Lab Practical Question Solution

Govind Kumar Yadav

PAS078BEI015

Database Management System(DBMS)

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) Ans ->

mysql> create database Ass1;

Query OK, 1 row affected (0.02 sec)

mysql> show databases;

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

| Database |

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

| ass1 |

| db |

| information\_schema |

| mysql |

| performance\_schema |

| sys |

| wrc |

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

7 rows in set (0.00 sec)

mysql> use ass1;

Database changed

mysql> CREATE TABLE employee (

-> emp\_id INT PRIMARY KEY,

-> first\_name VARCHAR(50),

-> last\_name VARCHAR(50),

-> phone\_no VARCHAR(15),

-> 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 (0.06 sec)

* 1. Insert five records into the table employees

Ans ->

mysql> INSERT INTO employee(emp\_id, first\_name,last\_name,phone\_no, hire\_date,job\_id,emp\_salary,commission\_pct, manager\_id,department\_id) VALUES

-> (1, 'Govind','Yadav','222-111','2024-03-09','AppDev',50000,10.0,1515,1),

-> (2, 'Dhiraj','Chaurasiya','222-101','2024-05-09','WebDev',60000,11.0,1516,2),

-> (3, 'Utkarsha','Gupta','222-121','2024-05-18','Design',40000,15.0,1011,3),

-> (4, 'Prabin','Thapa','222-333','2024-05-15','ML',90000,16.0,1033,4),

-> (5, 'Anil','Paudel','222-991','2024-01-09','DBA',50000,12.0,5586,5);

Query OK, 5 rows affected (0.02 sec)

Records: 5 Duplicates: 0 Warnings: 0

* 1. Display the table Employees

Ans ->

mysql> select \* from employee;

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

| emp\_id | first\_name | last\_name | phone\_no | hire\_date | job\_id | emp\_salary | commission\_pct | manager\_id | department\_id |

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

| 1 | Govind | Yadav | 222-111 | 2024-03-09 | AppDev | 50000.00 | 10.00 | 1515 | 1 |

| 2 | Dhiraj | Chaurasiya | 222-101 | 2024-05-09 | WebDev | 60000.00 | 11.00 | 1516 | 2 |

| 3 | Utkarsha | Gupta | 222-121 | 2024-05-18 | Design | 40000.00 | 15.00 | 1011 | 3 |

| 4 | Prabin | Thapa | 222-333 | 2024-05-15 | ML | 90000.00 | 16.00 | 1033 | 4 |

| 5 | Anil | Paudel | 222-991 | 2024-01-09 | DBA | 50000.00 | 12.00 | 5586 | 5 |

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

5 rows in set (0.00 sec)

* 1. Find out the employee id, names, salaries of all the employees

Ans ->

SELECT emp\_id, first\_name,last\_name,emp\_salary FROM employee;

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

| emp\_id | first\_name | last\_name | emp\_salary |

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

| 1 | Govind | Yadav | 50000.00 |

| 2 | Dhiraj | Chaurasiya | 60000.00 |

| 3 | Utkarsha | Gupta | 40000.00 |

| 4 | Prabin | Thapa | 90000.00 |

| 5 | Anil | Paudel | 50000.00 |

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

5 rows in set (0.00 sec)

* 1. Find the names of the employees who have a salary greater than or equal to 4800

Ans ->

mysql> select first\_name,last\_name from employee where emp\_salary >= 48000;

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

| first\_name | last\_name |

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

| Govind | Yadav |

| Dhiraj | Chaurasiya |

| Prabin | Thapa |

| Anil | Paudel |

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

4 rows in set (0.00 sec)

* 1. List out the employees whose last name is ‘Yadav’

Ans ->

mysql> select \* from employee where last\_name = 'Yadav';

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

| emp\_id | first\_name | last\_name | phone\_no | hire\_date | job\_id | emp\_salary | commission\_pct | manager\_id | department\_id |

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

| 1 | Govind | Yadav | 222-111 | 2024-03-09 | AppDev | 50000.00 | 10.00 | 1515 | 1 |

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

1 row in set (0.00 sec)

* 1. Find the names of the employees who works in departments 2, 3 and 4

Ans ->

mysql> select first\_name, last\_name, emp\_salary from employee where departme

nt\_id in (2,3,4);

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

| first\_name | last\_name | emp\_salary |

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

| Dhiraj | Chaurasiya | 60000.00 |

| Utkarsha | Gupta | 40000.00 |

| Prabin | Thapa | 90000.00 |

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

3 rows in set (0.00 sec)

* 1. Display the unique Manager\_Id from employees table

Ans ->

mysql> select distinct manager\_id from employee;

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

| manager\_id |

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

| 1515 |

| 1516 |

| 1011 |

| 1033 |

| 5586 |

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

5 rows in set (0.01 sec)

1. (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

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)

);

* 1. insert five records into the Client\_Master

INSERT INTO Client\_master (ClientNO, Name, Address, City, State, bal\_due)

VALUES

('C101', 'Hari Sharma', '12/3, Thapathali', 'Kathmandu', 'Bagmati', 4500.00),

('C102', 'Sita Rathi', '45, New Road', 'Pokhara', 'Gandaki', 6000.00),

('C103', 'Rajesh Bhattarai', '78, Bharatpur', 'Chitwan', 'Bagmati', 3000.00),

('C104', 'Gita Adhikari', '56, Biratnagar', 'Morang', 'Koshi', 7500.00),

('C105', 'Amit Koirala', '89, Butwal', 'Lumbini', 'Lumbini', 4200.00);

* 1. Display Client Master Table

SELECT \* FROM Client\_master;

* 1. Find the name of Clients whose balance\_due >5000

SELECT Name

FROM Client\_master

WHERE bal\_due > 5000;

* 1. Change the bal\_due of ClientNO “C123” to Rs. 5100

UPDATE Client\_master

SET bal\_due = 5100.00

WHERE ClientNO = 'C123';

* 1. Change the name of Client\_master to Client12

ALTER TABLE Client\_master

RENAME TO Client12;

* 1. Display the bal\_due heading as “BALANCE” Client master table

SELECT ClientNO, Name, Address, City, State, bal\_due AS BALANCE

FROM Client12;

1. 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)

CREATE TABLE Teacher (

Id INT PRIMARY KEY,

Name VARCHAR(100),

DeptNo INT,

Date\_of\_joining DATE,

DeptName VARCHAR(50),

Location VARCHAR(100),

Salary DECIMAL(10, 2)

);

* 1. Insert five records

INSERT INTO Teacher (Id, Name, DeptNo, Date\_of\_joining, DeptName, Location, Salary)

VALUES

(1, 'Suman Sharma', 101, '2020-06-15', 'Mathematics', 'Kathmandu', 50000.00),

(2, 'Maya Rai', 102, '2019-08-22', 'Commerce', 'Pokhara', 45000.00),

(3, 'Anil Kumar', 103, '2018-11-10', 'Science', 'Biratnagar', 55000.00),

(4, 'Priya Joshi', 104, '2021-01-12', 'Mathematics', 'Lalitpur', 48000.00),

(5, 'Rajesh Thapa', 105, '2017-03-05', 'Commerce', 'Bharatpur', 47000.00);

* 1. Give Increment of 25% salary for Mathematics Department.

UPDATE Teacher

SET Salary = Salary \* 1.25

WHERE DeptName = 'Mathematics';

* 1. Perform Rollback command

ROLLBACK;

* 1. Give Increment of 15% salary for Commerce Department

UPDATE Teacher

SET Salary = Salary \* 1.15

WHERE DeptName = 'Commerce';

* 1. Perform commit command

COMMIT;

1. (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)

CREATE TABLE Sales (

Sales\_No INT PRIMARY KEY,

Sales\_Name VARCHAR(100),

Branch VARCHAR(50),

Sales\_Amount DECIMAL(10, 2),

DOB DATE

);

* 1. Insert five records

INSERT INTO Sales (Sales\_No, Sales\_Name, Branch, Sales\_Amount, DOB)

VALUES

(1, 'Raj Kumar', 'Kathmandu', 120000.00, '1985-12-10'),

(2, 'Sita Rathi', 'Pokhara', 80000.00, '1990-12-22'),

(3, 'Anil Sharma', 'Kathmandu', 150000.00, '1982-05-15'),

(4, 'Maya Devi', 'Biratnagar', 90000.00, '1988-12-30'),

(5, 'Hari Prasad', 'Pokhara', 110000.00, '1995-03-08');

* 1. Calculate total salesamount in each branch

SELECT Branch, SUM(Sales\_Amount) AS Total\_Sales\_Amount

FROM Sales

GROUP BY Branch;

* 1. Calculate average salesamount in each branch

SELECT Branch, AVG(Sales\_Amount) AS Average\_Sales\_Amount

FROM Sales

GROUP BY Branch;

* 1. Display all the salesmen, DOB who are born in the month of December as day in character format i.e. 21-Dec-09

SELECT Sales\_Name, TO\_CHAR(DOB, 'DD-Mon-YY') AS DOB

FROM Sales

ORDER BY EXTRACT(MONTH FROM DOB);

* 1. Display the name and DOB of salesman in alphabetical order of the month.

SELECT Sales\_Name, TO\_CHAR(DOB, 'DD-Mon-YY') AS DOB

FROM Sales

ORDER BY EXTRACT(MONTH FROM DOB);

1. 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)

CREATE TABLE Emp (

Emp\_No INT PRIMARY KEY,

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)

);

* 1. Insert Five Records and calculate GrossPay and NetPay.

INSERT INTO Emp (Emp\_No, Emp\_Name, Designation, Basic, DA, HRA, PF, GrossPay, NetPay)

VALUES

(1, 'Deepak Sharma', 'Manager', 30000.00, 5000.00, 8000.00, 2500.00,

30000.00 + 5000.00 + 8000.00,

(30000.00 + 5000.00 + 8000.00) - 2500.00),

(2, 'Sita Rathi', 'Assistant', 25000.00, 4000.00, 6000.00, 2000.00,

25000.00 + 4000.00 + 6000.00,

(25000.00 + 4000.00 + 6000.00) - 2000.00),

(3, 'Rajesh Thapa', 'Clerk', 20000.00, 3000.00, 4000.00, 1500.00,

20000.00 + 3000.00 + 4000.00,

(20000.00 + 3000.00 + 4000.00) - 1500.00),

(4, 'Anita Koirala', 'Manager', 32000.00, 5500.00, 8500.00, 2700.00,

32000.00 + 5500.00 + 8500.00,

(32000.00 + 5500.00 + 8500.00) - 2700.00),

(5, 'Ravi Joshi', 'Assistant', 27000.00, 4200.00, 6200.00, 2200.00,

27000.00 + 4200.00 + 6200.00,

(27000.00 + 4200.00 + 6200.00) - 2200.00);

* 1. Adding column to table and Updating Attributes DA

ALTER TABLE Emp

ADD DA DECIMAL(10, 2);

UPDATE Emp

SET DA = (calculated value);

* 1. Adding column to table and Updating Attributes HRA

ALTER TABLE Emp

ADD HRA DECIMAL(10, 2);

UPDATE Emp

SET HRA = (calculated value);

* 1. Adding column to table and Updating Attributes PF

ALTER TABLE Emp ADD PF DECIMAL(10, 2);

UPDATE Emp SET PF = (calculated value);

* 1. Adding column to table and Updating Attributes Gross Pay

ALTER TABLE Emp

ADD GrossPay DECIMAL(10, 2);

UPDATE Emp

SET GrossPay = Basic + DA + HRA;

* 1. Adding column to table and Updating Attributes Net Pay

ALTER TABLE Emp

ADD NetPay DECIMAL(10, 2);

UPDATE Emp

SET NetPay = GrossPay - PF;

* 1. Display the employee table

SELECT \* FROM Emp;

* 1. Display the employees whose Basic is lowest in each department.

SELECT \*

FROM Emp e1

WHERE Basic = (

SELECT MIN(Basic)

FROM Emp e2

WHERE e1.Designation = e2.Designation

);

* 1. If NetPay is less than Rs. 10,000 add Rs. 1200 as special allowance

UPDATE Emp

SET NetPay = NetPay + 1200

WHERE NetPay < 10000;

* 1. Display the employees whose GrossPay lies between 10,000 & 20,000

SELECT \*

FROM Emp

WHERE GrossPay BETWEEN 10000 AND 20000;

* 1. Display all the employees who earn maximum salary.

SELECT \*

FROM Emp

WHERE GrossPay = (

SELECT MAX(GrossPay)

FROM Emp

);

1. 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)

CREATE TABLE Dept (

deptno INT PRIMARY KEY,

dname VARCHAR(100),

loc VARCHAR(100)

);

* 1. Create Dept table: Emp (empno, ename, job, mgr, hiredate, sal, comm, deptno)

CREATE TABLE Emp (

empno INT PRIMARY KEY,

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 Emp(empno)

);

* 1. Insert data int Dept and Emp tables

INSERT INTO Dept (deptno, dname, loc)

VALUES

(10, 'Sales', 'Kathmandu'),

(20, 'HR', 'Pokhara'),

(30, 'IT', 'Lalitpur'),

(40, 'Finance', 'Biratnagar');

INSERT INTO Emp (empno, ename, job, mgr, hiredate, sal, comm, deptno)

VALUES

(1, 'Ramesh Gupta', 'Manager', NULL, '1994-07-15', 80000.00, 5000.00, 10),

(2, 'Sita Sharma', 'Clerk', 1, '2000-01-21', 50000.00, 2000.00, 10),

(3, 'Rajesh Kumar', 'Analyst', 1, '1990-03-13', 70000.00, 3000.00, 30),

(4, 'Anita Thapa', 'Manager', NULL, '2005-05-23', 85000.00, 4000.00, 20),

(5, 'Pradeep Joshi', 'Clerk', 4, '2010-06-30', 55000.00, 1500.00, 20);

* 1. Update the employee salary by 15%, whose experience is greater than 30 years

UPDATE Emp

SET sal = sal \* 1.15

WHERE hiredate <= DATE\_SUB(CURDATE(), INTERVAL 30 YEAR);

* 1. Delete the employees, who completed 30 years of service.

DELETE FROM Emp

WHERE hiredate <= DATE\_SUB(CURDATE(), INTERVAL 30 YEAR);

* 1. Display the manager who is having maximum number of employees working under him?

SELECT mgr, COUNT(\*) AS num\_employees

FROM Emp

GROUP BY mgr

ORDER BY num\_employees DESC

LIMIT 1;

* 1. Create a view, which contain employee names and their manager

CREATE VIEW Employee\_Manager\_View AS

SELECT e1.ename AS Employee\_Name, e2.ename AS Manager\_Name

FROM Emp e1

LEFT JOIN Emp e2 ON e1.mgr = e2.empno;

1. Using Employee Database above perform the following queries
2. Determine the names of employee, who earn more than their managers.

SELECT e1.ename AS Employee\_Name, e2.ename AS Manager\_Name

FROM Emp e1

JOIN Emp e2 ON e1.mgr = e2.empno

WHERE e1.sal > e2.sal;

1. Determine the names of employees, who take highest salary in their departments.

SELECT e1.ename, e1.deptno

FROM Emp e1

JOIN (

SELECT deptno, MAX(sal) AS max\_salary

FROM Emp

GROUP BY deptno

) e2 ON e1.deptno = e2.deptno AND e1.sal = e2.max\_salary;

1. Determine the employees, who are located at the same place.

SELECT e1.ename AS Employee1, e2.ename AS Employee2, d.loc

FROM Emp e1

JOIN Emp e2 ON e1.deptno = e2.deptno AND e1.empno <> e2.empno

JOIN Dept d ON e1.deptno = d.deptno

WHERE d.loc = (SELECT loc FROM Dept WHERE deptno = e1.deptno);

1. Determine the employees, whose total salary is like the minimum Salary of any department.

SELECT e1.ename, e1.sal, e1.comm, e1.sal + COALESCE(e1.comm, 0) AS total\_salary

FROM Emp e1

WHERE (e1.sal + COALESCE(e1.comm, 0)) = (

SELECT MIN(sal)

FROM Emp e2

WHERE e1.deptno = e2.deptno

);

1. Determine the department which does not contain any employees

SELECT d.deptno, d.dname

FROM Dept d

LEFT JOIN Emp e ON d.deptno = e.deptno

WHERE e.deptno IS NULL;

1. Using the tables “DEPARTMENTS” and “EMPLOYEES” above perform the following queries
2. Display the employee details, departments that the departments are same in both the emp and dept.

SELECT e.\*, d.dname, d.loc

FROM EMPLOYEES e

JOIN DEPARTMENTS d ON e.deptno = d.deptno;

1. Display the employee name and Department name by implementing a left outer join.

SELECT e.ename, d.dname

FROM EMPLOYEES e

LEFT JOIN DEPARTMENTS d ON e.deptno = d.deptno;

1. Display the employee name and Department name by implementing a right outer join.

SELECT e.ename, d.dname

FROM EMPLOYEES e

RIGHT JOIN DEPARTMENTS d ON e.deptno = d.deptno;

1. Display the details of those who draw the salary greater than the average salary.

SELECT \*

FROM EMPLOYEES

WHERE sal > (

SELECT AVG(sal)

FROM EMPLOYEES

);