1. Basic Queries

Database and Table Creation

sql

Copy

Edit

CREATE DATABASE Practice1;

USE Practice1;

CREATE TABLE Mech (

s\_id INT,

s\_name VARCHAR(25)

);

CREATE DATABASE ORG123;

USE ORG123;

CREATE TABLE Worker (

WORKER\_ID INT NOT NULL PRIMARY KEY AUTO\_INCREMENT,

FIRST\_NAME CHAR(25),

LAST\_NAME CHAR(25),

SALARY INT(15),

JOINING\_DATE DATETIME,

DEPARTMENT CHAR(25)

);

Inserting Data

sql

Copy

Edit

INSERT INTO Mech VALUES (101, 'Jayanth'), (103, 'Karthick');

INSERT INTO Worker (WORKER\_ID, FIRST\_NAME, LAST\_NAME, SALARY, JOINING\_DATE, DEPARTMENT)

VALUES

(001, 'Monika', 'Arora', 100000, '2014-02-20 09:00:00', 'HR'),

(002, 'Niharika', 'Verma', 80000, '2014-06-11 09:00:00', 'Admin'),

(003, 'Vishal', 'Singhal', 300000, '2014-02-20 09:00:00', 'HR'),

(004, 'Amitabh', 'Singh', 500000, '2014-02-20 09:00:00', 'Admin'),

(005, 'Vivek', 'Bhati', 500000, '2014-06-11 09:00:00', 'Admin'),

(006, 'Vipul', 'Diwan', 200000, '2014-06-11 09:00:00', 'Account'),

(007, 'Satish', 'Kumar', 75000, '2014-01-20 09:00:00', 'Account'),

(008, 'Geetika', 'Chauhan', 90000, '2014-04-11 09:00:00', 'Admin');

Viewing Tables

sql

Copy

Edit

SELECT \* FROM Mech;

SELECT \* FROM Worker;

2. Constraints & Transactions

Using Transactions with Savepoints

sql

Copy

Edit

START TRANSACTION;

INSERT INTO Mech VALUES (104, 'Rahul');

SAVEPOINT A;

UPDATE Mech SET s\_name='Raj' WHERE s\_id=101;

SAVEPOINT B;

ROLLBACK TO A;

SELECT \* FROM Mech;

COMMIT;

3. Filtering with WHERE & Operators

sql

Copy

Edit

SELECT \* FROM Worker WHERE SALARY > 200000;

SELECT \* FROM Worker WHERE DEPARTMENT = 'HR';

SELECT \* FROM Worker WHERE SALARY BETWEEN 100000 AND 300000;

SELECT \* FROM Worker WHERE FIRST\_NAME LIKE 'M%';

SELECT \* FROM Worker WHERE SALARY > 300000 OR DEPARTMENT = 'Admin';

SELECT \* FROM Worker WHERE SALARY < 100000 AND DEPARTMENT = 'Account';

4. Sorting with ORDER BY

sql

Copy

Edit

SELECT \* FROM Worker ORDER BY SALARY DESC;

SELECT \* FROM Worker ORDER BY JOINING\_DATE ASC;

5. Grouping with GROUP BY and HAVING

sql

Copy

Edit

SELECT DEPARTMENT, AVG(SALARY) AS Avg\_Salary FROM Worker GROUP BY DEPARTMENT;

SELECT DEPARTMENT, COUNT(WORKER\_ID) AS Num\_Employees FROM Worker GROUP BY DEPARTMENT;

SELECT DEPARTMENT, SUM(SALARY) AS Total\_Salary FROM Worker GROUP BY DEPARTMENT HAVING SUM(SALARY) > 500000;

6. Aggregations (COUNT, SUM, AVG, MIN, MAX)

sql

Copy

Edit

SELECT COUNT(\*) AS Total\_Employees FROM Worker;

SELECT SUM(SALARY) AS Total\_Salary FROM Worker;

SELECT AVG(SALARY) AS Avg\_Salary FROM Worker;

SELECT MIN(SALARY) AS Min\_Salary, MAX(SALARY) AS Max\_Salary FROM Worker;

7. CASE Statement

sql

Copy

Edit

SELECT WORKER\_ID, FIRST\_NAME, DEPARTMENT,

CASE

WHEN SALARY > 300000 THEN 'Rich People'

WHEN SALARY BETWEEN 100000 AND 300000 THEN 'Middle Stage'

ELSE 'Poor People'

END AS People\_Stage

FROM Worker;

8. Using Joins

Inner Join

sql

Copy

Edit

SELECT W.WORKER\_ID, W.FIRST\_NAME, M.s\_name

FROM Worker W

INNER JOIN Mech M ON W.WORKER\_ID = M.s\_id;

Left Join

sql

Copy

Edit

SELECT W.WORKER\_ID, W.FIRST\_NAME, M.s\_name

FROM Worker W

LEFT JOIN Mech M ON W.WORKER\_ID = M.s\_id;

Right Join

sql

Copy

Edit

SELECT W.WORKER\_ID, W.FIRST\_NAME, M.s\_name

FROM Worker W

RIGHT JOIN Mech M ON W.WORKER\_ID = M.s\_id;

Full Outer Join (Using UNION)

sql

Copy

Edit

SELECT W.WORKER\_ID, W.FIRST\_NAME, M.s\_name

FROM Worker W

LEFT JOIN Mech M ON W.WORKER\_ID = M.s\_id

UNION

SELECT W.WORKER\_ID, W.FIRST\_NAME, M.s\_name

FROM Worker W

RIGHT JOIN Mech M ON W.WORKER\_ID = M.s\_id;

9. Using Views

sql

Copy

Edit

CREATE VIEW Worker\_View AS

SELECT WORKER\_ID, FIRST\_NAME, LAST\_NAME, SALARY, DEPARTMENT

FROM Worker WHERE SALARY > 100000;

SELECT \* FROM Worker\_View;

10. Using Subqueries

sql

Copy

Edit

SELECT FIRST\_NAME, SALARY FROM Worker

WHERE SALARY > (SELECT AVG(SALARY) FROM Worker);

SELECT WORKER\_ID, FIRST\_NAME FROM Worker

WHERE DEPARTMENT = (SELECT DEPARTMENT FROM Worker WHERE WORKER\_ID = 1);

11. Using IN and EXISTS

sql

Copy

Edit

SELECT \* FROM Worker WHERE DEPARTMENT IN ('HR', 'Admin');

SELECT \* FROM Worker WHERE EXISTS

(SELECT 1 FROM Mech WHERE Mech.s\_id = Worker.WORKER\_ID);

12. Using LIMIT and OFFSET

sql

Copy

Edit

SELECT \* FROM Worker ORDER BY SALARY DESC LIMIT 5;

SELECT \* FROM Worker ORDER BY SALARY DESC LIMIT 5 OFFSET 3;

13. Using UNION and UNION ALL

sql

Copy

Edit

SELECT FIRST\_NAME, SALARY FROM Worker WHERE SALARY > 100000

UNION

SELECT FIRST\_NAME, SALARY FROM Worker WHERE DEPARTMENT = 'Admin';

SELECT FIRST\_NAME, SALARY FROM Worker WHERE SALARY > 100000

UNION ALL

SELECT FIRST\_NAME, SALARY FROM Worker WHERE DEPARTMENT = 'Admin';

14. Indexing

sql

Copy

Edit

CREATE INDEX idx\_salary ON Worker(SALARY);

CREATE INDEX idx\_department ON Worker(DEPARTMENT);

15. Deleting Data & Dropping Tables

sql

Copy

Edit

DELETE FROM Worker WHERE WORKER\_ID = 2;

DROP TABLE Mech;

DROP DATABASE Practice1;