

Java Database Connectivity

Presented by

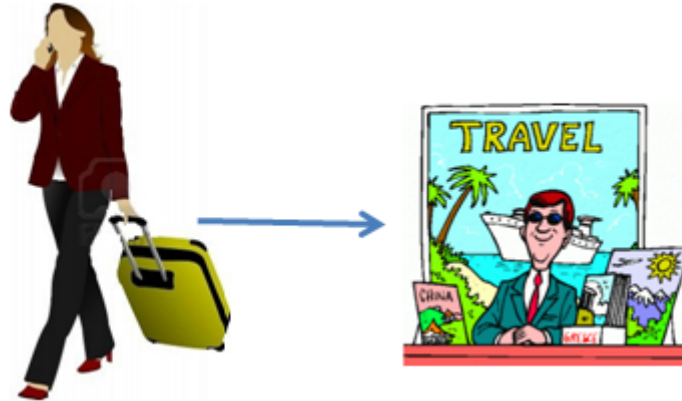


JDBC – Real World Analogy

A business woman needs to travel on a world tour and is in need of a VISA for U.S, Singapore and Italy.



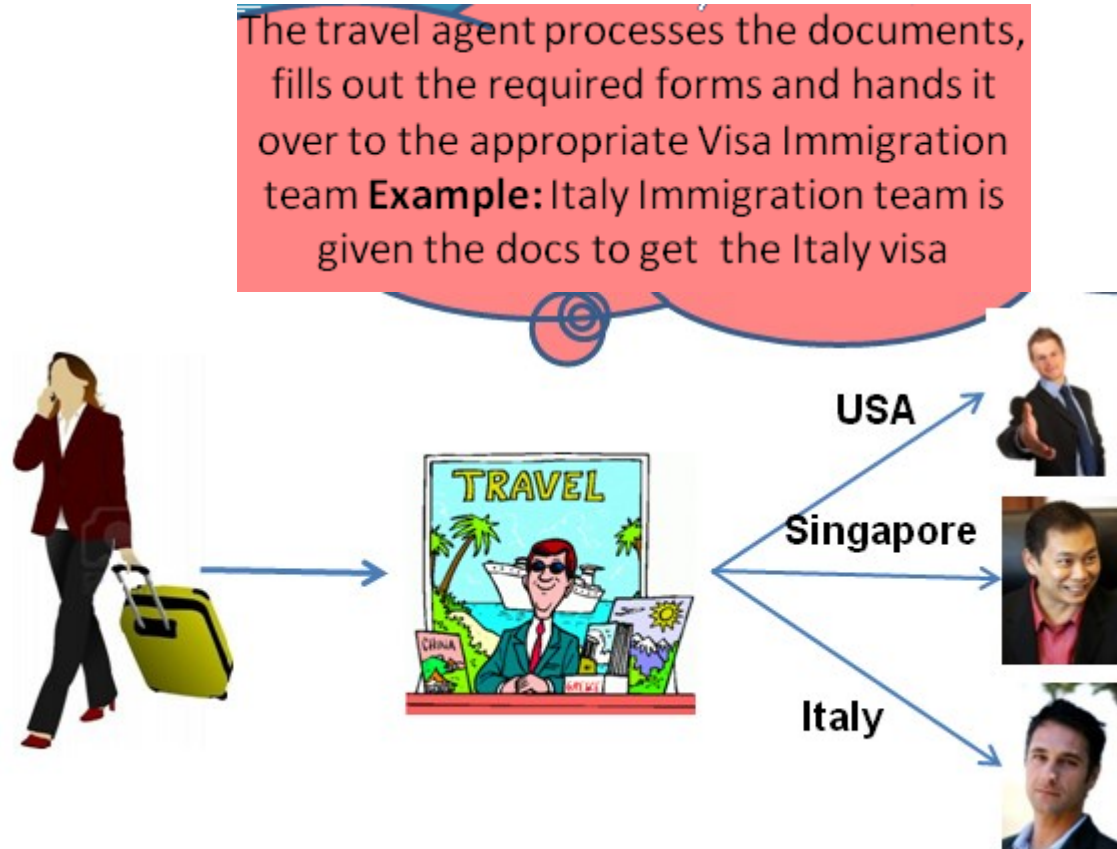
JDBC – Real World Analogy



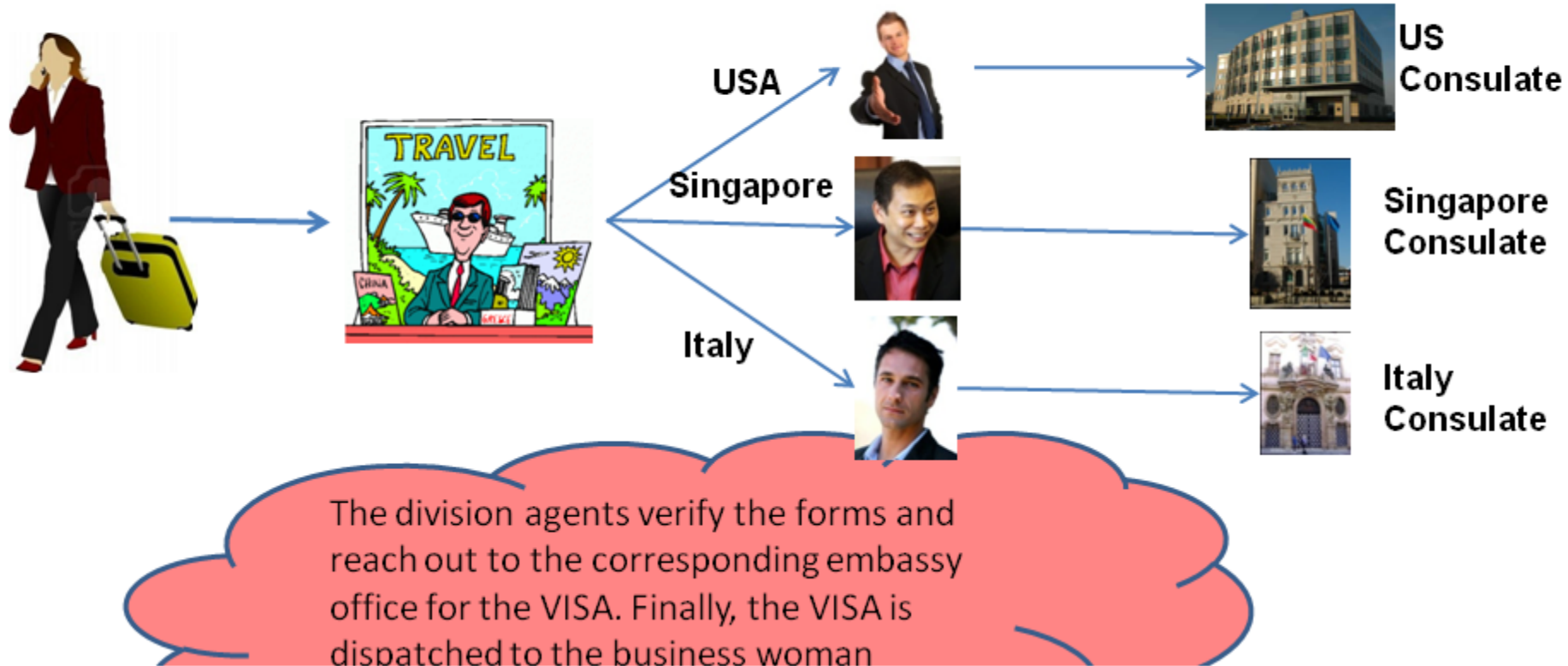
She approaches a travel agent who collects all the required information and documents from the business woman.



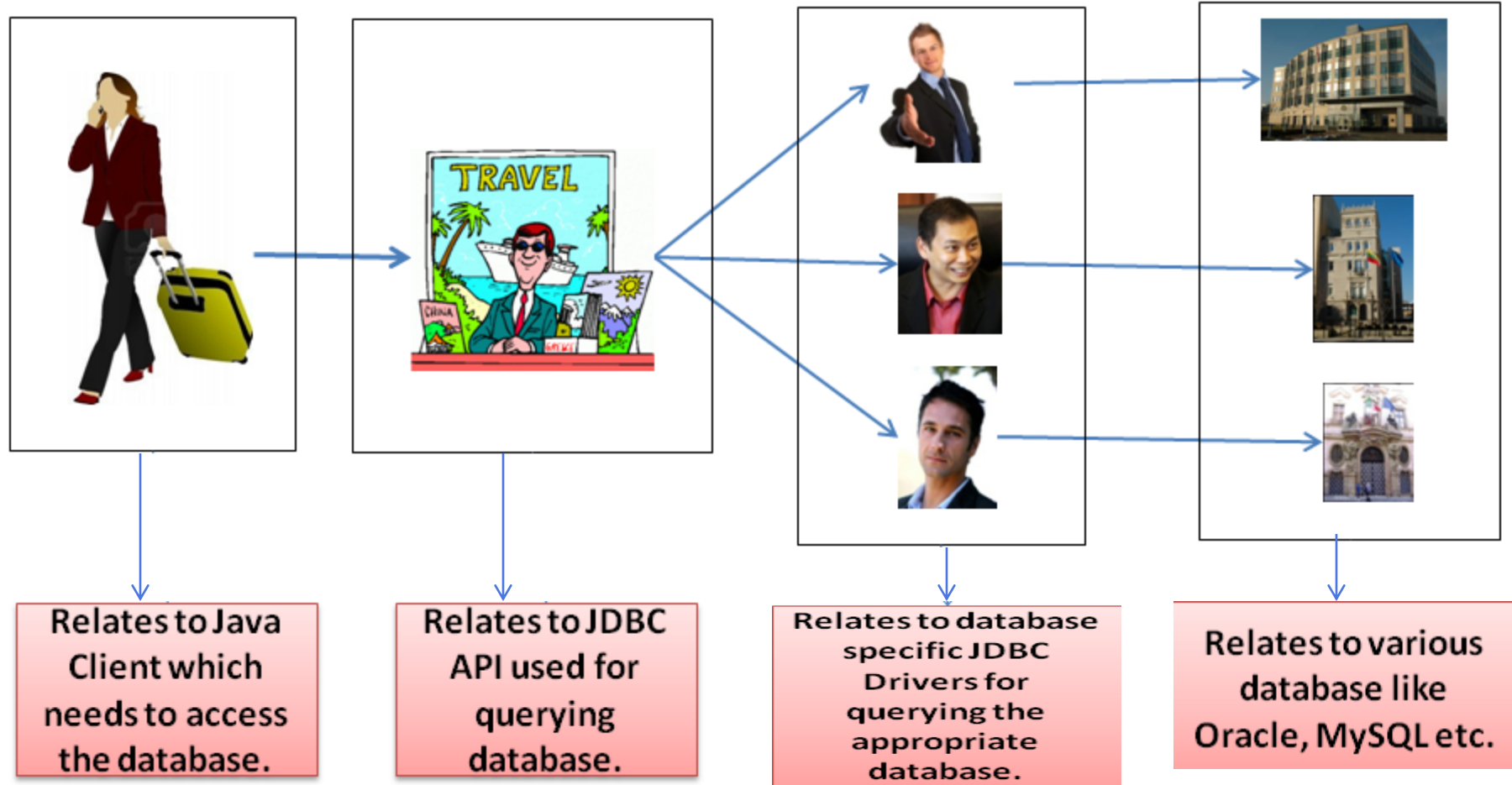
JDBC – Real World Analogy



JDBC – Real World Analogy

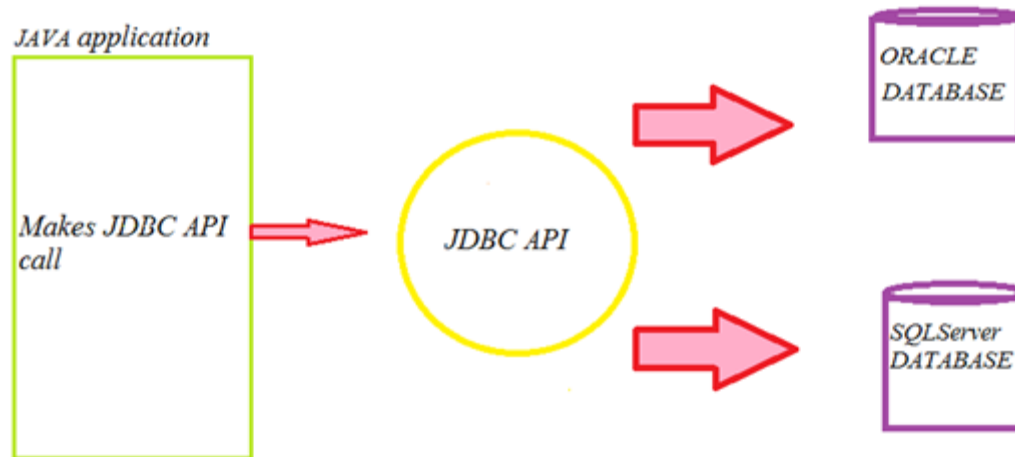


Analogy between JDBC and VISA



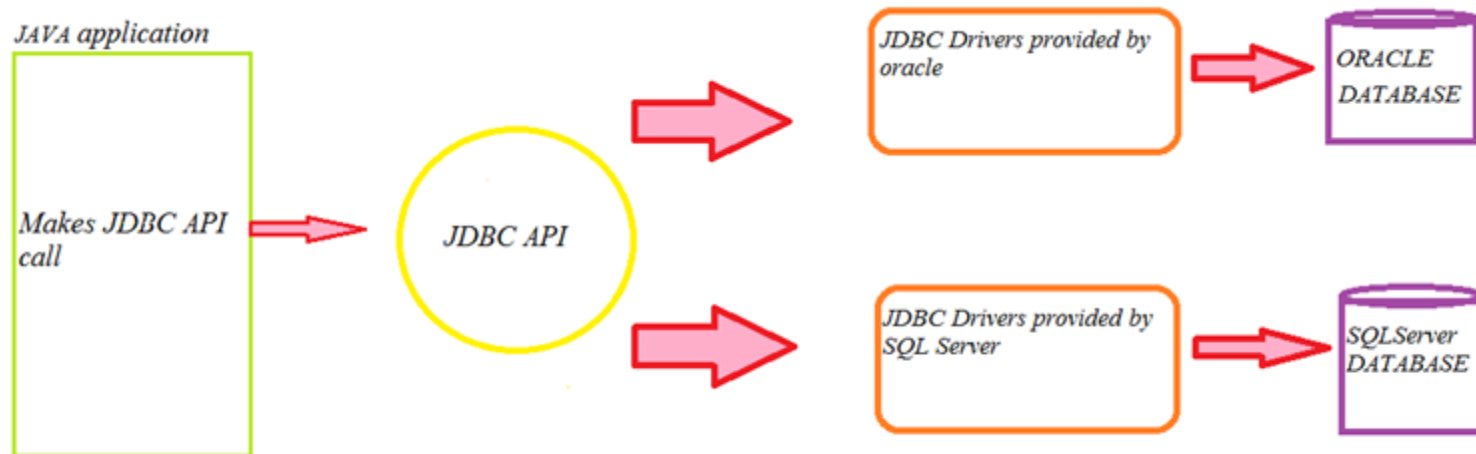
What is JDBC?

An API to access database



JDBC Drivers

A JDBC Component that enables the Java Application to interact with Database



The Drivers are available as .class files in a .jar file



JDBC Steps

1. Register the driver

For MySQL :

```
Class.forName("com.mysql.jdbc.Driver");
```

For Oracle :

```
Class.forName("oracle.jdbc.driver.OracleDriver");
```

localhost:3306 can be given

2. Establish the Connection to database

For MySQL :

```
Connection conn= DriverManager.getConnection(  
    "jdbc:mysql://localhost/library","root","root");
```

API Database Servername Databasename Username password

For Oracle :

```
Connection conn = DriverManager.getConnection(  
    "jdbc:oracle:thin:@localhost:1521:XE", "sagar","sagar");
```

API • Database Servername Port • Service Username password •



JDBC Steps ...

3. Create the Statement / PreparedStatement objects

```
Statement stmt=conn.createStatement(); // returns ResultSet object
```

- 4a. If the statement is Select – use ResultSet object

```
ResultSet rs=stmt.executeQuery("select * from emp");
```

- 4b. If the statement is DML statement – use executeUpdate() method

```
int rowCount=stmt.executeUpdate("delete from emp where id=7632");
```

```
System.out.println(rowCount+" records affected");
```

5. Close resultset object, statement and connection object

```
rs.close();  
stmt.close();  
conn.close();
```



JDBC Prepared Statements

Used to execute parameterized queries.

Ex :

```
PreparedStatement stmt=con.prepareStatement(
    "insert into Dept values(?,?,?)");
stmt.setInt(1,50);//1 specifies the first parameter in the query
stmt.setString(2, "Logistics");
stmt.setString(3, "Las Vegas");

int rowCount=stmt.executeUpdate();
```



Statement vs PreparedStatement

RDBMS handles a JDBC / SQL query in four steps:

1. Parse the incoming SQL query – JDBC format to SQL format
2. Compile the SQL query
3. Plan/optimize the data acquisition path – physical files
4. Execute the optimized query / acquire and return data

Statement object performs all the 4 steps.

Pre-compilation and DB-side caching of the SQL statement leads to overall faster execution

PreparedStatement will pre executes 1-3 steps (pre compilation).



JDBC Transactions

JDBC allows SQL statements to be grouped together into a single transaction

Transaction control is performed by the Connection object

Ex:

```
conn.rollback();
```

```
conn.commit();
```



ResultSetMetadata

- It represents an object that can be used to get information about the types and properties of the columns in a [ResultSet](#) object.
- Example:
 - `ResultSetMetaData rsmd = rs.getMetaData();`
 - `int cols = rsmd.getColumnCount();`
 - `rsmd.getColumnName(1);`
 - `rsmd.getColumnTypeName(1);`



