



# 제10장

## JDBC연동과 DB연동 객체들

# 5. PreparedStatement 객체 살펴보기

## 5.1 PreparedStatement 객체로 데이터 접근하기

- PreparedStatement 객체
  - 동적인 쿼리에 사용
  - **PreparedStatement** 객체는 하나의 객체로 여러 번의 쿼리를 실행할 수 있으며, 동일한 쿼리문을 특정 값만 바꾸어서 여러 번 실행해야 할 때, 매개변수가 많아서 쿼리문을 정리해야 할 때 유용하다.

```
PreparedStatement prepareStatement(String sql) throws SQLException
```

# 5. PreparedStatement 객체 살펴보기



## 5.2 PreparedStatement 객체의 메서드

메소드	반환 유형	설명
setString(int parameterIndex, String x)	void	필드 유형이 문자열인 경우
setInt(int parameterIndex, int x)	void	필드 유형이 정수형인 경우
setLong(int parameterIndex, long x)	void	필드 유형이 정수형인 경우
setDouble(int parameterIndex, double x)	void	필드 유형이 실수형인 경우
setFloat(int parameterIndex, float x)	void	필드 유형이 실수형인 경우
setObject(int parameterIndex, Object x)	void	필드 유형이 객체형인 경우
setDate(int parameterIndex, Date x)	void	필드 유형이 날짜형인 경우
setTimestamp(int parameterIndex, Timestamp x)	void	필드 유형이 시간형인 경우

메소드	반환 유형	설명
executeQuery()	ResultSet	SELECT 문을 실행할 때 사용합니다.(ResultSet 객체 반환).
executeUpdate()	int	삽입 수정 삭제와 관련된 SQL 문 실행에 사용합니다.
close();	void	Statement 객체를 반환할 때 사용합니다.

# 5. PreparedStatement 객체 살펴보기

## 5.3 executeQuery()메서드로 데이터 조회하기

- executeQuery( ) 메소드는 동적인 SELECT 쿼리문을 통해 데이터를 검색하는 데 사용

```
int executeQuery() throws SQLException
```

[executeQuery() 메소드 사용 예: SELECT 쿼리문]

```
<%  
    Connection conn = null;  
    ...(생략)...  
    String sql = "SELECT * FROM Member WHERE id =? ";  
    PreparedStatement pstmt = conn.prepareStatement(sql);  
    pstmt.setString(1,"1");  
    ResultSet rs = pstmt.executeQuery(sql);  
    ...(생략)...  
    pstmt.close();  
%>
```

# 5. PreparedStatement 객체 살펴보기

## 5.4 executeUpdate() 메소드로 데이터 삽입, 수정, 삭제하기

- executeUpdate() 메소드는 INSERT, UPDATE, SELECT 쿼리문을 통해 데이터를 삽입, 수정, 삭제하는 데 사용

[executeUpdate() 메소드 사용 예(삽입): INSERT 쿼리문]

```
<%  
    Connection conn = null;  
    ... (생략)...  
    String sql = "INSERT INTO Member(id, name, passwd) VALUES (?, ?, ?)";  
    PreparedStatement pstmt = conn.prepareStatement(sql);  
    pstmt.setString(1, "1");  
    pstmt.setString(2, "홍길순");  
    pstmt.setString(3, "1234");  
    pstmt.executeUpdate();  
    ... (생략)...  
    pstmt.close();  
%>
```

## 6. 쿼리문 실행 결과 값 가져오기

### 6.1 ResultSet 객체

- Statement 또는 PreparedStatement 객체로 SELECT 문을 사용하여 얻어온 레코드 값을 테이블 형태로 가진 객체

[Statement 객체를 사용하는 경우]

```
ResultSet executeQuery(String sql) throws SQLException
```

[PreparedStatement 객체를 사용하는 경우]

```
ResultSet executeQuery() throws SQLException
```

# 6. 쿼리문 실행 결과 값 가져오기



## 6.2 ResultSet 객체의 메서드

메소드	반환 유형	설명
getXxx(int ColumnIndex)	XXX	설정된 ColumnIndex(필드 순번)의 필드 값을 설정한 XXX 형으로 가져옵니다.
getXxx(String ColumnName)	XXX	설정된 ColumnName(필드 순번)의 필드 값을 설정한 XXX 형으로 가져옵니다.
absolute(int row)	boolean	설정된 row 행으로 커서를 이동합니다.
beforeFirst()	void	첫 번째 행의 이전으로 커서를 이동합니다.
afterLast()	void	마지막 행의 다음으로 커서를 이동합니다.
first()	void	첫 번째 행으로 커서를 이동합니다.
last()	void	마지막 행으로 커서를 이동합니다.
next()	boolean	다음 행으로 커서를 이동합니다.
previous()	boolean	현재 행의 이전 행으로 커서를 이동합니다.
close():	void	ResultSet 객체를 반환할 때 사용합니다.

감사합니다.

