SQL Queries and Solutions

SQL Problem statements for practical.pdf
PDF

give me ssolution for all the above problem with the code to solve it

Here are SQL solutions for each problem based on the details provided in your PDF file:

Problem 1: Employee Table Queries

1. Display all fields of the employee table:

```sql
SELECT * FROM Employee;
```

2. Retrieve employee number and their salary:

```sql
SELECT EMPNO, SALARY FROM Employee;
```

3. Retrieve average salary of all employees:

```sql
SELECT AVG(SALARY) AS AverageSalary FROM Employee;
```

4. Retrieve number of employees:

```sql
SELECT COUNT(*) AS EmployeeCount FROM Employee;
```

5. Retrieve distinct number of employees:

```sql
SELECT COUNT(DISTINCT EMP_NAME) AS DistinctEmployees FROM Employee;
```

6. Retrieve total salary of employees grouped by employee name and count similar names:

```sql
SELECT EMP_NAME, SUM(SALARY) AS TotalSalary, COUNT(*) AS NameCount
FROM Employee
GROUP BY EMP_NAME;
```

7. Retrieve total salary of employees with a salary greater than 120,000:

```sql
SELECT SUM(SALARY) AS TotalHighSalary FROM Employee WHERE SALARY > 120000;
```

8. Display names of employees in descending order:

```sql
SELECT EMP_NAME FROM Employee ORDER BY EMP_NAME DESC;
```

9. Display details of employees named "AMIT" with a salary greater than 50,000:

```sql
SELECT * FROM Employee WHERE EMP_NAME = 'AMIT' AND SALARY > 50000;
```

Problem 2: Company Database and Employee-Department Operations

1. Create the Company database:

```sql
CREATE DATABASE Company;
```

2. View all databases:

```sql
SHOW DATABASES;
```

3. View all tables in a database:

```sql
USE Company;
SHOW TABLES;
```

4. Create tables:

```sql
CREATE TABLE Employee (
    Name VARCHAR(50),
    SSN CHAR(9),
    Address VARCHAR(100),
    Salary DECIMAL(10, 2),
    Department VARCHAR(50)
);

CREATE TABLE Department (
    dname VARCHAR(50),
    DID INT PRIMARY KEY
);
```

5. Insert, update, and delete records:

```sql
-- Inserting records
INSERT INTO Employee (Name, SSN, Address, Salary, Department) VALUES ('John Doe', '123456789', '123 Elm St', 50000, 'HR');

-- Updating records
UPDATE Employee SET Salary = Salary * 1.1 WHERE Department = 'HR';

-- Deleting records
DELETE FROM Employee WHERE SSN = '123456789';
```

6. Retrieve the name of employees and their department name (using JOIN):

```sql
SELECT Employee.Name, Department.dname
FROM Employee
JOIN Department ON Employee.Department = Department.DID;
```

7. ER diagram for the Company database scenario:
    - Employee (Name, SSN, Address, Salary, Department)
    - Department (dname, DID)

(Note: This is a text-based response; you would need a diagram tool to create an ER diagram.)

This format provides example SQL commands to address each problem statement as outlined in the document. If you need solutions for more problems from the file or specific explanations, please let me know!

give these for all problems

Here are solutions for all the remaining problems as per your PDF file.

Problem 3: Company Database with Additional Requirements

1. Give a 10% raise to employees in the "Research" department:

```sql
UPDATE Employee SET Salary = Salary * 1.1 WHERE Department = 'Research';
```

2. Sum, maximum, minimum, and average salary for "Accounts" department:

```sql
SELECT SUM(Salary) AS TotalSalary, MAX(Salary) AS MaxSalary,
       MIN(Salary) AS MinSalary, AVG(Salary) AS AvgSalary
FROM Employee
WHERE Department = 'Accounts';
```

3. Retrieve employees controlled by department number 5:

```sql
SELECT Name FROM Employee WHERE EXISTS (
    SELECT 1 FROM Department WHERE DID = 5 AND Employee.Department = Department.DID
);
```

4. Retrieve department names and employee count for departments with at least 2 employees:

```sql
SELECT Department, COUNT(*) AS EmployeeCount
FROM Employee
GROUP BY Department
HAVING COUNT(*) >= 2;
```

5. Retrieve employees born in the 1990s:

```sql
SELECT Name FROM Employee WHERE YEAR(DOB) BETWEEN 1990 AND 1999;
```

6. Retrieve employees and their department names (using JOIN):

```
sql


  SELECT Employee.Name, Department.dname
  FROM Employee
  JOIN Department ON Employee.Department = Department.DID;
```

Problem 4: Views on the Employee Table

1. Create a view with and without `` `CHECK OPTION` ``:

```
sql


  CREATE VIEW Employee_View AS SELECT * FROM Employee;
  CREATE VIEW Employee_Salary_View AS
      SELECT * FROM Employee WHERE Salary > 30000
      WITH CHECK OPTION;
```

2. Selecting from a view:

```
sql


  SELECT * FROM Employee_View;
```

3. Dropping views:

```
sql


  DROP VIEW Employee_View;
```

Problem 5: Shoppe and Accessories Table Operations

1. Display name and price of all accessories in descending price order:

```
sql


  SELECT Name, Price FROM Accessories ORDER BY Price DESC;
```

2. Display IDs and shop names at "Nehru Place":

```
sql


  SELECT Id, Sname FROM Shoppe WHERE Location = 'Nehru Place';
```

3. Display minimum and maximum price for each accessory:

```sql
SELECT Name, MIN(Price) AS MinPrice, MAX(Price) AS MaxPrice FROM Accessories GROUP BY
Name;
```

4. Display accessory name, price, and shop name for items priced 5000 or more:

```sql
SELECT A.Name, A.Price, S.Sname FROM Accessories A
JOIN Shoppe S ON A.ShopId = S.Id WHERE A.Price >= 5000;
```

5. Display details of accessories containing "Board" in the name:

```sql
SELECT * FROM Accessories WHERE Name LIKE '%Board%';
```

Problem 6: Train and Passenger Queries

1. Display details of trains starting from "New Delhi":

```sql
SELECT * FROM Trains WHERE Start = 'New Delhi';
```

2. Display PNR, name, gender, and age for passengers under 50:

```sql
SELECT PNR, Pname, Gender, Age FROM Passengers WHERE Age < 50;
```

3. Display count of male and female passengers:

```sql
SELECT Gender, COUNT(*) AS PassengerCount FROM Passengers GROUP BY Gender;
```

4. Display passengers on train number 12015:

```sql
SELECT * FROM Passengers WHERE TNO = 12015;
```

Problem 7: Employee and Salary Grade Operations

1. Display employees in descending order of their DOJ:

```
sql

  SELECT * FROM Employee ORDER BY DOJ DESC;
```

2. Display names and designations for employees with SGrade "S02" or "S03":

```
sql

  SELECT Name, Desig FROM Employee WHERE SGrade IN ('S02', 'S03');
```

3. Display details for employees who joined in 2009:

```
sql

  SELECT Name, Desig, SGrade FROM Employee WHERE YEAR(DOJ) = 2009;
```

4. Display all salary grades with annual salary calculated:

```
sql

  SELECT SGrade, Salary * 12 AS Annual_Salary FROM SalGrade;
```

5. Count employees per salary grade:

```
sql

  SELECT SGrade, COUNT(*) AS EmployeeCount FROM Employee GROUP BY SGrade;
```

6. Display name, designation, salary, and HRA for employees with salary below 50000:

```
sql

  SELECT E.Name, E.Desig, S.Salary, S.HRA
  FROM Employee E
  JOIN SalGrade S ON E.SGrade = S.SGrade
  WHERE S.Salary < 50000;
```

Problem 8: Transactions and Account Queries

1. Display details of all "Withdraw" transactions:

```sql
SELECT * FROM Transact WHERE Type = 'Withdraw';
```

2. Display account number and amount of transactions done in May 2017:

```sql
SELECT ANO, Amount FROM Transact WHERE MONTH(DOT) = 5 AND YEAR(DOT) = 2017;
```

3. Display the first transaction date for account number 102:

```sql
SELECT MIN(DOT) AS FirstTransaction FROM Transact WHERE ANO = 102;
```

4. Display account number, name, amount, and date of transactions under 3000:

```sql
SELECT A.ANO, A.AName, T.Amount, T.DOT
FROM Account A
JOIN Transact T ON A.ANO = T.ANO
WHERE T.Amount <= 3000;
```

Problem 9: Employee Table Operations with Salary and Commission

1. Display names of employees starting with 'V' in ascending salary order:

```sql
SELECT EName FROM Employee WHERE EName LIKE 'V%' ORDER BY Salary ASC;
```

2. Display details of sales department employees with salary > 20000:

```sql
SELECT * FROM Employee WHERE Dept = 'SALES' AND Salary > 20000;
```

3. Count distinct departments:

```sql
SELECT COUNT(DISTINCT Dept) FROM Employee;
```

4. Change salary of "NITIN" to 20000:

```sql
UPDATE Employee SET Salary = 20000 WHERE EName = 'NITIN';
```

This should cover each problem from your file. Please let me know if you need explanations for specific queries or further assistance with SQL concepts!

give for the rest rpoblem

Here are the SQL solutions for the remaining problems as per your PDF.

Problem 10: Library Database with Author, Book, Adaptation, and Book_Review

1. Show the name of each author along with the title and publication year of their book:

```sql
SELECT a.name AS AuthorName, b.title AS BookTitle, b.publish_year AS PublicationYear
FROM author a
JOIN book b ON a.id = b.author_id;
```

2. Show the name of each author along with titles of books published after 2005:

```sql
SELECT a.name AS AuthorName, b.title AS BookTitle, b.publish_year AS PublicationYear
FROM author a
JOIN book b ON a.id = b.author_id
WHERE b.publish_year > 2005;
```

3. For each book, show its title, adaptation title, adaptation year, and publication year:

```sql
SELECT b.title AS BookTitle, ad.title AS AdaptationTitle, ad.release_year AS Adaptati
onYear, b.publish_year AS PublicationYear
FROM book b
LEFT JOIN adaptation ad ON b.id = ad.book_id;
```

4. Show book titles and their adaptations' release dates, even if a book has no adaptation:

```sql
SELECT b.title AS BookTitle, COALESCE(ad.title, 'No Adaptation') AS AdaptationTitle,
ad.release_year AS ReleaseYear
FROM book b
LEFT JOIN adaptation ad ON b.id = ad.book_id;
```

Problem 11: Employee Table Creation and Alteration Operations

1. Add a `commission` column to the Employee table:

```sql
ALTER TABLE Employee ADD commission DECIMAL(10, 2);
```

2. Insert five records into the Employee table:

```sql
INSERT INTO Employee (EMPNO, ENAME, JOB, MGR, SAL) VALUES
(101, 'Alice', 'Manager', 100, 75000),
(102, 'Bob', 'Analyst', 101, 50000),
(103, 'Charlie', 'Clerk', 102, 30000),
(104, 'David', 'Developer', 101, 60000),
(105, 'Eve', 'Executive', 100, 80000);
```

3. Update job details (for example, promote "Clerk" to "Senior Clerk"):

```sql
UPDATE Employee SET JOB = 'Senior Clerk' WHERE JOB = 'Clerk';
```

4. Rename the `MGR` column to `Manager_ID`:

```sql
ALTER TABLE Employee RENAME COLUMN MGR TO Manager_ID;
```

5. Delete the employee with `Empno` 105:

```sql
DELETE FROM Employee WHERE EMPNO = 105;
```

Problem 12: Employee Table with Basic Operations

1. Create Employee table with columns `E_id`, `E_name`, `Age`, `Salary`:

```
sql

  CREATE TABLE Employee (
      E_id INT PRIMARY KEY,
      E_name VARCHAR(50) NOT NULL,
      Age INT,
      Salary DECIMAL(10, 2)
  );
```

2. Count the number of employees by name:

```
sql

  SELECT COUNT(E_name) AS EmployeeNameCount FROM Employee;
```

3. Find the maximum age:

```
sql

  SELECT MAX(Age) AS MaxAge FROM Employee;
```

4. Find the minimum age:

```
sql

  SELECT MIN(Age) AS MinAge FROM Employee;
```

5. Display salaries in ascending order:

```
sql

  SELECT Salary FROM Employee ORDER BY Salary ASC;
```

6. Grouped salaries of employees:

```
sql

  SELECT Salary, COUNT(*) AS EmployeeCount FROM Employee GROUP BY Salary;
```

Problem 13: EMP Table with Constraints

1. Create EMP table with structure:

```sql
CREATE TABLE EMP (
    EMPNO NUMBER(6) CHECK (EMPNO > 100),
    ENAME VARCHAR2(20) NOT NULL,
    JOB VARCHAR2(10) NOT NULL,
    DEPTNO NUMBER(3) UNIQUE,
    SAL NUMBER(7,2),
    PRIMARY KEY (EMPNO)
);
```

2. Add primary key constraint on `EMPNO`:
    - Already specified in table creation above.

Problem 14: Sailors, Boats, and Reserves Table Queries

1. Find information on sailors who reserved boat number 101:

```sql
SELECT * FROM Sailors s
JOIN Reserves r ON s.sid = r.sid
WHERE r.bid = 101;
```

2. Find the name of the boat reserved by "Bob":

```sql
SELECT b.bname FROM Boats b
JOIN Reserves r ON b.bid = r.bid
JOIN Sailors s ON r.sid = s.sid
WHERE s.sname = 'Bob';
```

3. Find names of sailors who reserved a red boat, ordered by age:

```sql
SELECT s.sname FROM Sailors s
JOIN Reserves r ON s.sid = r.sid
JOIN Boats b ON r.bid = b.bid
WHERE b.color = 'red'
ORDER BY s.age;
```

4. Find names of sailors who reserved at least one boat:

```sql
SELECT DISTINCT s.sname FROM Sailors s
JOIN Reserves r ON s.sid = r.sid;
```

Problem 15: Employee Queries with Job Titles and Salaries

1. List E_no, E_name, and Salary of all employees working as "Manager":

```sql
SELECT Emp_no, E_name, Salary FROM Employee WHERE Designation = 'Manager';
```

2. Display all details of employees with a salary higher than any IT Professional:

```sql
SELECT * FROM Employee WHERE Salary > (SELECT MAX(Salary) FROM Employee WHERE Job_id
= 'IT_PROFF');
```

3. List employees in ascending order of designations who joined after 1981:

```sql
SELECT * FROM Employee WHERE HIREDATE > '1981-12-31' ORDER BY Designation ASC;
```

4. List employees with their experience and daily salary:

```sql
SELECT E_name, DATEDIFF(CURDATE(), HIREDATE) / 365 AS ExperienceYears, Salary / 30 AS
DailySalary FROM Employee;
```

Problem 16: Employee Table with Additional Columns

1. Insert at least 5 rows into the Employee table:

```sql
INSERT INTO Employee (Emp_no, E_name, E_address, E_ph_no, Dept_no, Dept_name, Job_id,
Salary)
VALUES (1, 'Alice', '123 Main St', '555-1234', 'D10', 'HR', 'HR_MGR', 45000),
       (2, 'Bob', '456 Elm St', '555-5678', 'D20', 'Finance', 'ANALYST', 50000),
       (3, 'Charlie', '789 Oak St', '555-9012', 'D10', 'HR', 'CLERK', 30000),
       (4, 'David', '101 Pine St', '555-3456', 'D30', 'Sales', 'EXEC', 60000),
       (5, 'Eve', '202 Birch St', '555-7890', 'D10', 'HR', 'HR_ASSIST', 35000);
```

2. Display all records in the Employee table:

```sql
SELECT * FROM Employee;
```

3. Display records of employees in department D10:

```sql
SELECT * FROM Employee WHERE Dept_no = 'D10';
```

4. Update the city of `Emp_no` 12 to "Nagpur":

```sql
UPDATE Employee SET E_address = 'Nagpur' WHERE Emp_no = 12;
```

5. Display details of employees in department "MECH":

```sql
SELECT * FROM Employee WHERE Dept_name = 'MECH';
```

6. Delete the `email_id` of employee named "James":

```sql
UPDATE Employee SET E_ph_no = NULL WHERE E_name = 'James';
```

Problem 17: Modifications on EMPLOYEE Table

1. Add a new column `HIREDATE`:

```sql
ALTER TABLE Employee ADD HIREDATE DATE;
```

2. Change the `JOB_ID` data type from `CHAR` to `VARCHAR2`:

```sql
ALTER TABLE Employee MODIFY JOB_ID VARCHAR2(20);
```

3. Rename column `Emp_no` to `E_no`:

```sql
ALTER TABLE Employee RENAME COLUMN Emp_no TO E_no;
```

4. Modify the column width of the job field:

```sql
ALTER TABLE Employee MODIFY Job VARCHAR2(15);
```

This completes the SQL solutions for all the problems in your document. Let me know if you need further explanations or have additional questions!

Is this conversation helpful so far?