

✱ ⊖ HD 4G ▬ 🔋 87%

The diagram consists of two circles. The left circle is light pink with a dark blue border and contains the text 'AC' in dark blue. The right circle is light green with a dark green border and contains the text 'AP' in dark green, with a small white icon of a person with arms raised in the bottom right corner.


✱ ⊖ HD 4G 87%



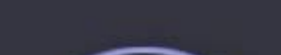
✱ ⊖ HD 4G ▬ 🔋 87%

Two circular icons are shown: a pink circle with the letters 'PB' in brown, and a light green circle with the letters 'AP' in dark green. The 'AP' circle also features a small white microphone icon with a diagonal line through it, indicating audio recording or a specific function.

✱ ⊖ HD 4G ▴ ▾ 🔋 87%



✱ ⊖ HD 4G ▬ 🔋 87%



SD

AP

✱ ⊖ HD 4G ▬ 🔋 87%

The diagram shows two circular icons. The left icon is pink with a blue border and contains the text 'PM'. The right icon is green with a blue border and contains the text 'AP' and a small white icon of a person holding a microphone.

✱ ⊖ HD 4G ▴ ▾ 🔋 87%

The diagram consists of two circles side-by-side. The left circle is pink with a dark blue border and contains the text 'PB' in dark blue. The right circle is green with a dark blue border and contains the text 'AP' in dark blue. A microphone icon is positioned at the bottom right of the green circle.



3-tier model for d/b access

Outline

Introduction to JDBC

JDBC Architecture

2-tier model

3-tier model

Types of JDBC drivers

JDBC APIs

DriverManager class

Connection interface

Statement interface

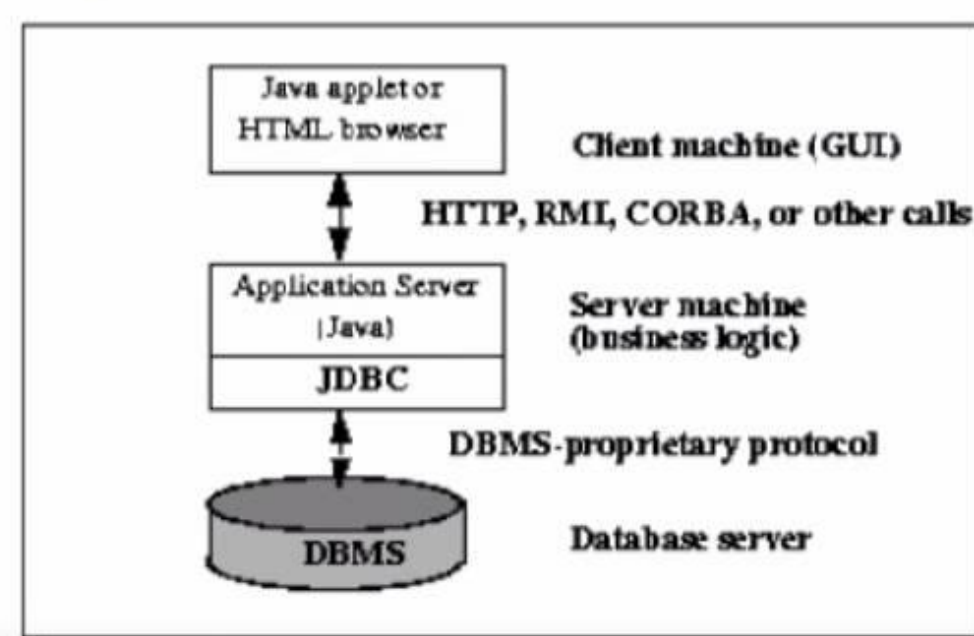
ResultSet interface

Steps to establish connection with d/b using JDBC APIs

PreparedStatement interface

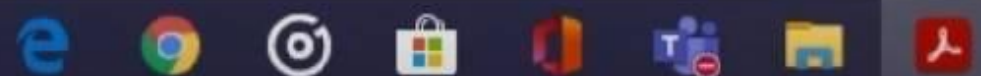
References


- In the 3-tier model, commands are sent to a “middle tier” of services, which then sends the commands to the d/b
- The d/b processes the commands and sends the results back to the middle tier, which then sends them to the user.
- It can connect to different d/bs without changing code
- In this model business logic is clearly separated from d/b
- There are three components of 3-tier model
 - 1 Client-tier
 - 2 Middle-tier
 - 3 Data Source Layer



Atul Chaudhari

Type here to search





Types of JDBC drivers

- Outline
- Introduction to JDBC
- JDBC Architecture
 - 2-tier model
 - 3-tier model
- Types of JDBC drivers**
- JDBC APIs
 - DriverManager class
 - Connection interface
 - Statement interface
 - ResultSet interface
- Steps to establish connection with d/b using JDBC APIs
- PreparedStatement interface
- References

Following are four types of JDBC drivers:


- 1 Type1: JDBC-ODBC bridge
- 2 Type2: Native-API \ Partial Java driver
- 3 Type3: Net-Protocol \ All Java drivers
- 4 Type4: Native-Protocol \ All Java driver \ Thin driver

January 6, 2021 Atul Chaudhari 9/

Atul Chaudhari

Type here to search





Outline

Introduction to JDBC

JDBC Architecture

2-tier model

3-tier model

Types of JDBC drivers

JDBC APIs

DriverManager class

Connection interface

Statement interface

ResultSet interface

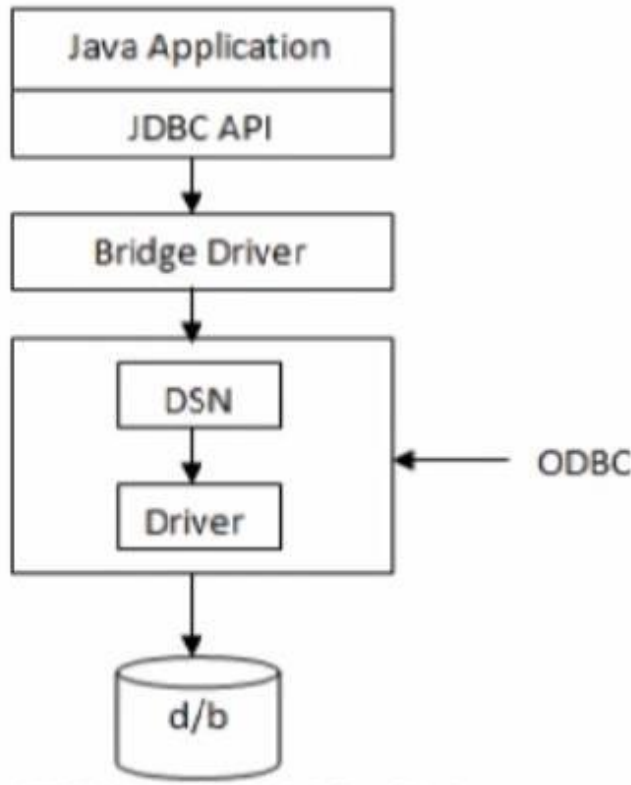
Steps to establish connection with d/b using JDBC APIs

PreparedStatement interface

References

Type1: JDBC-ODBC bridge

- JDBC-ODBC Bridge is a Type1 JDBC driver.
- Type 1 JDBC driver converts JDBC operation into C language ODBC API, ODBC calls are then pass to appropriate ODBC driver.



The diagram illustrates the architecture of a JDBC-ODBC Bridge Driver. It shows a 'Java Application' box containing a 'JDBC API' box. An arrow points from the 'JDBC API' box to a 'Bridge Driver' box. Below the 'Bridge Driver' box is a larger box containing a 'DSN' box and a 'Driver' box, with an arrow pointing from 'DSN' to 'Driver'. An arrow labeled 'ODBC' points from the right into this larger box. Finally, an arrow points from the 'Driver' box to a 'd/b' (database) cylinder at the bottom.

Fig: JDBC-ODBC Bridge Driver

January 6, 2021

Atul Chaudhari

10/

Atul Chaudhari

Type here to search



PB

AP



Type2: Native-API \ Partial Java driver

Outline

Introduction to JDBC

JDBC

Architecture

2-tier model

3-tier model

Types of JDBC drivers

JDBC APIs

DriverManager class

Connection interface

Statement interface

ResultSet interface

Steps to establish connection with d/b using JDBC APIs

PreparedStatement interface

References

- The Native API driver uses the client-side libraries of the database.
- The driver converts JDBC method calls into native C/C++ API calls, which are unique to the database.
- These drivers are typically provided by the database vendors
- The Native driver needs to be installed on the each client machine.
- The Vendor client library needs to be installed on client machine

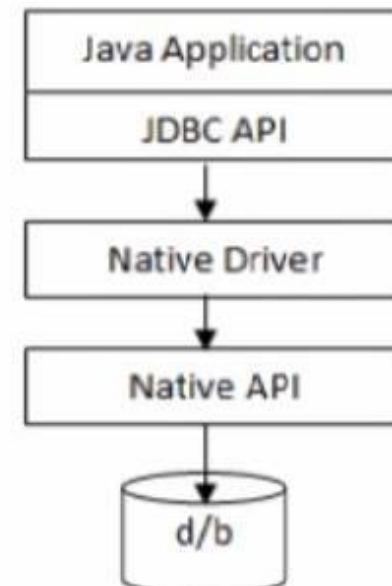


Fig: Native API / Partly Java Driver

January 6, 2021

Atul Chaudhari

11/

Atul Chaudhari

Type here to search





Type3: Net-Protocol \ All Java drivers

Outline

Introduction to JDBC

JDBC Architecture
2-tier model
3-tier model

Types of JDBC drivers

JDBC APIs

DriverManager class
Connection interface
Statement interface
ResultSet interface

Steps to establish connection with d/b using JDBC APIs

PreparedStatement interface

References

- These drivers can only be implemented with three tier architecture.
- The Type3 driver uses middleware (application server) that converts JDBC calls directly or indirectly into the vendor-specific database protocol.
- It is fully written in java.

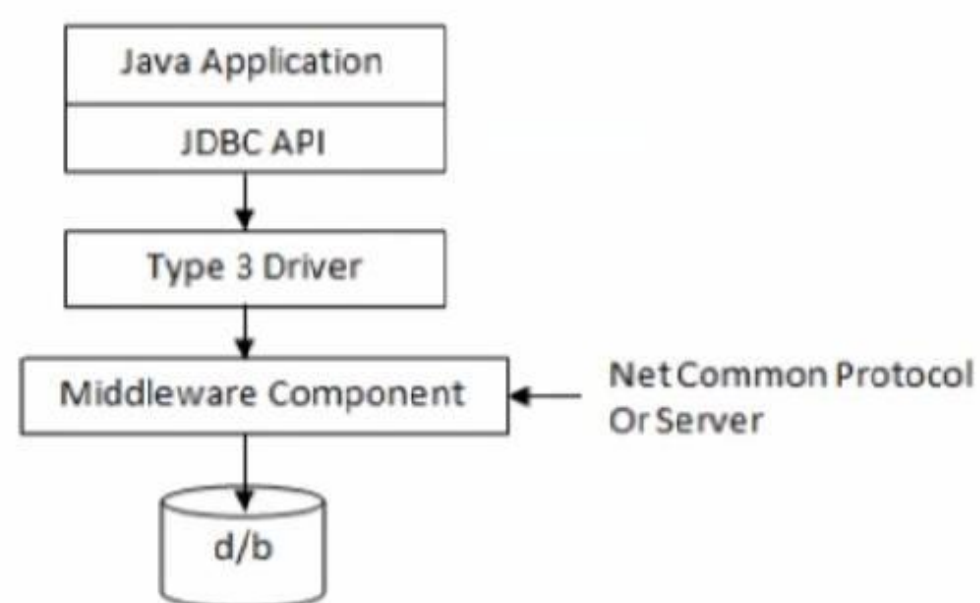


Fig: Net Protocol / All Java Drivers

January 6, 2021


Atul Chaudhari

12/

Atul Chaudhari

Type here to search





Outline

Introduction to JDBC

JDBC Architecture

2-tier model

3-tier model

Types of JDBC drivers

JDBC APIs

DriverManager class

Connection interface

Statement interface

ResultSet interface

Steps to establish connection with d/b using JDBC APIs

PreparedStatement interface

References

Type4: Native-Protocol \ All Java driver\Thin driver

- The thin driver converts JDBC calls directly into the vendor-specific database protocol.
- It is fully written in Java language.

Java Application

JDBC API

Thin/Type 4 Drivers

Native Protocol

d/b

Fig: Native Protocol / All Java Drivers


Atul Chaudhari

Type here to search



PB

AP

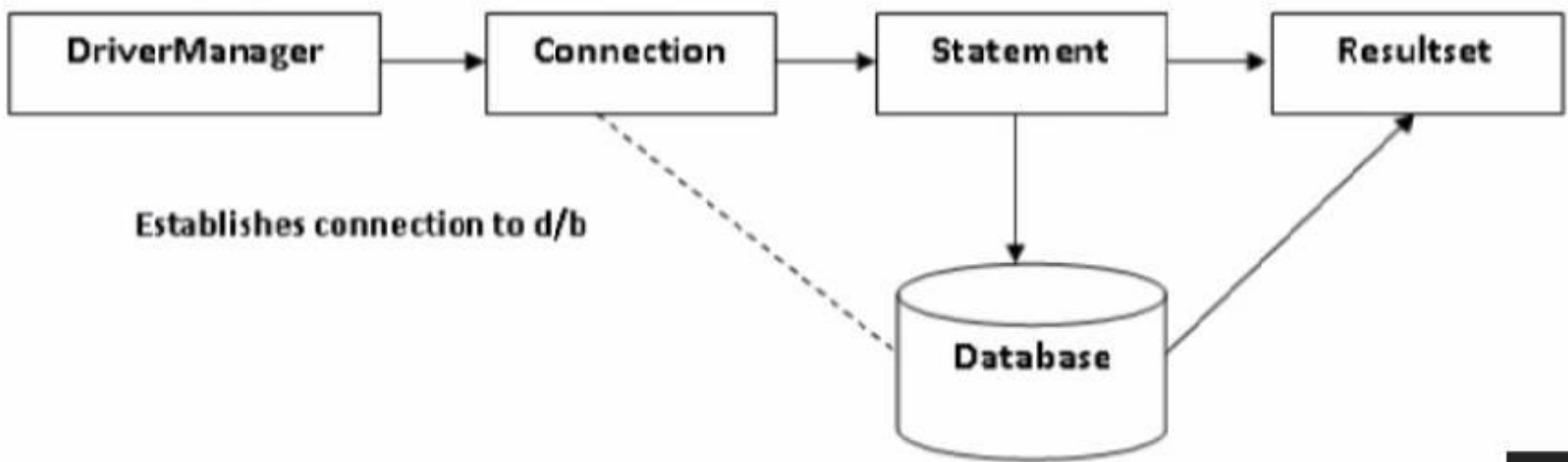


JDBC APIs

- Outline
- Introduction to JDBC
- JDBC Architecture
 - 2-tier model
 - 3-tier model
- Types of JDBC drivers
- JDBC APIs
 - DriverManager class
 - Connection interface
 - Statement interface
 - ResultSet interface
- Steps to establish connection with d/b using JDBC APIs
- PreparedStatement interface
- References

The JDBC API provides the following interfaces and classes:

- 1 DriverManager
- 2 Connection
- 3 Statement
- 4 ResultSet



```
graph LR; DM[DriverManager] --> C[Connection]; C --> S[Statement]; S --> RS[Resultset]; C -.-> DB[(Database)]; S --> DB; DB --> RS;
```

Establishes connection to d/b

January 6, 2021

Atul Chaudhari

14/

Atul Chaudhari

AC

AP



DriverManager class

Outline

Introduction to JDBC

JDBC

Architecture

2-tier model

3-tier model

Types of JDBC drivers

JDBC APIs

DriverManager class

Connection interface

Statement interface

ResultSet interface

Steps to establish connection with d/b using JDBC APIs

PreparedStatement interface

References

- DriverManager is a class in the Java Software Development Kit (J2SDK).
- DriverManager manages the set of JDBC drivers that are available for an application to use.
- Applications can use multiple JDBC drivers concurrently if necessary.
- By passing a URL for a specific JDBC driver to the DriverManager, the application informs the DriverManager about which type of JDBC connection should be returned to the application.
- DriverManager must be made aware of the available JDBC drivers so it can hand out connections.
- By making a call to the `Class.forName(String classname)` method, it loads a class into the running JVM based on its string name that is passed into the method.
- DriverManager object creates Connection object.

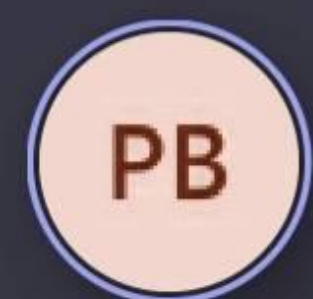
January 6, 2021

Atul Chaudhari

15/

Atul Chaudhari

Type here to search





Methods of DriverManager class

Outline

Introduction to JDBC

JDBC

Architecture

2-tier model

3-tier model

Types of

JDBC drivers

JDBC APIs

DriverManager class

Connection interface

Statement interface

ResultSet interface

Steps to

establish

connection

with d/b using

JDBC APIs

PreparedStatement interface

References

Following are the methods of DriverManager class:

- `Connection getConnection(String url)`: Attempt to establish a connection to the given database URL.
- `Connection getConnection(String url, String userName, String password)`: Attempt to establish a connection to the given database URL, username and password.

January 6, 2021

Atul Chaudhari

16/


Atul Chaudhari

Type here to search



12:11:32

HD 4G 86%



Connection interface

- Outline
- Introduction to JDBC
- JDBC Architecture
 - 2-tier model
 - 3-tier model
- Types of JDBC drivers
- JDBC APIs
 - DriverManager class
 - Connection interface**
 - Statement interface
 - ResultSet interface
- Steps to establish connection with d/b using JDBC APIs
- PreparedStatement interface
- References

- Connection is an interface.
- It is a session between an application and a database.
- Connection interface helps by providing the different methods for transactions and error handling.
- Connection object creates Statement object.

January 6, 2021

Atul Chaudhari

17/

Atul Chaudhari

Type here to search





Methods of Connection interface

Outline

Introduction to JDBC

JDBC

Architecture

2-tier model

3-tier model

Types of

JDBC drivers

JDBC APIs

DriverManager

class

Connection

interface

Statement

interface

ResultSet

interface

Steps to

establish

connection

with d/b using

JDBC APIs

PreparedStatement

interface

References

Following are the methods of Connection interface:

- **void commit():** Makes all changes made since the previous commit/rollback permanent.
- **void rollback():** Undoes all changes made in the current transaction.
- **void close():** Releases this Connection object's database and JDBC resources immediately instead of waiting for them to be automatically released.
- **Statement createStatement():** Creates a Statement object for sending SQL statements to the database.
- **PreparedStatement prepareStatement(String sql):** Creates a PreparedStatement object for sending parameterized SQL statements to the database.

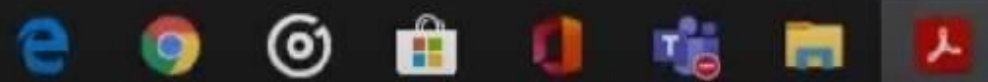
January 6, 2021

Atul Chaudhari

18/

Atul Chaudhari

Type here to search





Statement interface and its Methods

Outline

Introduction to JDBC

JDBC

Architecture

2-tier model

3-tier model

Types of

JDBC drivers

JDBC APIs

DriverManager

class

Connection

interface

Statement

interface

ResultSet

interface

Steps to

establish

connection

with d/b using

JDBC APIs

PreparedStatement

interface

References

- Statement is an interface.
- It is used for executing a static SQL statement on database.

Following are the methods of Statement interface:

- **ResultSet executeQuery(String qry):** It is used to execute given SQL statement, which returns single ResultSet object.

Generally it is used to execute SELECT query on database

- **int executeUpdate(String qry):** It is used to execute DML and DDL statements on database

In case of DML (INSERT, UPDATE, DELETE) statements it returns number of rows affected and

In case of DDL (CREATE) statement it returns 0

- **boolean execute(String qry):** Executes the given SQL statement, which may return multiple results.

January 6, 2021

Atul Chaudhari

19/

Atul Chaudhari

Type here to search





ResultSet interface

Outline

Introduction to JDBC

JDBC

Architecture

2-tier model

3-tier model

Types of JDBC drivers

JDBC APIs

DriverManager class

Connection interface

Statement interface

ResultSet interface

PreparedStatement interface

References

Steps to establish connection with d/b using JDBC APIs

PreparedStatement interface

References

- ResultSet is an interface.
- Resultset object contains result of SQL queries.
- Resultset object maintains cursor positioning to its current row of data.
- Initially the cursor is positioning before first row.
- ResultSet object provides corresponding getType() method for each data type to retrieve column value.
- Each method can be used with column name or by its ordinal position.

Syntax:

```
type rs.getType(String columnName)
```

```
type rs.getType(int columnPosition)
```

Here,

type: varies with SQL datatype.

rs: is object of ResultSet.

columnName: name of column in a table.

columnPosition: position of column in a table.

January 6, 2021


Atul Chaudhari

20/

Atul Chaudhari

Type here to search





Outline

Introduction to JDBC

JDBC Architecture

2-tier model

3-tier model

Types of JDBC drivers

JDBC APIs

DriverManager class

Connection interface

Statement interface

ResultSet interface

Steps to establish connection with d/b using JDBC APIs

PreparedStatement interface

References

Methods of ResultSet

Following are methods of resultset object to retrieve column value:

SQL Datatype	JDBC/Java Datatype	getType() Method
VARCHAR	java.lang.String	getString()
CHAR	java.lang.String	getString()
BIT	boolean	getBoolean()
NUMERIC	java.math.BigDecimal	getBigDecimal()
TINYINT	byte	getByte()
SMALLINT	short	getShort()
INTEGER	int	getInt()
BIGINT	long	getLong()
REAL	float	getFloat()
FLOAT	float	getFloat()
DOUBLE	double	getDouble()
DATE	java.sql.Date	getDate()
TIME	java.sql.Time	getTime()
CLOB	java.sql.Clob	
BLOB	java.sql.Blob	

Atul Chaudhari

PB

AP



Navigation Methods of ResultSet

Outline

Introduction to JDBC

JDBC Architecture
2-tier model
3-tier model

Types of JDBC drivers

JDBC APIs
DriverManager class
Connection interface
Statement interface
ResultSet interface

Steps to establish connection with d/b using JDBC APIs

PreparedStatement interface

References

Following are navigation methods of resultset object:

- **boolean first()**: Moves cursor to the first record in ResultSet. If first record is available then it returns true otherwise false.
- **boolean last()**: Moves cursor to the last record in ResultSet. If last record is available then it returns true otherwise false.
- **boolean next()**: Moves cursor to the next record in ResultSet. If next record is available then it returns true otherwise false.
- **boolean previous()**: Moves cursor to the previous record in ResultSet. If previous record is available then it returns true otherwise false.
- **void beforeFirst()**: Moves the cursor to the front of this ResultSet object, just before the first row.
- **void afterLast()**: Moves the cursor to the end of this ResultSet object, just after the last row.
- **boolean isFirst()**: Retrieves whether the cursor is on the first row of this ResultSet object. Returns true if the cursor is on the first row; false otherwise.
- **boolean isLast()**: Retrieves whether the cursor is on the last row of this ResultSet object.

January 6, 2021

Atul Chaudhari

22/

Atul Chaudhari

Type here to search





Navigation Methods of ResultSet contd...

Outline

Introduction to JDBC

JDBC Architecture

2-tier model

3-tier model

Types of JDBC drivers

JDBC APIs

DriverManager class

Connection interface

Statement interface

ResultSet interface

ResultSet interface

Steps to establish connection with d/b using JDBC APIs

PreparedStatement interface

PreparedStatement interface

PreparedStatement interface

PreparedStatement interface

PreparedStatement interface

PreparedStatement interface

PreparedStatement interface

PreparedStatement interface

PreparedStatement interface

PreparedStatement interface

PreparedStatement interface

PreparedStatement interface

PreparedStatement interface

PreparedStatement interface

PreparedStatement interface

PreparedStatement interface

PreparedStatement interface

PreparedStatement interface

PreparedStatement interface

PreparedStatement interface

PreparedStatement interface

PreparedStatement interface

PreparedStatement interface

PreparedStatement interface

PreparedStatement interface

PreparedStatement interface

PreparedStatement interface

PreparedStatement interface

PreparedStatement interface

PreparedStatement interface

PreparedStatement interface

PreparedStatement interface

PreparedStatement interface

PreparedStatement interface

PreparedStatement interface

PreparedStatement interface

PreparedStatement interface

PreparedStatement interface

PreparedStatement interface

PreparedStatement interface

PreparedStatement interface

PreparedStatement interface

Following are navigation methods of resultset object:

- **boolean isBeforeFirst():** Retrieves whether the cursor is before the first row in this ResultSet object.
Returns true if the cursor is before the first row; false otherwise.
- **boolean isAfterLast():** Retrieves whether the cursor is after the last row in this ResultSet object.
Returns true if the cursor is after the last row; false otherwise.
- **boolean absolute(int rn):** Moves the cursor to the given row number in this ResultSet object.
If the row number is positive, the cursor moves to the given row number with respect to the beginning of the result set.
If the given row number is negative, the cursor moves to an absolute row position with respect to the end of the result set.
If the row number specified is zero, the cursor is moved to before the first row.
Returns true if the cursor is on a row; false otherwise
- **boolean relative(int rows):** Moves the cursor a relative number of rows, either positive or negative.
Returns true if the cursor is on a row; false otherwise

January 6, 2021


Atul Chaudhari

23/

Atul Chaudhari

Type here to search





Types of ResultSet

Outline

Introduction to JDBC

JDBC Architecture

- 2-tier model
- 3-tier model

Types of JDBC drivers

JDBC APIs

- DriverManager class
- Connection interface
- Statement interface
- ResultSet interface**

Steps to establish connection with d/b using JDBC APIs

PreparedStatement interface

References

Following are three types of resultset:

Type	Description
ResultSet.TYPE_FORWARD_ONLY	The cursor can only move forward in the result set.
ResultSet.TYPE_SCROLL_INSENSITIVE	The cursor can scroll forward and backward, and the result set is not sensitive to changes made by others to the database that occur after the result set was created.
ResultSet.TYPE_SCROLL_SENSITIVE.	The cursor can scroll forward and backward, and the result set is sensitive to changes made by others to the database that occur after the result set was created.

January 6, 2021


Atul Chaudhari

24/

Atul Chaudhari

PB

AP



Concurrency of ResultSet

Outline

Introduction to JDBC

JDBC Architecture

- 2-tier model
- 3-tier model

Types of JDBC drivers

JDBC APIs

- DriverManager class
- Connection interface
- Statement interface
- ResultSet interface**

Steps to establish connection with d/b using JDBC APIs

PreparedStatement interface

References

Following are possible concurrency of resultset:

Concurrency	Description
ResultSet.CONCUR_READ_ONLY	Creates a read-only result set. This is the default
ResultSet.CONCUR_UPDATABLE	Creates an updateable result set.

January 6, 2021

Atul Chaudhari

25/


Atul Chaudhari

PB

AP

12:13:43

HD 4G 85%



Steps to establish connection with d/b using JDBC APIs

- Outline
- Introduction to JDBC
- JDBC Architecture
 - 2-tier model
 - 3-tier model
- Types of JDBC drivers
- JDBC APIs
 - DriverManager class
 - Connection interface
 - Statement interface
 - ResultSet interface
- Steps to establish connection with d/b using JDBC APIs
- PreparedStatement interface
- References


- 1 import java.sql package
- 2 Load and register the JDBC driver
- 3 Define Connection URL
- 4 Establish a connection
- 5 Create a statement object to perform a query
- 6 Execute the Queries
- 7 Process the result
- 8 Close the Connection

January 6, 2021Atul Chaudhari26/

Atul Chaudhari

Type here to search





PreparedStatement interface

- The PreparedStatement interface is a subinterface of Statement.
- It is used to execute parameterized query.
Example: "insert into stud values(?, ?, ?)";
- It is used to execute pre-compiled SQL statement.
- It provides different types of setter methods to set the input parameters for the query.
- The prepareStatement() method of Connection interface is used to return the object of PreparedStatement.

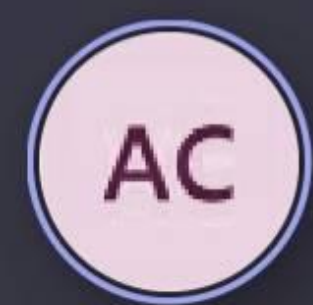
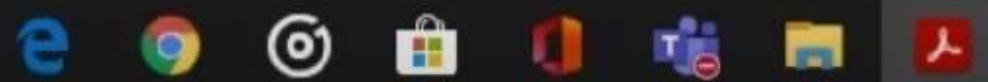
Syntax:


```
PreparedStatement prepareStatement(String query)
```

January 6, 2021 Atul Chaudhari 27/

Atul Chaudhari

Type here to search





Outline

Introduction to JDBC

JDBC Architecture

- 2-tier model
- 3-tier model

Types of JDBC drivers

JDBC APIs

- DriverManager class
- Connection interface
- Statement interface
- ResultSet interface

Steps to establish connection with d/b using JDBC APIs

PreparedStatement interface



References


Methods of PreparedStatement interface

Following are some important methods of PreparedStatement interface:

Method	Description
void setInt(int paramIndex, int value)	sets the integer value to the given parameter index.
void setLong(int paramIndex, long value)	sets the long value to the given parameter index.
void setFloat(int paramIndex, float value)	sets the float value to the given parameter index.
void setDouble(int paramIndex, double value)	sets the double value to the given parameter index.
void setString(int paramIndex, String value)	sets the String value to the given parameter index.
int executeUpdate()	executes the query. It is used for create, drop, insert, update, delete
ResultSet executeQuery()	executes the select query. It returns an instance of ResultSet.

Atul Chaudhari





References

Outline

- Introduction to JDBC
- JDBC Architecture
 - 2-tier model
 - 3-tier model
- Types of JDBC drivers
- JDBC APIs
 - DriverManager class
 - Connection interface
 - Statement interface
 - ResultSet interface
- Steps to establish connection with d/b using JDBC APIs
- PreparedStatement interface
- References

H. Schildt, *Complete Reference Java*. McGrawHill Education, New Delhi.

Holzner and Steven, *Java 2 Programming Black Book*. Dreamtech Press, New Delhi.

K. L. Solutions, *Java Sever Programming Tutorial Java EE6 Black Book*. Dreamtech Press, New Delhi.

"Website." [Online]. Available: <https://www.javatpoint.com/java-tutorial>

"Website." [Online]. Available: <https://www.tutorialspoint.com>

January 6, 2021


Atul Chaudhari


29/

Atul Chaudhari

Type here to search







✱ ⊖ HD 4G ▴ ▾ 🔋 85%

AC

AP