next()

- next() is an inbuilt method present in the ResultSet Interface.
- It is used to check whether the next record is present in the cursor or buffer memory or not.
- It returns Boolean value but not the record.

Syntax:

```
public boolean rs.next() throws SQLException;
```

Drawbacks of Statement Interface

- Whenever we pass the query inside the loop for multiple times of execution, each time the query is sent to database and compliled first, then executed, the result is returned to the program. The problem with this kind of execution is it decreases the performance of the program.
- SQL Injection attack chances are high.

SQL Injection

- In the Req 2 program, the malicious user will not be knowing the input to be given but 1=1 is always true. Hence, the hacker is able to steal the data of all the user.
- This type of SQL injection attacks are possible with Statement Interface.
- To prevent SQL Injection, PreparedStatement Interface can be used.

Ex: Consider the below query:

```
"SELECT * FROM EMP WHERE FNAME=" 'SIDDARTH'
```

As the hacker doesnot know the input to be given, he just injects a true condition inside the query.

```
"SELECT * FROM EMP WHERE FNAME="'unknown' OR 1=1;
```

PUNITH B

PreparedStatement Interface

- PreparedStatement is used to create a platform for executing the SQL queries.
- PreparedStatement pre compile the SQL query for once, and executed multiple times.
- Whenever we are dealing with the user defined values, we can go for PreparedStatment.
- PreparedStatement improves the performance of the program and avoids SQL Injection.
- We can make use of PreparedStatement to store images etc..
- PreparedStatement reference is created by a helper method called as prepareStatement() which is present in the Conenction Interface.
- It is available inside java.sql package.
- We use placeholders to set the user defined values.

Syntax:

```
PreparedStatement ps=con.prepareStatement(String query) throws SQLException;
```

prepareStatment()

- prepareStatement() is used to create the reference object of PreparedStatement.
- It is available inside Connection Interface.
- It takes query as a argument in the form of String.

Note: Incase of PreparedStatement Interface, the execute(),executeUpdate() and executeQuery() which is also present inside PreparedStatement doesnot take query as a argument.

Placeholders

- Placeholders are used to pass the user defined values for the given query.
- These placeholders reserves a place for the data which will be given by the user.
- Placeholders are represented by '?'.
- Each placeholder holds a single data.
- For placeholders, the values are set by using a pre defined method setXXX() which is present in PreparedStatement Interface.

setXXX()

- It is a predefined method present in PreparedStatement Interface.
- It is used to set the value for the placeholders.
- It takes parameter index(placeholder number) and the value(data to be assigned to placeholder) as a argument.
- setXXX() varies according to the type of data to be set for the placeholder.

Syntax:

```
void setXXX(int parameterIndex, XXX value) throws SQLException;
```

JDBC Transaction

- It is a single business unit.
- Multiple queries or transactions are written inside a single program.
- The backbone of JDBC transaction is ACID properties.

commit()

It is a predefined method which is present in Connection Interface.
It is used to permanently store the data inside the database.

```
void commit() throws SQLException;
```

Note: To enable or disable the auto-commit, we use setAutoCommit() which is present in Conenction interface.

```
void setAutoCommit(boolean autoCommit) throws SQLException;
```

rollback()

It is a predefined method which is present in Connection Interface.
It is used to rollout all the transactions until the previously used commit statement.

```
void rollback();
void rollback(Savepoint savepoint_name) throws SQLException;
```

Savepoint

This interface is used to maintain and manage the savepoints.
To set the savepoint,

```
Savepoint setSavepoint(String name) throws SQLException;
String getSavepointName() throws SQLException;
```