Database Connectivity with Java

Prativa Nyaupane

Outline

- JDBC Architecture
- JDBC Driver Types and Configuration
- Managing Connections and Statements
- Result Sets and Exception Handling
- DDL and DML Operations
- SQL Injection and Prepared Statements
- Row Sets and Transactions
- SQL Escapes

Introduction

- A database is an organized collection of data. Database Management System provides mechanisms for storing, retrieving and modifying data for many users.
- DBMS allow for the access and storage of data without concern for the internal representation of data. RDMBS stores data in tables. Tables are compose of rows and columns.
- Some popular RDBMSs are Microsoft SQL Server, Oracle, Sybase, PostgreSQL and MySQL.
- Java programs communicate with database and manipulate their data using the Java Database Connectivity(JDBC) API. It allows any java program to execute SQL(Structured Query Language) statements and retrieve results, modify data and delete data.

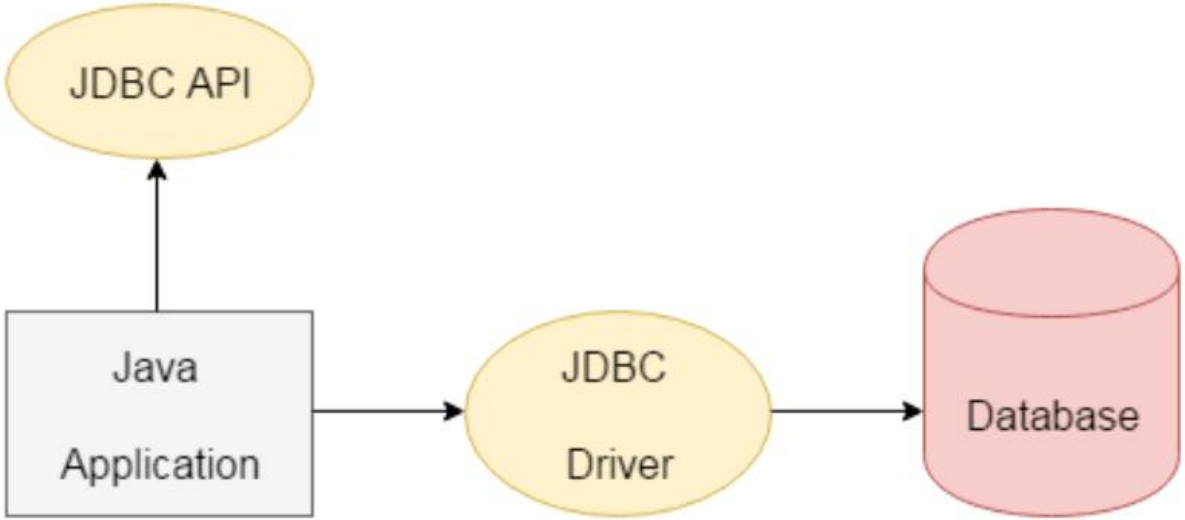
Fig: JDBC Link with Java Application and Database

Introduction Contd..

- **The JAVA API** is an interface which consists of description of all the characteristics of the application. It is just a set of procedures that carry out specific tasks and make development of applications easier.
- The java.sql package contains classes and interfaces for JDBC API
- **Java provides following APIs with respect to database:**
  - Defining the DriverManager for respective database drivers.
  - Establishing a connection to the database
  - Creating statements.
  - Executing queries in the database
  - Display and update of the resultant tuples.

JDBC simplifies database interaction by providing a set of interfaces for seamless portability across different databases.

JDBC Architecture Contd..



JDBC Driver Types and Configuration

- A database driver is a software component that acts as an interface between the Java application and a specific database management system (DBMS).
- It allows Java applications to communicate with the database by translating the JDBC calls into the appropriate DBMS-specific commands.
- Each database vendor provides its own database driver, which needs to be included in the Java application's classpath to establish a connection with that specific database.

Java Native Driver

- A Java Native Driver is a JDBC driver that is specifically designed to work with a particular database management system (DBMS) using native protocols.
- It is implemented in Java and provides direct communication between the Java application and the DBMS without the need for any intermediate layers or bridges.
- Java Native Drivers offer better performance and platform independence compared to JDBC-ODBC bridges.

Java Native Driver

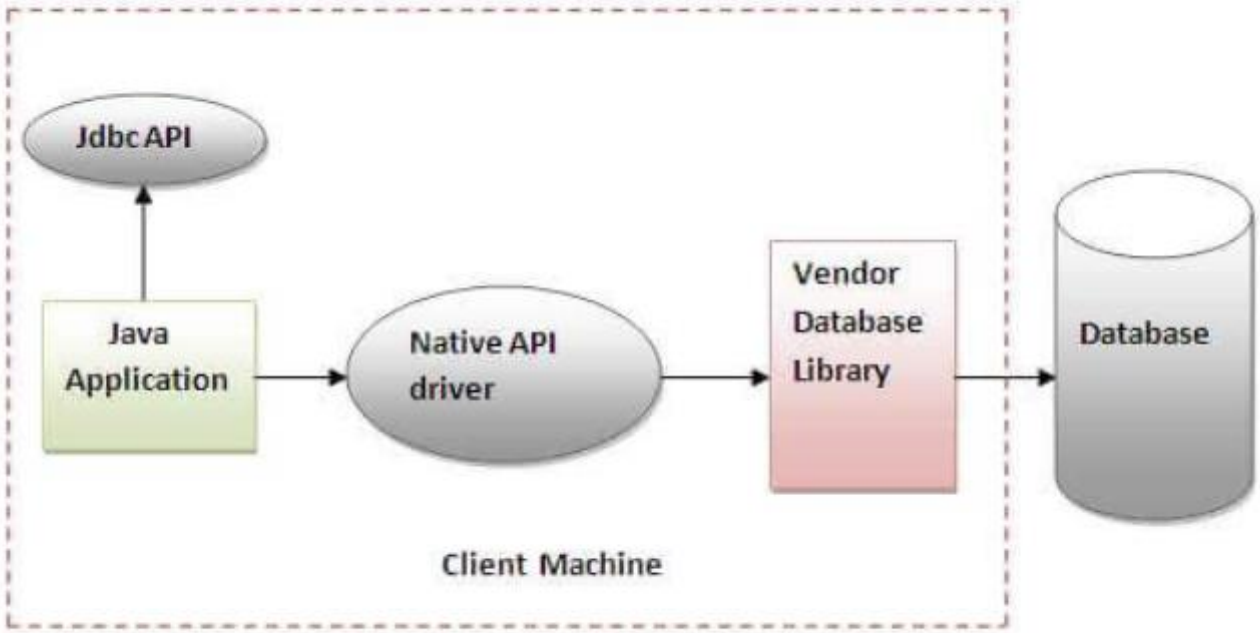


Figure- Native API Driver

Java Native Driver

- Advantage:
  - performance upgraded than JDBC-ODBC bridge driver.
- Disadvantage:
  - The Native driver needs to be installed on the each client machine.
  - The Vendor client library needs to be installed on client machine.

Lab

- JDBC Database Application: Create a Java application that connects to a database using JDBC, performs CRUD(Create, Read, Update, Delete) operations and handles exceptions.

Tutorial

- Database Connectivity with JDBC: Step-by-step tutorials on JDBC configuration, database connection management, executing SQL queries and handling result sets.