

- Altibase 3rd Party Connector Guide
  - 서문
    - 이 매뉴얼에 대하여
  - 1.Squirrel SQL 클라이언트
    - Altibase Plugin for Squirrel SQL Client 설치
    - Altibase Plugin 설치 및 제거
    - Altibase Plugin 특징
    - Squirrel SQL Client 설치
    - Altibase JDBC 드라이버 등록
    - Altibase와 연동
  - 2.Hibernate
    - Hibernate
  - 3.OpenLDAP
    - OpenLDAP 소개
    - 메타 데이터 설정
    - 환경 설정
    - 예제

Altibase® Application Development

# Altibase 3rd Party Connector Guide



Altibase Application Development Altibase 3rd Party Connector Guide

Release 7.1

Copyright © 2001~ 2019 Altibase Corp. All Rights Reserved.

본 문서의 저작권은 (주)알티베이스에 있습니다. 이 문서에 대하여 당사의 동의 없이 무단으로 복제 또는 전용할 수 없습니다.

**(주)알티베이스**

08378 서울시 구로구 디지털로 306 대륭포스트타워II 10층

전화: 02-2082-1114 팩스: 02-2082-1099

고객서비스포털: <http://support.altibase.com>

# 서문

## 이 매뉴얼에 대하여

이 매뉴얼은 Altibase와 연동하여 사용하는 방법을 기술한다.

## 대상 사용자

이 매뉴얼은 다음과 같은 사용자를 대상으로 작성되었다.

- 데이터베이스 관리자
- 데이터 분석가
- 시스템 및 응용 프로그램 개발자
- 시스템 관리자

다음과 같은 배경 지식을 가지고 이 매뉴얼을 읽는 것이 좋다.

- 컴퓨터, 운영 체제 및 운영 체제 유틸리티 운용에 필요한 기본 지식
- 관계형 데이터베이스 사용 경험 또는 데이터베이스 개념에 대한 이해
- 컴퓨터 프로그래밍 경험
- 데이터베이스 서버 관리, 운영 체제 관리 또는 네트워크 관리 경험

## 소프트웨어 환경

이 매뉴얼은 데이터베이스 서버로 Altibase 버전 7.1 이상을 사용한다는 가정 하에 작성되었다.

## 이 매뉴얼의 구성

이 매뉴얼은 다음과 같이 구성되어 있다.

- 제 1장 Squirrel SQL 클라이언트  
이 장에서는 Squirrel SQL Client와 Altibase Plugin for Squirrel SQL을 설치하고 Altibase JDBC를 통해 Altibase와 연동하는 방법을 설명한다.
- 제 2장 Hibernate
- 제 3장 OpenLDAP  
이 장은 openLDAP과 Altibase를 연동하는 방법을 설명한다.

## 문서화 규칙

이 절에서는 이 매뉴얼에서 사용하는 규칙에 대해 설명한다. 이 규칙을 이해하면 이 매뉴얼과 설명서 세트의 다른 매뉴얼에서 정보를 쉽게 찾을 수 있다.

여기서 설명하는 규칙은 다음과 같다.

- 샘플 코드 규칙

## 샘플 코드 규칙

코드 예제는 SQL, Stored Procedure, iSQL 또는 다른 명령 라인 구문들을 예를 들어 설명한다.

아래 테이블은 코드 예제에서 사용된 인쇄 규칙에 대해 설명한다.

규칙	의미	예제
[ ]	선택 항목을 표시	VARCHAR [(size)] [[FIXED ]] VARIABLE]
{ }	필수 항목 표시. 반드시 하나 이상을 선택해야 되는 표시	{ ENABLE   DISABLE   COMPILE }
	선택 또는 필수 항목 표시의 인자 구분 표시	{ ENABLE   DISABLE   COMPILE } [ ENABLE   DISABLE   COMPILE ]
...	그 이전 인자의 반복 표시 예제 코드들의 생략되는 것을 표시	SQL> SELECT ename FROM employee; ENAME ----- SWNO HJNO HSCHOI . . . 20 rows selected.
그 밖에 기호	위에서 보여진 기호 이 외에 기호들	EXEC :p1 := 1; acc NUMBER(11,2);
기울임 꼴	구문 요소에서 사용자가 지정해야 하는 변수, 특수한 값을 제공해야만 하는 위치	SELECT * FROM <i>table_name</i> ; CONNECT <i>userID/password</i> ;
소문자	사용자가 제공하는 프로그램의 요소들, 예를 들어 테이블 이름, 칼럼 이름, 파일 이름 등	SELECT ename FROM employee;
대문자	시스템에서 제공하는 요소들 또는 구문에 나타나는 키워드	DESC SYSTEM_.SYS_INDICES_;

## 관련 자료

자세한 정보를 위하여 다음 문서 목록을 참조한다.

- Installation Guide
- Getting Started Guide
- Administrator's Manual
- General Reference
- Error Message Reference

## Altibase는 여러분의 의견을 환영합니다.

이 매뉴얼에 대한 여러분의 의견을 보내주시기 바랍니다. 사용자의 의견은 다음 버전의 매뉴얼을 작성하는데 많은 도움이 됩니다. 보내실 때에는 아래 내용과 함께 고객센터포털(<http://support.altibase.com/kr/>)로 보내주시기 바랍니다.

- 사용 중인 매뉴얼의 이름과 버전
- 매뉴얼에 대한 의견
- 사용자의 성함, 주소, 전화번호

이 외에도 Altibase 기술지원 설명서의 오류와 누락된 부분 및 기타 기술적인 문제들에 대해서 이 주소로 보내주시면 정성껏 처리하겠습니다. 또한, 기술적인 부분과 관련하여 즉각적인 도움이 필요한 경우에도 고객센터포털을 통해 서비스를 요청하시기 바랍니다.

여러분의 의견에 항상 감사드립니다.

## 1.Squirrel SQL 클라이언트

이 장에서는 Squirrel SQL Client와 Altibase Plugin for Squirrel SQL을 설치하고 Altibase JDBC를 통해 Altibase와 연동하는 방법을 설명한다.

### Altibase Plugin for Squirrel SQL Client 설치

Altibase Plugin for Squirrel SQL Client(이하 Altibase Plugin) 은 Altibase에 접속하여 Squirrel SQL Client를 사용하기 위한 Plugin이다. Altibase Plugin은 Squirrel SQL Client의 공식 Plugin이 아니기 때문에, Squirrel SQL Client와 별도로 사용자가 직접 다운로드 받아 설치해야 한다.

### 시스템 요구사항

Altibase plugin을 설치하고 실행하기 위해 필요한 소프트웨어 요구사항과 Altibase Plugin과 호환되는 데이터베이스 관리 시스템을 열거한다.

---

소프트웨어 요구사항	Squirrel SQL Client 3.7.1 또는 3.8.0
호환 가능한 데이터베이스 시스템	Altibase 6.5.1 또는 그 이상의 버전

## Altibase Plugin 설치 및 제거

Altibase Plugin을 사용하기 위한 설치 및 준비 방법과 제거 방법에 대해서 기술한다.

### 설치 절차

1. Squirrel SQL Client 다운로드: <http://squirrel-sql.sourceforge.net/> 에서 Squirrel SQL Client를 다운로드하여 설치한다.
2. Altibase Plugin 다운로드: <http://support.altibase.com> 다운로드 항목에서 altibase4squirrel.zip 을 다운 받은 후 압축을 푼다. 압축 해제된 파일들 중 altibase4squirrel.jar(Altibase Plugin) 파일을, 설치한 Squirrel SQL Client 디렉토리의 plugins 하위 디렉토리에 복사한다.(예, C:\Program Files\squirrel-sql-3.8.0\plugins\altibase4squirrel.jar)
3. Squirrel SQL Client 실행: 이제 Altibase Plugin을 사용할 준비가 완료되었다. Squirrel SQL Client를 실행시키면 altibase4squirrel.jar가 자동으로 로딩된다. Plugin 로딩은 Squirrel SQL Client 수행 후 , 메인 메뉴 중 ‘플러그인-요약’ 을 클릭하여 확인 가능하다. Squirrel SQL Client 구동은 Windows는 squirrel-sql.bat를, Linux는 [squirrel-sql.sh](#) 를 실행시킨다. Squirrel SQL Client 구동에 대한 자세한 내용은 Squirrel SQL Client 매뉴얼을 참조한다.□

### 제거

Altibase Plugin의 제거는 Squirrel SQL Client 프로그램을 종료한 후, Squirrel SQL Client plugin 디렉토리에서 altibase4squirrel.jar를 삭제하면 된다.

## Altibase Plugin 특징

Altibase Plugin의 특징은 다음과 같다.

### Altibase 전용 객체(Object)

Altibase의 database 객체를 Squirrel SQL Client Object Tree에서 보여준다.

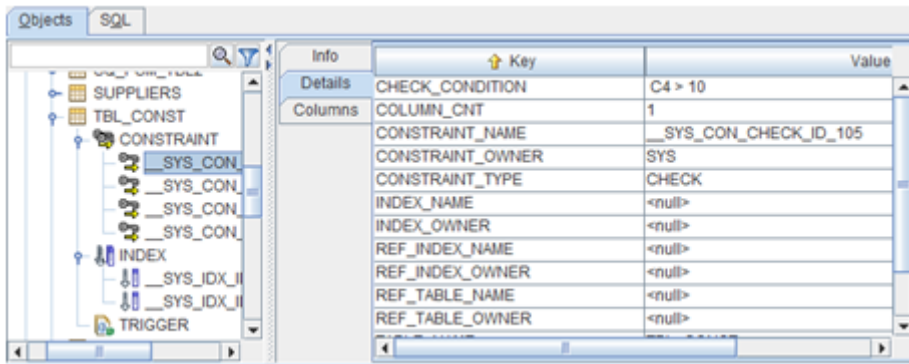
보여주는 객체는 다음과 같다.

객체 유형	종 류
Table	Table, Queue, Materialized View, System Table, System View

객체 유형	종 류
PSM	Stored Procedure, Function, Packages, Trigger
기타	Index, Sequence, Synonym, Library, Job

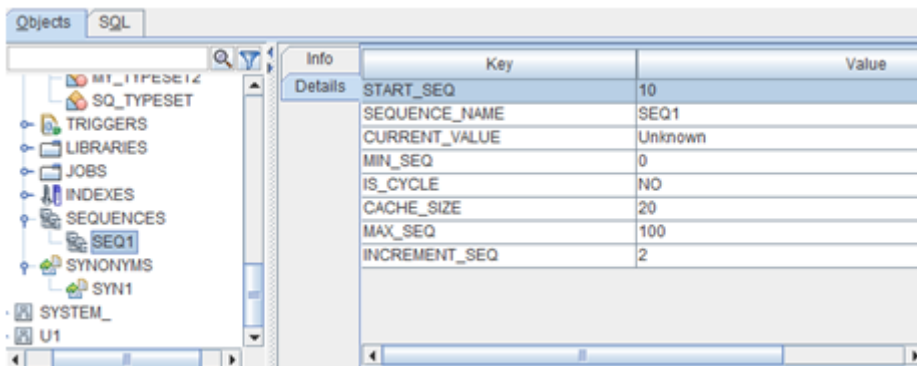
## Table 및 종속 객체(Constraint, Index, Trigger)

Table 객체에 대해 해당 테이블에 종속된 Constraint, Index, Trigger 객체를 테이블의 하위 객체로 표현한다.



## 객체 상세정보

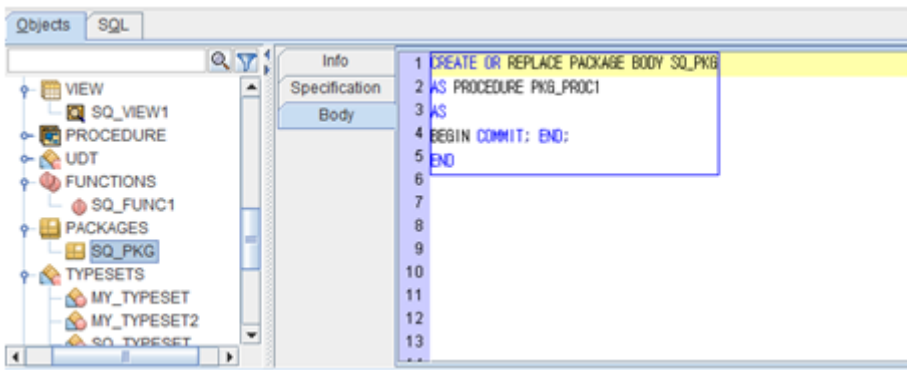
데이터베이스 객체에 대한 상세 정보를 보여준다. 예를 들어 Index, Sequence, Synonym, Constraint등이 객체들에 대해 상세 정보를 지원한다.



## Source View

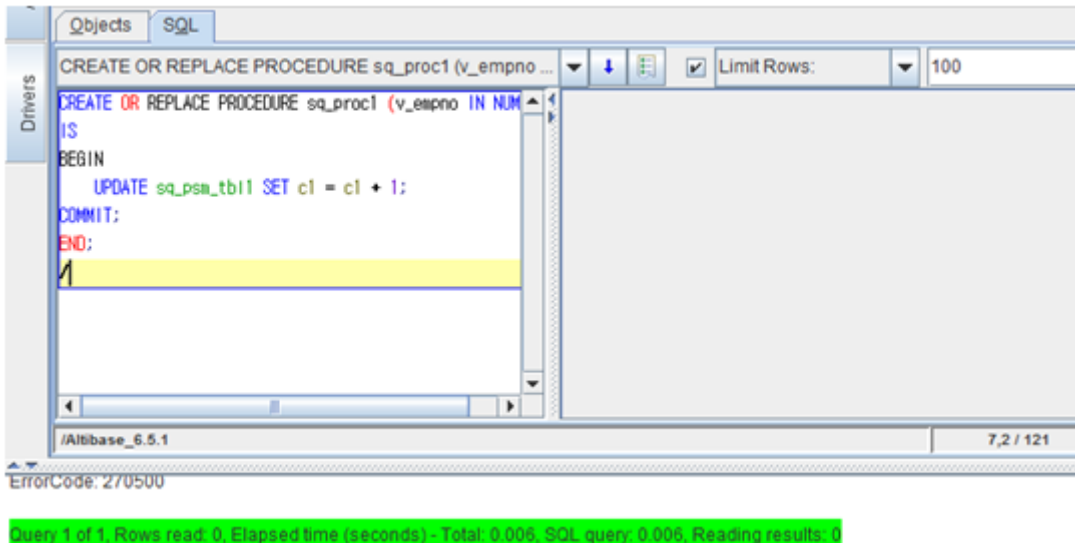
**PSM 객체 타입 및 필요한 객체들에 대해 다음의 객체 생성 DDL을 보여준다.**

Stored Procedure, Function, Package, Typeset, Trigger, Library, Synonym, View, Materialized View



## PSM DDL grammar 지원

Squirrel SQL Client의 SQL editor에서 Altibase PSM DDL grammar를 지원한다. altibase.jar를 설치하지 않은 경우, Squirrel SQL Client 자체 parser로는 Altibase PSM 생성 DDL을 파싱할 수 없어 실패하게 된다.



## Squirrel SQL Client 설치

이 장에서는 Squirrel SQL Client를 설치하는 방법을 설명한다.

### 설치 절차

1. 아래 링크에서 자신의 시스템에 설치 가능한 Squirrel SQL Client를 다운받는다.  
<http://squirrel-sql.sourceforge.net/#installation>
2. 다운로드 받은 "squirrel-sql-3.7.1-standard.jar"을 더블 클릭하여 설치를 시작한다.

## Altibase JDBC 드라이버 등록

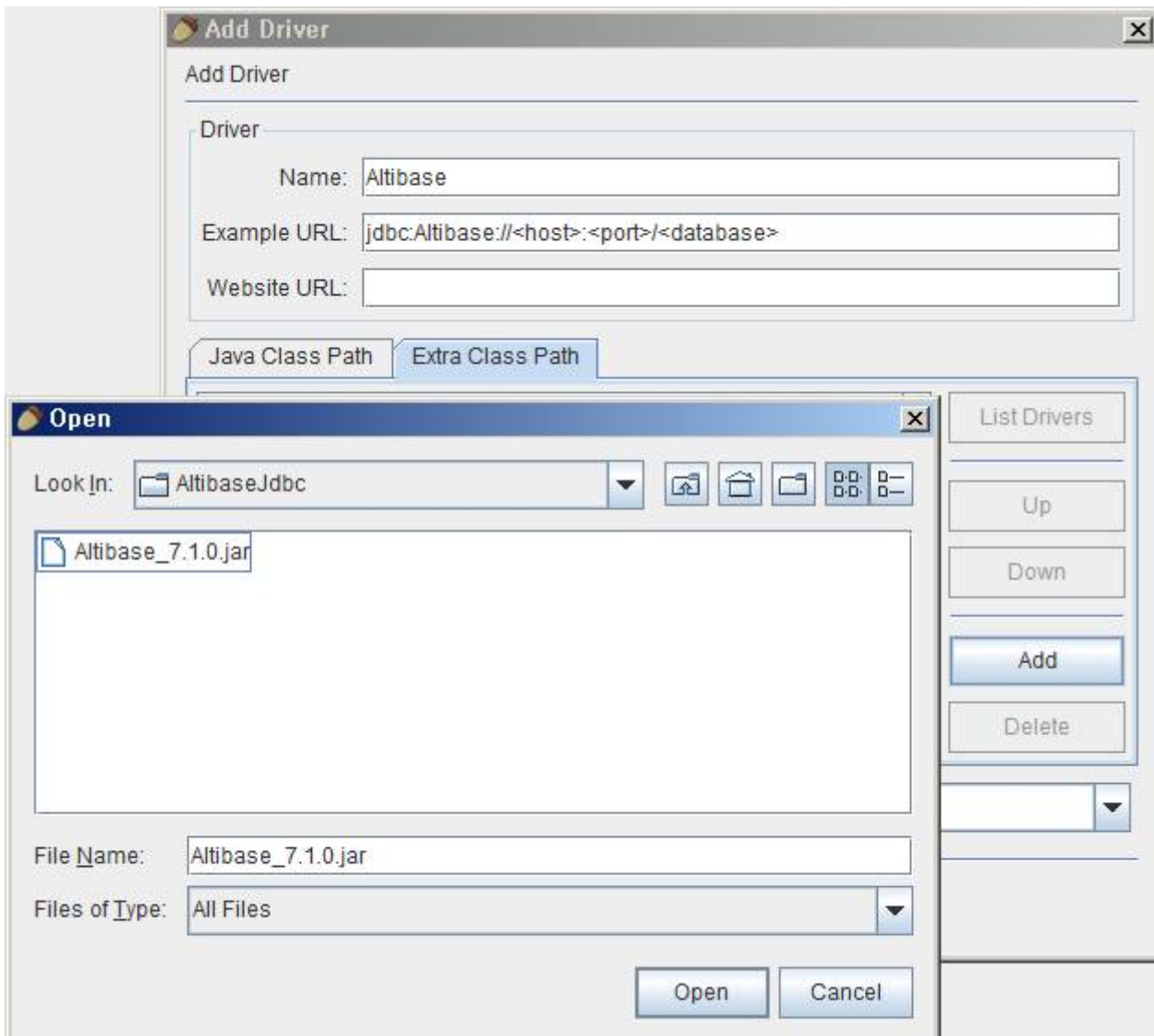
### 등록 방법

1. Squirrel SQL Client를 설치한 디렉토리의 "squirrel-sql.bat"를 더블 클릭하여 프로그램을 구동시킨다.
2. Drivers -> New Driver를 클릭하여 Altibase JDBC driver를 등록한다. Drivers 메뉴가 비활성화 되어 있다면 Windows -> View Drivers를 클릭하여 활성화 시킨다.
3. Add Driver 창의 Driver항목에 아래와 같이 입력한다.

Name: Altibase

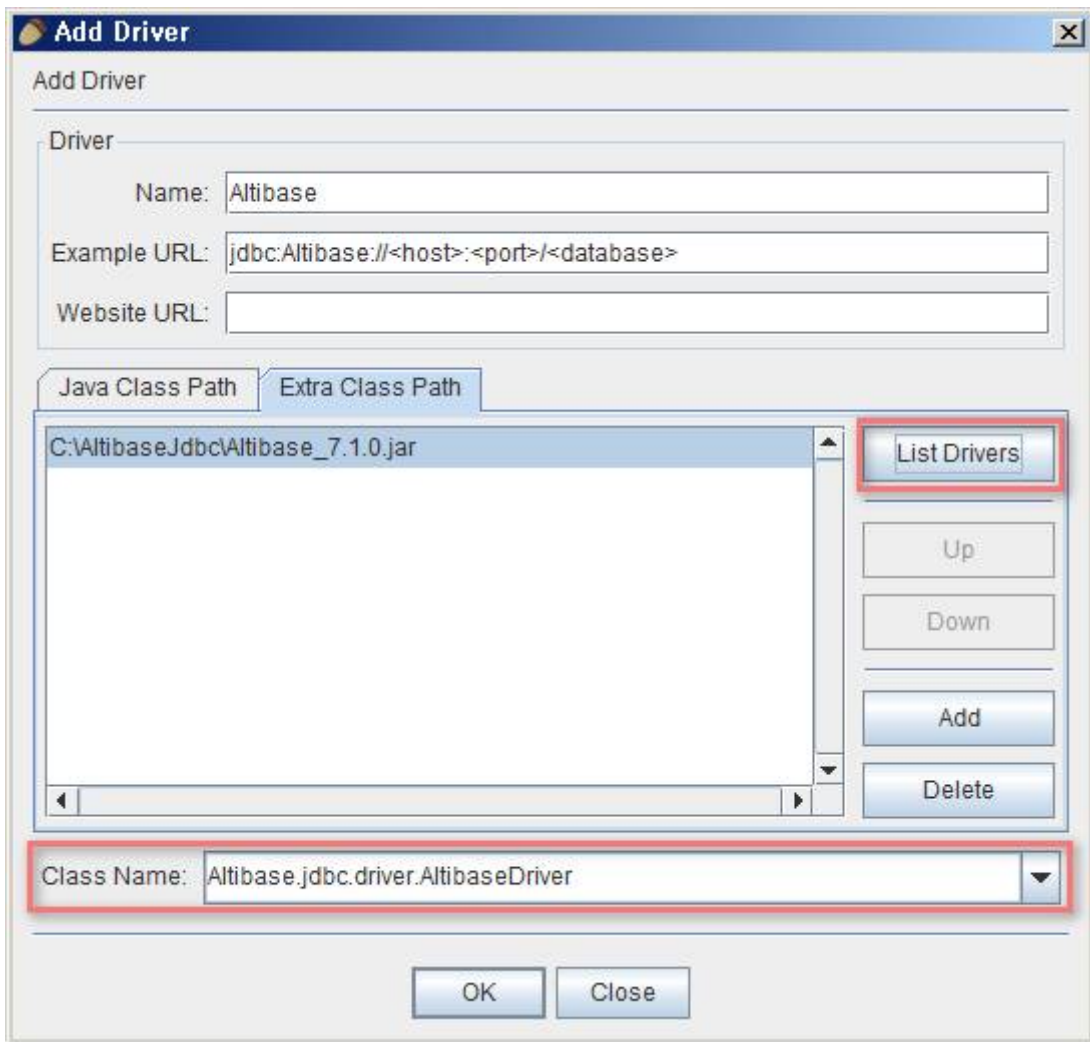
Example URL: jdbc:Altibase://<host>:<port>/<database>

4. 하단의 Extra Class Path 탭의 "Add" 버튼을 클릭하여 접속에 사용할 Altibase JDBC driver file을 선택한다.



파일 선택 후 "List Drivers" 버튼을 눌러, 하단의 Class Name 콤보 박스에 "Altibase.jdbc.driver.AltibaseDriver"가 추가되었는지 확인한다.



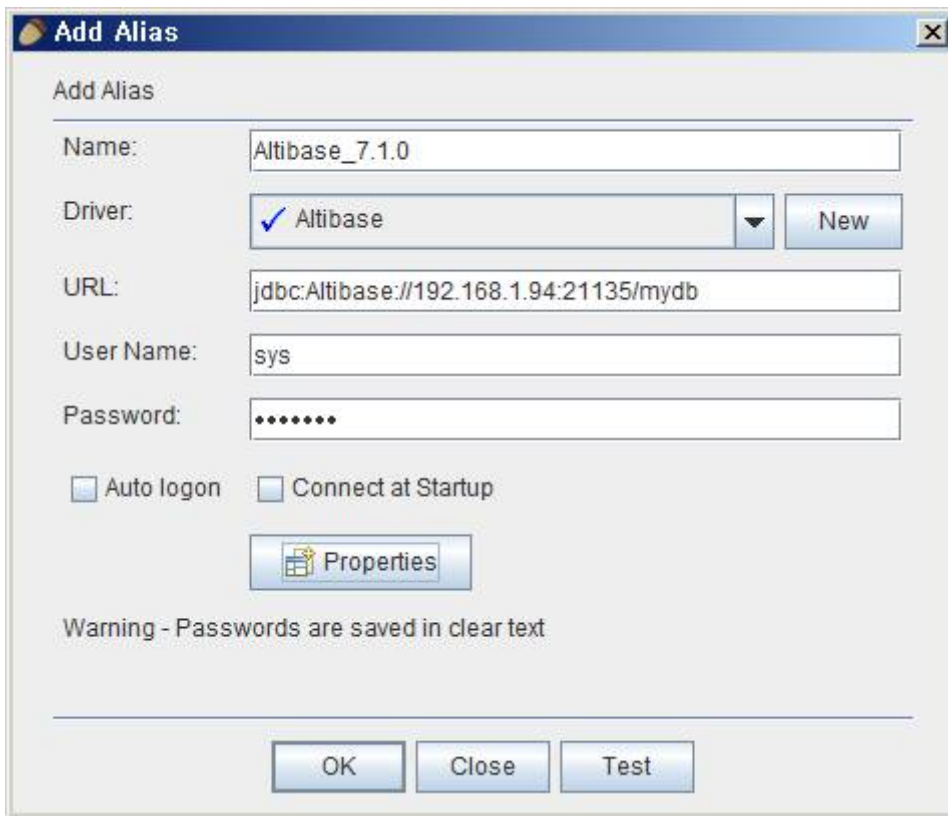


5. "OK" 버튼을 눌러 Altibase JDBC driver 등록을 완료한다

## Altibase와 연동

### 데이터베이스 접속 정보 등록

1. 메인 메뉴의 Aliases -> New Alias를 클릭하여 접속할 데이터베이스 정보를 등록한다. Aliases 메뉴가 비활성화되어 있으면, Windows -> View Aliases를 클릭하여 활성화시킨다.
2. 아래 정보를 입력한다.  
Name: 데이터베이스 접속 이름  
Driver: 앞서 등록한 Altibase를 선택한다.  
URL: 접속할 host, port, databasename을 입력한다  
User Name: 접속에 이용할 데이터베이스 사용자 이름  
Password: 접속에 이용할 데이터베이스 사용자의 암호
3. 입력완료 후 하단의 "Test" 버튼을 눌러 입력한 접속정보가 올바른지 확인한다.



**Add Alias**

Add Alias

Name: Altibase\_7.1.0

Driver: ✓ Altibase New

URL: jdbc:Altibase://192.168.1.94:21135/mydb

User Name: sys

Password: .....

☐ Auto logon ☐ Connect at Startup

Properties

Warning - Passwords are saved in clear text

OK Close Test

## 데이터베이스 사용

1. 메인 메뉴의 Aliases -> Connect를 클릭하여 접속창을 연다.



**Connect to: Altibase\_7.1.0**

Connect to: Altibase\_7.1.0

Alias: Altibase\_7.1.0

Driver: Altibase

URL: jdbc:Altibase://192.168.1.94:21135/...

User: sys

Password: .....

Properties

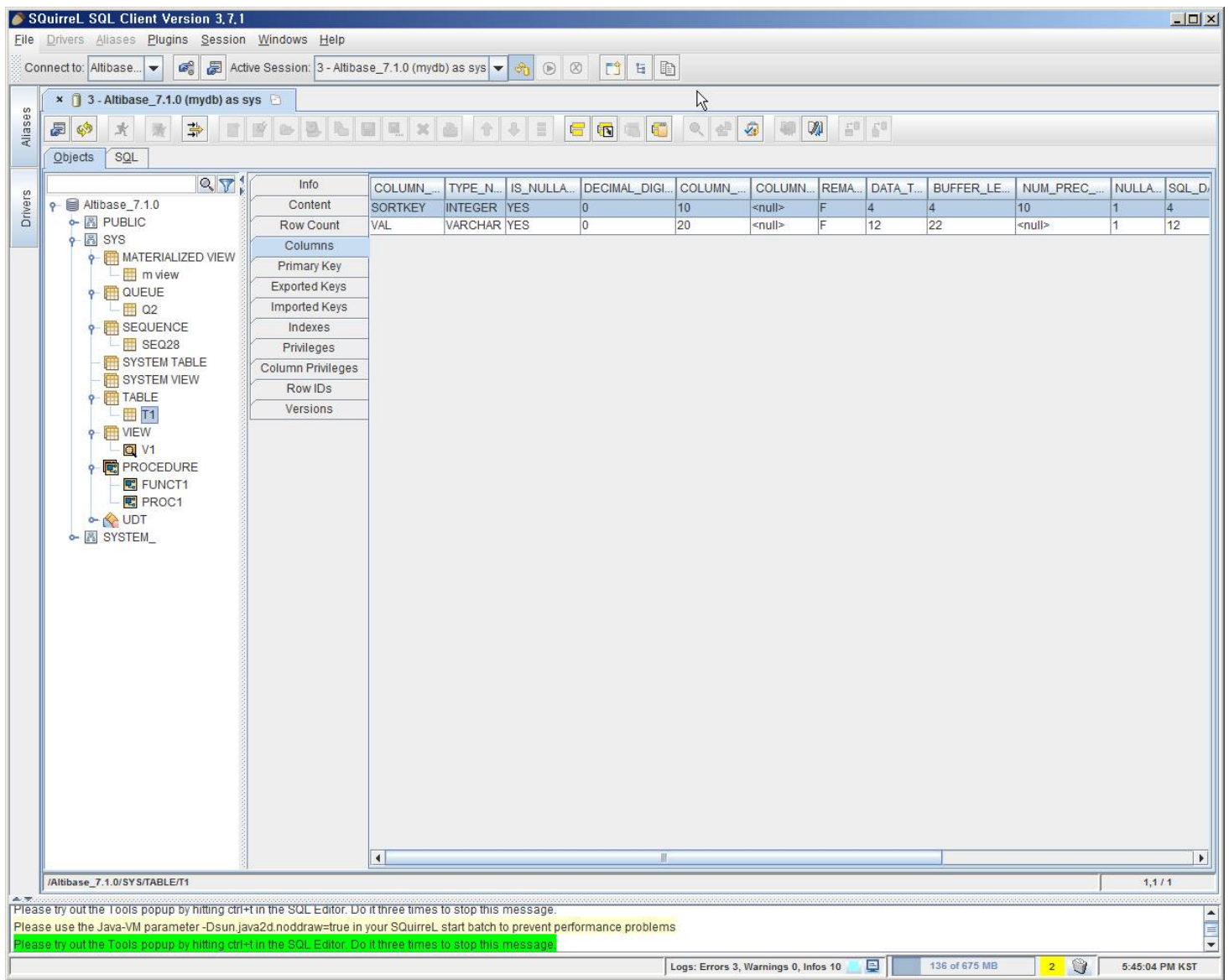
Warning - Caps lock may interfere with passwords

Connect Close

2. "Connect" 버튼을 눌러 등록된 데이터베이스에 접속한다.

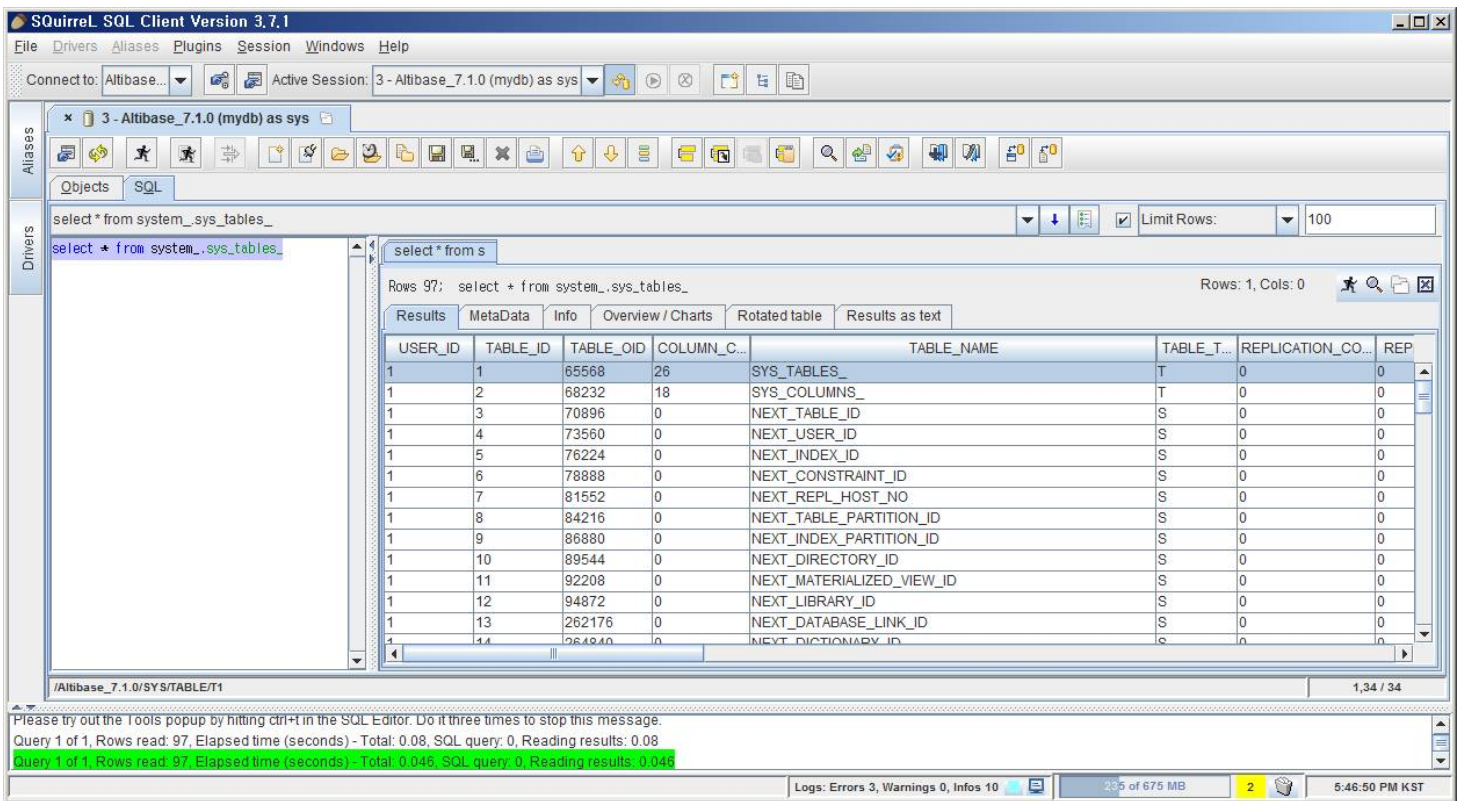
## Objects

접속에 성공하면 아래와 같이 두 개의 탭으로 구성된 창이 열린다. Object창이 기본으로 선택되는데, 연결된 데이터베이스에 존재하는 객체 및 상세정보를 볼 수 있다.



## SQL

상단의 2개 탭 중 SQL을 선택하여 사용자가 쿼리를 입력하고 실행할 수 있다.



## 2.Hibernate

### Hibernate

Altibase는 비표준SQL을 제공하며, Hibernate는 이러한 기능을 수행할 수 있도록 Dialect 클래스를 지원한다.

Hibernate에서 Altibase를 연동하려면 Altibase의 JDBC Driver를 설정하고, Hibernate의 configuration에 AltibaseDialect.class를 지정해야 한다.

### AltibaseDialect

Hibernate가 공식적으로 제공하는 라이브러리는 AltibaseDialect.class를 포함하지 않기 때문에 AltibaseDialect.java파일 (필요에 따라

[AltibaseLimitHandler.java](#)포함)을

컴파일하고 Hibernate가 제공하는 파일에 포팅해야 사용할 수 있다.

AltibaseDialect.java파일과

[AltibaseLimitHandler.java](#)파일은

Altibase Github사이트에서 제공한다.

상세한 사용 방법은 [AltibaseDialect 포팅방법](#)

([https://github.com/ALTIBASE/hibernate-orm/blob/master/ALTIBASE\\_DIALECT\\_PORTING.md](https://github.com/ALTIBASE/hibernate-orm/blob/master/ALTIBASE_DIALECT_PORTING.md))을 참고한다.

# 3.OpenLDAP

이 장은 openLDAP과 Altibase를 연동하는 방법을 설명한다.

## OpenLDAP 소개

OpenLDAP은 경량형 디렉터리 접근 프로토콜(Light-weight Directory Access Protocol, LDAP)을 오픈 소스로 구현한 것이다.

OpenLDAP은 backend 저장소로 RDBMS를 지원하는데 이를 SQL Backend 또는 back-sql이라고 한다. back-sql은 RDBMS와 연동시 ODBC를 사용한다.

## OpenLDAP 설치

1. Download OpenLDAP

<http://www.openldap.org/>

2. Unpack

```
$ tar xvfz name_of.tgz
```

3. Configure, Make, and Install

```
$ ./configure --prefix=... --enable-sql
./configure --help 참조
$ make depend
$ make
$ make install (su root -c 'make install')
```

## Altibase 설치

Altibase는 6.5.1 이상을 설치할 것을 권장한다.

설치에 대한 자세한 설명은 Installation Guide를 참조한다.

## 메타 데이터 설정

LDAP용 메타 테이블에 LDAP의 objectClass와 attributeType들을 Altibase의 어떤 테이블과 컬럼으로 매핑할 것인지 저장해야 한다.

Altibase는 4개의 LDAP용 메타 테이블을 설정하여 사용할 수 있다.

- ldap\_oc\_mappings
- ldap\_attr\_mappings
- ldap\_entries

- ldap\_entry\_objclasses

## ldap\_oc\_mappings

objectClass 맵핑 테이블로써 structuralObjectClass와 테이블간의 관계를 저장한다.

컬럼 이름	설명
ID	고유 ID, 다른 ldap_* 테이블을 참조할 때 사용됨
NAME	objectClass 이름. slapd의 schema에 로딩된 objectClass 중 하나의 이름과 일치해야 함.
KEYTBL	objectClass에 대응하는 테이블의 이름. 예제에서는 inetOrgPerson objectClass가 person 테이블로 맵핑됨
KEYCOL	"keytbl" 테이블의 primary key 컬럼의 이름
CREATE_PROC	엔트리를 추가할 때 수행할 SQL 문
CREATE_KEYVAL	"CREATE_PROC" SQL 문으로 입력된 새로운 ID 값을 가져오는 쿼리문. 이 쿼리를 사용하려면 slapd.conf에서 create_needs_select를 "yes"로 설정해야 함.
DELETE_PROC	엔트리를 삭제할 때 수행할 SQL문 "keytbl.keycol"을 호스트 변수로 취한다.
EXPECT_RETURN	<p>CREATE_PROC과 DELETE_PROC의 SQL문이 리턴 값을 반환하는지 여부를 나타내는 비트열의 정수값.  첫 번째 비트는 ADD_PROC, 두 번째 비트는 DELETE_PROC에 해당한다.</p> <p>사용 가능한 값  0: CREATE_PROC과 DELETE_PROC의 SQL문 모두 리턴값이 없음  1: CREATE_PROC의 SQL문이 값을 반환함  2: DELETE_PROC의 SQL문이 값을 반환함  3: CREATE_PROC과 DELETE_PROC의 SQL문 모두 값을 반환함</p> <p>리턴 값이 있는 SQL문(프로시저 또는 함수) 작성시 지켜야 할 사항  CREATE_PROC의 리턴 값은 입력된 새로운 ID 값이어야 함.  DELETE_PROC의 리턴 값은 성공하면 0, 그렇지 않으면 1을 반환해야 함.  리턴 값을 받는 outbinding 파라미터를 맨 앞에 위치시켜야 함.</p>

## ldap\_attr\_mappings

attributeType 맵핑 테이블로써 특정 objectClass를 위한 attributeType이 어떻게 SQL 문과 맵핑되는지를 저장한다.

컬럼 이름	설명
ID	고유 ID
OC_MAP_ID	ldap_oc_mappings 테이블의 ID를 가리키며, 해당 attributeType이 어느 objectClass에 속하는지를 나타냄.
NAME	attributeType의 이름. slapd의 schema에 로딩된 attributeType 중 하나의 이름과 일치해야 함.
SEL_EXPR	아래 attribute를 select 하기 위한 expression. ("select <sel_expr> from ..." 부분)
FROM_TBLS	아래 attribute를 가져올 테이블. ("select ... from <from_tbls > where ..." 부분)
JOIN_WHERE	아래 attribute를 select 하기 위한 조건. ("select ... where <join_where> ..." 부분)
ADD_PROC	해당 attribute에 입력하기 위한 SQL문. 추가될 attribute의 값과 그것에 연관된 엔트리의 "keytbl.keycol"을 호스트 변수로 취한다.
DELETE_PROC	해당 attribute에서 삭제하기 위한 SQL문. 삭제될 attribute의 값과 그것에 연관된 엔트리의 "keytbl.keycol" 을 호스트 변수로 취한다.

컬럼 이름	설명
PARAM_ORDER	<p>ADD_PROC과 DELETE_PROC의 SQL문에서 keytbl.keycol 값이 attribute의 값보다 앞에 오는지 뒤에 오는지를 나타내는 비트열의 정수값. keytbl.keycol 값이 attribute의 값보다 앞에 오면 0, 뒤에 오면 1을 설정하면 된다. 첫 번째 비트는 ADD_PROC, 두 번째 비트는 DELETE_PROC에 해당한다.</p> <p>사용 가능한 값  0: CREATE_PROC과 DELETE_PROC의 SQL문 모두 keytbl.keycol 값이 attribute의 값보다 앞에 옴.  1: CREATE_PROC의 SQL문은 keytbl.keycol 값이 attribute의 값보다 뒤에 옴.  2: DELETE_PROC의 SQL문은 keytbl.keycol 값이 attribute의 값보다 뒤에 옴.  3: CREATE_PROC과 DELETE_PROC의 SQL문 모두 keytbl.keycol 값이 attribute의 값보다 뒤에 옴.</p>
EXPECT_RETURN	<p>ADD_PROC과 DELETE_PROC의 SQL문이 리턴값을 반환하는지 여부를 나타내는 비트열의 정수값.</p> <p>사용 가능한 값  0: CREATE_PROC과 DELETE_PROC의 SQL문 모두 리턴값이 없음  1: CREATE_PROC의 SQL문이 값을 반환함  2: DELETE_PROC의 SQL문이 값을 반환함  3: CREATE_PROC과 DELETE_PROC의 SQL문 모두 값을 반환함</p> <p>리턴값이 있는 SQL문(프로시저 또는 함수) 작성시 지켜야 할 사항  CREATE_PROC과 DELETE_PROC의 리턴값은 성공하면 0, 그렇지 않으면 1을 반환해야 함.  리턴값을 받는 outbinding 파라미터를 맨 앞에 위치시켜야 함.</p>

## ldap\_entries

엔트리 매핑 테이블이다. LDAP 트리의 각 엔트리를 식별할 수 있는 DN 값을 저장한다.

컬럼 이름	설명
ID	고유 ID
DN	엔트리의 DN이 "pretty" form으로 저장됨.



컬럼 이름	설명
OC_MAP_ID	엔트리의 메인 objectClass에 해당하는 테이블의 ID. ldap_oc_mappings 테이블의 ID 값 중에 하나에 해당함.
PARENT	부모 엔트리의 ID. "suffix" 엔트리는 이 값이 0이다.
KEYVAL	엔트리의 데이터를 갖고 있는 테이블의 해당 로우의 primary key 컬럼의 값. 즉, 이 엔트리의 objectClass에 해당하는 테이블의 primary key 컬럼 값 ("keytbl.keycol" 값)

## ldap\_entry\_objclasses

objectClass 맵핑 테이블이다. 엔트리가 어떤 보조 objectClass(auxiliary objectClasses)를 가지는지를 저장한다.

컬럼 이름	설명
ENTRY_ID	ldap_entries.id의 값
NAME	auxiliary objectClass의 이름. slapd의 schema에 로딩된 objectClass 중 하나의 이름과 일치해야 함.

## 환경 설정

OpenLDAP 환경 설정을 위해 ODBC 설정 및 slapd.conf 파일을 편집한다.

### .odbc.ini

```
[ldap_altibase]
Description = ODBC for Altibase
Driver = /home/altibase/altibase_home/lib/libaltibase_odbc-64bit-ul64.so
server = 127.0.0.1
port = 20030
```

odbc 추적 설정: back-end 오류 발생시 문제 추적을 위해 아래의 설정을 이용할 수 있다.

### .odbcinst.ini

```
[ODBC]
TraceFile = /home/altibase/odbc.log
Trace = Yes
```

# slapd.conf

Altibase용 샘플로 제공된 slapd.conf를 참고하여 slapd.conf를 편집한다.

```
# $OpenLDAP$
#
# See slapd.conf(5) for details on configuration options.
# This file should NOT be world readable.
#
include      /usr/local/etc/openldap/schema/core.schema
include      /usr/local/etc/openldap/schema/cosine.schema
include      /usr/local/etc/openldap/schema/inetorgperson.schema

# Define global ACLs to disable default read access.

# Do not enable referrals until AFTER you have a working directory
# service AND an understanding of referrals.
#referral    ldap://root.openldap.org

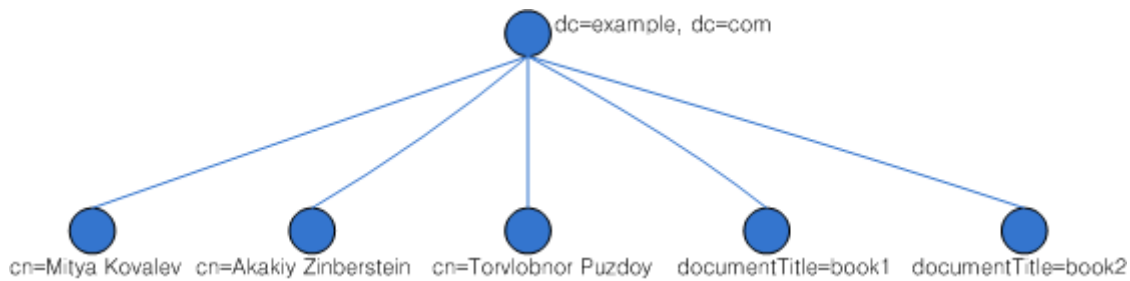
pidfile      /usr/local/var/slapd.pid
argsfile     /usr/local/var/slapd.args

#####
# sql database definitions
#####

database     sql
suffix       "dc=example,dc=com"
rootdn       "cn=Manager,dc=example,dc=com"
rootpw       secret
dbname       ldap_altibase --ODBC dsn 이름
dbuser       ldap --Altibase 사용자 이름
dbpasswd     ldap --db 사용자 pw
subtree_cond "upper(ldap_entries.dn) LIKE CONCAT('%',upper(?))"
insentry_stmt "insert into ldap_entries (id,dn,oc_map_id,parent,keyval) values (ldap_entry_ids.nextval,?,?,?,?)"
has_ldapinfo_dn_ru no
upper_func   UPPER
```

## 예제

아래 그림과 같은 DIT(Directory Information Tree)를 RDBMS인 Altibase에 구축한다.



root DN은 dc=example,dc=com이며, Idif (LDAP Data Interchange Files, 트리 데이터를 텍스트로 표현한 파일)는 아래와 같다. root DN "dc=example,dc=com" 외에는 샘플 파일을 참고하기 바란다. 샘플 파일은

[https://github.com/ALTIBASE/openldap/tree/master/servers/slapd/back-sql/rdbms\\_depend/altibase](https://github.com/ALTIBASE/openldap/tree/master/servers/slapd/back-sql/rdbms_depend/altibase) 에 있다.

# Mitya Kovalev, example.com  
dn: cn=Mitya Kovalev,dc=example,dc=com  
objectClass: inetOrgPerson  
cn: Mitya Kovalev  
sn: Kovalev  
seeAlso: documentTitle=book1,dc=example,dc=com  
seeAlso: documentTitle=book2,dc=example,dc=com  
givenName: Mitya  
userPassword:: bWl0  
telephoneNumber: 222-3234  
telephoneNumber: 332-2334

# Akakiy Zinberstein, example.com  
dn: cn=Akakiy Zinberstein,dc=example,dc=com  
objectClass: inetOrgPerson  
cn: Akakiy Zinberstein  
sn: Zinberstein  
givenName: Akakiy

# Torvlobnor Puzdoy, example.com  
dn: cn=Torvlobnor Puzdoy,dc=example,dc=com  
objectClass: inetOrgPerson  
cn: Torvlobnor Puzdoy  
sn: Puzdoy  
seeAlso: documentTitle=book1,dc=example,dc=com  
givenName: Torvlobnor  
telephoneNumber: 545-4563

# book1, example.com  
dn: documentTitle=book1,dc=example,dc=com  
objectClass: document  
description: abstract1  
documentTitle: book1  
documentAuthor: cn=Mitya Kovalev,dc=example,dc=com  
documentAuthor: cn=Torvlobnor Puzdoy,dc=example,dc=com  
documentIdentifier: document 1

# book2, example.com  
dn: documentTitle=book2,dc=example,dc=com  
objectClass: document  
description: abstract2  
documentTitle: book2  
documentAuthor: cn=Mitya Kovalev,dc=example,dc=com  
documentIdentifier: document 2

# example.com  
dn: dc=example,dc=com  
objectClass: organization  
objectClass: dcObject  
o: Example  
dc: example

이 엔트리는 structural objectClass "organization"과 auxiliary objectClass인 "dcObject"를 포함(contain)하며, 각각의 정의는 아래와 같다.

objectClass를 DB 테이블로 매핑하기 위해서는 objectClass의 attribute member들 중에서 선택해서 컬럼화해야 한다. 이 예제에서는 "o"와 "dc" attribute를 선택했다.

```
objectclass ( 2.5.6.4 NAME 'organization'
  DESC 'RFC2256: an organization'
  SUP top STRUCTURAL
  MUST o
  MAY ( userPassword $ ... ) )

objectclass ( 1.3.6.1.4.1.1466.344 NAME 'dcObject'
  DESC 'RFC2247: domain component object'
  SUP top AUXILIARY MUST dc )
```

아래는 위의 objectClass를 Altibase의 테이블로 매핑한 "institutes" 테이블의 생성 구문과 데이터 삽입, 검색, 변경 등을 위한 메타 데이터를 생성하는 SQL문이다.

```

CREATE TABLE institutes (
    id int NOT NULL,
    name varchar(255)
);
insert into institutes (id,name) values (institute_ids.nextval,'Example');

insert into ldap_oc_mappings (id,name,keytbl,keycol,create_proc,delete_proc,expect_return)
values (3,'organization','institutes','id','exec create_org(?)','exec delete_org(?)',1);

insert into ldap_attr_mappings (id,oc_map_id,name,sel_expr,from_tbls,join_where,add_proc,delete_proc,param_order,
values (11,3,'o','institutes.name','institutes',NULL,'exec set_org_name(?,?)',NULL,0,0);

insert into ldap_attr_mappings (id,oc_map_id,name,sel_expr,from_tbls,join_where,add_proc,delete_proc,param_order,
values (12,3,'dc','lower(institutes.name)','institutes,ldap_entries dcObject,ldap_entry_objclasses auxObjectClass',
'institutes.id=dcObject.keyval AND dcObject.oc_map_id=3 AND dcObject.id=auxObjectClass.entry_id AND auxObjectClass.oc_map_id=12',
NULL,NULL,0,0);

insert into ldap_entries (id,dn,oc_map_id,parent,keyval)
values (ldap_entry_ids.nextval,'dc=example,dc=com',3,0,1); -- id = 1

insert into ldap_entry_objclasses (entry_id,oc_name) values (1,'dcObject');

CREATE OR REPLACE PROCEDURE create_org(keyval OUT INTEGER) AS
BEGIN
INSERT INTO institutes (id,name) VALUES (institute_ids.nextval,' ');
SELECT institute_ids.currval INTO keyval FROM DUAL;
END;
/

CREATE OR REPLACE PROCEDURE delete_org(keyval IN INTEGER) AS
BEGIN
DELETE FROM institutes WHERE id=keyval;
END;
/

CREATE OR REPLACE PROCEDURE set_org_name(keyval IN INTEGER, new_name IN varchar(255)) AS
BEGIN
UPDATE institutes SET name=new_name WHERE id=keyval;
END;
/

```

위 샘플 예제는 스크립트 파일로 제공되며, 아래와 같이 샘플 데이터를 구축할 수 있다. 사용된 스크립트 파일은

[https://github.com/ALTIBASE/openldap/tree/master/servers/slapd/back-sql/rdbms\\_depend/altibase](https://github.com/ALTIBASE/openldap/tree/master/servers/slapd/back-sql/rdbms_depend/altibase)에

서  
다운로드할 수 있다.

## Idap 사용자 생성

sys 유저로 isql 접속 후 아래 SQL 문 실행

```
drop user ldap cascade;  
create user ldap identified by ldap;
```

## 데이터 생성

```
$ isql -s localhost -u ldap -p ldap -f backsql_create.sql  
$ isql -s localhost -u ldap -p ldap -f testdb_create.sql  
$ isql -s localhost -u ldap -p ldap -f testdb_metadata.sql  
$ isql -s localhost -u ldap -p ldap -f testdb_data.sql
```

- backsql\_create.sql: 맵핑용 메타 데이터 저장을 위한 테이블 생성
- testdb\_create.sql: 사용할 objectClass를 DB 테이블로 생성
- testdb\_metadata.sql: 맵핑용 메타데이터 입력
- testdb\_data.sql: objectClass 데이터 입력