JDBC

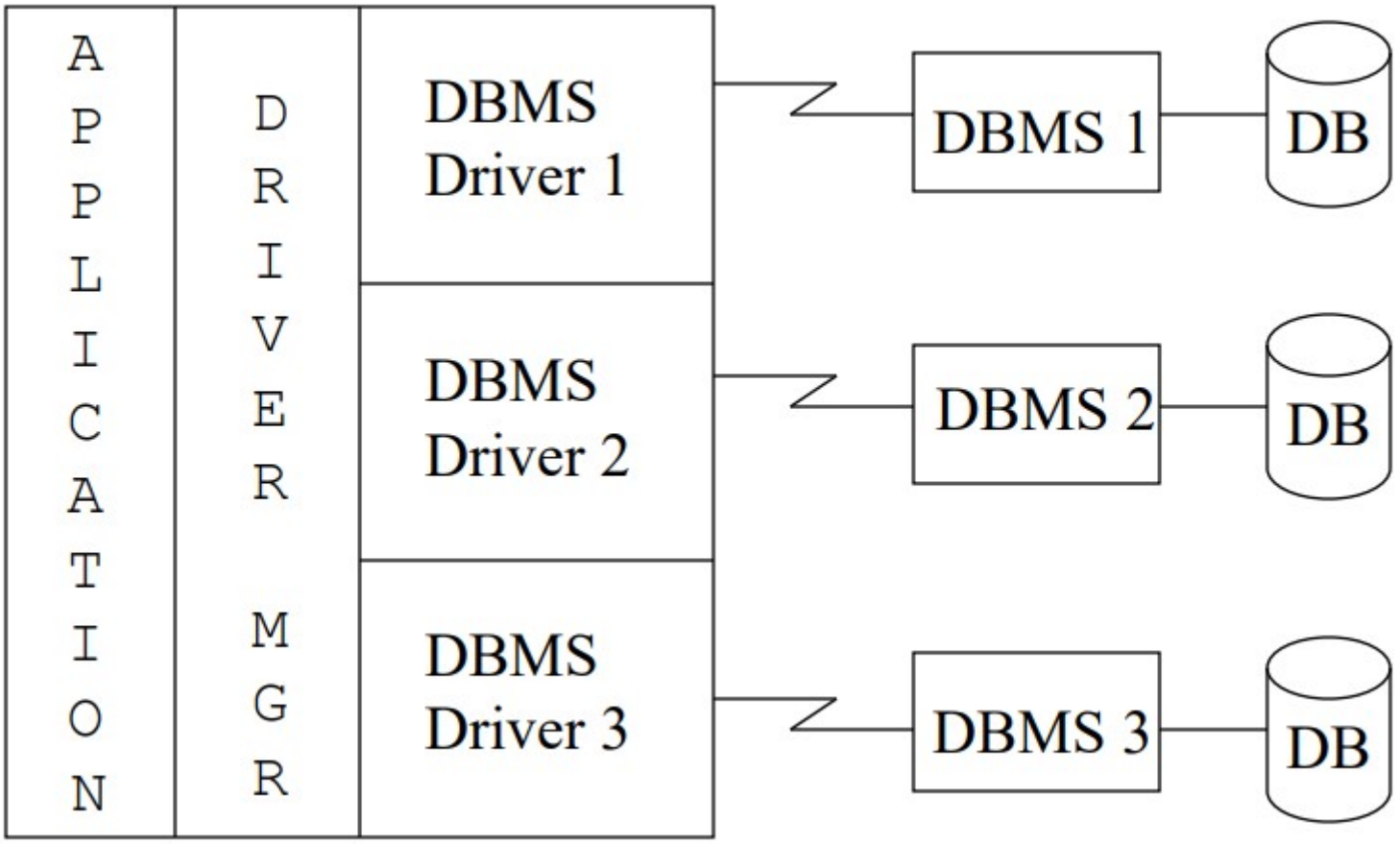
By
Uday Kumar

Introduction

- Most popular form of database system is the relational database system.

- Examples: MS Access, Sybase, Oracle, MS SQL Server, MySQL.
- Structured Query Language (SQL) is used among relational databases to construct queries

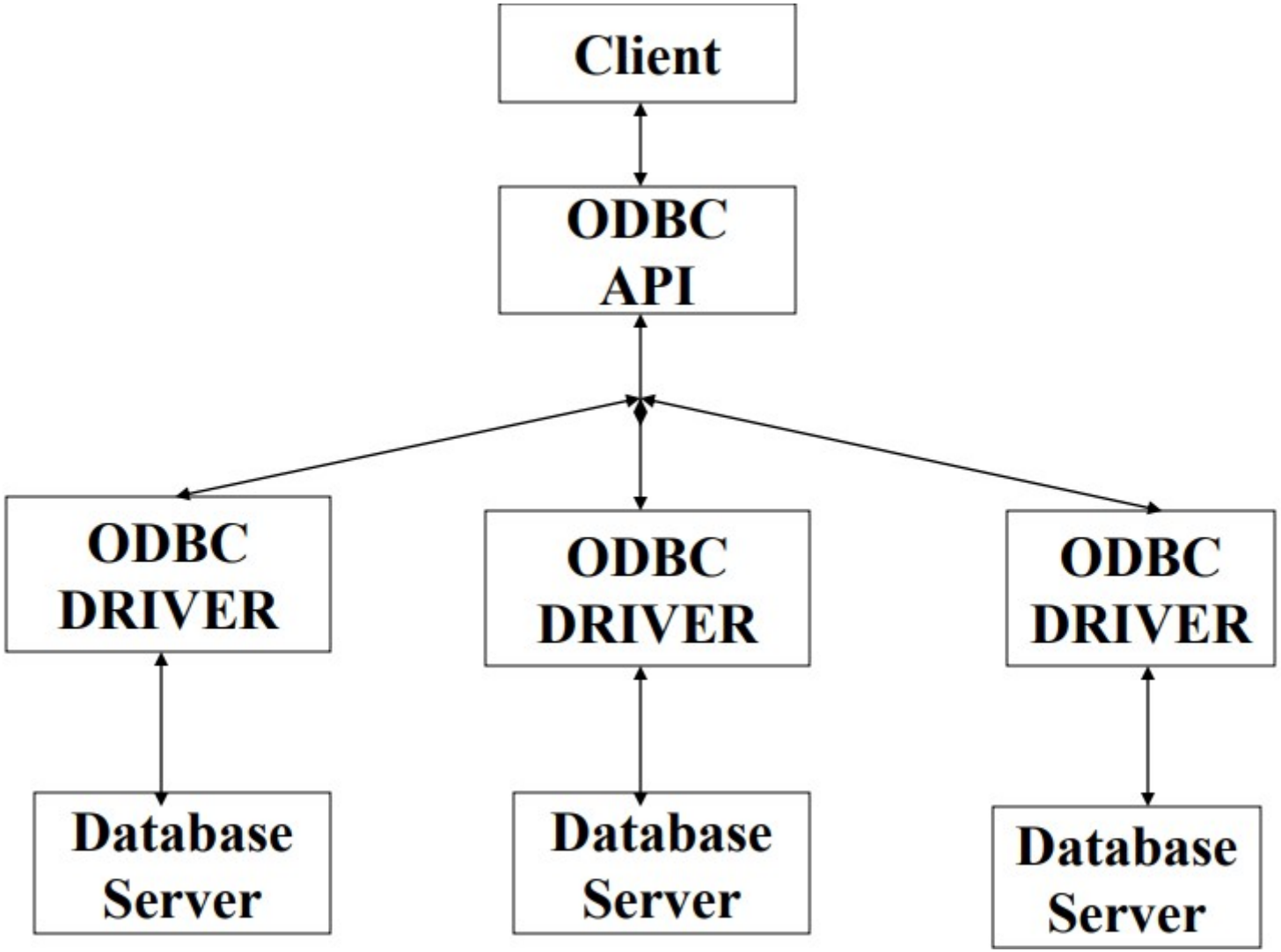
Standard Access to DB



ODBC

- ODBC (Open Database Connectivity) is a Microsoft standard from the mid 1990‟s.
- The database driver bridges the differences between your underlying system calls and the ODBC interface functionality
- ODBC offers a set of specifications to the server side database engines. Any client can Connect to the server, if the server follows these specifications properly

ODBC Architecture



Drawbacks of ODBC

- ODBC API was completely written in C language.
- Calls to native C code have a number of drawbacks in the security.
- ODBC drivers must be installed on Client's machine

What is JDBC?

- Java Database Connectivity (JDBC) is a Java API for connecting and executing queries with a database
- Provides a standard way to connect to different types of databases
- Part of the Java Standard Edition platform

Overall Architecture

Key JDBC Components

- JDBC Driver: Software that enables Java to interact with a specific database
- Connection: Establishes a session with a specific database
- Statement: Used to submit SQL statements to the database
- ResultSet: Represents data returned from a database query

JDBC Driver Types

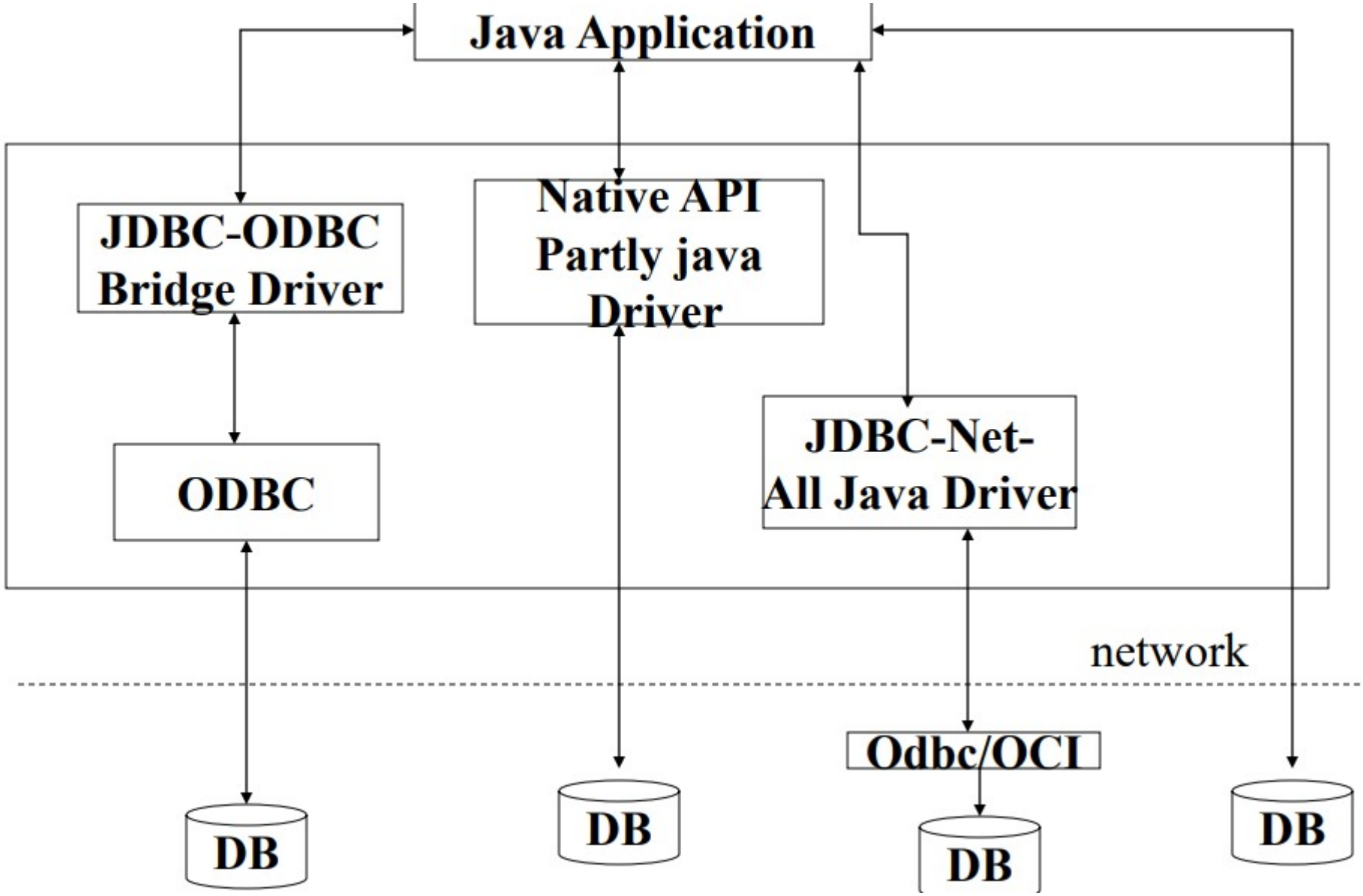
- Type 1 (JDBC-ODBC Bridge):
  - Uses ODBC drivers to connect to databases
    - Translates Java to the ODBC API
  - Partially implemented, considered legacy
- Type 2 (Native API):
  - Uses database vendor's native library
    - Translate Java to the database's own API
  - Partially Java, partially native code

JDBC Driver Types Contd..

- Type 3 (Network Protocol):
  - Uses middleware(usually TCP/IP) to convert JDBC calls to database-specific protocol
  - Pure Java, but requires middleware server
  - Required for networked applications

- Type 4 (Thin Driver):
  - Pure Java implementation
  - Direct communication with database
  - Most commonly used in modern applications

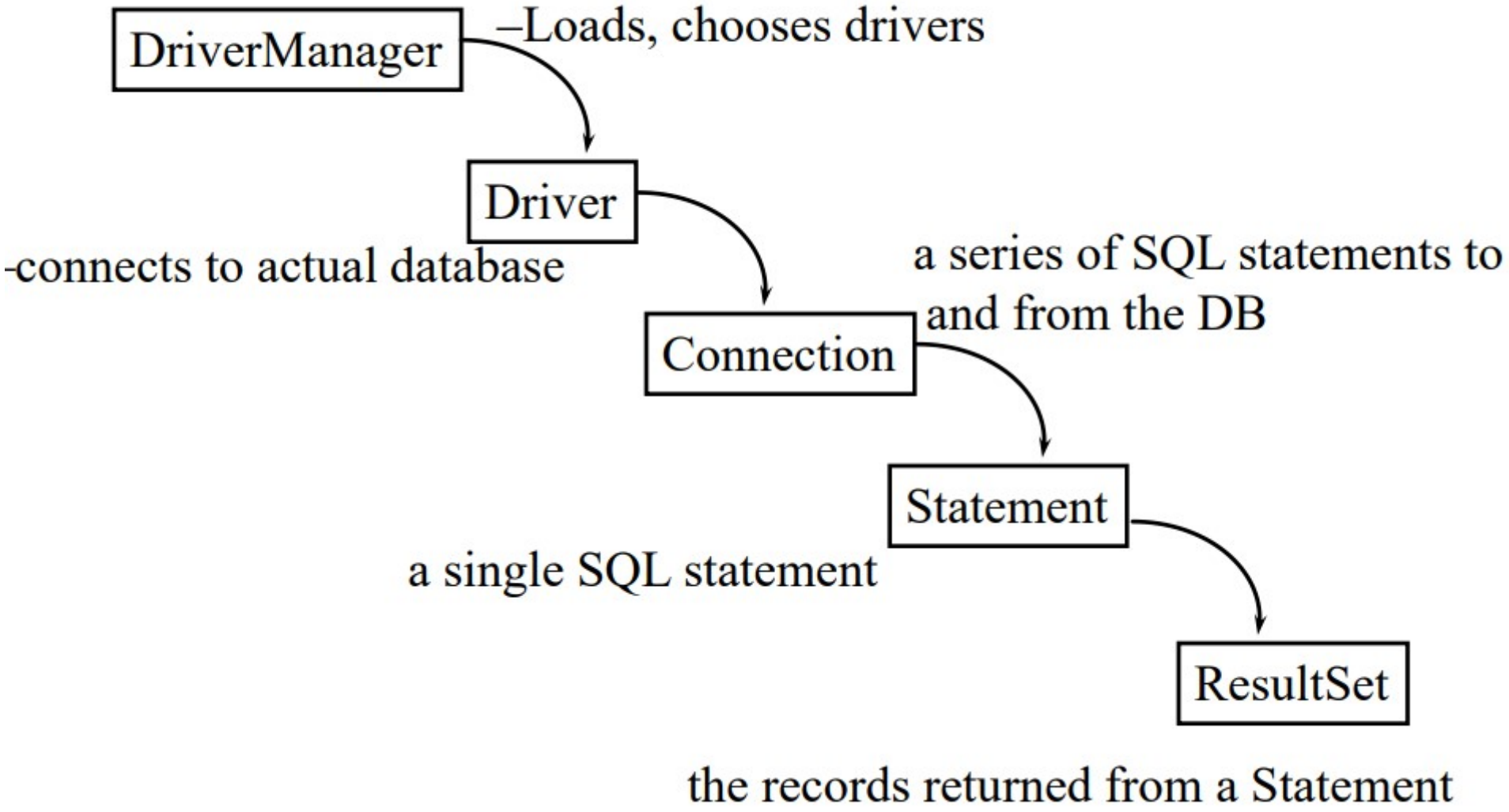
JDBC Driver Types Contd..



Basic JDBC Connection Steps

1. Load the JDBC driver
2. Establish a connection
3. Create a statement
4. Execute the query
5. Process the result set
6. Close the connection

JDBC Process



JDBC Connection Code

```java
try {
    // Load driver
    Class.forName("com.mysql.jdbc.Driver");

    // Establish connection
    Connection conn = DriverManager.getConnection(
        "jdbc:mysql://localhost:3306/mydatabase",
        "username",
        "password"
    );

    // Create statement
    Statement stmt = conn.createStatement();

    // Execute query
    ResultSet rs = stmt.executeQuery("SELECT * FROM users");

    // Process results
    while(rs.next()) {
        System.out.println(rs.getString("username"));
    }

    // Close resources
    rs.close();
    stmt.close();
    conn.close();
} catch (Exception e) {
    e.printStackTrace();
}
```

JDBC URLs

- jdbc:subprotocol:source
  - each driver has its own subprotocol
  - each subprotocol has its own syntax for the source

- *jdbc:odbc:DataSource*

– e.g. jdbc:odbc:cdac

- *jdbc:mysql://host[:port]/database*

Connection

- Connection getConnection(String url, String user, String password)
- Connects to given JDBC URL with given user name and password
  - Throws java.sql.SQLException
  - returns a Connection object

Statement

- The Statement object provides a "workspace" where SQL queries can be created and results collected.
- ***Methods:***
  - ***ResultSet executeQuery(String)***
    - Execute a SQL statement that returns a single ResultSet.
  - ***int executeUpdate(String)***
    - Execute a SQL INSERT, UPDATE or DELETE statement. Returns
    - the number of rows changed.
  - ***boolean execute(String)***
    - Execute a SQL statement that may return multiple results.

Driver Loading Process

- When Class.forName() is called, it dynamically loads the MySQL JDBC driver class
  - This method does three important things:
    a) Loads the driver class into memory
    b) Initializes the class
    c) Registers the driver with the DriverManager automatically

```
// When this line is executed
Class.forName("com.mysql.jdbc.Driver");

// It's equivalent to what happens behind the
scenes:
Driver mysqlDriver = new com.mysql.jdbc.Driver();
DriverManager.registerDriver(mysqlDriver);
```

ResultSet

- A ResultSet provides access to a table of data generated by executing a Statement.
- Only one ResultSet per Statement can be open at once.
  - The table rows are retrieved in sequence.
- A ResultSet maintains a cursor pointing prior to its current row of data.
- The 'next' method moves the cursor to the next row.

Prepared Statements

- More secure alternative to regular statements
- Prevents SQL injection
- Allows parameterized queries
  - It treats all input as literal values
- Improves performance through query precompilation

```
PreparedStatement pstmt =
conn.prepareStatement("INSERT INTO users (name,
email) VALUES (?, ?)");

pstmt.setString(1, "abc");

pstmt.setString(2, "abc@cdac.in");

pstmt.executeUpdate();
```

Transactions

- A logical unit of work
- Either a single or set of SQL statements either all execute successfully or all of them fail compulsory.
- ACID Properties
  - **Atomic** - All the statements treated as a single unit
  - **Consistent** – user should see the same data until the transaction completes
  - **Isolation** – each transaction should be treated separately
  - **Durability** – changes can be made permanent

Transaction Management

- Connection.setAutoCommit(boolean)
  - if *AutoCommit* is false, then every statement is added to an ongoing transaction
  - You must explicitly commit or rollback the transaction using *Connection.commit()* and *Connection.rollback()*

Best Practices

- Always use try-with-resources for automatic resource management
- Close database connections after use
- Use prepared statements to prevent SQL injection
- Handle exceptions properly
- Use connection pooling for better performance