

#### 4. 도입 JDBC

데이터 액세스 방법

- 데이터 액세스 처리를 버퍼나 쿼리 등에서 분리하는 것.

DAO(Data Access Object)

- 데이터 액세스 처리에 특화된 인터페이스

자신의 데이터 액세스 기술

- JDBC, Hibernate, Mybatis, JPA

DAO 관점에서 도입 방법

- 데이터 액세스 기술을 함께 사용하기 위한 연계 가능 제형

데이터 소스(Data Source)

- 데이터 액세스 기술/프레임워크의 상관없이 DB 접속을 관리해주는 인터페이스
- 애플리케이션은 구체적인 풀에 의해 구체적인 연결객체 재사용.

데이터 소스 구현

- 커넥터가 제공하는 데이터 소스 commons - dbcp
  - Apache Commons DBCP → Bean 정의 파일에 DBCP 데이터 소스 설정.  
applicationContext.xml
  - 여러 양에 설정.
- 애플리케이션이 제공하는 데이터 소스
  - Tomcat, Oracle WebLogic 등
  - JNDI 이용하여 데이터 소스 연결도 취득
- 애플리케이션 DB가 제공하는 도입된 자원 데이터 소스
  - HSQLDB, H2, Apache Derby

JDBC 이용 문제점

- 다양한 소스코드 기술.
- 다양한 예외 원인을 파악하기 위해 코딩 필요.
- DB 자원이 다 예외 코드가 달라서 일관성 유지가 어렵음.

## 프로그 JDBC

- JDBC로 다양한 API 제공하 소프트웨어 개발자

- JDBC로 각종 사용자의 데이터는 강력한 코딩

→ 커넥션 연결 관리

→ SQL문 실행

→ SQL문 결과 행에 대해 반복 처리

→ 예외 처리

## 프로그 JDBC가 제공하는 주요 특징

- JdbcTemplate

- NamedParameterTemplate

JdbcTemplate 클래스는 기본 메서드

~~queryForObject~~ queryForObject 한개의 결과 레코드 중에서 하나의 값을 가져올 때  
RowMapper에 의해 사용 한개의 레코드 객체로 객체 매핑

queryForMap 한개의 결과 레코드 객체를 Map 형태로 매핑해 주 있음

queryForList 여러 개의 Map 형태의 결과 레코드로 가져올 수 있음.

~~query~~ query 여러 개의 레코드로 가져올 때 ResultExtractor (ResultSetExtractor)

update 데이터 변경 (INSERT, UPDATE, DELETE) 실행할 때.

## SELECT 문 : 취득 결과가 레코드 한 개 or 특정 컬럼만 취득할 경우

- queryForObject SQL 문 반환할 컬럼의 타입 (int)

int count = jdbcTemplate.queryForObject("SELECT COUNT(\*) FROM STUDENT", Integer.class);

String name = jdbcTemplate.queryForObject("SELECT username FROM STUDENT WHERE id=?", String.class, id);  
SQL문 반환할 컬럼의 타입 (String) 파라미터 값

## SELECT 문 : 취득 결과가 한 레코드 값을 취득할 경우

- queryForMap : 한 레코드 값을 Map (컬럼 이름을 키로 값을 저장) 데이터로

Map<String, Object> student = jdbcTemplate.queryForMap("SELECT \* FROM STUDENT WHERE id=?", id);  
String name = (String) student.get("user name");

- queryForList : 여러 레코드 값을 Map 데이터로.

List<Map<String, Object>> studentList = jdbcTemplate.queryForMap("SELECT \* FROM STUDENT");

SELECT 문 : 데이터로 데이터를 가져오기

- queryForObject 와 query 이용

- queryForObject - 한 레코드 가져올 때.

vo = jdbcTemplate.queryForObject("SELECT \* FROM STUDENT WHERE ID=?",

new BeanPropertyRowMapper<StudentVO>(StudentVO.class), id);

데이터 가져오기 위한 SQL을 작성합니다.

사용자 StudentVO의 프로퍼티명과 테이블 컬럼명이 같아야 합니다.

그러나 다른 경우, RowMapper 인터페이스를 구현하여 StudentVO로 매핑 처리.

- query → 여러 레코드 가져올 때

→ RowMapper 인터페이스를 구현한 클래스를 사용.

→ 클래스 내 mapRow() 구현 방식에 따라

```
public List<StudentVO> readList() throws Exception {
    List<StudentVO> studentList = jdbcTemplate.query(
        "SELECT * FROM STUDENT",
        new RowMapper<StudentVO>() {
            public StudentVO mapRow(ResultSet rs, int rowNum) throws SQLException {
                StudentVO vo = new StudentVO();
                vo.setId(rs.getString("ID"));
                vo.setPassword(rs.getString("PASSWORD"));
                vo.setUsername(rs.getString("USERNAME"));
                vo.setSnum(rs.getString("SNUM"));
                vo.setDepart(rs.getString("DEPART"));
                vo.setMobile(rs.getString("MOBILE"));
                vo.setEmail(rs.getString("EMAIL"));
                return vo;
            }
        });
    return studentList;
}
```

INSERT / UPDATE / DELETE 문.

- update 데이터를 수정

• INSERT 문

```
StudentVO vo;
jdbcTemplate.update(
    "INSERT INTO STUDENT (ID, PASSWORD, USERNAME, SNUM, DEPART, MOBILE, EMAIL) "
    + "VALUES (?, ?, ?, ?, ?, ?, ?)", vo.getId(), vo.getPassword(), vo.getUsername(),
    vo.getSnum(), vo.getDepart(), vo.getMobile(), vo.getEmail()
);
```

• DELETE 문

```
StudentVO vo;
jdbcTemplate.update("DELETE FROM STUDENT WHERE ID=?", vo.getId());
```

## 스프링 테스트 (Spring - Test)

- 스프링 프레임워크에서 만든 클래스 ( @Controller, @Service, @Repository, @Component 등이 붙은 클래스)를 테스트하는 모듈.
- 단위 테스트, 통합테스트를 자월하기 위해 데이터베이스나 외부에 기능 제공
  - JUnit 테스트 프레임워크 사용하여 스프링 DI 컨테이너 동작 검증
  - 테스트용 테스트 상황에 맞게 제어 가능
  - 웹 서버 사용X 스프링 MVC 동작 검증 가능
  - RestTemplate 이용하 HTTP요청에 대한 응답을 받아볼 수 있는 것

POM.xml (spring-test, junit).

① Runwith (SpringJUnit4ClassRunner.class) // 테스트용 DI 컨테이너를 동작시키기 위한 Runner 클래스

② Context Configuration (locations = "classpath:/applicationContext.xml") // 테스트용 DI 컨테이너가 사용하는 설정파일

③ Auto wired // DI 컨테이너에 등록할 테스트 대상 빈을 주입

④ Test // 테스트 마스터 선언.