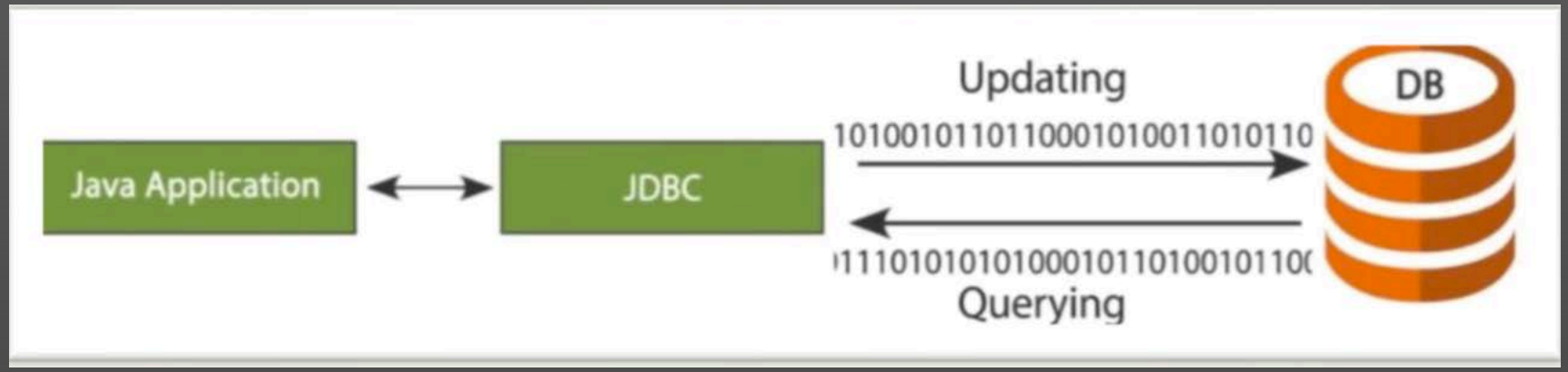
JDBC

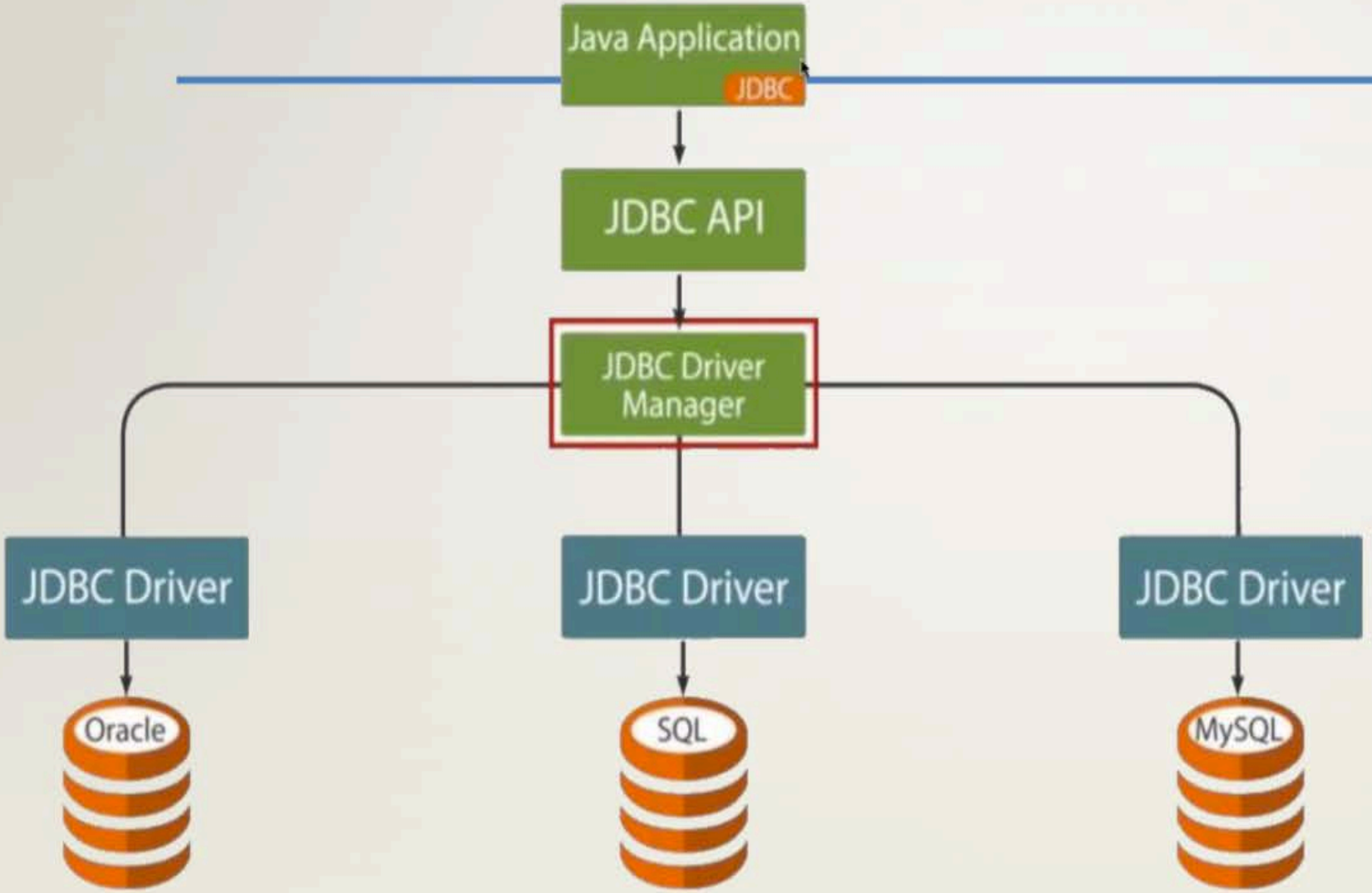
What is JDBC?

- Java Database Connectivity (JDBC) is the industry standard for database-independent connectivity between the Java programming language and a wide range of databases



Cybertek

ARCHITECTURE OF JDBC



Cybertek

JDBC Maven Dependency

```
<dependency>
        <groupId>oracle</groupId>
        <artifactId>ojdbc6</artifactId>
        <version>11.2.0.3</version>
</dependency>
```

Connection String

- dbUrl = "**jdbc:oracle:thin:@yourIPaddressandport:xe**";
  dbUsername = "**hr**";
  dbPassword = "**hr**";

3 Important Steps in JDBC

- Connection ➔ Helps our java project connect to database
- Statement ➔ Helps to write and execute SQL query
- ResultSet ➔ A DataStructure where we can store the data that came from database

Cybertek

Create Connection

- Connection
```
import java.sql.Connection;
```
- DriverManager
```
import java.sql.DriverManager;
```

```
Connection connection = DriverManager.getConnection(url, userName, passWord);
```

- URL syntax : jdbc:DataBaseType:thin:@Host:port:SID

Cybertek

Create Statement

- After we successfully created the connect next step is Statement

```
import java.sql.Statement;
```

```
Statement statement = connection.createStatement();
```

- We use createStatement( ) method to create the statement from our connection

Create ResultSet

- Once we have statement we can run the query and get the result to ResultSet format

```
import java.sql.ResultSet;
```

- We use the method executeQuery( ) to execute our queries

```
ResultSet result = statement.executeQuery("Select * from employees");
```

Printing the Result

- We use next() method to iterate each rows

```
while(result.next()) {

    System.out.println(result.getString("first_name"));

}
```

- While the next row is not Null we can continue to iterate
- If next row is Null (empty), it exits the loop immediately

Cybertek

ResultSet Methods

- next()
- getString(ColumName)
- getString(Index)
- getInt(ColumName)
- getInt(Index)
- getDouble(ColumName)
- getDouble(Index)
- getDate(ColumName)
- getDate(Index)

Execute Update

- Can help us run the query statements which can be an INSERT, UPDATE, , DELETE, ALTER, TRUNCATE, DROP statements

```
Statement statement = connection.createStatement();

statement.executeUpdate("Update developers Set names='Erhan' Where names = 'Madina'");
```

Cybertek

Working with MetaData

- Metadata Programming is useful to know the capabilities, limitations and facilities of underlying database software and its resources

- JDBC metadata programming Supports :
  - DatabaseMetadata
  - ResultSetMetaData

DataBase MetaData

```
DatabaseMetaData metaData = connection.getMetaData();

System.out.println("User: " + metaData.getUserName());
System.out.println("Database Type: "+metaData.getDatabaseProductName());
System.out.println("Database Version: "+metaData.getDatabaseProductVersion());
System.out.println("Driver Name: "+metaData.getDriverName());
System.out.println("Driver Version: "+metaData.getDriverVersion());
```

```
User: HR
Database Type: Oracle
Database Version: Oracle Database 11g Express Edition Release 11.2.0.2.0 - Production
Driver Name: Oracle JDBC driver
Driver Version: 12.1.0.1.0
```

Cybertek

ResultSet MetaData

```
ResultSet result = statement.executeQuery("Select * from employees");

ResultSetMetaData rsMetadata = result.getMetaData();

System.out.println("Columns count: " + rsMetadata.getColumnCount());
System.out.println("Column Name: " + rsMetadata.getColumnName(1));
```

```
Columns count: 11
Column Name: EMPLOYEE_ID
```

Cybertek