

SQL Commands

→ DDL → Data Definition Language → Structure

CREATE
ALTER
DROP
TRUNC

\rightarrow DMC \rightarrow Data Manipulation Language — Rows

INSERT
UPDATE
DELETE

\rightarrow DQL \rightarrow Data Query Language - Fetching Rows

SELECT

→ TCL → Transaction Control Language

ROLLBACK
COMMIT

→ DCL → Data Control Language

The diagram illustrates a legal relationship between a 'Grantee' and a 'Servitor'. The 'Grantee' box contains the word 'Revoke'. An arrow points from 'Revoke' to the 'Servitor' circle. Below the boxes are three numbers: 1, 2, and 3.

DDL

```
CREATE TABLE table-name
(
    column1 Data-type (width) constraints,
    column2 Data-type (width) constraints,
    ...
);
```

```
CREATE TABLE employee
(
    empid int primary key,
    name varchar(50) not null,
    phone varchar(20) unique,
    salary float
);
```

ALTER

```
ALTER TABLE table-name
[ADD  
MODIFY  
DROP] column-name data-type (width)
```

1) ALTER TABLE employee
ADD age int;

2) ALTER TABLE employee
MODIFY phone char(15);

3) ALTER TABLE employee
DROP age;

DROP

```
DROP TABLE table-name
```

DROP TABLE employee;

TRUNCATE TABLE employee

DMLINSERT

INSERT INTO Table-Name [(Column₁, Col₂, Col₃, ...)] VALUES (Val₁, Val₂, Val₃, ...)

INSERT INTO employee VALUES (1, 'Anil', 2500)

INSERT INTO employee VALUES (1, 'Anil', null);

INSERT INTO employee (name, empid) VALUES ('Anil', 1);

| | | | |
|-----|--------|------|----|
| 1 | Anil | 2500 | 10 |
| 2 | Ram | 1500 | 20 |
| → 3 | Rakesh | 1000 | 10 |
| 4 | Pawan | 3500 | 30 |

UPDATE

UPDATE Table-Name

SET Column₁ = Value₁,
Column₂ = Value₂,

⋮

WHERE condition

UPDATE employee
SET name = 'Rakesh'
WHERE empid = 3

UPDATE employee
SET salary = salary + salary * 0.1
WHERE deptno = 10;

UPDATE employee
SET name = 'Rakesh',
salary = 2000
WHERE empid = 3;

DELETE

DELETE FROM Table-name
WHERE condition;

e.g. DELETE FROM employee;

DELETE FROM employee WHERE deptno ≥ 20;

DELETE FROM employee WHERE salary < 1000;

SELECT

```
SELECT * | (column, Expression, Literal)
FROM Table-Name
WHERE Condition
ORDER BY Column1, Column2, ...;
```

→ SELECT * FROM employee;

→ SELECT empid, name FROM employee;

→ SELECT empid, name, salary * 12 FROM employee;

→ SELECT empid AS EmployeeID, name AS EmployeeName, salary * 12 AS AnnualSalary
FROM employee

→ SELECT empid, -, name FROM employee L - Anil
 2 - Ram

→ SELECT * FROM employee WHERE depthno = 10;

→ SELECT * FROM employee
WHERE salary > 5000;

→ SELECT * FROM employee
WHERE depthno = 10
AND salary > 5000;

→ SELECT * FROM employee
ORDER BY salary DESC

→ SELECT * FROM employee
ORDER BY depthno, salary DESC

→ SELECT * FROM employee
ORDER BY depthno DESC, salary

→ SELECT * FROM employee
WHERE depthno < 50
ORDER BY depthno DESC, salary

→ SELECT count(*) FROM employee

→ SELECT distinct depthno FROM employee

| | | SI | PF |
|---|---|------|----|
| 1 | A | 2000 | 20 |
| 2 | B | 1800 | 10 |
| 3 | C | 1000 | 20 |
| 4 | D | 2000 | 10 |
| 5 | E | 1800 | 10 |
| 6 | F | 2500 | 20 |
| 7 | G | 1000 | 20 |

10
20

| | | | | | Group |
|---|---|-----|----|--|----------------|
| | | | | | Group By depth |
| 1 | A | 100 | 10 | | |
| 2 | B | 200 | 20 | | |
| 3 | C | 200 | 10 | | |
| 4 | D | 300 | 20 | | |

SUM, MAX, MIN, AVG, COUNT

```
SELECT sum(sal), MAX(sal), MIN(sal), COUNT(sal)
FROM Exp;
```

800 300 100 4

Select sum(sal), MAX(sal), MIN(sal), COUNT(sal)
FROM exp
GROUP BY depth;
→ 300 200 100 2
→ 800 300 200 2

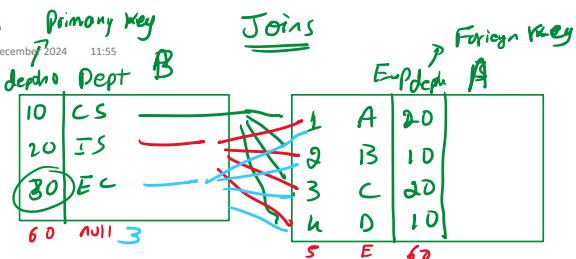
SELECT depth, MAX(sal)
FROM employee
WHERE sal > 100
GROUP BY depth
Having MAX(sal) > 200

20 300

| | | |
|----|-----|---|
| 10 | 200 | ✓ |
| 20 | 300 | ✗ |

SELECT

```
FROM emp
GROUP BY depth;
```



SELECT A.empno, A.name, A.deptno, B.deptno, B.depname
FROM Emp AS A, Dept AS B
WHERE A.deptno = B.deptno;

↑

Equi-Join

| | | | | |
|---|---|----|----|----|
| 1 | A | 20 | 10 | CS |
| 2 | B | 10 | 10 | CS |
| 3 | C | 20 | 10 | CS |
| 4 | D | 10 | 10 | CS |
| 1 | A | 20 | 20 | IS |
| 2 | B | 10 | 20 | IS |
| 3 | C | 20 | 20 | IS |
| 4 | D | 10 | 20 | IS |
| 1 | A | 20 | 30 | EC |
| 2 | B | 10 | 30 | EC |
| 3 | C | 20 | 30 | EC |
| 4 | D | 10 | 30 | EC |

| | | | | |
|---|---|----|----|----|
| 2 | B | 10 | 10 | CS |
| 4 | D | 10 | 10 | CS |
| 1 | A | 20 | 20 | IS |
| 3 | C | 20 | 20 | IS |
| - | - | 30 | EC | |

SELECT empno, name, deptno, depname
FROM Emp
NATURAL JOIN Dept

| eno | name | Emp | FK |
|-----|------|------|----|
| 1 | A | null | |
| 2 | B | 1 | |
| 3 | C | 1 | |
| 4 | D | 2 | |
| 5 | E | 3 | |

| | | | |
|---|---|------|---|
| 1 | A | null | - |
| 2 | B | 1 | A |
| 3 | C | 1 | A |
| 4 | D | 2 | B |
| 5 | E | 3 | C |

Self Join

SELECT A.empno, A.name, A.mgr, B.empno, B.name
FROM Emp A, Emp B
WHERE A.mgr = B.empno;

| mgr | eno |
|-----|-----|
| 1 | 1 |
| 2 | 1 |
| 3 | 1 |
| 4 | 2 |
| 5 | 2 |

| A | eno | name | sal | B | | | Salgrade |
|---|-----|------|-----|-------|-------|-------|----------|
| | | | | Grade | Local | Hired | |
| 1 | A | 300 | | A | 301 | 800 | |
| 2 | B | 500 | | B | 1 | 300 | |
| 3 | C | 250 | | | | | |
| 4 | D | 400 | | | | | |

| | | | |
|---|---|-----|---|
| 1 | A | 300 | B |
| 2 | B | 500 | A |
| 3 | C | 250 | B |
| 4 | D | 400 | A |

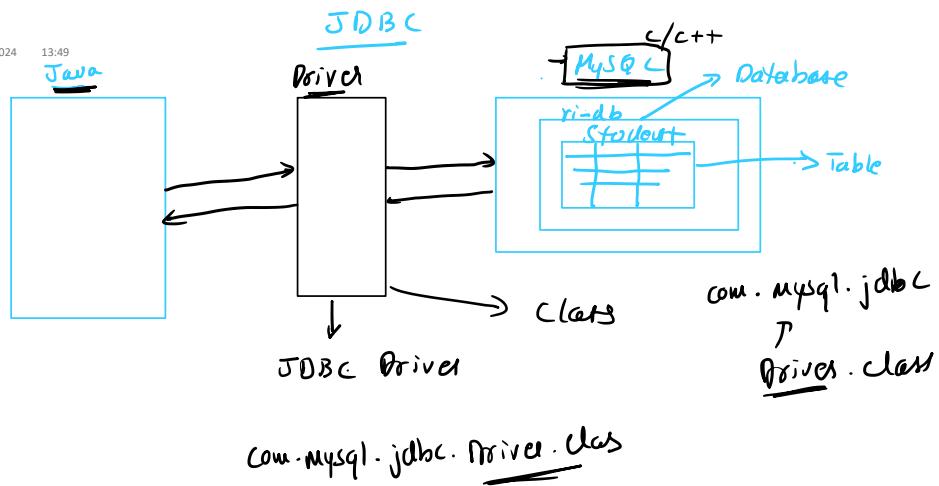
SELECT A.empno, A.name, B.sal, B.grade
FROM Emp A, Salgrade B

WHERE $A.Sal \geq B.Sal$ AND $A.Sal \leq B.HireD$;

| | <u>A</u> | <u>B</u> | |
|---|----------|-------------|---|
| 1 | A 300 | A 301 800 X | 1 |
| 2 | B 500 | A 301 800 X | |
| 3 | C 750 | A 301 800 X | |
| 4 | D 600 | A 301 800 X | 2 |
| 5 | A 300 | B 1 300 X | 3 |
| 6 | B 500 | B 1 300 X | |
| 7 | C 750 | B 1 300 X | 4 |
| 8 | D 600 | B 1 300 X | |

07

18 December 2024 13:49



1) Load JDBC Driver

```
Class.forName("class-name")
    ↳ ClassNotFoundException
```

↳ Checked Exception.

```
try {
    Class.forName("com.mysql.jdbc.Driver");
} catch (ClassNotFoundException e) {
    e.printStackTrace();
}
```

2) Establish connection to mysql database

URL
username → `jdbc:mysql://` `localhost:3306/` `ri-db`
password → Protocol servername portno Database
 ↳ URL

```
Connection conn=DriverManager.getConnection(url, user, pwd)
    ↳ SQLException
```

```
String url = "jdbc:mysql://localhost/ri-db";
String username = "test";
String password = "test123";
try {
    Connection conn = DriverManager.getConnection(url, username, password);
} catch (SQLException e) {
    e.printStackTrace();
}
```

3) Create either Statement or PreparedStatement class object to issue SQL command

Connection

```

↳ Statement createStatement()
    PreparedStatement prepareStatement(sql)
  
```

Statement st = conn.createStatement();

4) Issue SQL Commands

Statement → int executeUpdate("insert/update/delete")

ResultSet executeQuery("select")

String sql = "SELECT * FROM contacts";

ResultSet rs = st.executeQuery(sql);

while(rs.next())

```

        System.out.println("Id:" + rs.getInt(1) + "Name:" + rs.getString(2)
            + ", Email:" + rs.getString(3) + ", Phone number:"
            + rs.getString(4) + ", City:" + rs.getString(5));
      
```



ResultSet
→ boolean next()

| 1 | 2 | 3 | 4 | 5 |
|----|---------------|---------------------|--------------|-----------|
| id | name | email | phone_number | city |
| 1 | Ravi Kumar | ravi@example.com | 9876543210 | Delhi |
| 2 | Priya Sharma | priya@example.com | 9876543211 | Mumbai |
| 3 | Amit Patel | amit@example.com | 9876543212 | Bangalore |
| 4 | Neha Gupta | neha@example.com | 9876543213 | Kolkata |
| 5 | Sandeep Singh | sandeep@example.com | 9876543214 | Chennai |

5) Close the Connection

try

 st.close()

 conn.close()

 }

 } → SQL Exception

```

import java.util.*;
import java.sql.*;
class Main{
    public static void main(String[] args){
        Connection conn = null;
        PreparedStatement ps = null;
        String url = "jdbc:mysql://localhost/hi_db";
        String username = "test";
        String password = "test123";
        try{
            conn = DriverManager.getConnection(url,username,password);
            System.out.println("Database connected successfully");
            String sql = "INSERT INTO office VALUES(?, ?, ?)"
            int eid = &conn.createStatement();
            &conn.nextLine();
            String ername = &conn.nextLine();
            int esal = &conn.nextInt();
            ps = conn.prepareStatement(sql);
            ps.setInt(1, eid);
            ps.setString(2, ername);
            ps.setInt(3, esal);
            int count = ps.executeUpdate();
        }catch(SQLException e){
            e.printStackTrace();
        }finally{
            try{
                ps.close();
                conn.close();
                System.out.println("Connection closed successfully");
            }catch(SQLException e){
                e.printStackTrace();
            }
        }
    }
}

```

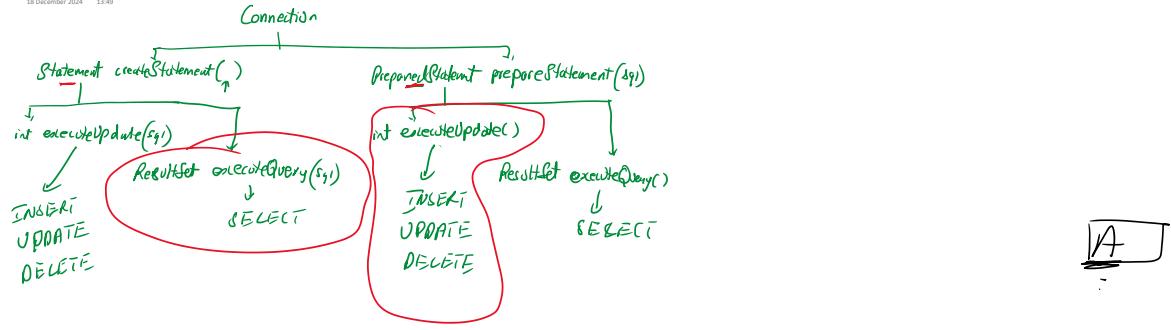
3

```

ps = conn.prepareStatement(" INSERT INTO office VALUES(?, ?, ?)");
int n = &conn.nextInt();
for (int i=1; i<=n; i++) {
    ps.setInt(1, &conn.nextInt());
    &conn.nextLine();
    ps.setString(2, &conn.nextLine());
    ps.setInt(3, &conn.nextInt());
    ps.executeUpdate();
}

Statement st = conn.createStatement();
ResultSet rs = st.executeQuery(" select * from office");
while(rs.next()){
    System.out.println(rs.getInt(1) + " " + rs.getString(2) + " " + rs.getInt(3));
}

```



A

```

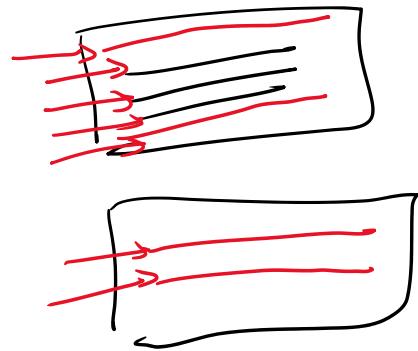
Statement st = conn.createStatement();
int sal = scan.nextInt();
rs = st.executeQuery("SELECT * FROM employee WHERE salary > " + sal);
String gender = scan.nextLine();
rs = st.executeQuery("SELECT * FROM employee WHERE gender = '" + gender + "'");

char ch = scan.nextLine().charAt(0);
rs = st.executeQuery("SELECT * FROM employee WHERE name LIKE '" + ch + "%'");
  
```

Name LIKE 'A%'

```
if(rs.next())  
{  
    do  
    {  
        ...  
    } while(rs.next());  
}
```

else



11

UPDATE locar

SET interestrate = ?

WHERE loadid = ? "