', 'Tania Chauhan', 'Manager', '778', '1983-08-14', 90, 235.90, 2921.37),
       ('89', 'Alicia Jones', 'Manager', '778', '1982-10-26', 95, NULL, 980.35),
       ('232', 'Isa Jaques', 'Manager', '778', '1996-07-23', 45, NULL, 651.05),
       ('520', 'King Hethron', 'Manager', '778', '2015-10-13', 90, NULL, 497.18),
       ('188', 'Sherline Harbord', 'Manager', '778', '1995-08-05', 40, NULL, 926.42),
       ('146', 'Donielle Tuite', 'Intern', '520', '2016-06-12', 90, NULL, 171.60),
       ('125', 'Gurpreet Singh', 'Intern', NULL, '2008-05-15', NULL, NULL, 521.37),
       ('243', 'Marie Brahms', 'Engineer', '23', '1983-06-22', 40, NULL, 876.78),
       ('299', 'Mahala Burtwhistle', 'Analyst', '89', '1990-07-20', 95, NULL, 602.86),
       ('354', 'Bryna McKenny', 'Analyst', '153', '2001-10-11', 90, NULL, 693.63),
       ('421', 'Ulberto Gosz', 'Assistant', NULL, '2017-02-07', 10, NULL, 381.64),
       ('442', 'Netti Dellenbrok', 'Designer', '23', '2014-05-14', 30, NULL, 790.82),
       ('492', 'Salmon Gough', 'Assistant', '86', '2008-12-13', 90, 148.80, 998.32),
       ('504', 'Berny Bonas', 'Accountant', '778', '2008-05-24', 40, NULL, 779.34),
       ('516', 'Lowell Paule', 'Systems Administrator', '23', '2008-03-31', 30, 324.39, 969.66),
       ('518', 'Nicholas Comelini', 'Accountant', '188', '1997-03-20', 90, 546.04, 882.53);
```

```
INSERT INTO EMPLOYEE
VALUES ('528', 'Laurence Grosvener', 'Intern', '520', '2019-09-06', 90, 125.45, 244.37),
       ('611', 'Brandtr Wildbore', 'Engineer', '23', '2000-04-14', 30, NULL, 704.23),
       ('697', 'Druci Sixsmith', 'Technician', NULL, '2005-07-27', 10, 530.53, 387.43),
       ('718', 'Oneida Hicklingbottom', 'Intern', '89', '2006-03-25', 95, NULL, 789.82),
       ('732', 'Gannie Quantrell', 'Architect', '23', '2019-04-29', 30, NULL, 174.12),
       ('783', 'Fran Lening', 'Analyst', '23', '1981-05-27', 30, NULL, 471.17),
       ('79', 'Rahul Sahay', 'Intern', '23', '2004-08-14', 30, NULL, 2921.37),
       ('811', 'Shay Restieaux', 'Lawyer', '232', '1999-04-16', 45, 574.68, 500.22),
       ('823', 'Marcille Occleshaw', 'Lawyer', '232', '2003-12-27', 45, 453.52, 507.10),
       ('832', 'Desiri Allott', 'Intern', '89', '1997-11-04', 95, 239.84, 683.09),
       ('87', 'Blake Lewis', 'Intern', '86', '2008-05-02', 90, 125.45, 244.37),
       ('908', 'Gustav Waring', 'Engineer', '23', '2002-12-23', 30, NULL, 141.05),
       ('960', 'Virge Swancott', 'Developer', '23', '1984-01-23', 30, NULL, 292.48),
       ('969', 'Lee Prattin', 'Developer', '23', '1986-01-23', 30, NULL, 296.48),
       ('997', 'Mag Giacomo', 'Architect', '23', '2002-12-23', 30, NULL, 427.98),
       ('785', 'Chandal McBrier', 'Consultant', '23', '1998-02-10', 30, NULL, 697.68),
       ('659', 'Martha Annies', 'Developer', '23', '2008-11-03', 30, NULL, 200.80),
       ('932', 'Selestina Herries', 'Engineer', '23', '2013-05-23', 30, NULL, 239.27),
       ('837', 'Ashil Burney', 'Engineer', '23', '1987-03-29', 30, NULL, 586.13);
```

```
INSERT INTO EMPLOYEE
VALUES ('763', 'Rodrick Haighton', 'Analyst', '23', '2000-05-08', 30, NULL, 627.74),
       ('981', 'Nina Fisby', 'Systems Administrator', '23', '2017-07-26', 30, 324.39, 920.84),
       ('498', 'Skell Woolerton', 'Engineer', '23', '2004-04-04', 30, NULL, 506.44),
       ('809', 'Cornie Munby', 'Consultant', '23', '1986-06-07', 30, NULL, 688.37),
       ('915', 'Layton Murrie', 'Designer', '23', '2006-06-24', 30, NULL, 640.31),
       ('842', 'Rivy Buzek', 'Technician', NULL, '1982-05-25', 30, NULL, 295.31);
```

## QUERY - 31

### Description

Query to display the total number of supervisors without listing their names.

### Query

```
-- Q31
SELECT COUNT(DISTINCT SupervisorEno)
FROM EMPLOYEE;
```

### Results

```
mysql> SELECT COUNT(DISTINCT SupervisorEno)
      → FROM EMPLOYEE;
+-----+
| COUNT(DISTINCT SupervisorEno) |
+-----+
| 8 |
+-----+
1 row in set (0.00 sec)
```

## QUERY - 32

### Description

Query to display the Department Name, Location Name, No. of Employees and the average salary for all employees in that department.

### Query

```
-- Q32
SELECT Dname,
       Location,
       COUNT(*) AS NumberOfEmployees,
       ROUND(AVG(Salary), 2) AS AvgSalary
FROM EMPLOYEE
     NATURAL JOIN DEPARTMENT
GROUP BY EMPLOYEE.Dno;
```

### Results

```
mysql> SELECT Dname,
→         Location,
→         COUNT(*) AS NumberOfEmployees,
→         ROUND(AVG(Salary), 2) AS AvgSalary
→ FROM EMPLOYEE
→         NATURAL JOIN DEPARTMENT
→ GROUP BY EMPLOYEE.Dno;
```

| Dname          | Location  | NumberOfEmployees | AvgSalary |
|----------------|-----------|-------------------|-----------|
| Headquarters   | Mumbai    | 3                 | 2189.77   |
| IT             | Bengaluru | 21                | 717.06    |
| Administration | New Delhi | 5                 | 1190.15   |
| Legal          | Mumbai    | 3                 | 552.79    |
| Sales          | New Delhi | 8                 | 831.67    |
| Purchase       | Dallas    | 4                 | 764.03    |

6 rows in set (0.00 sec)

## QUERY - 33

### Description

Query to display Name and Hire Date for all employees in the same department as Blake.

### Query

```
-- Q33
SELECT Ename,
       Hire_date
FROM EMPLOYEE
WHERE Dno IN (
    SELECT Dno
    FROM EMPLOYEE
    WHERE Ename LIKE 'Blake%'
);
```

### Results

```
mysql> SELECT Ename,
→         Hire_date
→ FROM EMPLOYEE
→ WHERE Dno IN (
→     SELECT Dno
→     FROM EMPLOYEE
→     WHERE Ename LIKE 'Blake%'
→ );
```

| Ename              | Hire_date  |
|--------------------|------------|
| Donielle Tuite     | 2016-06-12 |
| Bryna McKenny      | 2001-10-11 |
| Salmon Gough       | 2008-12-13 |
| Nicholas Comelini  | 1997-03-20 |
| King Hethron       | 2015-10-13 |
| Laurence Grosvener | 2019-09-06 |
| Tania Chauhan      | 1983-08-14 |
| Blake Lewis        | 2008-05-02 |

8 rows in set (0.00 sec)



## QUERY -34

### Description

Query to display the Employee No. And Name for all employees who earn more than the average salary.

### Query

```
-- Q34
SELECT Ename,
       Salary
FROM EMPLOYEE
WHERE Salary > (
    SELECT AVG(Salary)
    FROM EMPLOYEE
);
```

### Results

```
mysql> SELECT Ename,
→         Salary
→ FROM EMPLOYEE
→ WHERE Salary > (
→         SELECT AVG(Salary)
→         FROM EMPLOYEE
→ );
```

| Ename             | Salary  |
|-------------------|---------|
| Sherline Harbord  | 926.42  |
| Sudipto Ghosh     | 2465.92 |
| Anmol Gupta       | 2865.92 |
| Salmon Gough      | 998.32  |
| Lowell Paule      | 969.66  |
| Nicholas Comelini | 882.53  |
| John Smith        | 5800.23 |
| Rahul Sahay       | 2921.37 |
| Tania Chauhan     | 2921.37 |
| Alicia Jones      | 980.35  |
| Nina Fisby        | 920.84  |

11 rows in set (0.00 sec)



## QUERY - 35

### Description

Query to display Employee Number and Name for all employees who work in a department with any employee whose name contains a 'T'.

### Query

```
-- Q35
SELECT Eno,
       Ename
FROM EMPLOYEE
WHERE Dno IN
      (
        SELECT Dno
        FROM EMPLOYEE
        WHERE Ename LIKE '%T%'
      );
```

### Results

```
mysql> SELECT Eno,
→         Ename
→ FROM EMPLOYEE
→ WHERE Dno IN
→       (
→         SELECT Dno
→         FROM EMPLOYEE
→         WHERE Ename LIKE '%T%'
→       )
→ LIMIT 10;
```

| Eno | Ename              |
|-----|--------------------|
| 146 | Donielle Tuite     |
| 354 | Bryna McKenny      |
| 492 | Salmon Gough       |
| 518 | Nicholas Comelini  |
| 520 | King Hethron       |
| 528 | Laurence Grosvener |
| 86  | Tania Chauhan      |
| 87  | Blake Lewis        |
| 23  | Sudipto Ghosh      |
| 442 | Netti Dellenbrok   |

10 rows in set (0.00 sec)

## QUERY - 36

### Description

Query to display the names and salaries of all employees who report to supervisor named 'King'.

### Query

```
-- Q36
SELECT Ename,
       Salary
FROM EMPLOYEE
WHERE SupervisorEno IN (
  SELECT Eno
  FROM EMPLOYEE
  WHERE Ename LIKE 'King%'
);
```

### Results

```
mysql> SELECT Ename,
→       Salary
→ FROM EMPLOYEE
→ WHERE SupervisorEno IN (
→       SELECT Eno
→       FROM EMPLOYEE
→       WHERE Ename LIKE 'King%'
→ );
```

| Ename              | Salary |
|--------------------|--------|
| Donielle Tuite     | 171.60 |
| Laurence Grosvener | 244.37 |

2 rows in set (0.00 sec)

## QUERY - 37

### Description

Query to display the department no, name and job for all employees in the Sales department.

### Query

```
-- Q37
SELECT Dno,
       Ename,
       Job_type
FROM (EMPLOYEE
      NATURAL JOIN DEPARTMENT)
WHERE Dname = 'Sales';
```

### Results

```
mysql> SELECT Dno,
→       Ename,
→       Job_type
→ FROM (EMPLOYEE
→       NATURAL JOIN DEPARTMENT)
→ WHERE Dname = 'Sales';
```

| Dno | Ename              | Job_type   |
|-----|--------------------|------------|
| 90  | Donielle Tuite     | Intern     |
| 90  | Bryna McKenny      | Analyst    |
| 90  | Salmon Gough       | Assistant  |
| 90  | Nicholas Comelini  | Accountant |
| 90  | King Hethron       | Manager    |
| 90  | Laurence Grosvener | Intern     |
| 90  | Tania Chauhan      | Manager    |
| 90  | Blake Lewis        | Intern     |

8 rows in set (0.00 sec)

## QUERY - 38

### Description

Display names of employees along with their department name who have more than 20 years' experience.

### Query

```
-- Q38
SELECT Ename,
       Dname
FROM (EMPLOYEE
      NATURAL JOIN DEPARTMENT)
WHERE TIMESTAMPDIFF
      (
        YEAR,
        Hire_date,
        NOW()
      ) > 20;
```

### Results

```
mysql> SELECT Ename,
→          Dname
→ FROM (EMPLOYEE
→       NATURAL JOIN DEPARTMENT)
→ WHERE TIMESTAMPDIFF
→       (
→         YEAR,
→         Hire_date,
→         NOW()
→       ) > 20;
```

| Ename              | Dname          |
|--------------------|----------------|
| John Smith         | Headquarters   |
| Sudipto Ghosh      | IT             |
| Fran Lening        | IT             |
| Chandal McBrier    | IT             |
| Cornie Munby       | IT             |
| Ashil Burney       | IT             |
| Rivy Buzek         | IT             |
| Virge Swancott     | IT             |
| Lee Prattin        | IT             |
| Sherline Harbord   | Administration |
| Marie Brahms       | Administration |
| Anmol Gupta        | Administration |
| Isa Jaques         | Legal          |
| Shay Restieaux     | Legal          |
| Nicholas Comelini  | Sales          |
| Tania Chauhan      | Sales          |
| Mahala Burtwhistle | Purchase       |
| Desiri Allott      | Purchase       |
| Alicia Jones       | Purchase       |

19 rows in set (0.00 sec)

## QUERY - 39

### Description

Display total number of departments at each location.

### Query

```
-- Q39
SELECT Location,
       COUNT(*)
FROM DEPARTMENT
GROUP BY Location;
```

### Results

```
mysql> SELECT Location,
→       COUNT(*)
→ FROM DEPARTMENT
→ GROUP BY Location;
```

| Location  | COUNT(*) |
|-----------|----------|
| Mumbai    | 2        |
| Dallas    | 2        |
| Bengaluru | 1        |
| New Delhi | 3        |

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

## QUERY - 40

### Description

Find the department name in which at least 20 employees work in.

### Query

```
-- Q40
SELECT Dname
FROM (EMPLOYEE
      NATURAL JOIN DEPARTMENT)
GROUP BY Dno
HAVING COUNT(*) > 20;
```

### Results

```
mysql> SELECT Dname
       → FROM (EMPLOYEE
       →        NATURAL JOIN DEPARTMENT)
       → GROUP BY Dno
       → HAVING COUNT(*) > 20;
```

```
+-----+
| Dname |
+-----+
| IT    |
+-----+
```

```
1 row in set (0.00 sec)
```

# QUERY - 41

## Description

Query to find the employees' name who is not supervisor and name of supervisor supervising more than 5 employees.

## Query

```
-- Q41
(
  SELECT Ename
  FROM EMPLOYEE
  WHERE Eno NOT IN (
    SELECT DISTINCT SupervisorEno
    FROM EMPLOYEE
    WHERE SupervisorEno IS NOT NULL
  )
)
UNION
(
  SELECT Ename
  FROM EMPLOYEE
  WHERE Eno IN (
    SELECT SupervisorEno
    FROM EMPLOYEE
    WHERE SupervisorEno IS NOT NULL
    GROUP BY SupervisorEno
    HAVING COUNT(*) > 5
  )
);
```

## Results

```
mysql> (
→      SELECT Ename
→      FROM EMPLOYEE
→      WHERE Eno NOT IN (
→          SELECT DISTINCT SupervisorEno
→          FROM EMPLOYEE
→          WHERE SupervisorEno IS NOT NULL
→      )
→ )
→ UNION
→ (
→      SELECT Ename
→      FROM EMPLOYEE
→      WHERE Eno IN (
→          SELECT SupervisorEno
→          FROM EMPLOYEE
→          WHERE SupervisorEno IS NOT NULL
→          GROUP BY SupervisorEno
→          HAVING COUNT(*) > 5
→      )
→ ) LIMIT 20;
```



| Ename                 |
|-----------------------|
| Gurpreet Singh        |
| Donielle Tuite        |
| Marie Brahms          |
| Anmol Gupta           |
| Mahala Burtwhistle    |
| Bryna McKenny         |
| Ulberto Gosz          |
| Netti Dellenbrok      |
| Salmon Gough          |
| Skell Woolerton       |
| Berny Bonas           |
| Lowell Paule          |
| Nicholas Comelini     |
| Laurence Grosvener    |
| Brandtr Wildbore      |
| Martha Annies         |
| Druci Sixsmith        |
| Oneida Hicklingbottom |
| Gannie Quantrell      |
| Rodrick Haighton      |

20 rows in set (0.00 sec)

## QUERY - 42

### Description

Query to display the job type with maximum and minimum employees.

### Query

```
-- Q42
WITH JOBCOUNT AS (
    SELECT COUNT(*) AS ECount
    FROM EMPLOYEE
    GROUP BY Job_type
)
SELECT Job_type,
       COUNT(*)
FROM EMPLOYEE
GROUP BY Job_type
HAVING COUNT(*) IN (
    (
        SELECT MAX(ECount)
        FROM JOBCOUNT
    )
    UNION
    (
        SELECT MIN(ECount)
        FROM JOBCOUNT
    )
);
```

### Results

```
mysql> WITH JOBCOUNT AS (
  →     SELECT COUNT(*) AS ECount
  →     FROM EMPLOYEE
  →     GROUP BY Job_type
  → )
  → SELECT Job_type,
  →        COUNT(*)
  → FROM EMPLOYEE
  → GROUP BY Job_type
  → HAVING COUNT(*) IN (
  →   (
  →     SELECT MAX(ECount)
  →     FROM JOBCOUNT
  →   )
  →   UNION
  →   (
  →     SELECT MIN(ECount)
  →     FROM JOBCOUNT
  →   )
  → );
```

| Job_type          | COUNT(*) |
|-------------------|----------|
| Manager           | 8        |
| Managing Director | 1        |

2 rows in set (0.00 sec)