

Simple Query

1. Select

```
3 Statement preStat = null;
4 boolean isValidUser = false;
5 String sqlQuery = "Select * from User where userName= "+userName;
6 try {
7     preStat = connection.createStatement();
8     ResultSet resultSet = preStat.executeQuery(sqlQuery);
```

```
3 PreparedStatement preStat = null;
4 boolean isValidUser = false;
5 String sqlQuery = "Select * from User where userName=? ";
6 try {
7     preStat = connection.prepareStatement(sqlQuery);
8     preStat.setObject(1, userName);
9     ResultSet resultSet = preStat.executeQuery();
```

2. Update

```
statement =
connection.createStatement(
ResultSet.TYPE_SCROLL_INSENSITIVE,
ResultSet.CONCUR_UPDATABLE
);
String sql ="update Book set name = '' +book.getName() +", idAuthor = " +book.g
getIdAuthor() +"where id =" +book.getId();
statement.executeUpdate(sql);
```

```
4 String sql = "update Book set name = '?', idAuthor =?where id =?";
5 statement =
6 connection.prepareStatement(
7 sql,
8 ResultSet.TYPE_SCROLL_INSENSITIVE,
9 ResultSet.CONCUR_UPDATABLE
10 );
11 statement.setObject(1, book.getName());
12 statement.setObject(2, book.getIdAuthor());
13 statement.setObject(3, book.getId());
14 statement.executeUpdate();
```

3. Insert

```
3 Statement preStat = null;
String sqlQuery = "Insert into User(id,firstName,lastName,userName,email,password
rd,mobileNo)
values("+userInfo.getId()+","+userInfo.getFirstName()+","+userInfo.getLastName
()+","+userInfo.getUserName()+","+userInfo.getEmail()+","+userInfo.getPassword
()+","+userInfo.getMobileNo()+")";
5 try {
6     preStat = connection.createStatement();
7     preStat.executeUpdate(sqlQuery);
```

```
3 PreparedStatement preStat = null;
String sqlQuery = "Insert into User(id,firstName,lastName,userName,email,password
d,mobileNo)
values(?,?,?,?,?,?)";
5 try {
6     preStat = connection.prepareStatement(sqlQuery);
7     preStat.setObject(1, userInfo.getId());
8     preStat.setObject(2, userInfo.getFirstName());
9     preStat.setObject(3, userInfo.getLastName());
10    preStat.setObject(4, userInfo.getUserName());
11    preStat.setObject(5, userInfo.getEmail());
12    preStat.setObject(6, userInfo.getPassword());
13    preStat.setObject(7, userInfo.getMobileNo());
14    preStat.executeUpdate();
```

4. Delete

```
3 String SQL = "DELETE FROM COUNTRY WHERE COUNTRY_ID = "+country.getCountry_id();
4 try {
5     Statement p = connection.createStatement();
6     p.execute(SQL);
```

```
3 String SQL = "DELETE FROM COUNTRY WHERE COUNTRY_ID = ?";
4 try {
5     PreparedStatement p = connection.prepareStatement(SQL);
6     p.setObject(1, country.getCountry_id());
7     p.execute();
```

Complex Query

1. Order by

```
4 Statement stmt = conn.createStatement(); {
5 ResultSet rs = stmt.executeQuery("SELECT * FROM message WHERE from_id = "+mid+"
ORDER BY sent_date ASC");
```

```
4 PreparedStatement stmt = conn.prepareStatement("SELECT
* FROM message WHERE from_id = "+?+" ORDER BY sent_date ASC"); {
5 stmt.setObject(1, mid);
6 ResultSet rs = stmt.executeQuery();
```

2. Count

```
String query = "SELECT COUNT(*) AS facilityInstances FROM Facility " + " WHERE i
id<>
'" + id + "' AND name= '" + name + "'";
7 System.out.println(query);
8 try {
9 Statement stmt = conn.createStatement();
10 stmt.setQueryTimeout(5);
11 results = stmt.executeQuery(query);
```

```
String query = "SELECT COUNT(*) AS facilityInstances FROM Facility " + " WHERE i
d<>
'? AND name= '?'";
7 System.out.println(query);
8 try {
9 PreparedStatement stmt = conn.prepareStatement(query);
10 stmt.setQueryTimeout(5);
11 stmt.setObject(1,id);
12 stmt.setObject(2,name);
13 results = stmt.executeQuery();
```

3. Limit

```
4 String sql = "select * from inception_preference where userId = "+userId+"
order by id desc limit "+start+", "+count;
5 Statement ps = conn.createStatement();
6 ResultSet rs = ps.executeQuery(sql);
```

```
4 String sql = "select * from inception_preference where userId = ?
order by id desc limit ?, ?";
5 PreparedStatement ps = conn.prepareStatement(sql);
6 ps.setObject(1, userId);
7 ps.setObject(2, start);
8 ps.setObject(3, count);
9 ResultSet rs = ps.executeQuery();
```

4. Distinct

```
6 Statement ps = null;
7 ResultSet rs = null;
8 try {
9 conn = datasource.getConnection();
10 String sql = "SELECT COUNT(DISTINCT order_num) FROM bespeak WHERE order_state NO
T IN(7, 8, 9)";
11 ps = conn.createStatement();
12 rs = ps.executeQuery(sql);
```

```
6 PreparedStatement ps = null;
7 ResultSet rs = null;
8 try {
9 conn = datasource.getConnection();
10 String sql = "SELECT COUNT(DISTINCT order_num) FROM bespeak WHERE order_state NO
T IN(7, 8, 9)";
11 ps = conn.prepareStatement(sql);
12 rs = ps.executeQuery();
```

Multiple Queries in Single function

1.

```
5 Statement s = conn.createStatement();
6 ResultSet rs = s.executeQuery("SELECT * FROM virtual_slave_node_table WHERE
host='"+ hostname + "'");
7 if (rs.next()) {
8 conn.close();
9 throw new Exception("Can not remove. Host is being used.");
10 }
11 rs = s.executeQuery("SELECT * FROM virtual_master_node_table WHERE
host='"+ hostname + "'");
12 if (rs.next()) {
13 conn.close();
14 throw new Exception("Can not remove. Host is being used.");
15 }
16 s.executeUpdate("DELETE FROM virtual_machine_host_table where
hostname='"+ hostname + "'");
```

```
5 PreparedStatement s = conn.prepareStatement("SELECT
* FROM virtual_slave_node_table WHERE host=?");
6 s.setObject(1,hostname);
7 ResultSet rs = s.executeQuery();
8 if (rs.next()) {
9 conn.close();
10 throw new Exception("Can not remove. Host is being used.");
11 }
12 s = conn.prepareStatement("SELECT * FROM virtual_master_node_table WHERE
host=?");
13 s.setObject(1,hostname);
14 rs = s.executeQuery();
15 if (rs.next()) {
16 conn.close();
17 throw new Exception("Can not remove. Host is being used.");
18 }
19 s = conn.prepareStatement("DELETE FROM virtual_machine_host_table where
hostname=?");
20 s.setObject(1,hostname);
21 s.executeUpdate();
```

2.

```
3 Statement pStatement;
4 ResultSet rs;
5 String queryString;
6 ArrayList<String> list = new ArrayList<String>();
7 try {
8     queryString = "set search_path to imdb";
9     pStatement = connection.createStatement();
10    pStatement.executeQuery(queryString);
11    queryString = "select distinct name " + "from people, roles " + "where people.p
12    person_id = roles.person_id and name <>
13    '" + person + "' and
14    movie_id in " + "(select distinct movie_id " + "from people, roles " + "where na
15    me =
16    '" + person + "' AND people.person_id = roles.person_id)";
17    pStatement = connection.createStatement();
18    rs = pStatement.executeQuery(queryString);
```

```
3 PreparedStatement pStatement;
4 ResultSet rs;
5 String queryString;
6 ArrayList<String> list = new ArrayList<String>();
7 try {
8     queryString = "set search_path to imdb";
9     pStatement = connection.prepareStatement(queryString);
10    pStatement.execute();
11    queryString = "select distinct name " + "from people, roles " + "where people.p
12    erson_id = roles.person_id and name <>
13    '?' and
14    movie_id in " + "(select distinct movie_id " + "from people, roles " + "where na
15    me =
16    '?' AND people.person_id = roles.person_id)";
17    pStatement = connection.prepareStatement(queryString);
18    pStatement.setObject(1, person);
19    pStatement.setObject(2, person);
20    rs = pStatement.executeQuery();
```