JDBC

JDBC Architecture

Mrs. Apurwa Barve
(IMCC)

- JDBC stands for Java Database Connectivity

- It is an API provided by Java to enable database independent connectivity between Java applications and the underlying databases.

- JDBC is a specification that provides a complete set of interfaces which can be used for achieving the following tasks :

    a. Making a connection to a database

    b. Creating statements

    c. Executing those statements / queries in the database

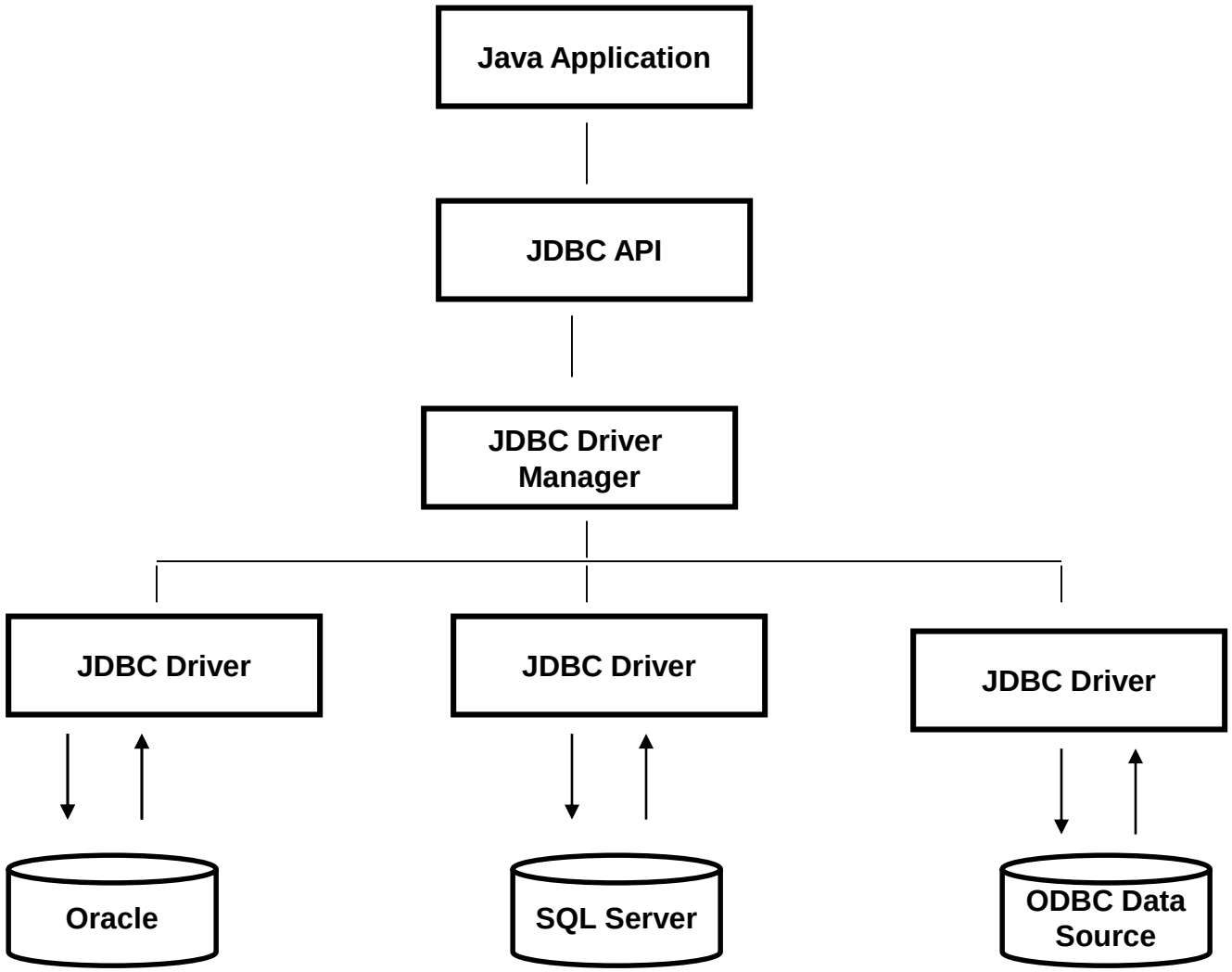
    d. Viewing the results or modifying those results.

- For executing each of the above tasks JDBC API has various inbuilt classes and methods.

JDBC Architecture

Mrs. Apurwa Barve
(IMCC)

- Consider,



JDBC Architecture

Mrs. Apurwa Barve
(IMCC)

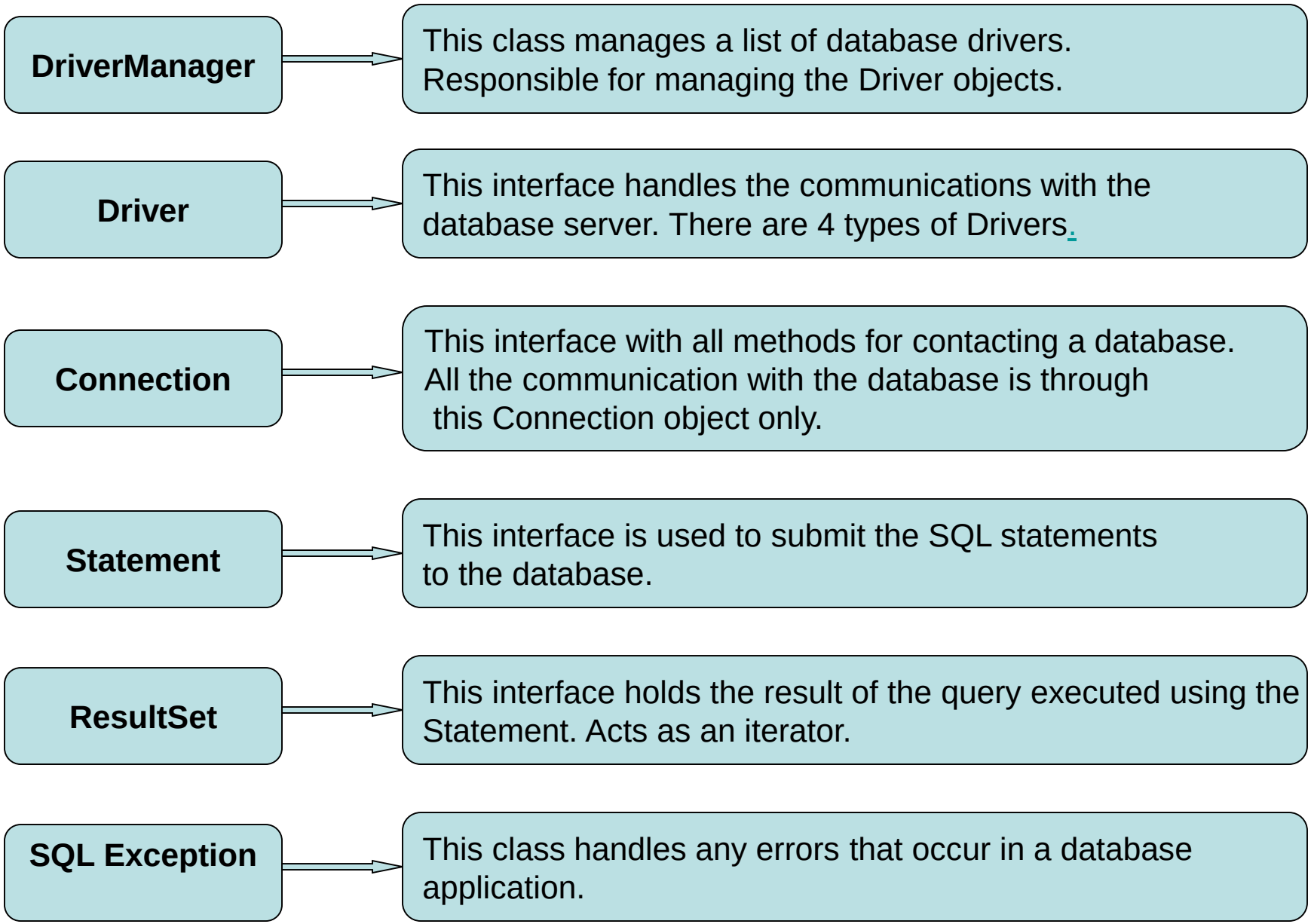
- The architecture consists of 2 layers, viz :

a. **JDBC API:** This provides the application-to-JDBC Manager connection.

b. **JDBC Driver API:** This supports the JDBC Manager-to-Driver Connection.
It ensures that the correct driver is used to access each data source. The driver manager is capable of supporting multiple concurrent drivers connected to multiple heterogeneous databases.

JDBC Interface and classes

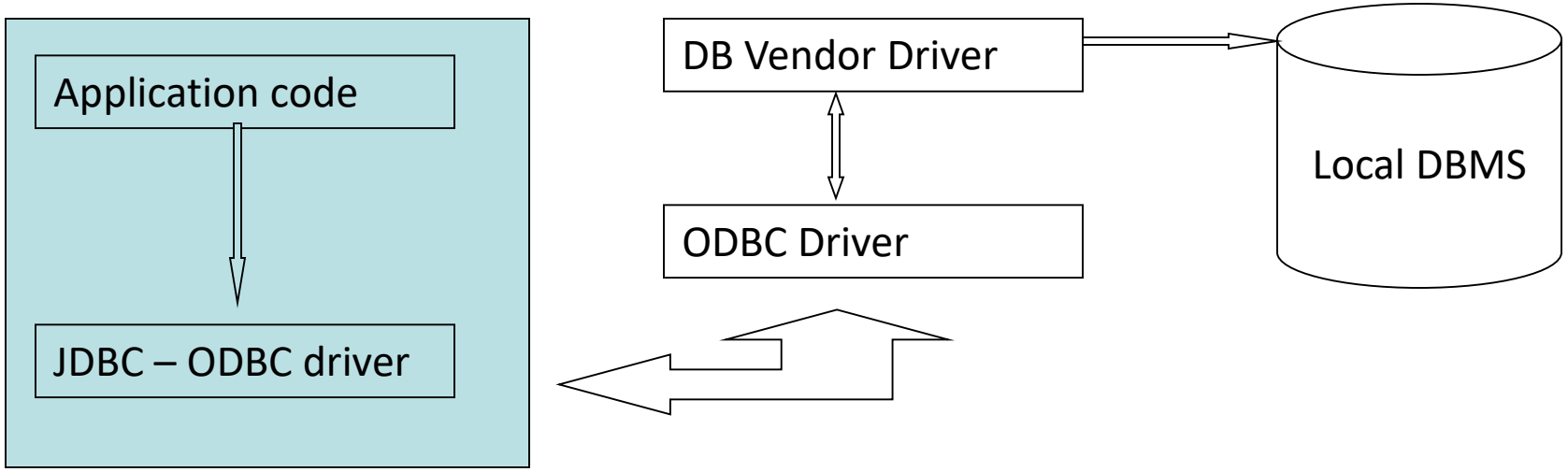
Mrs. Apurwa Barve
(IMCC)



JDBC – Types of drivers

Mrs. Apurwa Barve
(IMCC)

Type 1: The JDBC-ODBC bridge

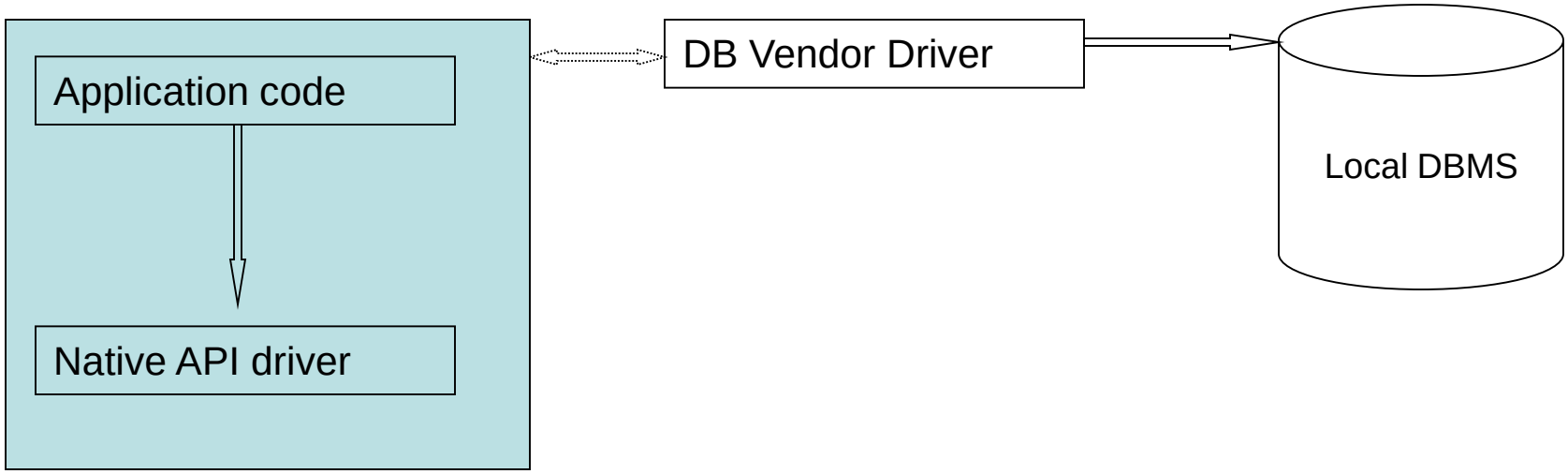


- Converts JDBC method calls into ODBC function calls.
- Requires configuring on your system a Data Source Name (DSN) that represents the target database.
- Recommended only for experimental use.

JDBC – Types of drivers

Mrs. Apurwa Barve
(IMCC)

Type 2: The Native-API Driver

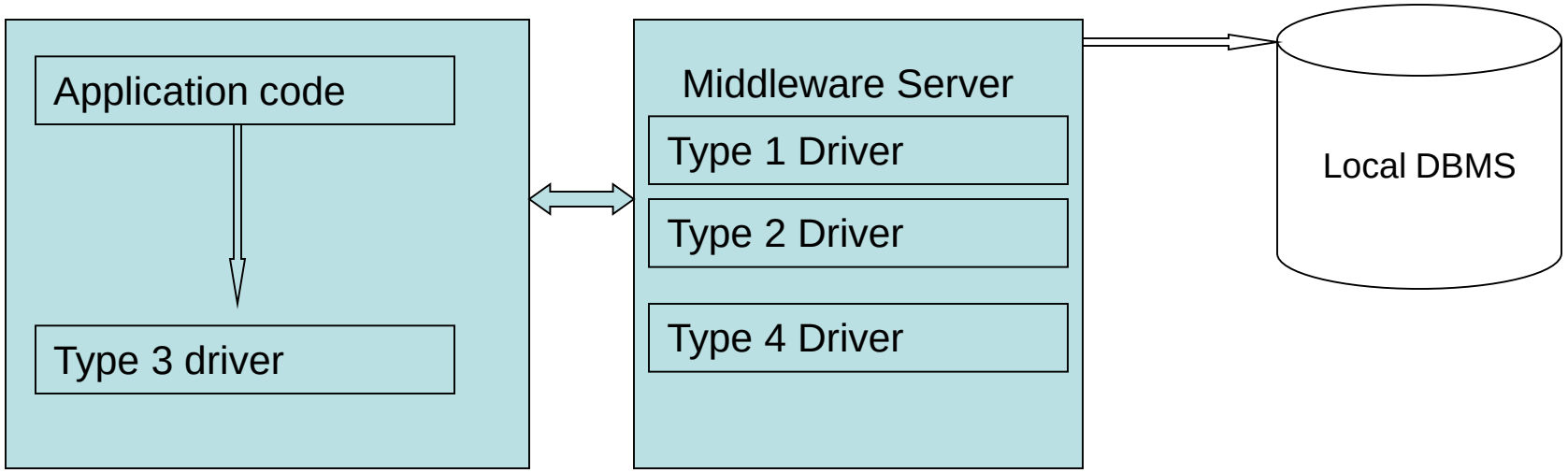


- Converts JDBC method calls into native C/C++ api function calls.
- Provided by the database and need to be installed on each client machine.
- Recommended when Type3 and Type4 drivers are not available.

JDBC – Types of drivers

Mrs. Apurwa Barve
(IMCC)

Type 3: The JDBC-Net pure Java Driver

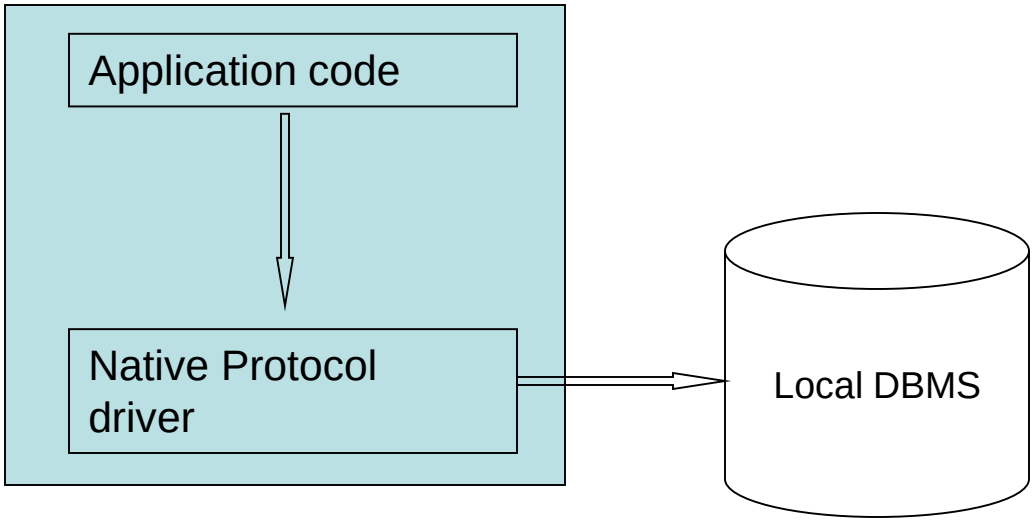


- The JDBC clients use standard network sockets to communicate with a middleware application server.
- Extremely flexible, since it requires no code installed on the client and a single driver can actually provide access to multiple databases.
- Recommended to be used when the application talks to more than one databases.

JDBC – Types of drivers

Mrs. Apurwa Barve
(IMCC)

Type 4: The Native-Protocol Driver



- Communicates directly with vendor's database through socket connection
- Completely written in Java and is hence platform independent. It is installed inside the Java Virtual Machine of the client**.**
- Does not have the overhead of conversion of calls into ODBC or database API calls.
- Better performing than other drivers.
- Very flexible and can be downloaded dynamically.

JDBC – Statement and its types

Mrs. Apurwa Barve
(IMCC)

- **Statement :**

  - ✓ This interface will be used when we want to execute simple queries. This executes static SQL statements at runtime. This interface can not accept parameters.
  - ✓ One of three methods can be used to execute SQL statements :

    a. **boolean execute(String SQL)**: The String parameter is the SQL query which can be formed before and passed to the execute function. It returns true if a resultset is found else it returns false. Can be used to execute typical DDL type SQL statements or dynamic statements.

    b. **int executeUpdate(String SQL)** : Returns the numbers of rows affected by the execution of the SQL statement like INSERT, UPDATE, or DELETE statement.

    c. **ResultSet executeQuery(String SQL)** : Returns a ResultSet object.
    Can be used with a SELECT statement.

JDBC – Statement and its types

Mrs. Apurwa Barve
(IMCC)

- **Prepared Statement :**
  - ✓ This interface extends Statement interface.
  - ✓ It is possible to pass parameters dynamically to it.
  - ✓ All parameters in JDBC are represented by the **?** symbol, which is known as the parameter marker. It is mandatory to supply values to all these parameters for the query to execute.
  - ✓ The **setXXX()** methods bind values to the parameters, where **XXX** represents the Java data type of the value you wish to bind to the input parameter.
  - ✓ Each parameter marker is referred to by its ordinal position. The first marker represents position 1, the next position 2, and so forth.
  - ✓ Can be executed in the same way as Statement object is executed.

JDBC – Statement and its types

Mrs. Apurwa Barve
(IMCC)

- **Callable Statement :**

  - ✓ Would be used to execute a call to a database stored procedure.
  - ✓ Refer CallableStmtDemo.java, Employee.java and PrpStatementDemo.java to understand how different statements are written.

Thank you.