

Module 03

JDBC를 사용한 CRUD

NHN Academy



Overview

- CRUD Operation
- 데이터 삽입
- 데이터 수정
- 데이터 삭제

CRUD Operation

- CRUD 개요
- SQL 매핑

CRUD 개요

- Create, Read, Update, Delete
- 대부분의 컴퓨터 소프트웨어가 가지는 기본적인 데이터 처리 기능
- 유사 용어
 - ABCD: Add, Browse, Change, Delete
 - ACID: Add, Change, Inquire, Delete
 - BREAD: Browse, Read, Edit, Add, Delete
 - VADE(R): View, Add, Delete, Edit (트랜잭션에서 Restore 추가)
 - SIUD: Select, Insert, Update, Delete

SQL 매핑

- 각 문자는 표준 SQL문으로 대응 가능함

이름	조작	SQL
Create	생성	INSERT
Read(또는 Retrieve)	읽기(또는 인출)	SELECT
Update	갱신	UPDATE
Delete(또는 Destroy)	삭제(또는 파괴)	DELETE

데이터 삽입

- 데이터 삽입
- 저장 프로시저 사용

데이터 삽입

- Statement의 ExecuteUpdate 메소드 사용
 - 표준 SQL의 INSERT문과 매핑

```
String sqlQuery = "INSERT INTO Passenger VALUES (" + passengerNo + ", '";
sqlQuery += passengerName + "',";
sqlQuery += grade + ",";
sqlQuery += age + ")";

int result = 0;

try (Connection myConnection = DriverManager.getConnection(databaseURL, userName, password);
    PreparedStatement statement = myConnection.prepareStatement(sqlQuery)) {
    result = statement.executeUpdate();
}
catch (SQLException e) {
    e.printStackTrace();
}
```

저장 프로시저 사용

- CallableStatement의 execute 또는 executeUpdate 메소드 사용
 - 파라미터와 함께 저장 프로시저 호출

```
# Stored Procedure in MySQL database
DELIMITER $$
CREATE PROCEDURE InsertPassenger (
    PassengerNo int,
    PassengerName varchar(10),
    Grade int,
    Age int)
BEGIN
    INSERT INTO Passenger
    VALUES (PassengerNo, PassengerName, Grade, Age);
END $$
DELIMITER ;
```

```
try (Connection myConnection = ...;
    CallableStatement statement = ...) {
    statement.setInt(1, passengerNo);
    statement.setString(2, passengerName);
    statement.setInt(3, grade);
    statement.setInt(4, age);
    statement.execute();
}
catch (SQLException e) {
    e.printStackTrace();
}
```


데이터 갱신

- 데이터 갱신
- 저장 프로시저 사용

데이터 삽입

- Statement의 ExecuteUpdate 메소드 사용
 - 표준 SQL의 UPDATE문과 매핑

```
String sqlQuery = "INSERT INTO Passenger VALUES (" + passengerNo + ", '";
sqlQuery += passengerName + "',";
sqlQuery += grade + ",";
sqlQuery += age + ")";

int result = 0;

try (Connection myConnection = DriverManager.getConnection(databaseURL, userName, password);
    PreparedStatement statement = myConnection.prepareStatement(sqlQuery)) {
    result = statement.executeUpdate();
}
catch (SQLException e) {
    e.printStackTrace();
}
```

저장 프로시저 사용

- CallableStatement의 execute 또는 executeUpdate 메소드 사용
 - 파라미터와 함께 저장 프로시저 호출

```
# Stored Procedure in MySQL database
CREATE PROCEDURE UpdatePassengerGrade (
    m_PassengerNo int,
    m_Grade        int
)
BEGIN
    UPDATE Passenger SET
        Grade = m_Grade
    WHERE
        PassengerNo = m_PassengerNo;
END $$
DELIMITER ;
```

```
int result = 0;

try (Connection myConnection = ...;
     CallableStatement statement = ...) {
    statement.setInt(1, passengerNo);
    statement.setInt(2, grade);
    result = statement.executeUpdate();
}
catch (SQLException e) {
    e.printStackTrace();
}
```

Lab – 승객 정보 응용 프로그램 작성

