**iBATIS in Action (1/14)**

**1. iBATIS의 탄생 철학**

**1.1. 복합적인 솔루션**

**iBATIS 는 어플리케이션에서 데이터베이스에 SQL 을 실행시키는 다양한 방법들로부터다양한 개념을 빌려다 만든 독보적인 솔루션이다.**

**1.1.1. iBATIS 의 기원 답사**

**그림 1 1. iBATIS가 개발 프로세스를 단순화하기 위해 끌어다 모은 몇몇 개념들**

**iBATIS 는 관계형 데이터 베이스에 접근하는 가장 잘 알려진 방법들로부터 가장 좋은 특징과 아이디어들을 차용하고, 그것들로부터 시너지를 이끌어낸다.**

**데이터베이스와 소통하는 다양한 접근 방법들을 논의하고, iBATIS가 응용한 각 부분들에 대하여 알아본다.**

**-SQL**

**두 가지 언어가 합쳐졌다고 볼수 있는데, 첫째로는 데이터 정의 언어(Data Definition Language, DDL) 로 CREATE, DROP, ALTER 같은 구문을 포함하고 있다. 데이터 베이스의구조와 설계를 정의하는데 사용된다.**

**둘 째로는 데이터 조작 언어(Data Manipulation Language,DML)이다. 이것은 SELECT, INSERT, UPDATE, DELETE 같은 구문들을 포함하고 있다. DML은 데이터를 직접 조작하기위해 사용한다.**

**iBATIS 에서는 관계형 데이터베이스에 접근하는 1차적인 방법으로 SQL사용을 채택하였다.**

**-옛날식의 저장 프로시저(Stored Procedure)**

**관계형 데이터베이스로 애플리케이션 프로그램을 작성하는 가장 오래된 방법. 2-티어설계라고 불리는 방식을 사용했다.성능과 확장성에는 문제가 있다.**

**-현대적인 저장 프로시저**

**중간 티어에서 호출되는 원격 프로시저처럼 사용되고 있다.성능에 관한 많은 제약들이커넥션 풀이나 데이터베이스의 자원을 관리함으로써 해결되었다.**

**-인라인 SQL**

**일반 프로그래밍 언어에 SQL을 내장시키는 방법이다.**

**폭넓게 받아들여지지는 못했다. 보통 실제 프로그래밍 언어의 일부로써 구현되지 못하는 점이 있다. 언어 측면에서 인라인 SQL의 이러한 문제를 해결하는 한가지 방법은 애플리케이션에서 SQL을 문자열 같은 자료구조로 나타내는 것이다. 이러한 접근 방법을 보통 동적 SQL 이라고 한다.**

**인라인SQL : 일반적인 소스 코드내에 SQL명령문을 삽입하는 방식이다.**

**-동적 SQL**

**전처리기를 사용하지 않고 인라인 SQL을 다루는 방법이다.SQL이 문자타입이므로 인라인 SQL처럼 직접 프로그래밍 언어의 특징을 사용할 수는 없다.**

**동적 SQL : 동적 SQL 은 한 예로 asp, php 등과 같은 서버 스크립트에서 SQL문을 만들 때,테이블 명이나 필드명을 변수 처리 해서 SQL 문을 유연하게 만든 것과 같다.**

**동적SQL은 Pre-Compiler를 이용한 프로그램이나 Stored Procedure 내에서**

**SQL문 자체가 동적으로 변하는 경우를 일컫는 말입니다.**

**즉 Select 문장에서 조회되는 컬럼명이나 테이블, 조건절에서의 컬럼명등이**

**상황에 따라 변할 때 사용된다는 것이다.**

**-객체 관계 매핑**

**객체 관계 매핑(ORM) 은 SQL을 개발자의 책임 영역에서 완전히 제거함으로써 객체의 영구적인 저장을 단순화하도록 설계돼 있다. SQL의 자동생성과 트랜잭션 관리기능이 있다.**

**객체관계 매핑 : SQL을 개발자의 책임 영역에서 완전히 제거함으로써 객체의 영구적인저장을 단순화하도록 설계되어 있다. 대신 SQL은 자동 생성된다. 어떤 개발 도구들은 컴파일시에 SQL을 정적으로 자동 생성하기도 하지만, 보통은 실행시 동적으로 자동 생성한다. SQL은 애플리케이션의 클래스와 관계형 데이터베이스 테이블 간의 매핑을 기반으로 하여 생성된다. ORM의 API는 SQL을 없앨 수 있을 뿐만 아니라 전형적인 SQL API보다훨씬 단순하기도 하다.**

**1.1.2. iBATIS 의 장점 이해하기**

**iBATIS 는 이제껏 살펴본 방법론에서 장점들을 차용하여 만들어졌다.**

**그리고 iBATIS의 외부 저장과 캡슐화라는 두 개념이 합쳐져서 iBATIS가 이룩한 많은 가치와 고급 기능들을 제공한다.**

**-외부로 뺀 SQL**

**SQL을 코드로부터 분리한다. iBATIS의 핵심 장점은 SQL을 쓰고자 하는 대로 그대로 쓰면된다는 점이다.**

**-캡슐화된 SQL**

**저장 프로시저 안에 SQL을 숨기는 것이다. 코드를 응집성있는 모듈로 조직하는 것뿐만아니라 또한 세부적인 구현을 숨기고 호출하는 코드에게 인터페이스만을 노출시키는 모듈화의 한 형태이다. XML을 사용한다.**

**1. iBATIS란 무엇인가?**

**1.1. SQL 매핑하기**

**아래는 간단한 XML 서술 파일을 이용하여 SQL 구문의 입력과 출력을 매핑하는 예이다.**

**여기서 주소 데이터를 반환하는 SQL SELECT 구문을 볼 수 있다.**

**parameterClass="java.lang.Integer"**

**resultClass="Employee">**

**SELECT ID as id,**

**EMPLOYEE\_NUMBER as employeeNumber,**

**FIRST\_NAME as firstName,**

**LAST\_NAME as lastName,**

**TITLE as title**

**FROM EMPLOYEE**

**WHERE EMPLOYEE\_NUMBER = #empNum#**

**이 쿼리문을 실행하는 방법은 자바코드에서 다음을 실행하는 것이다.**

**Employee emp = (Employee) sqlMap.queryForObject("getEmployee",**

**new Integer(5));**

**이 코드는 구문을 실행하고 파라미터를 셋팅하며 실제 자바 객체로 결과를 가져온다.**

**2.2.1작고 간단한 시스템을 위한 iBATIS**

**iBATIS 가 소규모 어플리케이션과도 잘 작동하는 이유는**

**첫째, iBATIS 자체가 작고 간단하다.**

**둘째, iBATIS 는 이미 존재하는 어플리케이션이나 데이터베이스의 설계 변경을 강요하지 않는다.**

**셋째, 현재 오랜 기간 운영중인 소프트웨어라면, 자그마한 규모의 소프트웨어가 큰 규모의 소프트웨어로 성장하는 것은 피할 수 없는 사실이다. iBATIS가 대규모 시스템에서도잘 작동하기 때문에 iBATIS를 사용하는 소규모 어플리케이션이 전사적 애클리케이션의요구 조건을 만족시키는 시스템으로 성장해가는 데도 문제가 없다.**

**1.3. 왜 iBATIS를 사용하나?**

**2.3.1간단함**

**iBATIS의 매우 견고한 토대가 되는 JDBC와 SQL을 계속 유지함으로써 이러한 간결성을 확보하게 되었다. iBATIS는 훨씬 더 적은 코드로도 JDBC처럼 작동하기 때문에 자바 개발자들이 다루기가 쉽다.**

**2.3.2생산성**

**iBATIS는 개발자의 생산성을 높여주는 데 성공했다. 코드량의 감소는 주소 작성할 필요가 없는 JDBC 코드 때문이다. SQL은 여전히 손으로 작성해야 하지만, SQL은 문제가 되지않는다.**

**2.3.3성능**

**iBATIS는 나중에 자세히 언급할 많은 성능 최적화 기법을 지원한다. 지금은 iBATIS를 간단한 방법으로 설정해서 사용할 수 있고, 나아가 JDBC만큼 잘, 혹은 더 뛰어나게 수행된다는 것을 아는 것이 중요하다.**

**2.3.4관심사의 분리**

**iBATIS는 데이터베이스 커넥션과 PreparedStatement 그리고 결과셋 같은 퍼시스턴스 계층과 관련된 리소스를 관리하여 계층화를 지원하도록 도와준다. iBATIS는 데이터베이스에 독립적인 인터페이스와 API를 제공하여 애플리케이션을 어떠한 퍼시스턴스 계층 관련 리소스에도 의존하지 않게 해준다.**

**2.3.5 작업의 분배**

**SQL문이 애플리케이션 소스 코드로부터 완전히 분리되기 때문에, SQL 프로그래머는 문자열 이어 붙이기에 대해 걱정할 필요없이 SQL을 작성할 수 있다.**

**2.3.6 이식성: 자바, .NET 그리고 그 외…**

**이식성이 높다는 것은 모든 애플리케이션에서 설계를 일관성 있게 할 수 있다는 의미이다. iBATIS는 애플리케이션의 설계와 무관하게 더 많은 언어와 다른 프레임워크보다 더많은 형태의 애플리케이션에서 작동한다.**

**2.3.7 오픈 소스와 정직성**

**오픈 소스 소프트웨어의 가장 큰 장점 중 하나는 정직하다는 점이다. 진실이나 거짓을과장할 까닭이 없다. 그래서 iBATIS를 사용할 필요가 없는 경우에 대해 몇 가지 이야기해보고 적절한 대안을 제시해 본다.**

**1.4. iBATIS를 사용하지 않는 경우**

**2.4.1 개발자가 모든 것에 대해 영원한 결정권을 갖고 있을 때…**

**모든 결정권을 갖고 있을 때라면, 하이터 네이트와 같은 완전한 객체 관계 매핑 솔루션을 사용하는 것이 좋다. 이 때는 객체 관계 매퍼가 제공해 주는 설계상의 이점과 생산성향상을 제대로 누릴 수 있다.**

**2.4.2 어플리케이션이 완전히 동적인 SQL을 요구할 때**

**애플리케이션의 핵심 기능이 SQL을 동적으로 생성하는 것이라면, iBATIS는 좋지 못한 선택이다. 그냥 JDBC를 사용하거나 혹은 자신만의 프레임워크를 직접 만드는 것도 좋다.**

**2.4.3 관계형 데이터베이스를 사용하지 않을 때**

**iBATIS는 환경에 대해 많은 전제조건을 걸지는 않는다. 하지만 트랜잭션과 전통적인 SQL그리고 저장 프로시저 구문을 지원하는 진짜 관계형 데이터 베이스를 사용한다고 어느정도는 가정하고 있다. 실제로 관계형 데이터 베이스를 사용하지 않는다면 순수한 JDBC나 저수준 파일 입출력 API 사용을 추천한다.**

**2.4.4 그냥 작동하지 않을 경우**

**iBATIS는 커뮤니티가 성장하면서 나타나는 요구사항에 맞춰 지속적으로 많은 훌륭한 기능들을 구현하고 있다. 하지만 iBATIS도 몇몇 애플레케이션의 요구사항과는 상반되는개발 방향이나 설계 지침을 분명히 가지고 있다.이러한 경우에 대비하여 요구사항에 맞아 떨어지도록 iBATIS를 확장할 수 있는 추가 가능한 인터페이스를 제공하도록 시도한다.**

**1.5. 5분 내에 사용 가능한 iBATIS**

**간단한 데이터베이스 테이블을 조회하는 정적인 SQL 문을 설정하고, 아래와 같이 콘솔에 그 결과를 출력한다.**

**java -classpath <…> Main**

**Selected 2 records.**

**{USERNAME=LMEADORS, PASSSWORD=PICKLE, USERID=1, GROUPNAME=EMPLOYEE}**

**{USERNAME=JDOE, PASSSWORD=TEST, USERID=2, GROUPNAME=EMPLOYEE}**

**2.5.1데이터 베이스 준비하기**

**우리가 사용할 테이블을 생성하고 몇 가지 샘플 데이터를 추가하는 MySQL 스크립트이다.**

**#**

**# Table structure for table 'user'**

**#**

**CREATE TABLE USER\_ACCOUNT (**

**USERID INT(3) NOT NULL AUTO\_INCREMENT,**

**USERNAME VARCHAR(10) NOT NULL,**

**PASSSWORD VARCHAR(30) NOT NULL,**

**GROUPNAME VARCHAR(10),**

**PRIMARY KEY (USERID)**

**);**

**#**

**# Data for table 'user'**

**#**

**INSERT INTO USER\_ACCOUNT (USERNAME, PASSSWORD, GROUPNAME)**

**VALUES ('LMEADORS', 'PICKLE', 'EMPLOYEE');**

**INSERT INTO USER\_ACCOUNT (USERNAME, PASSSWORD, GROUPNAME)**

**VALUES ('JDOE', 'TEST', 'EMPLOYEE');**

**COMMIT;**

**2.5.2 코드 작성하기**

**우리의 첫 번째 완전한 형태의 예제이자 iBATIS 사용법에 대한 소개이다. 타입 안정성과예외 처리는 나중에 알아보도록 하고 여기서는 고려하지 않을 것이다.**

**import com.ibatis.sqlmap.client.\*;**

**import com.ibatis.common.resources.Resources;**

**import java.io.Reader;**

**import java.util.List;**

**public class Main {**

**public static void main(String arg[]) throws Exception {**

**String resource = "SqlMapConfig.xml";**

**Reader reader = Resources.getResourceAsReader (resource);**

**SqlMapClient sqlMap = SqlMapClientBuilder.buildSqlMapClient(reader);**

**List list = sqlMap.queryForList("getAllUsers", "EMPLOYEE");**

**System.out.println("Selected " + list.size() + " records.");**

**for(int i = 0; i < list.size(); i++) {**

**System.out.println(list.get(i));**

**}**

**}**

**}**

**10줄 정도의 자바 코드로 iBATIS 를 설정하고 구문을 실행하였으며 결과를 출력했다. 위의 자바 코드만으로 완벽하게 작동하는 iBATIS 애플리케이션을 만든 것이다. 나중에 더개선할 것이지만, 일단은 기본적인 설정 방법을 알아보자.**

**2.5.3 iBATIS 설정하기(미리보기)**

**3장에서 iBATIS의 설정에 대해 깊이 있게 다룰 것이기 때문에 여기서 간단하게만 살펴본다. 먼저 SqlMapConfig.xml 파일을 보자. 다음은 우리의 간단한 애플리케이션을 위한SqlMapConfig.xml 파일을 보여준다.**

**PUBLIC "-//ibatis.apache.org//DTD SQL Map Config 2.0//EN"**

**"http://ibatis.apache.org/dtd/sql-map-config-2.dtd">**

**value="com.mysql.jdbc.Driver"/>**

**value="jdbc:mysql://localhost/test"/>**

**value="root"/>**

**value="blah"/>**

**위 설정은 데이터 베이스에 연결하는 방법과 사용 가능한 SQL Map 파일들을 iBATIS에 지정해주는 역할을 한다. 핵심이 되는 설정파일을 보았으니, 이제 우리는 SqlMap.xml 파일을 살펴보자.**

**"http://ibatis.apache.org/dtd/sql-map-2.dtd">**

**GROUPNAME 파라미터를 위해 하나의 문자열 파라미터를 받고 결과를 HashMap에 매핑한다.**

**2.5.4 애플리케이션 빌드하기**

**빌드하기 위해 클래스 패스에 둘 필요가 있는 JAR 파일은 ibatis-common-2.jar 와 ibatis-sqlmap-2.jar 이다. 그러고 나서 자바 컴파일러에 클래스 패스를 지정하기 위해 다음과같이 명령줄을 작성할 것이다.**

**javac -classpath ibatis-common-2.jar;**

**ibatis-sqlmap-2.jar Main.java**

**2.5.5 애플리케이션 실행하기**

**애플리케이션을 실행할 때 몇 개의 추가적인 jar 파일이 있어야 하지만 많지는 않다. 애플리케이션을 실행하기 위해 클래스패스에 두어야 할 jar 파일은 ibatis-common-2.jar, ibatis-sqlmap-2.jar, commons-logging.jar 그리고 JDBC 드라이버(이 경우, mysql-connector-java.jar)이다. 이제 다음으로 아래 명령을 입력한다.**

**java -classpath ;mysql-connector.jar;commons-logging.jar;**

**ibatis-common-2.jar;ibatis-sqlmap-2.jar;.**

**Main**

**프로그램이 실행될 것이고, 얼마나 많은 행이 선택되었는지 알려주고, 다음과 같이 대강의 형태로 데이터를 출력한다.**

**Selected 2 records.**

**{USERID=1, USERNAME=LMEADORS, PASSSWORD=PICKLE, GROUPNAME=EMPLOYEE}**

**{USERID=2, USERNAME=JDOE, PASSSWORD=TEST, GROUPNAME=EMPLOYEE}**

**1.6. 미래: iBATIS는 어디로 가는가?**

**2.6.1 Apache 소프트웨어 재단**

**iBATIS 는 기술보다는 소프트웨어를 둘러싼 커뮤니티에 집중한다.Apache는 iBATIS를 보호하기 위해 필요한 것들을 제공하고, iBATIS를 원하는 사람들이 있는한 계속해서 우리곁에 있을 것이다.**

.