

Condition & query native to Oracle

Nguyễn Anh Tuấn

KTECH
COLLEGE



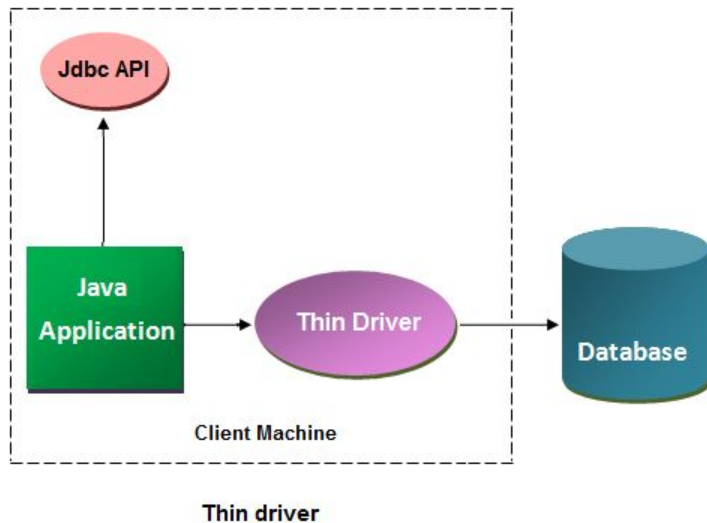
Nội dung bài giảng

- 1 Tạo connection
- 2 PreparedStatement, ResultSet
- 3 Các lệnh điều kiện
- 4 Thực hành
- 5 Các lệnh thao tác dữ liệu

Tạo Connection

Tạo Connection

- Import thư viện ojdbc11 của oracle để có thể kết nối tới OracleDB.
- <https://mvnrepository.com/artifact/com.oracle.database.jdbc/ojdbc11/23.4.0.24.05>
 - **Cách 1:** Tải về file jar trong link trên để add trực tiếp vào thư viện jar của dự án.
 - **Cách 2:** Sử dụng file pom.xml của Maven hoặc build.xml của Gradle để import thư viện ojdbc11



Tạo connection

// Thông tin kết nối

String url = "jdbc:oracle:thin:@localhost:1521:orcl"; *// Url kết nối đến OracleDB*

String user = "system"; *// Tên user đã tạo trong OracleDB*

String pass = "1"; *// Mật khẩu user đã tạo trong OracleDB*

try {

// Mở kết nối

Lớp Connection nằm trong package java.sql.Connection

Connection conn = DriverManager.getConnection(this.url, this.user, this.pass);

System.out.println("Đã mở connection.");

// Đóng kết nối

conn.close();

System.out.println("Đã đóng connection.");

} catch (SQLException sqlException) { sqlException.printStackTrace(); }

PreparedStatement, ResultSet

PreparedStatement, ResultSet

- **PreparedStatement:**

- Là một đối tượng trong JDBC được sử dụng để thực thi câu lệnh SQL có tham số.
- Được sử dụng khi muốn thực hiện một câu lệnh SQL nhiều lần với các giá trị tham số khác nhau mà không cần phải tạo lại câu lệnh SQL từ đầu và giúp bảo vệ ứng dụng khỏi các lỗ hổng bảo mật như SQL injection.

- **ResultSet:**

- Là một đối tượng trong JDBC được sử dụng để biểu diễn kết quả của một câu lệnh SELECT.
- Cho phép duyệt qua các hàng của kết quả truy vấn và lấy dữ liệu từ các cột tương ứng.

PreparedStatement, ResultSet

// Gọi method lấy kết nối dựa theo class tùy chỉnh

```
conn.openConn();
```

// Thực hiện câu lệnh truy vấn

```
PreparedStatement sta = conn.getConnection().prepareStatement("select * from AAA");
```

// Biểu diễn kết quả

```
ResultSet res = sta.executeQuery();
```

```
while (res.next()) {
```

```
    System.out.println("ID: " + res.getString(1));
```

// Số cột lấy theo index từ 1

```
    System.out.println("Name: " + res.getString(2));
```

```
    System.out.println("-----");
```

```
}
```

// Gọi method đóng kết nối dựa theo class tùy chỉnh

```
conn.closeConn();
```


PreparedStatement, ResultSet

// Gọi method lấy kết nối dựa theo class tùy chỉnh

```
conn.openConn();
```

// Thực hiện câu lệnh thêm dữ liệu

```
PreparedStatement sta = conn.getConnection().prepareStatement("insert into aaa values (4, ?)");
```

```
sta.setString(1,"haha"); // Vị trí của dấu tính từ 1
```

// Thực thi câu lệnh thêm dữ liệu

```
int status = sta.executeUpdate();
```

```
if (status > 0) {
```

```
    System.out.println("Đã thêm thành công");
```

```
}
```

// Gọi method đóng kết nối dựa theo class tùy chỉnh

```
conn.closeConn();
```

Các lệnh điều kiện

Điều kiện Logic (AND/OR)

// Có thể sử dụng các điều kiện AND và OR để kết hợp nhiều điều kiện trong câu lệnh SELECT, UPDATE, DELETE, và các câu lệnh khác

- **Syntax sử dụng AND / OR:**

```
SELECT column1, column2, ...  
FROM table_name  
WHERE condition1 AND condition2 OR ...;
```

- **Ví dụ:**

```
SELECT *  
FROM employees  
WHERE department = 'IT' AND salary > 50000 OR salary < 10000;
```

Điều kiện Tập hợp (IN/NOT IN)

// IN và NOT IN là hai toán tử điều kiện được sử dụng để kiểm tra xem một giá trị có nằm trong 1 danh sách giá trị cho trước hay không

- **Syntax sử dụng AND / OR:**

SELECT column1, column2, ...

FROM table_name

WHERE column_name **IN** (value1, value2, ...);

- **Ví dụ:**

SELECT *

FROM employees

WHERE department **IN** ('IT', 'HR', 'Finance');

Điều kiện Phạm vi (BETWEEN)

// BETWEEN là một toán tử điều kiện được sử dụng để kiểm tra xem một giá trị có nằm trong khoảng giá trị được xác định hay không.

- **Syntax sử dụng AND / OR:**

```
SELECT column1, column2, ...  
FROM table_name  
WHERE column_name BETWEEN value1 AND value2;
```

- **Ví dụ:**

```
SELECT *  
FROM employees  
WHERE salary BETWEEN 50000 AND 80000;
```

// Toán tử BETWEEN bao gồm cả giá trị biên đầu và cuối. Do đó, nếu cần lựa chọn các giá trị nằm trong một khoảng nhưng không bao gồm giá trị biên, cần phải sử dụng các toán tử khác như > và < hoặc >= và <=.

Điều kiện Mẫu (LIKE/NOT LIKE)

// LIKE được sử dụng để kiểm tra xem một chuỗi có khớp với một mẫu chuỗi (pattern) cho trước hay không.

// NOT LIKE ngược lại với LIKE.

- **Syntax sử dụng AND / OR:**

```
SELECT column1, column2, ...
```

```
FROM table_name
```

```
WHERE column_name LIKE 'pattern';
```

- **Ví dụ:**

```
SELECT *
```

```
FROM employees
```

```
WHERE last_name LIKE 'Sm%';
```

// % đại diện cho bất kỳ chuỗi ký tự nào (bao gồm cả chuỗi rỗng).

// _ đại diện cho một ký tự duy nhất.

Điều kiện NULL (IS NULL/IS NOT NULL)

// IS NULL được sử dụng để kiểm tra xem một giá trị có là null hay không.

// IS NOT NULL ngược lại với IS NULL

- **Syntax sử dụng AND / OR:**

```
SELECT column1, column2, ...  
FROM table_name  
WHERE column_name IS NULL;
```

- **Ví dụ:**

```
SELECT *  
FROM employees  
WHERE email IS NULL;
```

Thực hành 60'

Tạo 1 chương trình trong đó:

- Phải đăng nhập trước mới được sử dụng các chức năng của chương trình.
- Lấy danh sách Học viên từ file txt trong bài thực hành về File Handling, đưa vào Database.
- Thống kê học viên có mặt trong ngày.
- Thống kê học viên vắng mặt trong ngày.
- Cho phép xem lại danh sách học viên có mặt, vắng mặt hoặc tất cả.

Lưu ý:

- Tài khoản đăng nhập phải mã hoá mật khẩu.
- Chương trình dùng để điểm danh, thống kê vào database hàng ngày.

Nâng cao (60'):

- Có 2 role: Giảng viên và Trợ giảng. Giảng viên có 2 user, Trợ giảng có 2 user.
- Giảng viên A có full quyền trong chương trình.
- Giảng viên B chỉ có quyền xem danh sách Học viên, danh sách Học viên có mặt, danh sách Học viên vắng mặt và danh sách Học viên trong ngày (cả vắng cả có mặt)
- Trợ giảng X có quyền điểm danh (ghi lại Học viên có mặt và vắng mặt).
- Trợ giảng Y có quyền như Giảng viên B.

ROAD TO KOREA

Nếu có bất kỳ thắc mắc nào, hãy đặt câu hỏi qua

mail@mail.com hoặc Zalo 0xxx xxx xxx