**6. ResultSet Interface**

1. **What is ResultSet in JDBC?**

ResultSet in JDBC is an interface that represents the result set of a database query. When you execute a SQL statement that retrieves data, such as a SELECT query, the data returned by the database is stored in a ResultSet object. This object allows you to access and manipulate the data row by row.

|  |
| --- |
| Java ResultSet Tutorial | DigitalOcean |

1. **Navigating through ResultSet (first, last, next, previous)**

* **first()**
* Moves the cursor to the first row in the ResultSet.
* Returns true if the ResultSet is not empty; returns false if the ResultSet is empty.

**Example**:

if (rs.first()) {

// Access data from the first row

int id = rs.getInt("id");

String name = rs.getString("name");

System.out.println("First Row - ID: " + id + ", Name: " + name); }

* **last()**
* Moves the cursor to the last row in the ResultSet.
* Returns true if the ResultSet is not empty; returns false if the ResultSet is empty.

**Example**:

if (rs.last()) {

// Access data from the last row

int id = rs.getInt("id");

String name = rs.getString("name");

System.out.println("Last Row - ID: " + id + ", Name: " + name);

}

* **next()**
* Moves the cursor to the next row in the ResultSet.
* Returns true if the new current row is valid; returns false if there are no more rows.

**Example**:

while (rs.next()) {

// Access data from the current row

int id = rs.getInt("id");

String name = rs.getString("name");

System.out.println("ID: " + id + ", Name: " + name);

}

* **previous()**
* Moves the cursor to the previous row in the ResultSet.
* Returns true if the new current row is valid; returns false if the cursor is before the first row.

**Example**:

if (rs.last()) { // Move to last row first

do {

// Access data from the current row

String name = rs.getString("name");

System.out.println("Name: " + name);

} while (rs.previous());

}

1. **Working with ResultSet to retrieve data from SQL queries**

* A ResultSet in Java is an interface that holds the data returned by an SQL query executed through JDBC. It allows you to read the data row by row from a database.
* **Steps to use ResultSet:**
* **Execute a query using Statement or PreparedStatement:**

Eg:- ResultSet rs = stmt.executeQuery("SELECT \* FROM employees");

* **Iterate through the result using rs.next() to move the cursor to the next row:**

Eg:- while (rs.next()) {

int id = rs.getInt("id");

String name = rs.getString("name");

System.out.println(id + " | " + name);

}

* **Close the resources (ResultSet, Statement, and Connection) when done:**

Eg:- rs.close();

stmt.close();

con.close();

* **ResultSet provides methods like getInt(), getString(), and getDouble() to retrieve data of various types from the columns.**