# 목 차


Ⅰ. 시작하기 전에 ..... 5

오즈 쿼리 디자이너 개요 ..... 6

본 매뉴얼의 목적 ..... 6

본 매뉴얼 구성 ..... 7

매뉴얼 로드맵 ..... 10

Ⅱ. 오즈 쿼리 디자이너 실행 ..... 11

시스템 요구사항 ..... 12

오즈 쿼리 디자이너 파일 구성 ..... 13

오즈 쿼리 디자이너 실행 ..... 16

Ⅲ. 오즈 쿼리 디자이너 윈도우 ..... 19

오즈 쿼리 디자이너 화면 구성 ..... 20

타이틀바 ..... 22

메뉴바 ..... 23

툴바 ..... 37

데이터 창, 리파지토리 창 ..... 39

쿼리 창 ..... 42

속성 창 ..... 44

스크립트 창 ..... 45

Ⅳ. 오즈 쿼리 디자이너 사용 ..... 49

작업 폴더 설정 ..... 50

스크립트 입력, 비교 ..... 54

파일 경로 설정 ..... 64

Ⅴ. 데이터 소스와 데이터 셋 ..... 70
데이터 소스 개요 ..... 71
데이터 셋 개요 ..... 71
데이터 소스 종류 ..... 72
데이터 소스별 속성 ..... 75
데이터베이스 연결 ..... 104
동적 데이터베이스 연결 ..... 122

Ⅵ. 쿼리 데이터 셋 디자인 ..... 128
매뉴얼 모드로 쿼리 디자인하기 ..... 130
디자인뷰 모드로 쿼리 디자인하기 ..... 132
사용자 지정 패러미터 추가하기 ..... 140
패러미터 마법사 사용하기 ..... 142
쿼리 마법사 사용하기 ..... 144
수식필드 추가하기 ..... 151
INCLUDE ODI를 통한 필드 추가하기 ..... 157
마스터/디테일 셋 디자인하기 ..... 160
멀티 데이터 셋 디자인하기 ..... 163
다이나믹 데이터 셋 디자인하기 ..... 167
프로시저 데이터 셋 디자인하기 ..... 171
컴파일된 질의문 디자인하기 ..... 177
행 삽입/삭제/변경 쿼리 디자인하기 ..... 179

Ⅶ. 사용자 데이터 스토어 ..... 192
사용자 데이터 스토어 개요 ..... 193
사용자 데이터 스토어 이용 ..... 195

Ⅷ. 그룹 데이터 스토어 ..... 202
그룹 데이터 스토어 개요 ..... 203
그룹 데이터 스토어 이용 ..... 207

Ⅸ. 파일/HTTP 스토어 ..... 211

파일/HTTP 스토어 개요 ..... 212

파일 스토어 이용 ..... 215

HTTP 스토어 이용 ..... 228

Ⅹ. SOAP 스토어 ..... 239

SOAP 개요 ..... 240

SOAP 스토어 이용 ..... 241

XI. ADO DataSet 웹서비스 스토어 ..... 249

ADO DataSet 웹서비스 개요 ..... 250

ADO DataSet 웹서비스 스토어 이용 ..... 251

XⅡ. XML 스토어 ..... 255

XML 개요 ..... 256

XML 스토어 이용 ..... 257

XIII. SAP 스토어 ..... 265

SAP 개요 ..... 266

SAP 스토어 이용 ..... 267

XIV. CLEAR QUEST 스토어 ..... 274

CLEAR QUEST 개요 ..... 275

CLEAR QUEST 스토어 이용 ..... 276

XV. 서버 리파지토리 사용 ..... 288

서버 리파지토리 개요 ..... 289

서버 리파지토리 연결 ..... 290
서버로부터 파일 열기 ..... 295
카테고리 및 아이템 관리 ..... 298
체크인, 체크아웃 ..... 303
히스토리 보기 ..... 306
아이템 찾기 ..... 308

Appendix. 문자셋과 유효한 앨리어스 ..... 311

# Ⅰ. 시작하기 전에

- 오즈 쿼리 디자이너 개요
- 본 매뉴얼의 목적
- 본 매뉴얼 구성
- 매뉴얼 로드맵

## 오즈 쿼리 디자이너 개요

오즈 쿼리 디자이너는 데이터베이스를 비롯한 다양한 데이터 소스에 대한 연결 정보 설정과 쿼리문 등을 통한 데이터 셋 생성 기능을 지원하는 툴로써 직관적인 UI 환경에서 보고서 및 애플리케이션 개발자들이 손쉽게 비즈니스 로직을 개발할 수 있도록 도와줍니다.

오즈 쿼리 디자이너로부터 생성되는 오즈 데이터 정보 파일(ODI)은 오즈 리포트 디자이너와 오즈 애플리케이션 디자이너에서 공통으로 활용될 수 있습니다.

## 본 매뉴얼의 목적

오즈 데이터 소스의 개념과 오즈 쿼리 디자이너를 이용하여 오즈 데이터 정보 파일 즉, ODI 파일을 만들고 필요한 데이터 셋을 만드는 방법에 대해 설명합니다.

# 본 매뉴얼 구성

본 매뉴얼 구성과 각 장에서 설명하는 내용은 다음과 같습니다.

## 오즈 쿼리 디자이너 실행

오즈 쿼리 디자이너의 파일 구성 및 실행 방법에 대해 설명합니다.

## 오즈 쿼리 디자이너 윈도우

오즈 쿼리 디자이너의 화면 구성과 사용자 인터페이스에 대해 설명합니다.

## 오즈 쿼리 디자이너 사용

오즈 쿼리 디자이너에서 작업 폴더를 설정하는 방법, 스크립트를 입력하는 방법, 파일 경로를 설정하는 방법 등을 설명합니다.

## 데이터 소스와 데이터 셋

오즈 쿼리 디자이너에서 지원되는 데이터 소스의 종류와 각 속성에 대한 개요, 데이터 소스 별 연결 방법 및 각 데이터 소스별로 데이터 셋을 생성하는 방법 등을 설명합니다.

## 쿼리 데이터 셋 디자인

매뉴얼 모드 및 디자인뷰 모드를 이용하여 쿼리 데이터 셋으로부터 쿼리문을 디자인하는 방법에 대해 설명합니다

## 사용자 데이터 스토어

다양한 사용자 데이터 소스 연동 방법에 대해 설명합니다.

### 그룹 데이터 스토어

하나의 데이터 셋을 분할하여 그룹 데이터 셋을 생성하는 방법을 설명합니다.

### 파일 / HTTP 스토어

XML, CSV 등의 파일 형식 데이터 소스를 네트워크 연결이나 HTTP 프로토콜을 통해 전달받아 이를 연동하는 방법에 대해 설명합니다.

### SOAP 스토어

웹 서비스를 통하여 얻어온 데이터를 데이터 소스로 사용하는 SOAP 스토어와 연동하는 방법에 대해 설명합니다.

### ADO DataSet 웹서비스 스토어

닷넷 웹 서비스에서 ADO DataSet 객체를 연동하는 ADO DataSet 웹서비스 스토어와 연동하는 방법에 대해 설명합니다.

### XML 스토어

XML-DTD 혹은 XML-XSD 데이터 소스와 연동하는 방법에 대해 설명합니다.

### SAP 스토어

SAP에 저장된 데이터와 연동하는 방법에 대해 설명합니다.

### CLEAR QUEST 스토어

형상 관리 툴인 IBM Rational Clear Quest 프로그램에서 생성한 DB와 연동하는 방법에 대해 설명합니다.

## 서버 리파지토리 사용

오즈 서버 리파지토리 서비스의 개념을 설명하고 서버 리파지토리에 연결하고 파일을 업로드하고 다운로드하는 방법을 설명합니다.

## 문자셋과 유효한 앨리어스

다양한 사용자 데이터 소스 연동 방법에 대해 설명합니다.

# 매뉴얼 로드맵

보고서 개발자와 웹 애플리케이션 개발자, 시스템 관리자가 알아야 할 내용을 전달하기 위해 다음과 같은 매뉴얼을 제공합니다.

| Manual | Description |
|---|---|
| **오즈 리포트 Getting Started** | 오즈 리포트 소개와 간단한 보고서 작성을 통해 오즈 리포트의 개념에 대해 설명합니다. |
| **오즈 리포트 디자이너 사용자 가이드** | 오즈 리포트 디자이너 사용 방법과 보고서 작성하고 작성된 보고서를 웹에 배포하는 방법에 대해 설명합니다. |
| **오즈 리포트 디자이너 사용자 레퍼런스** | 오즈 리포트에서 제공하는 컴포넌트와 마법사에 대해 자세히 설명합니다. |
| **오즈 리포트 뷰어 사용자 가이드** | 보고서를 웹을 통해 조회하고 사용자 애플리케이션과 연동할 수 있도록 뷰어 패러미터와 활용 방법 등에 대해 설명합니다. |
| **오즈 애플리케이션 Getting Started** | 오즈 애플리케이션에 대한 이해와 오즈 애플리케이션 디자이너를 이용하여 프로젝트를 만들고 웹에 적용하는 기본적인 과정을 소개합니다. |
| **오즈 애플리케이션 디자이너 사용자 가이드** | 오즈 애플리케이션 디자이너 사용 방법과 컴포넌트와 마법사를 이용하는 방법에 대해 설명합니다. |
| **오즈 애플리케이션 뷰어 사용자 가이드** | 오즈 애플리케이션 뷰어를 설치하고 실행하는 방법과 뷰어 패러미터와 활용 방법 등에 대해 설명합니다. |
| **오즈 리파지토리 매니저 사용자 가이드** | 오즈 리파지토리 매니저를 사용하여 오즈 폼 및 데이터 파일 등 각종 아이템을 포괄적으로 관리하는 방법을 설명합니다. |
| **스크립트 가이드** | 오즈 리포트에서 사용할 수 있는 스크립트의 문법과 그 활용 방법에 대해 설명합니다. |
| **에러 메시지 레퍼런스** | 오즈 제품에서 보여지는 에러 메시지에 대해 설명합니다. |

# Ⅱ. 오즈 쿼리 디자이너 실행

- 시스템 요구사항
- 오즈 쿼리 디자이너 파일 구성
- 오즈 쿼리 디자이너 실행하기

# 시스템 요구 사항

## OS

Windows NT(Service pack4 이상)/2000(Service pack2 이상)/XP/2003

❖ **참고사항** ❖

Windows 95/98/ME에서는 Unicode를 지원하지 않으므로 오즈 쿼리 디자이너를 사용할 수 없습니다.

## CPU

Pentium Ⅲ 500MHz 이상

## Main Memory

최소 256MB (512MB 이상 권장)

## HDD

최소 50MB 이상의 빈 공간 필요

## 모니터

컬러 XGA (1024 x 768 이상의 해상도 권장)

# 오즈 쿼리 디자이너 파일 구성

다음은 오즈 쿼리 디자이너를 설치한 다음 생성되는 디렉토리와 파일의 종류 및 기능에 대해 설명합니다.

| 디렉토리 | 파일명 | 설명 |
|---|---|---|
| %OZQ_HOME% | Console.txt | 디버그용 로그 파일 |
| | dbconfig.xml | DB 벤더 연결 정보 설정 파일 |
| | dbconfig.xsd | |
| %OZQ_HOME%bin | OZQueryDesigner.exe | 오즈 쿼리 디자이너 실행 파일 |
| | OZRepositoryManager.ini | 오즈 리파지토리 매니저 환경 설정 파일 |
| | OZUSLClientDes128.dll | USL 관련 라이브러리 |
| | OZUSLClientSession.dll | |
| | user.cif | 사용자 정보 파일 |
| %OZQ_HOME%bin/res | MultiLanguageList.xml | 언어 리소스 |
| | QDJavaScriptConfig.xml | 자바 스크립트 정보 파일 |
| | qd_언어명_국가명.dll | 쿼리 디자이너 라이브러리 |
| | qd_error_msg_언어명_국가명.xml | 쿼리 디자이너 에러 발생시 코드별 에러 메시지 |
| | rm_언어명_국가명.dll | 리파지토리 매니저 라이브러리 |
| | rm_error_msg_언어명_국가명.xml | 리파지토리 매니저 에러 발생시 코드별 에러 메시지 |
| | server_error_msg_언어명_국가명.xml | 서버 에러 발생시 코드별 에러 메시지 |
| %OZQ_HOME%bin/soap | mssoap1.dll | SOAP 관련 라이브러리 |
| | MSSOAP30.dll | |
| | SOAPIS30.dll | |
| | WHSC30.dll | |
| | WISC10.dll | |
| | WISC30.dll | |
| | register_soap.bat | SOAP 관련 라이브러리 레지스트리 등록 파일 |
| | unregister_soap.bat | SOAP 관련 라이브러리 레지스트리 등록 해지 파일 |

| | | |
|---|---|---|
| **%OZQ_HOME%bin/soap/Resources** | SOAP 관련 리소스 파일 | |
| **%OZQ_HOME%config** | assembly.properties | 어셈블리 정보 설정 파일 |
| | launch.cfg | 오즈 쿼리 디자이너 라이브러리 설정 파일 |
| | LayoutConfig.xml | 오즈 쿼리 디자이너 레이아웃 정보 파일 |
| | OZOpenHistory.xml | 폴더 히스토리 정보 저장 파일 |
| | OZServerList.xml | 오즈 리파지토리 서버 리스트 정보 파일 |
| | QDConfig.ini | 쿼리 디자이너 환경 설정 파일 |
| | XMLDesignerConfig.xml | XML 스토어 관련 파일 |
| | xml.properties | XML 파서 설정 파일 |
| **%OZQ_HOME%doc** | OZQueryAPI_언어명_국가명.chm | 오즈 쿼리 자바 스크립트 도움말<br>예) OZQueryAPI_ko_kr.chm |
| **%OZQ_HOME%lib** | activation.jar | 메일 관련 라이브러리 |
| | mail.jar | |
| | crimson.jar | XML 크림손 파서 라이브러리 |
| | dtdparser120.jar | DTD 파일 파싱 클래스 라이브러리 |
| | jcert.jar | J2EE 라이브러리(HTTPS/SSL 지원을 위해 필요한 라이브러리)<br>JRE 1.3.X의 경우에만 필요함 |
| | jnet.jar | |
| | jsse.jar | |
| | jsr173_api.jar | 파일 또는 HTTP 스토어에서 XML 지원 라이브러리 파일 |
| | log4j.jar | Log 사용에 대한 라이브러리 |
| | ozsfw50.jar | 오즈 서버 라이브러리 |
| | parser.jar | XML 파서 관련 클래스 라이브러리 |
| | xerces.jar | |
| | dom4j-1.6.1.jar | |
| | jaxen-1.1.6.jar | |
| | xalan.jar | XML XPath 관련 클래스 라이브러리 |
| | xercesImpl.jar | |
| | xml-apis.jar | |
| | soap2_3_1.jar | SOAP 관련 라이브러리 |
| | jazzlib.jar | 파일 압축 라이브러리 |

| | | |
|---|---|---|
| | js.jar | Java Script 구동 라이브러리 |
| **%OZA_HOME% license** | ozlicense.xml | 라이센스 파일 |
| **UniversalDriver** | classes12.jar | Oracle 8i, 9i용 JDBC Driver |
| | classes12.zip | |
| | db2java.jar | IBM DB2 용 JDBC Driver |
| | db2java.zip | |
| | db2jcc.jar | IBM DB2용 JDBC 드라이버 Type 8 (DB2 버전 8 이상) |
| | db2jcc_license_ cisuz.jar | |
| | db2jcc_license_cu.jar | |
| | ifxjdbc.jar | Infomix용 JDBC Driver |
| | ijava.jar | Oracle 7i 관련 JDBC Driver |
| | jconn2.jar | Sybase ASE용 JDBC Driver |
| | jkjdbc.jar | MSSQL 97 JDBC 드라이버 |
| | jt400.jar | IBM JT400 용 JDBC 드라이버 |
| | msbase.jar | MSSQL 2000 JDBC 드라이버 |
| | mssqlserver.jar | |
| | msutil.jar | |
| | mysql5.1.5.jar | Mysql용 JDBC 라이브러리 |
| | ozsapjco2.jar | SAPJCO2 용 라이브러리 |
| | ozsapjco3.jar | SAPJCO3 용 라이브러리<br>※ 주의사항 : SAPJCO3 버전은 JRE 1.5 이상에서만 동작하므로, launch.cfg에 설정된 JRE 버전을 변경한 후 사용하시기 바랍니다. |
| | sqljdbc.jar | MSSQL 2008 JDBC 드라이버 |
| | unisqljdbc10.jar | UniSQL용 JDBC 드라이버 |
| | unsqljdbc205.zip | |

※ 주의사항 : JDBC 드라이버 호환

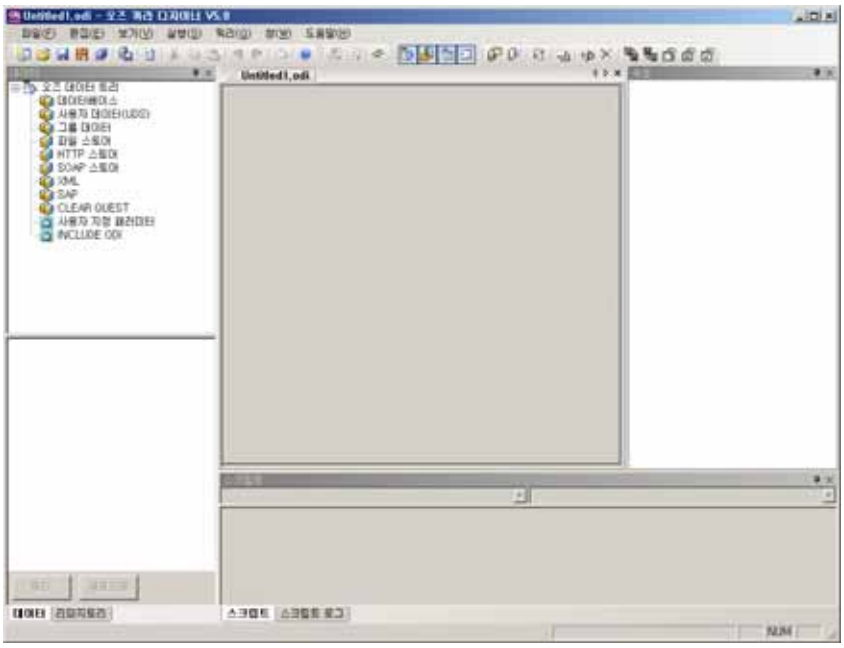
"UniversalDriver" 폴더의 파일은 기본적으로 제공하는 JDBC 드라이버로, 이미 사용중인 RDB의 JDBC 드라이버와 호환되지 않거나 필드 정보 등이 다르게 표시되는 등의 충돌이 발생할 수 있습니다. 그러므로 이미 사용중인 RDB의 JDBC 드라이버를 사용할 것을 권장합니다.

# 오즈 쿼리 디자이너 실행

윈도우 시작 프로그램의 [프로그램] - [OZ XStudio 5.0] - [OZ Query Designer 5.0]를 클릭하거나 윈도우 바탕 화면의 'OZ Query Designer 5.0' 아이콘을 클릭하여 오즈 쿼리 디자이너를 실행합니다.




오즈 쿼리 디자이너가 실행되면 다음과 같이 오즈 쿼리 디자이너 초기 화면이 나타납니다.



오즈 쿼리 디자이너를 실행시키면 각 설정 클래스 파일들이 로딩되게 됩니다. 클래스 패스 등의 설정은 config 폴더의 launch.cfg 파일에서 변경할 수 있습니다.

launch.cfg의 내용은 다음과 같습니다.

```
CLASSPATH
=          .\lib\ozsfw50.jar;.\lib\log4j.jar;.\lib\xalan.jar;.\lib\xercesImpl.jar;.\lib\xml-
apis.jar;.\lib\parser.jar;.\lib\xerces.jar;.\lib\dtdparser120.jar;.\lib\jcert.jar;.\lib\jnet
.jar;.\lib\jsse.jar;.\lib\jazzlib.jar;.\lib\js.jar;.\lib\soap2_3_1.jar;.\lib\activation.jar;.\
lib\mail.jar;.\UniversalDriver\classes12.zip;.\UniversalDriver\db2java.zip;.\Universal
Driver\ifxjdbc.jar;.\UniversalDriver\jconn2.jar;.\UniversalDriver\msbase.jar;.\Univer
salDriver\mssqlserver.jar;.\UniversalDriver\msutil.jar;.\UniversalDriver\mysql5.1.5.j
ar;.\UniversalDriver\unisqljdbc10.jar;.\lib\jsr173_api.jar;.\lib\stax-
1.2.0.jar;.\lib\dom4j-1.6.1.jar;.\lib\jaxen-1.1.6.jar
JRE_PATH = ..\jre\bin\hotspot\jvm.dll
```

※ 주의사항

- 사이트에서 사용하지 않는 JDBC 드라이버는 CLASSPATH에서 삭제하시기 바랍니다.
- 오즈에서 기본적으로 제공되지 않는 JDBC 드라이버를 사용할 경우 lib 폴더에 해당 데이터베이스의 JDBC 드라이버 파일을 복사한 후 launch.cfg 파일에 클래스 패스를 추가하여야 합니다.

- JDBC 드라이버의 버전을 변경하면 변경 전에 저장된 ODI 파일의 필드 정보가 제대로 반영되지 않을 수 있습니다. 이럴 경우에는 ODI 파일을 열어서 다시 저장한 후 사용하시기 바랍니다. (이런 번거로움을 줄이기 위해서는 쿼리 디자이너의 JDBC를 데이터베이스의 JDBC 드라이버로 먼저 변경한 후 쿼리 디자이너를 사용하시기 바랍니다.)

※ 참고사항 : Java 옵션 설정하는 방법

오즈 쿼리 디자이너가 실행될 때 특별한 Java 옵션을 설정하여 실행하고자 할 경우에는 launch.cfg 파일에 실행할 Java 옵션을 등록하면 됩니다.

오즈 쿼리 디자이너에서는 각 줄의 맨 첫 글자가 "-"로 시작하면 해당 줄의 내용을 Java 옵션으로 인식하므로, Java 옵션의 첫 글자에 "-"를 추가하여 launch.cfg 파일에 등록합니다.

Java 옵션이 추가된 launch.cfg 파일의 예는 다음과 같습니다.

```
CLASSPATH
= .\lib\ozsfw50.jar;.\lib\log4j.jar;.\lib\xalan.jar;.\lib\xercesImpl.jar;.\lib\xml-apis.jar;.\lib\parser.jar;.\lib\xerces.jar;.\lib\dtdparser120.jar;.\lib\jcert.jar;.\lib\jnet.jar;.\lib\jsse.jar;.\lib\jazzlib.jar;.\lib\js.jar;.\lib\soap2_3_1.jar;.\lib\activation.jar;.\lib\mail.jar;.\UniversalDriver\classes12.zip;.\UniversalDriver\db2java.zip;.\UniversalDriver\ifxjdbc.jar;.\UniversalDriver\jconn2.jar;.\UniversalDriver\msbase.jar;.\UniversalDriver\mssqlserver.jar;.\UniversalDriver\msutil.jar;.\UniversalDriver\mysql5.1.5.jar;.\UniversalDriver\unisqljdbc10.jar;.\lib\jsr173_api.jar;.\lib\stax-1.2.0.jar;.\lib\dom4j-1.6.1.jar;.\lib\jaxen-1.1.6.jar
JRE_PATH = "C:\Program Files\Java\j2re1.4.2_03\bin\client\jvm.dll"-Djava.version=AAA
-Djava.vendor=ABC
-Dos.name=DEF
-Dos.arch=GHI
-Dos.version=JKL
-Duser.name=MNO
-Duser.home=PQR
-Dsystem.IPs=STU
-Dforcs=Forcs
```

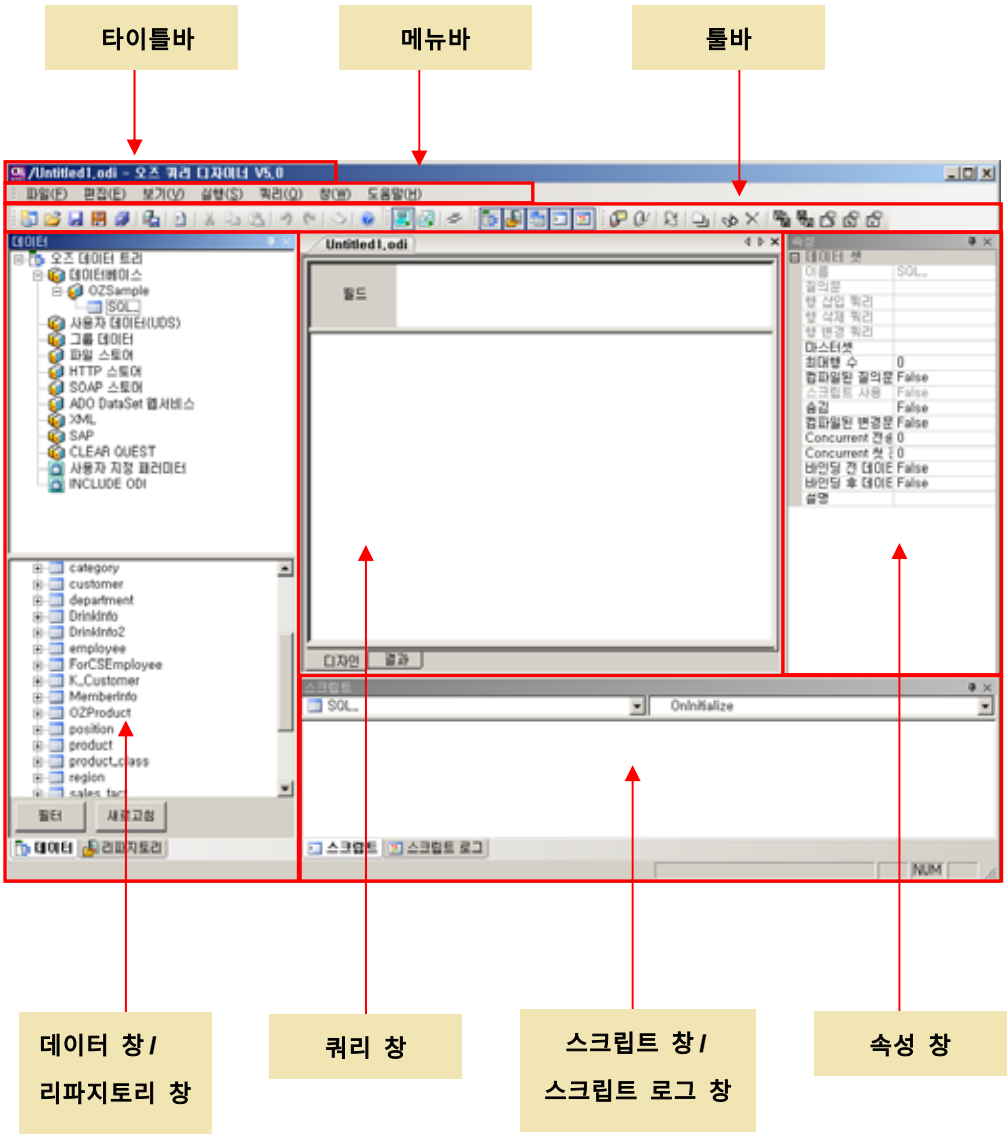
# Ⅲ. 오즈 쿼리 디자이너 윈도우

- 오즈 쿼리 디자이너 화면 구성
- 타이틀바
- 메뉴바
- 툴바
- 데이터 창, 리파지토리 창
- 쿼리 창
- 속성 창
- 스크립트 창

**본 장에서는 오즈 쿼리 디자이너의 화면 구성과 사용자 인터페이스에 대해 설명합니다.**

## 오즈 쿼리 디자이너 화면 구성

다음은 오즈 쿼리 디자이너의 사용자 인터페이스 화면이며 각 구성 요소의 이름과 기능은 다음과 같습니다.



| 인터페이스 | 설명 |
|---|---|
| **타이틀바** | 현재 오즈 쿼리 디자이너에서 편집 중인 ODI(OZ Data Info) 파일의 카테고리와 파일명을 표시합니다. |
| **메뉴바** | 메뉴바는 오즈 쿼리 디자이너의 상단에 풀다운(pull down)형식으로 제공되며, 파일, 편집, 보기, 실행, 쿼리, 창, 도움말로 구성됩니다. |
| **툴바** | 툴바는 오즈 쿼리 디자이너의 일반적인 기능, 쿼리 실행 기능 등을 쉽고 빠르게 설정할 수 있도록 메뉴바 아래에 아이콘 형태로 제공됩니다. 툴바는 새로 만들기, 열기, 저장 등 파일 관련 기능과 잘라내기, 복사하기, 붙여넣기 등의 편집 기능을 아이콘 형태로 나타낸 '표준 툴바', 디자인 모드를 디자인 뷰 또는 매뉴얼 모드로 변경하고 쿼리문을 실행하는 '데이터 셋 툴바', 서버 리파지토리 관련 기능을 실행하는 '리파지토리 툴바'로 로 구성됩니다. |
| **데이터 창 / 리파지토리 창** | 트리 창은 데이터 창과 리파지토리 창으로 구성됩니다.<br>데이터 창은 현재 ODI(OZ Data Info) 파일에 추가된 스토어, 데이터 셋, 사용자 지정 패러미터 등을 트리 형태로 표시하고 선택된 스토어의 테이블 정보를 표시하는 창입니다.<br>리파지토리 창은 로컬 및 서버 리파지토리의 폴더 및 파일을 탐색기 형태로 표시하는 창입니다. |
| **쿼리 창** | 쿼리를 작성하고 쿼리를 실행한 결과 데이터 셋을 보여줍니다. |
| **스크립트 창 / 스크립트 로그 창** | 스크립트 창은 스크립트를 입력하거나 입력된 스크립트를 표시합니다.<br>스크립트 로그 창은 자바 스크립트 함수 중 _DEBUG(), _ERROR(), _INFO() 함수를 이용하여 남긴 로그를 표시하는 창입니다. |
| **속성 창** | 선택된 아이템에 관한 속성을 표시하거나 설정합니다. |

# 타이틀바

현재 오즈 쿼리 디자이너에서 편집중인 ODI(OZ Data Info) 파일의 파일명을 표시합니다.

타이틀바에 표시되는 형식은 아래와 같습니다.
*/카테고리/파일명* - 오즈 쿼리 디자이너 V5.0

■ /카테고리/파일명

작업 폴더에서 현재 작업 ODI를 열었을 경우 작업 폴더를 기준으로 ODI 파일의 경로와 파일명이 표시되며 작업 폴더가 아닌 경로에서 ODI를 열었을 경우 ODI 파일의 절대 경로와 파일명이 표시됩니다.

새 파일에서 작업할 경우에는 파일명이 표시되지 않고 Untitled#으로 표시됩니다.

■ 적용 예

- OZProduct.odi 파일을 작업 폴더에서 열었을 경우




- OZProduct.odi 파일을 작업 폴더가 아닌 경로에서 열었을 경우



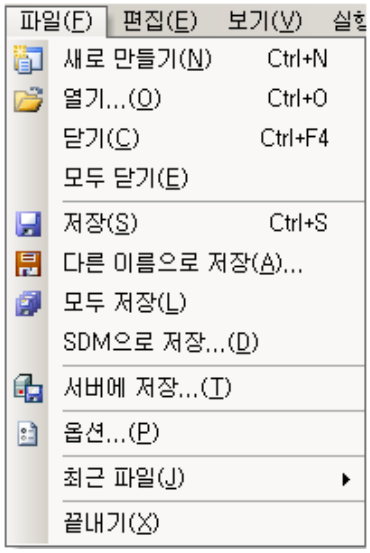

- 새파일에서 작업할 경우



# 메뉴바

## 파일

메뉴바의 [파일]을 선택하면 아래와 같은 메뉴가 나타납니다.




- **새로 만들기(Ctrl+N)**

  새로운 ODI 파일을 만듭니다.

- **열기(Ctrl+O)**

  만든 ODI 파일을 읽어 옵니다.

- **닫기**

  편집 중인 파일을 닫습니다.

- **모두 닫기**

  편집 중인 모든 파일을 닫습니다.

- **저장(Ctrl+S)**

  편집 중인 파일을 저장합니다.

- **다른 이름으로 저장**

편집 중인 파일을 새로운 이름으로 저장합니다.

■ **모두 저장**

편집 중인 모든 파일을 저장합니다.

■ **SDM으로 저장**

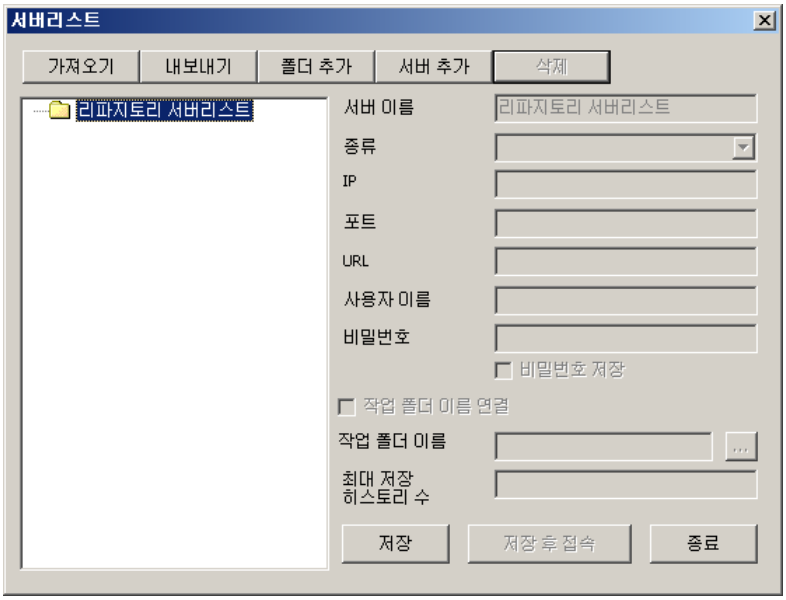
편집 중인 ODI 파일의 SDM(Server Data Module) 파일로 저장합니다.

SDM 파일로 저장하면 데이터 셋의 필드 정보 뿐만 아니라 쿼리를 실행한 후에 얻어진 결과 데이터까지 저장됩니다.

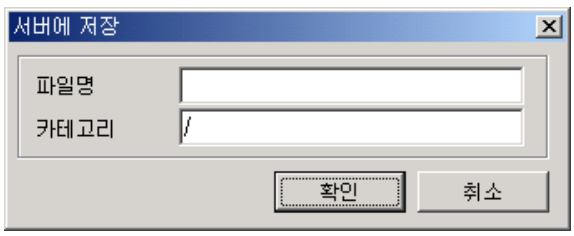
■ **서버에 저장**

작성 중인 파일을 서버 리파지토리에 저장합니다.

만일 서버 리파지토리에 접속되어 있지 않은 상태에서 [서버에 저장] 메뉴를 클릭하면 서버 리스트를 선택하는 다이얼로그가 표시됩니다.




리파지토리 서버에 접속되어있는 상태에서 [서버에 저장] 메뉴을 클릭하면 파일명과 카테고리를 설정하는 '서버에 저장 다이얼로그'가 표시됩니다.



| | |
|---|---|
| **파일명** | 저장할 파일명을 입력합니다. 현재 편집 중인 ODI 파일의 이름이 자동으로 표시됩니다. |
| **카테고리** | 서버 리파지토리의 카테고리를 입력합니다. |

※ 주의사항

- [서버에 저장]은 편집 중인 ODI 파일을 서버 리파지토리의 특정 카테고리에 저장하는 기능으로 지정한 카테고리에 동일한 파일명이 없을 경우에는 해당 카테고리에 파일이 새로 생성됩니다. 만약 지정한 카테고리에 동일한 파일명이 존재할 경우에는 기존 파일을 지우고 새로운 파일로 교체됩니다.
- [서버에 저장] 기능을 통해 파일을 저장하면 파일의 히스토리 정보가 삭제되므로 주의해야 합니다. 파일의 히스토리 관리를 위해서 가급적 리파지토리의 체크아웃, 체크인 기능을 이용하시기 바랍니다.

■ **옵션**

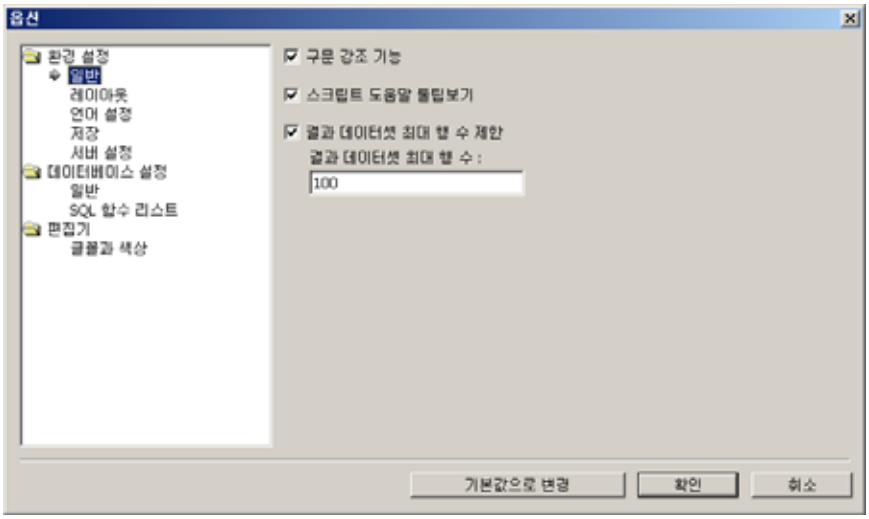
[옵션] 메뉴를 클릭하면 '옵션 다이얼로그'가 표시되며, '옵션 다이얼로그'를 이용하여 오즈 쿼리 디자이너의 실행 환경, 데이터베이스 설정 등의 옵션을 설정합니다. [기본 값으로 변경] 버튼을 클릭하면 '옵션 다이얼로그'에서 설정한 값을 기본 값으로 변경합니다.

- 환경 설정

  환경 설정에서는 레이아웃, 언어 설정, 서버 설정 옵션 등 오즈 쿼리 디자이너의 실행 환경을 설정합니다.

  ➢ 일반

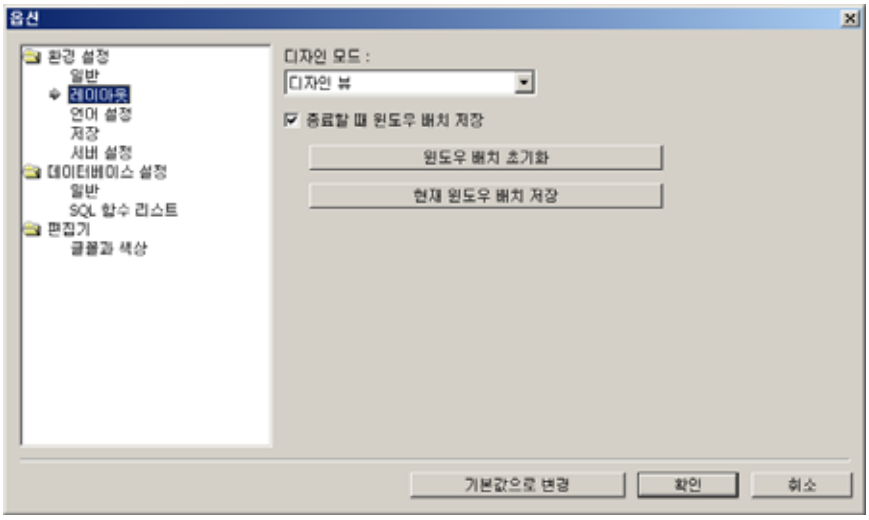
  구문 강조 기능, 결과 데이터셋 최대 행 수 제한 등을 설정합니다.



| | |
|---|---|
| **구문 강조 기능** | 매뉴얼 모드로 쿼리를 작성할 경우 입력한 문자열을 문법 체크를 하여 키워드 별로 텍스트 색을 달리하여 표시할지 여부를 설정합니다. |
| **스크립트 도움말 툴팁보기** | 스크립트 창에서 함수의 프로토 타입을 툴팁으로 표시할지 여부를 설정합니다. |
| **결과 데이터셋 최대행 수 제한** | 쿼리 실행 결과로 표시되는 데이터의 최대 행 수의 제한 여부를 설정하고 최대 행 수를 입력합니다.<br>※ 참고사항 : 쿼리 창의 [결과] 탭에만 적용됩니다.<br>※ 주의사항 : 0 ~ 2147483647 이외의 값은 설정할 수 없습니다. |

➢ 레이아웃

오즈 쿼리 디자이너의 레이아웃을 설정합니다.



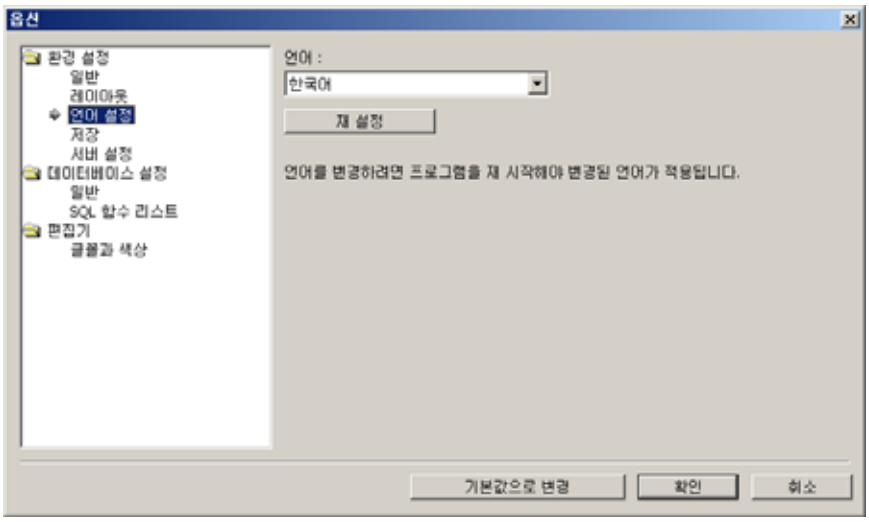

| | |
|---|---|
| **디자인 모드** | 오즈 쿼리 디자이너 시작 시 쿼리 모드를 디자인뷰 모드 또는 매뉴얼 모드로 선택합니다. |
| **종료할 때 윈도우 배치 저장** | 오즈 쿼리 디자이너 종료 시 윈도우 배열을 저장합니다. |
| **윈도우 배치 초기화** | 윈도우의 배열을 초기화합니다. |
| **현재 윈도우 배치 저장** | 현재 윈도우 배치를 저장합니다. |

➢ 언어 설정

메뉴바나 툴팁, 기타 메시지 창에 표시되는 언어를 설정합니다. 원하는 언어를 콤

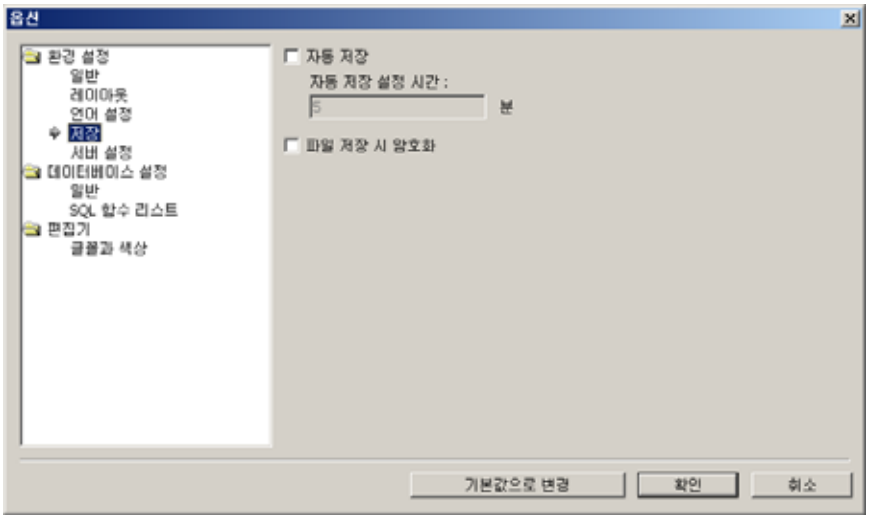
보 박스에서 선택한 후 [확인] 버튼을 클릭합니다.

언어를 콤보박스에서 변경한 후 변경전의 언어로 다시 선택하려면 [재설정] 버튼을 클릭합니다.




➢ 저장

파일 저장 관련 옵션을 설정합니다.

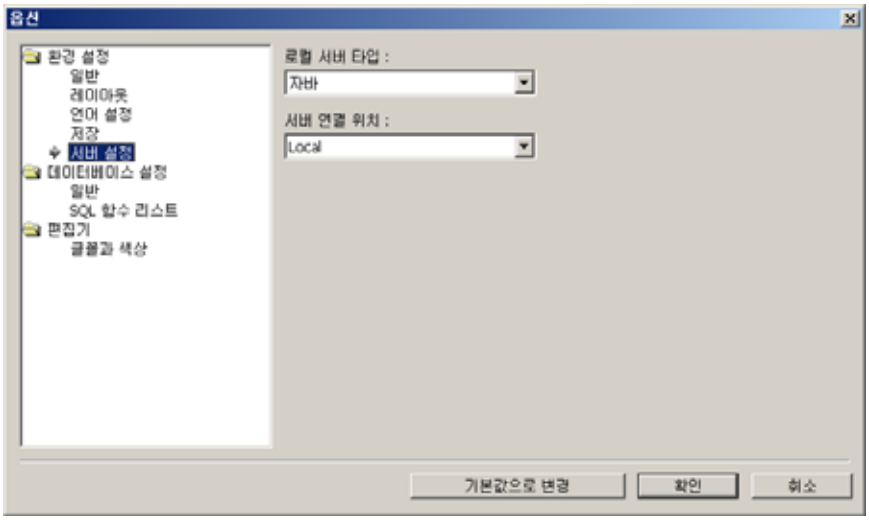



| | |
|---|---|
| **자동 저장** | 작업 중인 ODI 파일을 백업하는 옵션으로 지정된 시간마다 백업 파일을 생성합니다.<br>※ 주의사항 : 1 ~ 60 이외의 값은 설정할 수 없습니다. |

| | |
|---|---|
| **파일 저장 시 암호화** | 보고서 파일 저장 시 암호화하여 저장할지 여부를 설정합니다.<br>특정 사이트에서만 파일을 열 수 있도록 파일을 암호화하고 복호화할 수 있는 기능을 제공합니다.<br>해당 기능에 대한 자세한 사항은 오즈 기술 지원 센터로 문의하시기 바랍니다. |

➢ 서버 설정

서버 관련 옵션을 설정합니다.




| | | |
|---|---|---|
| **로컬 서버 타입** | 오즈 쿼리 디자이너의 로컬 서버 타입은 자바, 닷넷 중 하나로 설정할 수 있습니다. | |
| **서버 연결 위치** | 쿼리문 실행 시 데이터를 바인딩할 서버의 위치를 설정합니다.<br>※ 주의사항 : "ForceLocal" 값을 제외한 나머지 값은 오즈 데이터 트리의 "서버 연결 위치" 속성 값에 따라 다르게 설정될 수 있으며, 자세한 내용은 오즈 데이터 트리의 속성 설명 부분을 참조하시기 바랍니다. | |
| | **OpenType** | ODI 파일을 열기한 위치에 따라 연결할 서버를 설정합니다. |
| | **Local** | 로컬 서버로 설정합니다. |
| | **OZServer** | 연결되어있는 오즈 서버로 설정합니다. |
| | **ForceLocal** | 무조건 로컬 서버로 설정합니다. |
| | **Force OZServer** | 연결되어있는 오즈 서버로 설정합니다. |

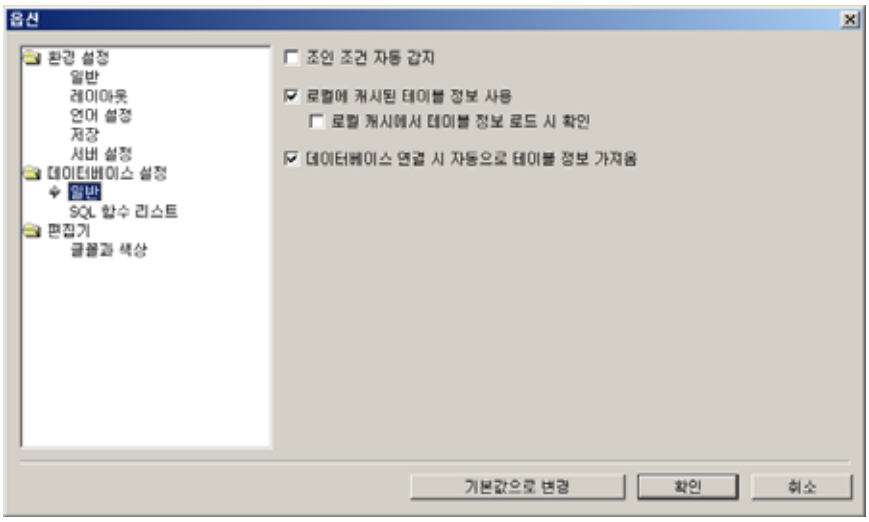
| | | |
|---|---|---|
| | **Force OpenType** | ODI를 열기한 위치에 따라 연결할 서버를 설정합니다. |

- 데이터베이스 설정

  데이터베이스, SQL 함수 리스트 옵션을 설정합니다.

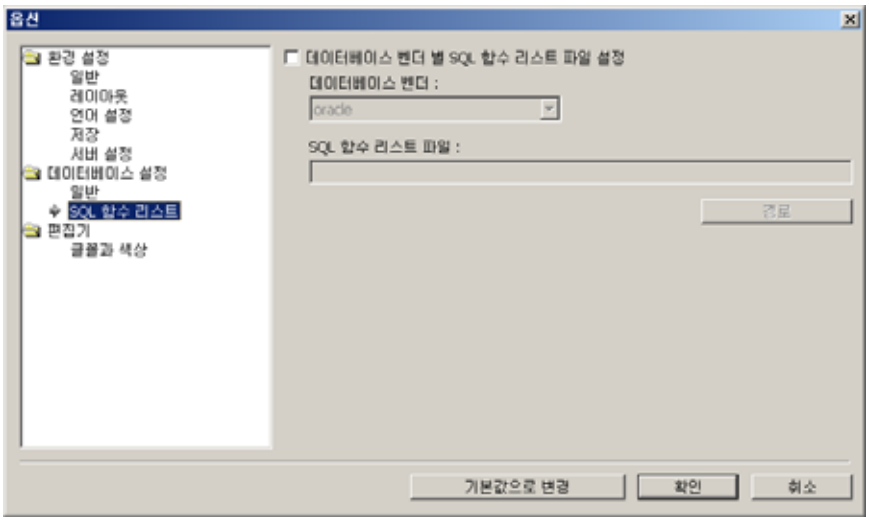
  ➢ 일반

  데이터베이스 관련 옵션을 설정합니다




| | |
|---|---|
| **조인 조건 자동 감지** | 데이터베이스에 설정된 테이블 간의 조인을 자동으로 감지하도록 설정합니다. |
| **로컬에 캐시된 테이블 정보 사용** | 최초 데이터베이스 접속 시 가져온 테이블 정보를 로컬 캐시에 저장하고 다음 접속 시부터는 로컬에 저장된 테이블 정보를 사용할 것인지 여부를 설정합니다. |
| **로컬 캐시에서 테이블 정보 로드 시 확인** | 로컬 캐시에서 테이블 정보 로딩 시 확인 창을 띄울 것인지 여부를 설정합니다. |
| **데이터베이스 연결 시 자동으로 테이블 정보 가져옴** | 데이터베이스 접속 시 테이블 정보를 자동으로 가져올지 여부를 설정합니다. |

  ➢ SQL 함수 리스트




| | | |
|---|---|---|
| **데이터베이스 벤더 별 SQL 함수 리스트 파일 설정** | **데이터베이스 벤더** | 데이터베이스 종류를 선택합니다. |
| | **SQL 함수 리스트 파일** | 가져올 SQL 함수 리스트 파일을 설정합니다. |

※ 주의사항

- 데이터베이스 별 SQL 함수 리스트는 쿼리 마법사에서 사용하며, 파일을 설정하지 않을 경우 해당 데이터베이스의 JDBC Driver에서 지원하는 함수 이름이 자동으로 사용됩니다. 그러나 JDBC Driver에 따라 함수명이 정확하지 않을 수도 있습니다.

- SQL 함수 리스트 파일 작성 예

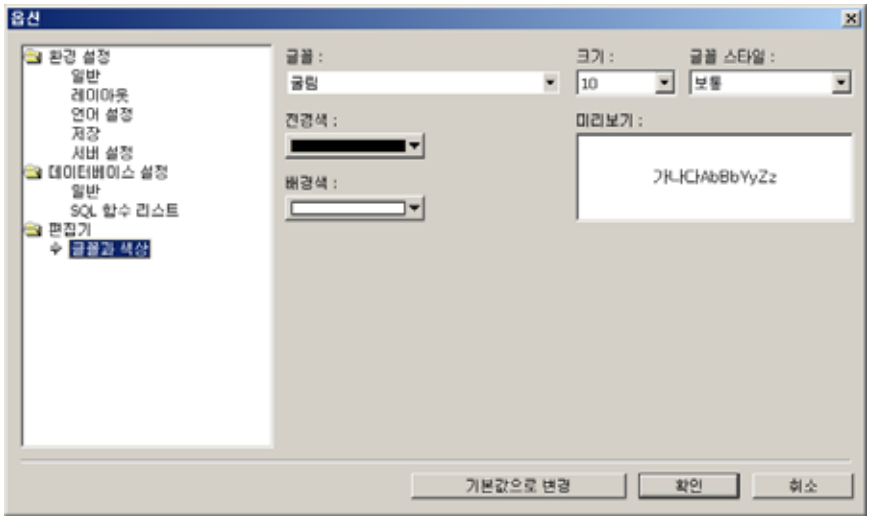
```
ASCII REPLACE
SUM MIN
USER DUMP
SYSDATE LAST_DAY NEXT_DAY
```

각 함수들은 공백 문자(Space 혹은 Tab)으로 구분되며, 라인은 4개입니다. 또한 함수 개수에 관계없이 1라인은 스트링 함수, 2라인은 수학 함수, 3라인은 시스템 함수, 4라인은 시간 함수를 표현합니다.

- 편집기

매뉴얼 모드의 쿼리 창, 디자인 뷰 모드의 쿼리 영역 등에서 쿼리문 및 스크립트를

표시하는 텍스트의 글꼴, 전경색, 배경색 등을 설정합니다.



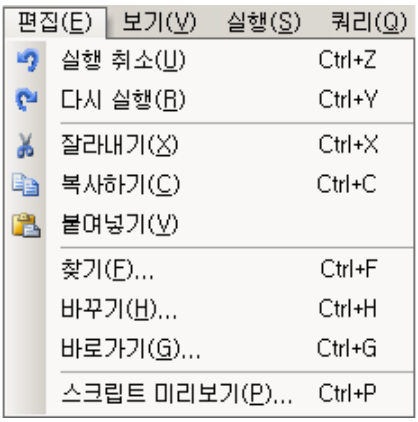

■ **최근 파일**

최근 열어본 ODI 파일을 최대 4개까지 서브 메뉴로 표시합니다. 서브 메뉴에 나타난 ODI 파일명을 클릭하면 해당 파일이 열립니다.

■ **끝내기**

오즈 쿼리 디자이너를 종료합니다.

## 편집



■ **실행 취소(Ctrl+Z)**

바로 전에 실행했던 작업을 취소하고 이전 상태로 되돌립니다.

※ 주의사항 : 디자인뷰 모드에서는 작업 프레임 단위로 Undo 작업이 이루어지므로 다양한 작업 프레임을 지닌 환경내에서는 취소에 각별한 주의가 요구됩니다.

■ **다시 실행(Ctrl+Y)**

실행 취소했던 작업을 다시 복구하며, 실행 취소한 만큼 복구 가능합니다.

■ **잘라내기(Ctrl+X)**

선택된 항목을 클립보드에 복사하고 현재 문서에서는 삭제합니다. [붙여넣기] 메뉴를 이용하여 다른 문서에 동일한 내용을 삽입할 수 있습니다. 매뉴얼 모드에서만 활성화됩니다.

■ **복사하기(Ctrl+C)**

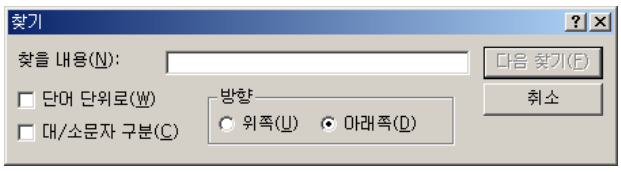
선택된 항목을 클립보드에 복사합니다. 복사한 내용은 문서에는 그대로 남겨지며, [붙여넣기] 메뉴를 통하여 다른 문서에 삽입할 수 있습니다. 매뉴얼 모드에서만 활성화됩니다.

■ **붙여넣기(Ctrl+V)**

클립보드에 복사된 항목을 문서에 삽입합니다. 매뉴얼 모드에서만 활성화됩니다.

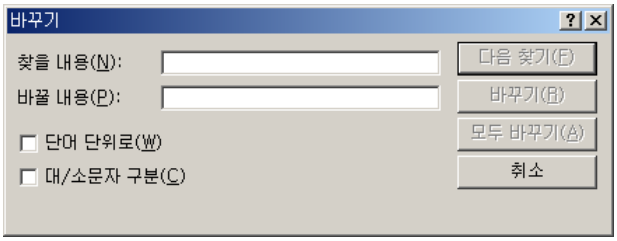
■ **찾기(Ctrl+F)**

지정한 문자열을 찾습니다. [찾기] 메뉴는 매뉴얼 모드에서만 활성화됩니다.



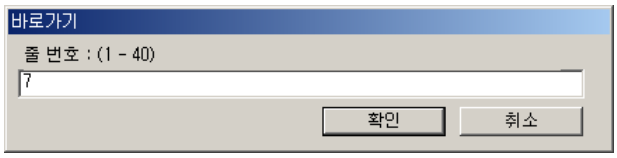

■ **바꾸기(Ctrl+H)**

지정한 문자열을 찾아서 다른 문자열로 변경합니다. [바꾸기] 메뉴는 매뉴얼 모드에서만 활성화됩니다.



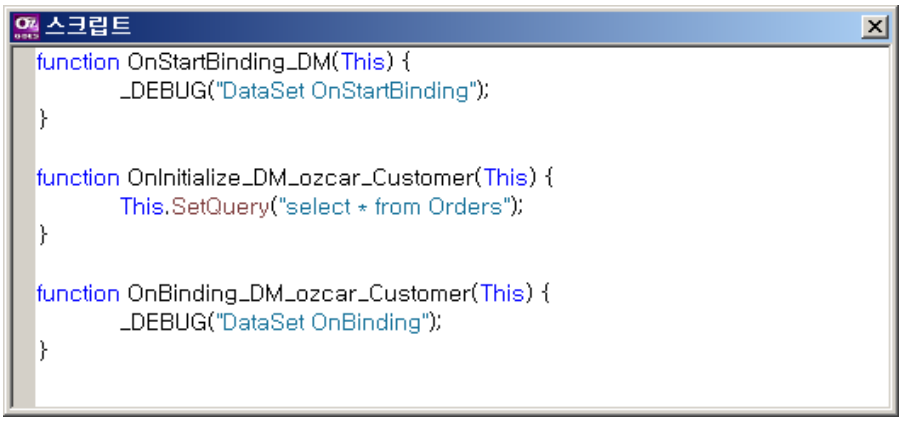
■ **바로가기(Ctrl+G)**

쿼리문의 특정 줄로 커서를 이동합니다. [바로가기] 메뉴는 매뉴얼 모드에서만 활성화됩니다.




■ **스크립트 미리보기(P)**

ODI 전체 스크립트 내용을 미리보기합니다.




## 보기




■ **데이터 창(Alt+0)**

오즈 쿼리 디자이너의 데이터 창의 표시 여부를 설정합니다.

■ **리파지토리 창(Alt+1)**

오즈 쿼리 디자이너의 리파지토리 창의 표시 여부를 설정합니다.

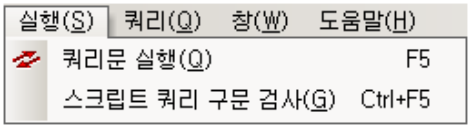
■ **속성 창(Alt+2)**

오즈 쿼리 디자이너의 속성 창의 표시 여부를 설정합니다.

■ **스크립트 창(Alt+3)**

오즈 쿼리 디자이너의 스크립트 창의 표시 여부를 설정합니다.

## 실행



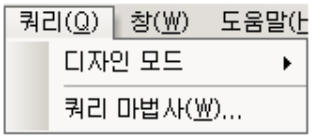

■ **쿼리문 실행(F5)**

쿼리 데이터 셋에서 디자인한 쿼리문을 실행하여 결과 데이터 셋을 받아옵니다.

■ **스크립트 쿼리 구문 검사(Ctrl+F5)**

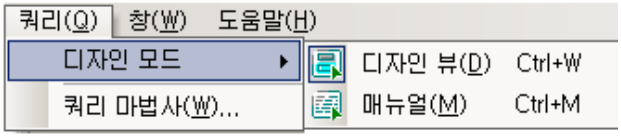
스크립트가 사용된 쿼리문의 구문 검사를 실행합니다. [스크립트 쿼리 구문 검사] 메뉴는 매뉴얼 모드에서만 사용 가능하며, 현재 작업 중인 데이터 셋의 속성 값 중 '스크립트 사용'이 'True'인 경우에만 활성화됩니다.

## 쿼리




■ **디자인 모드**

[디자인 모드] 메뉴는 쿼리를 디자인뷰 모드로 디자인할지 매뉴얼 모드로 디자인할지를 선택하는 메뉴로 2개의 서브 메뉴로 구성됩니다.



- **디자인 뷰(Ctrl+W)**

  데이터 셋을 생성하기 위한 쿼리문 작성을 쿼리 마법사를 이용하여 디자인합니다.

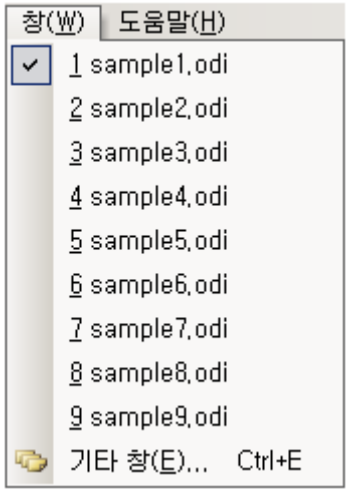
- **매뉴얼(Ctrl+M)**

  데이터 셋을 생성하기 위해 쿼리문을 직접 입력하여 디자인합니다.

■ **쿼리 마법사**

디자인뷰 모드에서만 활성화되며, 단계별 쿼리 작성을 위한 쿼리 마법사를 실행합니다.

## 창

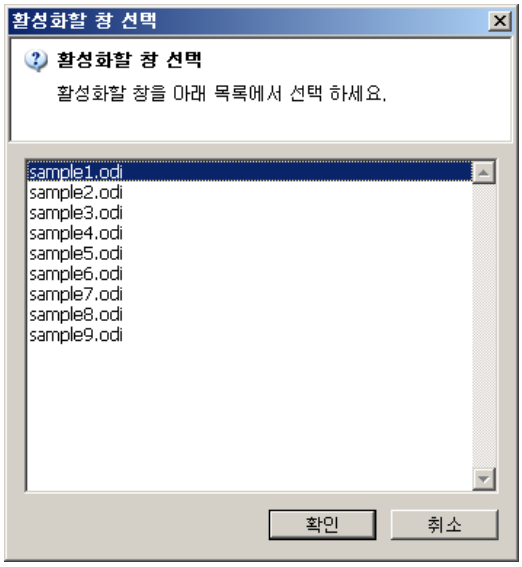



■ **파일 목록**

현재 열려 있는 파일 중 최대 9개까지 파일명을 서브 메뉴 형태로 표시합니다. 파일명을 클릭하면 해당 파일이 쿼리 디자이너의 쿼리 창에 활성화되어 표시되며, 서브 메뉴의 파일명 앞에 체크 표시됩니다.

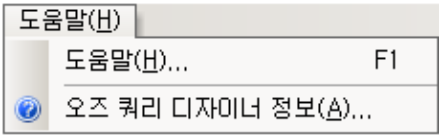
■ **기타 창**

현재 열려 있는 파일의 전체 목록을 기타 창에 표시합니다.




활성화할 파일을 선택한 후 [확인] 버튼을 클릭하면 해당 파일이 쿼리 디자아너의 쿼리 창에 활성화되어 표시됩니다.

## 도움말




■ **도움말**

오즈 쿼리 자바 스크립트의 기본 문법/함수 등에 대한 설명과 예제를 보여주는 도움말 창이 나타납니다.

■ **오즈 쿼리 디자이너 정보**

오즈 쿼리 디자이너의 버전 정보와 사용자 PC정보를 나타냅니다.

오즈 쿼리 디자이너의 버전 정보 창에서 [PC 정보 보기] 버튼을 클릭하면 'PC 정보 다이얼로그'가 표시되고 사용자 PC의 IP Address, Mac Address, HDD Serial을 확인할 수 있습니다. [복사] 버튼을 클릭하여 사용자 PC 정보를 복사한 후 일반 편집 프로그램에 붙여넣기 할 수 있습니다.

# 툴바

## 표준 툴바

새로 만들기, 열기, 저장 등 파일 관련 기능과 잘라내기, 복사하기, 붙여넣기 등의 편집 기능, 도움말 등을 실행합니다.

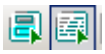


| 아이콘 | 설명 |
|---|---|
|  | 새로 만들기, 열기, 저장, 다른 이름으로 저장, 모두 저장 |
|  | 서버에 저장 |
|  | '옵션 다이얼로그' 실행 |
|  | 잘라내기, 복사, 붙여넣기 |
|  | 실행 취소, 다시 실행 |
|  | 기타 창 표시 |
|  | 오즈 리포트 디자이너 정보 |

## 데이터 셋 툴바

디자인 모드 변경 기능과 쿼리문 실행 기능을 합니다.



| 아이콘 | 설명 |
|---|---|
|  | 디자인 뷰 모드, 매뉴얼 모드 |
|  | 쿼리문 실행 |

## 보기 툴바

데이터 창, 리파지토리 창, 속성 창, 스크립트 창의 표시 여부를 설정합니다.



| 아이콘 | 설명 |
|---|---|
|  | 데이터 창, 리파지토리 창, 속성 창, 스크립트 창 |

## 리파지토리 툴바

서버 리파지토리와 관련된 기능을 실행합니다.

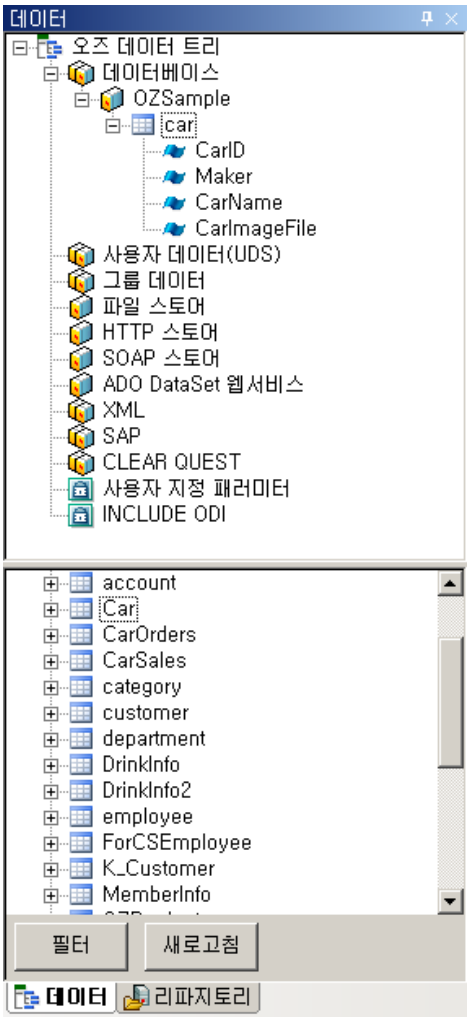


| 아이콘 | 설명 |
|---|---|
|  | 리파지토리 열기, 리파지토리 닫기 |
|  | 새로고침 |
|  | 작업 폴더 설정 |
|  | 카테고리 생성, 삭제 |
|  | 파일 업로드, 파일 다운로드, 체크아웃, 체크인, 체크아웃 취소 |

# 데이터 창, 리파지토리 창

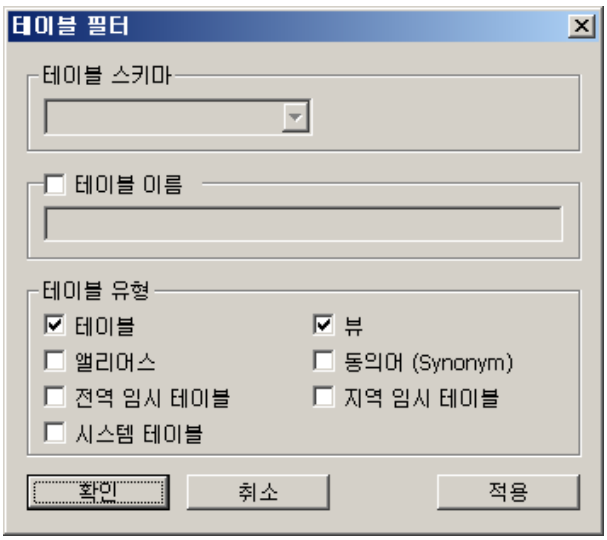
## 데이터 창

데이터 창은 현재 ODI(OZ Data Info) 파일에 추가된 스토어, 데이터 셋, 사용자 지정 패러미터 등을 트리 형태로 표시하고 선택된 스토어의 테이블 정보를 표시하는 창입니다.




| UI | 설명 |
|---|---|
| 필터 | 선택된 스토어의 테이블 정보를 필터링하여 표시합니다. |
| 새로 고침 | 선택된 스토어의 테이블 정보를 새로고침하여하여 표시합니다. |

[필터] 버튼을 클릭하면 '테이블 필터 다이얼로그'가 표시되며 해당 다이얼로그에서 설정한 값으로 선택된 스토어의 테이블 정보를 필터링하여 표시할 수 있습니다.




| UI | 설명 |
|---|---|
| **테이블 스키마** | 접속한 사용자가 소유한 스키마별로 필터링하여 테이블을 가져오도록 설정합니다. |
| **테이블 이름** | 특정 문자열로 시작되는 테이블만을 표시하도록 설정합니다. |
| **테이블 유형** | 테이블, 뷰, 앨리어스, 동의어, 전역 임시 테이블, 지역 임시 테이블, 시스템테이블 등 리스트하기 원하는 테이블 유형을 선택합니다.<br>※ 참고사항<br>▪ "테이블 유형"을 변경한 후 [확인] 또는 [적용] 버튼을 클릭하면 로컬에 캐시된 테이블 정보가 있을 경우 캐시된 정보를 가져오며 캐시된 정보가 없을 경우에는 데이터베이스에 연결하여 선택한 테이블 유형의 테이블 정보를 가져옵니다.<br>▪ 데이터 창의 [새로고침] 버튼을 클릭하면 로컬에 캐시된 테이블 정보를 초기화 시킨 후 데이터베이스에 연결하여 선택한 테이블 유형의 테이블 정보를 가져옵니다. |

※ 주의사항 : DB2 접속 시 다음과 같은 제약 사항이 있습니다.

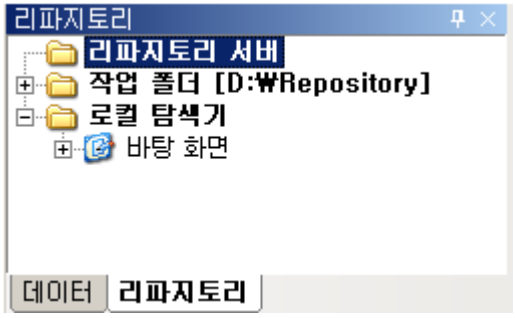
- DB에 최초 접근하는 스토어를 생성하는 경우 테이블 목록을 받아오고 스키마 에 따라 필터링하게 되는데, 이 경우 사용자가 스키마를 선택하는 과정없이 임의로 하나가 선택되게 됩니다. 이 때 접속한 사용자 이름과 같은 스키마가 있는 경우에는 접

속한 사용자 이름의 스키마가 사용되며, 사용자 이름과 같은 스키마가 없는 경우에는 스키마 목록에서 가장 마지막 스키마를 임의로 지정하게 됩니다.

- 이 두 경우 모두 사용자가 테이블을 만들어 놓은 스키마와 다른 스키마가 선택될 수 있기 때문에 원하는 테이블 목록이 안 보이는 경우가 발생할 수 있습니다.
- 그러므로, DB2 접속시에는 오즈 쿼리 디자이너에서 [필터]를 선택한 후 [테이블 스키마]에서 테이블들이 들어있는 스키마를 사용자가 직접 선택하는 것을 권장합니다.
- 이렇게 최초 접속 이후 사용자가 스키마를 선택하게 되면 이 스키마를 새로고침 하더라도 계속 사용되며, 다른 스키마를 사용자가 선택하지 전까지는 사용자가 현재 선택한 스키마를 계속 사용하게 됩니다.

## 리파지토리 창

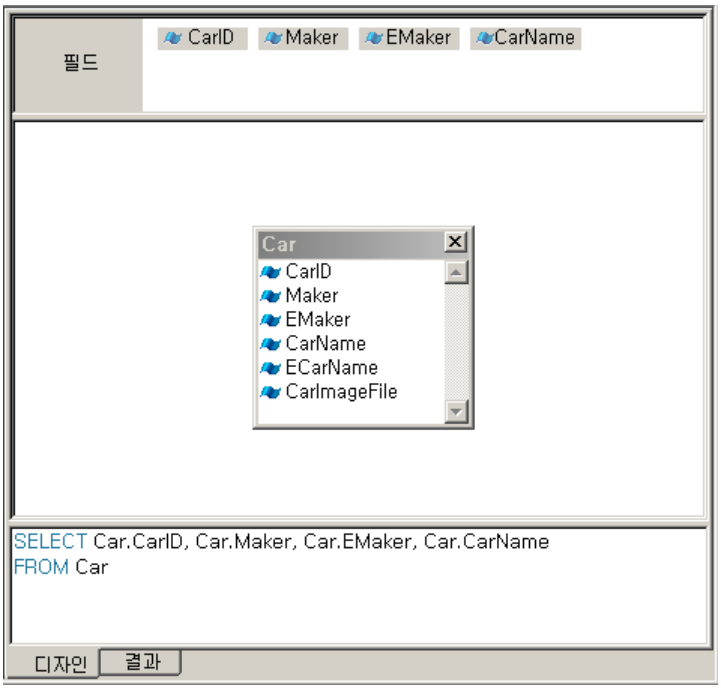
리파지토리 창은 로컬 및 서버 리파지토리의 폴더 및 파일을 탐색기 형태로 표시하는 창입니다.



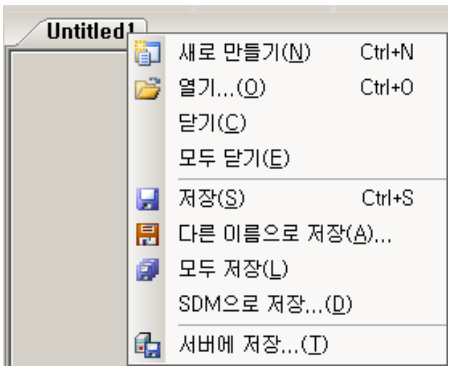
# 쿼리 창

## 디자인

쿼리를 작성하는 창으로 설정한 디자인 모드에 따라 화면이 표시됩니다. 디자인 모드에 대한 자세한 내용은 본 매뉴얼의 "Ⅲ. 오즈 쿼리 디자이너 윈도우 - 메뉴바 - 쿼리" 부분을 참조하시기 바랍니다.

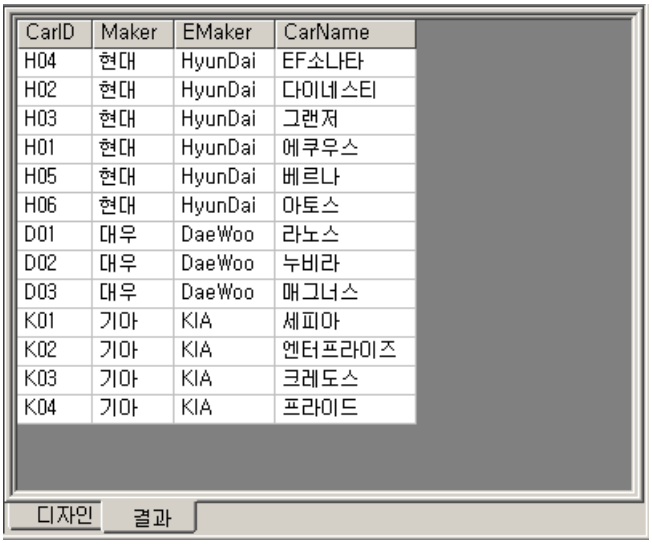



디자인 탭의 ODI 탭을 마우스 오른쪽 클릭하면 아래와 같은 메뉴가 나타납니다. 각 메뉴에 대한 자세한 설명은 "Ⅲ. 오즈 쿼리 디자이너 윈도우 - 메뉴바 - 파일" 부분을 참조하시기 바랍니다.



## 결과

쿼리를 실행한 결과 데이터 셋을 보여줍니다.



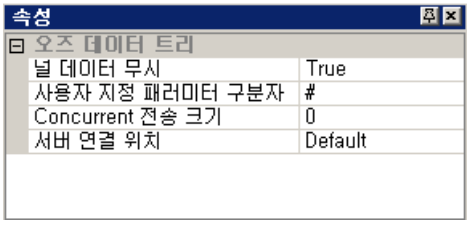
| CarID | Maker | EMaker | CarName |
|---|---|---|---|
| H04 | 현대 | HyunDai | EF소나타 |
| H02 | 현대 | HyunDai | 다이네스티 |
| H03 | 현대 | HyunDai | 그랜저 |
| H01 | 현대 | HyunDai | 에쿠우스 |
| H05 | 현대 | HyunDai | 베르나 |
| H06 | 현대 | HyunDai | 아토스 |
| D01 | 대우 | DaeWoo | 라노스 |
| D02 | 대우 | DaeWoo | 누비라 |
| D03 | 대우 | DaeWoo | 매그너스 |
| K01 | 기아 | KIA | 세피아 |
| K02 | 기아 | KIA | 엔터프라이즈 |
| K03 | 기아 | KIA | 크레도스 |
| K04 | 기아 | KIA | 프라이드 |

디자인 결과

※ 참고사항 : 결과 창에 표시된 쿼리 결과는 마우스로 드래그 & 드롭하여 선택한 후 메뉴바의 [편집] – [복사하기] 메뉴를 클릭하거나 <Ctrl + C> 키를 눌러 복사하여 텍스트 편집기 또는 엑셀에 붙여넣기할 수 있습니다. (결과 창의 타이틀은 복사되지 않으며, 데이터 내용만 복사됨)

# 속성 창

데이터 스토어, 데이터 셋, 데이터 필드 등에 대한 속성과 해당 속성에 설정된 값을 표시합니다.



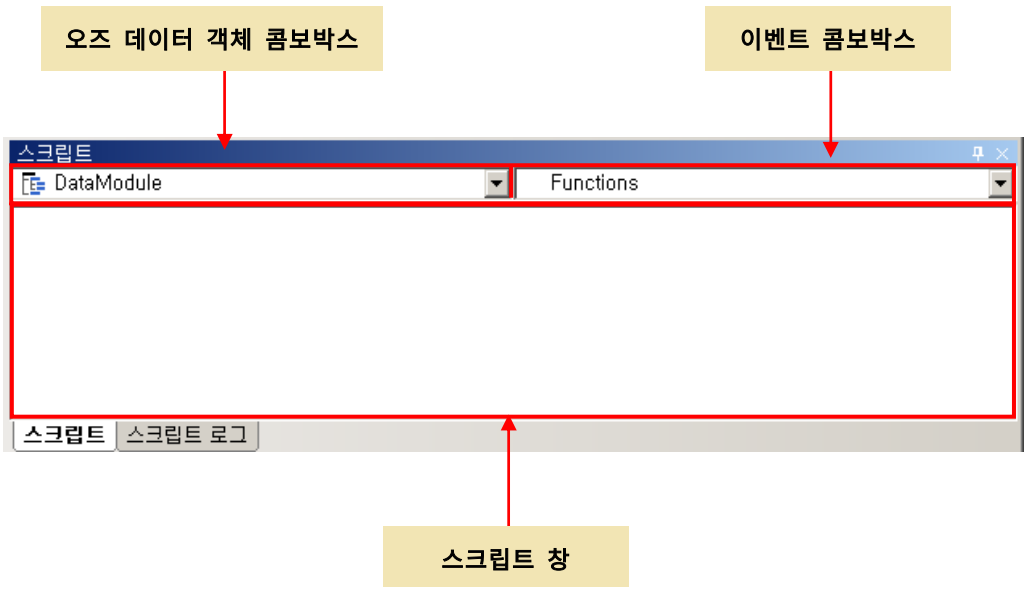
# 스크립트 창

스크립트 창은 스크립트를 입력하는 스크립트 창과 스크립트 실행 후 로그를 표시하는 스크립트 로그 창으로 나뉘어 있습니다.

## 스크립트 창

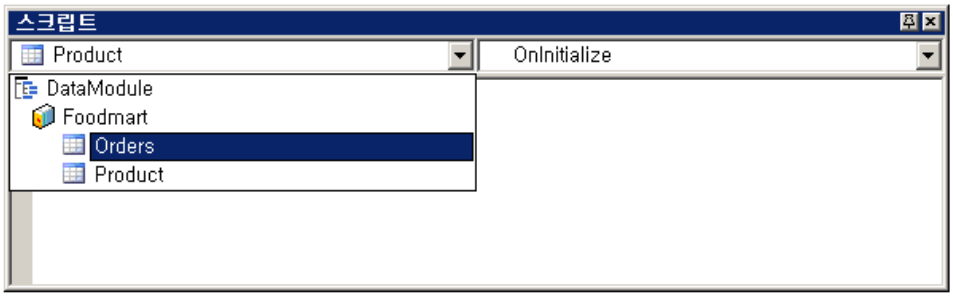
오즈 데이터 객체의 각 이벤트에 대한 스크립트를 코딩하거나 코딩된 내용을 표시하는 창입니다.

스크립트 창은 아래와 같은 화면으로 구성되어 있습니다.




■ **오즈 데이터 객체 콤보박스**

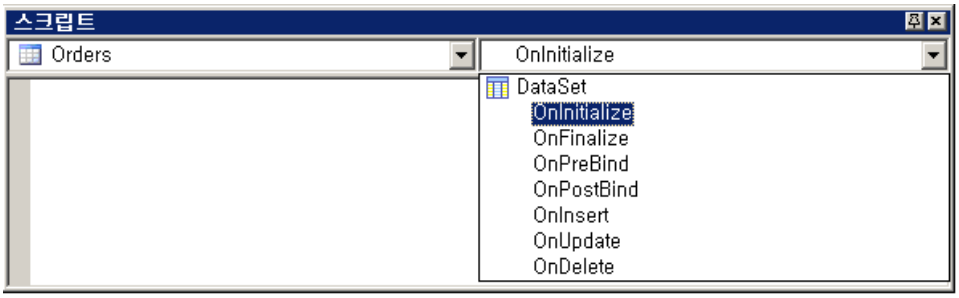
현재 선택된 오즈 데이터 객체의 이름을 나타냅니다. 콤보박스를 클릭하면 현재 ODI에 추가되어 있는 모든 오즈 데이터 객체의 이름이 리스트 형태로 나타납니다.




오즈 데이터 객체 이름을 클릭하면 리스트가 닫히면서, 해당 오즈 데이터 객체의 스크립트를 입력할 수 있습니다.

■ **이벤트 콤보박스**

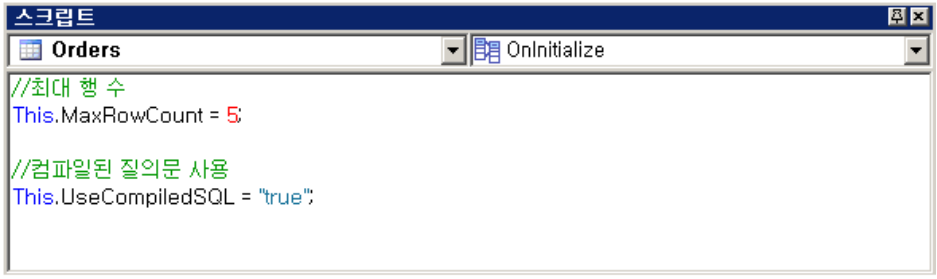
현재 선택된 이벤트를 나타냅니다. 콤보박스를 클릭하면 오즈 데이터 객체 콤보박스에 선택된 오즈 데이터 객체가 가질 수 있는 모든 이벤트가 리스트 형태로 나타납니다.




이벤트를 클릭하면 이벤트 리스트가 닫히면서, 해당 이벤트의 스크립트를 입력할 수 있습니다.

■ **스크립트 창**

스크립트를 입력할 오즈 데이터 객체와 이벤트를 선택한 후 스크립트를 입력합니다.



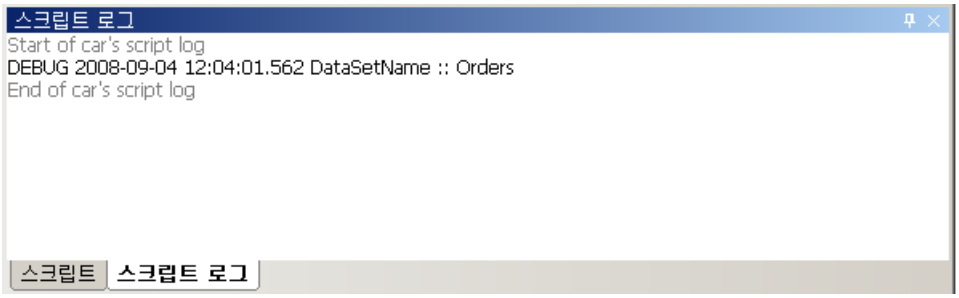

※ 참고사항 : 스크립트 창에 스크립트 입력 시 입력한 문자열에 따라 함수, 속성, 상수가

자동 완성되어 표시됩니다. 함수, 속성, 상수가 자동 완성되는 규칙은 다음 표와 같습니다.

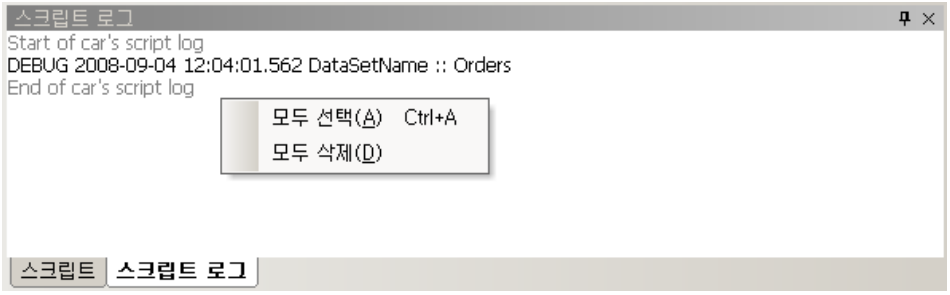
| 동작 | 자동 완성 규칙 |
|---|---|
| 컴포넌트 이름 없이 <Ctrl + Space> 키를 눌렀을 경우 | 모든 함수, 속성, 상수가 팝업 헬프 창에 표시됩니다. |
| 컴포넌트 이름 없이 "*문자열*" 입력 후 <Ctrl + Space> 키를 눌렀을 경우 | 입력한 문자열로 시작되는 모든 함수, 속성, 상수가 팝업 헬프 창에 자동으로 선택되어 표시됩니다. |
| "*컴포넌트명*." 입력 후 <Ctrl + Space> 키를 눌렀을 경우 | 해당 컴포넌트에서 사용할 수 있는 함수 또는 속성이 팝업 헬프 창에 표시됩니다. |
| "*컴포넌트명.문자열*" 입력 후 <Ctrl + Space> 키를 눌렀을 경우 | 해당 컴포넌트에서 사용할 수 있는 함수 또는 속성 중 입력한 문자열로 시작되는 함수 또는 속성이 팝업 헬프 창에 자동으로 선택되어 표시됩니다.<br>만일, 입력한 문자열로 시작되는 함수 또는 속성이 유일한 경우에는 팝업 헬프 창이 뜨지 않고 함수 또는 속성의 전체 이름으로 입력됩니다. |
| "*상수명*." 입력 후 <Ctrl + Space> 키를 눌렀을 경우 | 해당 상수 값이 팝업 헬프 창에 표시됩니다. |
| "*상수명.문자열*" 입력 후 <Ctrl + Space> 키를 눌렀을 경우 | 해당 상수 값 중 입력한 문자열로 시작되는 상수 값이 팝업 헬프 창에 자동으로 선택되어 표시됩니다.<br>만일, 입력한 문자열로 시작되는 상수 값이 유일한 경우에는 팝업 헬프 창이 뜨지 않고 상수 값의 전체 이름으로 입력됩니다. |

## 스크립트 로그 창

스크립트 함수 중 _DEBUG(), _ERROR(), _INFO() 함수를 이용하여 남긴 로그를 표시합니다.




스크립트 로그 창을 마우스 오른쪽 버튼으로 클릭하면 아래 그림과 같이 두 개의 팝업 메뉴가 나타납니다.




- 모두 선택

  스크립트 로그 창의 로그 내용을 모두 선택합니다.

- 모두 삭제

  스크립트 로그 창의 로그 내용을 모두 삭제합니다.

  ※ 참고사항 : 모두 선택하지 않고 [모두 삭제] 메뉴만 클릭하여도 모두 삭제됩니다.

※ 참고사항 : 각 함수 별로 표시되는 로그 포맷은 다음과 같습니다.

| 함수명 | 로그 포맷 |
|---|---|
| _DEBUG("로그 내용") | DEBUG yyyy-MM-dd hh:mm:ss.sss 로그 내용<br>ex) DEBUG 2008-05-20 20:17:45.031 로그 내용 |
| _ERROR("로그 내용") | ERROR yyyy-MM-dd hh:mm:ss.sss 로그 내용<br>ex) ERROR 2008-05-20 20:17:45.471 로그 내용 |
| _INFO("로그 내용") | INFO yyyy-MM-dd hh:mm:ss.sss 로그 내용<br>ex) INFO 2008-05-20 20:17:45.765 로그 내용 |

# Ⅳ. 오즈 쿼리 디자이너 사용

- 작업 폴더 설정
- 스크립트 입력, 비교
- 파일 경로 설정

# 작업 폴더 설정

작업 폴더는 리파지토리 클라이언트가 리파지토리 서버로부터 파일을 복사 받아 저장하는 로컬 작업 폴더입니다. 작업 폴더는 리파지토리를 이용한 파일의 공유 과정에서 매우 중요한 역할을 담당하는 부분입니다.

## 작업 폴더 기본 값

오즈 쿼리 디자이너가 처음 실행되면 오즈 쿼리 디자이너가 설치된 디렉토리의 Repository라는 폴더가 작업 폴더 기본 값으로 설정됩니다. 만일 해당 디렉토리에 Repository라는 폴더가 없으면 오즈 쿼리 디자이너가 실행될 때 자동으로 생성됩니다.

예를 들어, 오즈 쿼리 디자이너가 설치된 폴더가 다음과 같다면,

C:\Program Files\Forcs\OZ XStudio 5.0\OZ Query Designer 5.0

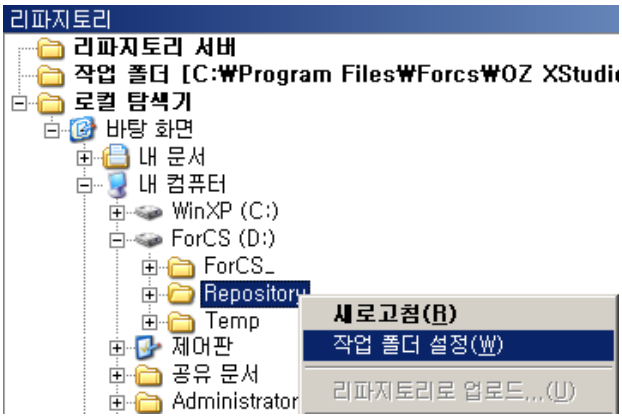
작업 폴더 기본 값은 아래와 같이 설정됩니다.

C:\Program Files\FORCS\OZ Xstudio 5.0\Repository

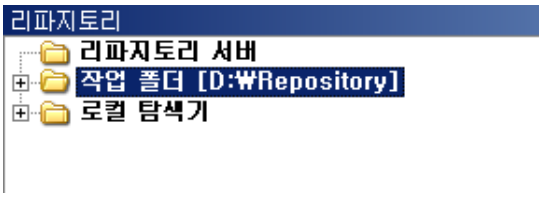
## 작업 폴더 설정

작업 폴더는 쿼리 디자이너의 리파지토리 창에서 쉽게 변경할 수 있습니다.

리파지토리 창에서 작업 폴더로 설정할 폴더를 선택한 후 마우스 오른쪽 버튼을 눌러 나타나는 팝업 메뉴에서 [작업 폴더 설정] 메뉴를 클릭하면 해당 폴더가 작업 폴더로 설정됩니다.

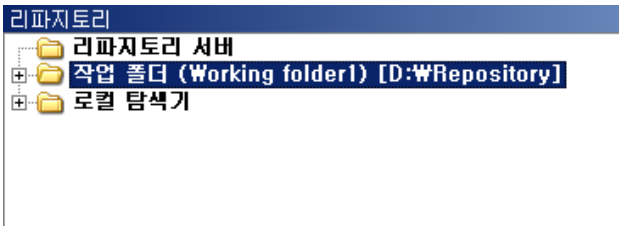


리파지토리 창의 작업 폴더가 변경된 것을 확인할 수 있으며 로컬 탐색기에서 작업 폴더를 설정할 경우 작업 폴더는 아래 그림과 같이 "작업 폴더 [작업 폴더 위치]"의 형태로 표시됩니다.




만일 선택한 경로가 작업 폴더 매니저에 등록되어 있는 경로일 경우 작업 폴더 이름과 작업 폴더 경로가 함께 표시됩니다.

예를 들어 "D:/Repository" 경로가 "Working folder1"로 작업 폴더 매니저에 등록되어 있을 경우 작업 폴더를 설정하면 작업 폴더 이름과 경로가 함께 표시됩니다.

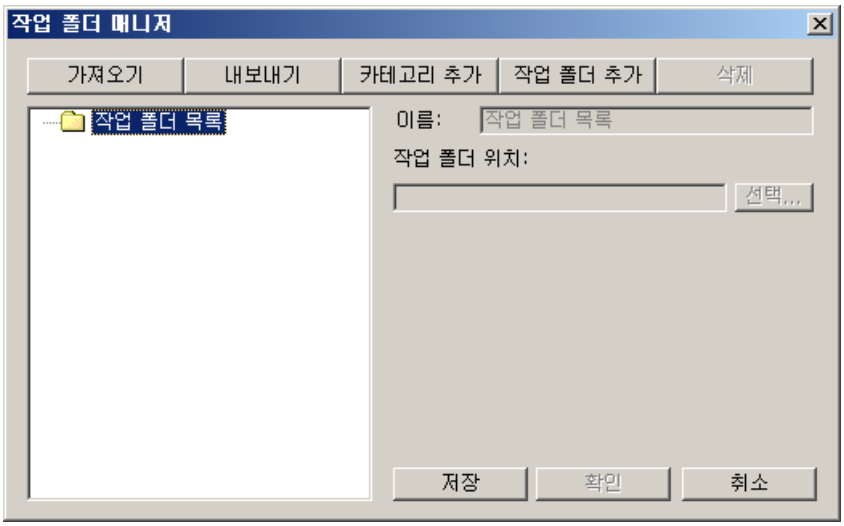



## 작업 폴더 매니저 사용

작업 폴더 매니저는 로컬 탐색기에서 여러 개의 작업 폴더를 추가하여 관리할 수 있는 관리 도구입니다.

리파지토리 창에서 작업 폴더를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [작업 폴더 매니저] 메뉴를 클릭하여 실행합니다.





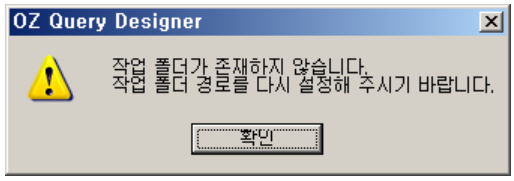

'작업 폴더 매니저'의 버튼 및 옵션에 대한 설명은 아래와 같습니다.

- 버튼

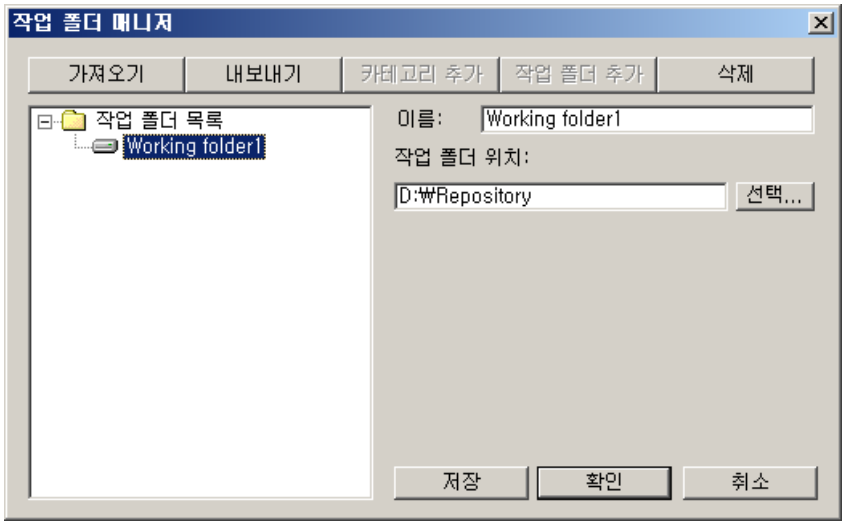
| Button | Description |
|---|---|
| **가져오기** | 작업 폴더 리스트 정보를 저장하고 있는 XML 파일을 가져와서 서버 리스트를 표시합니다. |
| **내보내기** | 현재 다이얼로그에 추가되어 있는 서버 리스트 정보를 XML 파일로 저장합니니다. |
| **카테고리 추가** | 작업 폴더 리스트에 카테고리를 추가합니다. |
| **작업 폴더 추가** | 새로운 작업 폴더를 추가합니다. [작업 폴더 추가] 버튼을 클릭하면 새로운 작업 폴더를 추가할 수 있도록 옵션 설정이 활성화 됩니다. |
| **삭제** | 선택된 작업 폴더 또는 카테고리를 삭제합니다. |
| **저장** | 현재 작업 폴더 정보를 저장합니다. |
| **확인** | 작업 폴더 매니저에서 설정한 옵션 정보를 저장 및 적용한 후 다이얼로그를 닫습니다. |
| **취소** | 작업 폴더 매니저에서 설정한 옵션 정보를 적용하지 않고 다이얼로그를 닫습니다. |

- 옵션

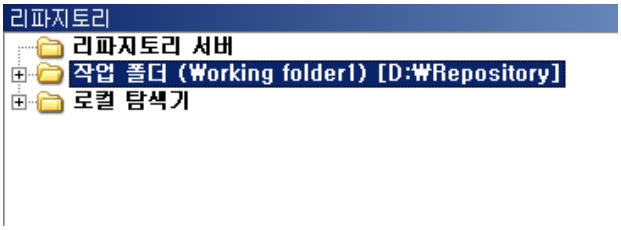
| Option | Description |
|---|---|
| **이름** | 설정할 작업 폴더의 이름을 입력합니다. |

| | |
|---|---|
| **작업 폴더 위치** | 작업 폴더 다이얼로그에서 작업 폴더의 경로를 설정합니다.<br>※ 주의사항 : 마침표(.)가 포함되어 있는 경로는 작업 폴더로 설정할 수 없습니다.<br>※ 참고사항 : 작업 폴더 위치가 존재하지 않거나 작업 폴더로 설정할 수 없는 경로일 경우 [확인] 버튼을 클릭하면 아래와 같은 에러 메시지가 표시됩니다.<br> |
| **선택** | 경로 선택 다이얼로그를 표시합니다. |

작업 폴더 정보를 추가한 후 작업 폴더로 설정할 목록을 더블 클릭하거나 작업 폴더 목록을 선택한 후 [확인] 버튼을 클릭하여 작업 폴더를 설정합니다.




설정된 작업 폴더 경로는 리파지토리 창에서 확인할 수 있으며 작업 폴더 매니저에서 작업 폴더를 설정할 경우 작업 폴더는 아래 그림과 같이 "작업 폴더 (작업 폴더 이름) [작업 폴더 위치]"의 형태로 표시됩니다.



# 스크립트 입력, 비교

오즈 쿼리 디자이너에서는 자바 스크립트 기반의 함수를 지원합니다. 자바 스크립트 기반의 함수를 입력하고, 스크립트를 비교하는 방법에 등에 대해 설명합니다.

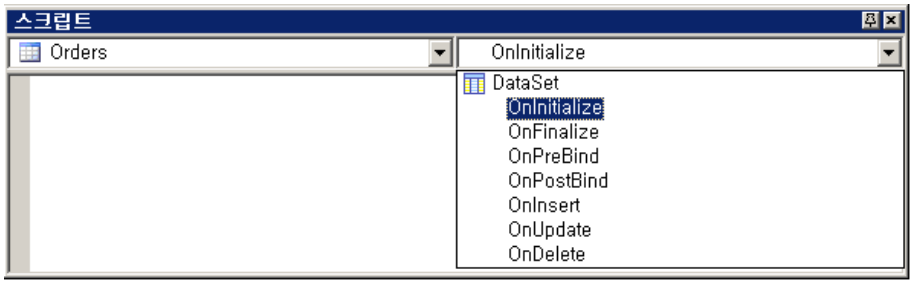
## 스크립트 입력

자바 스크립트 기반의 함수를 스크립트 편집 창에 입력하고, 스크립트 팝업 헬프 창을 사용하는 방법 등에 대해 설명합니다.

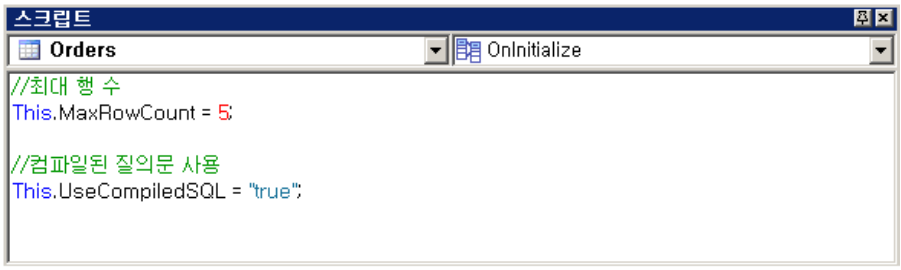
### ■ 스크립트 입력 방법

스크립트를 입력할 오즈 데이터 객체의 이벤트를 선택한 후 스크립트 편집 창에 스크립트를 입력합니다.

- 오즈 데이터 객체 선택, 스트립트 편집 창 활성화
  오즈 데이터 트리 창 또는 스크립트 편집 창의 오즈 데이터 객체 콤보박스에서 스크립트를 입력할 오즈 데이터 객체를 선택한 후 스크립트 편집 창의 이벤트 콤보박스를 선택하여 스크립트 창을 활성화합니다.

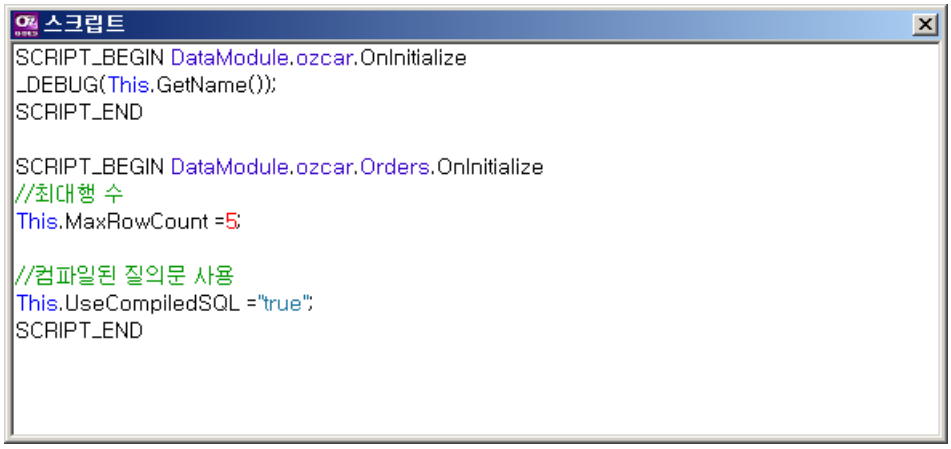



- 스크립트 입력
  스크립트 편집 창의 상단 콤보박스에 스크립트를 입력할 오즈 데이터 객체의 이름이 맞는지 확인한 후 스크립트를 입력합니다.




## ■ 스크립트 미리보기

메뉴바에서 [편집] - [스크립트 미리보기] 메뉴를 클릭하면 ODI에 입력되어 있는 모든 스크립트를 표시하는 스크립트 미리보기 창이 화면에 나타납니다.

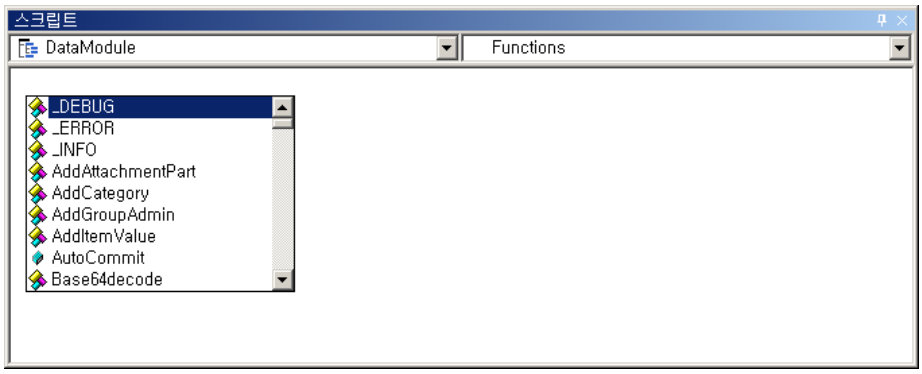



## ■ 스크립트 팝업 헬프 창

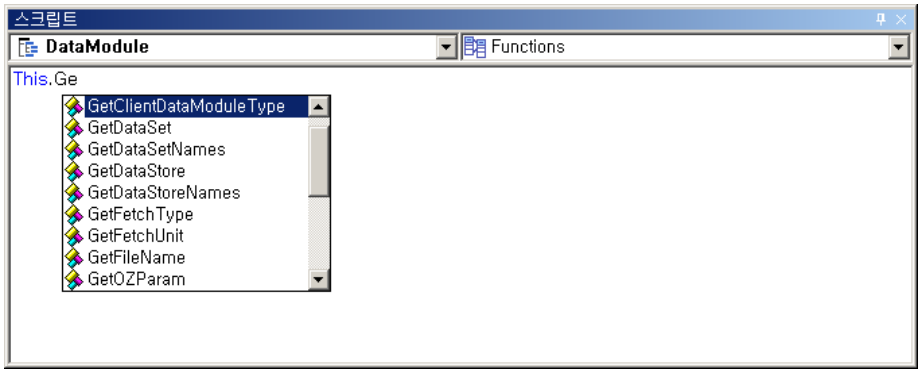
스크립트 팝업 헬프창은 자바 스크립트의 함수, 속성, 상수를 리스트 형태로 나타낸 팝업 창으로, 자동 완성 기능으로 스크립트 작성을 쉽게 할 수 있도록 도와 줍니다.

- 전체 보기

  스크립트 편집 창에서 빈 문자 다음에 <Ctrl + Space> 키를 누르면 스크립트 팝업 헬프 창이 나타납니다. 이 때 나타난 스크립트 팝업 헬프 창에는 모든 함수, 속성, 상수가 아이템으로 표시됩니다.




스크립트 팝업 헬프 창이 뜬 상태에서 함수명, 속성명, 상수명을 입력하면 입력한 문자열에 해당하는 아이템으로 자동으로 포커스가 이동하고, 이 때 <Tab> 키를 누르거나 마우스로 더블 클릭하면 자동으로 입력됩니다.



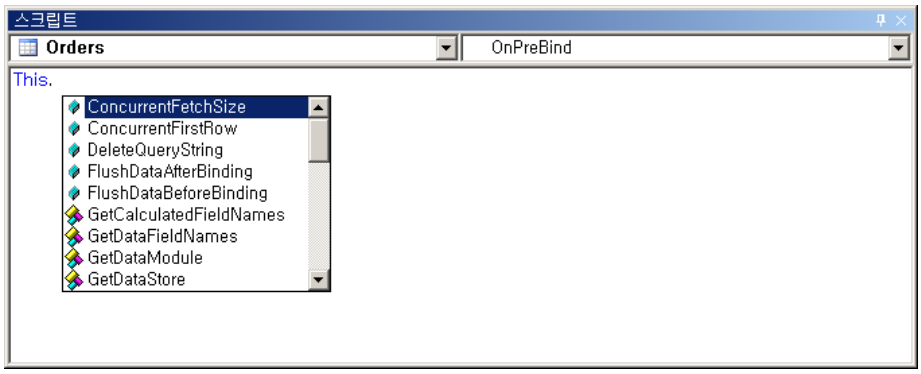

스크립트 팝업 헬프창 각각의 아이템 앞에는 해당 아이템이 어떤 내용인지 나타내는 아이콘이 있어서 해당 아이템이 함수, 속성, 상수인지 또는 오즈 데이터 객체인지를 쉽게 식별할 수 있습니다.

각각의 아이콘에 대한 간략한 설명은 다음과 같습니다.

| Icon | Description |
|---|---|
|  | 함수 |
|  | 상수 |
|  | 속성 |
|  | 오즈 데이터 모듈 |
|  | 오즈 데이터 스토어 |
|  | 오즈 데이터 셋 |

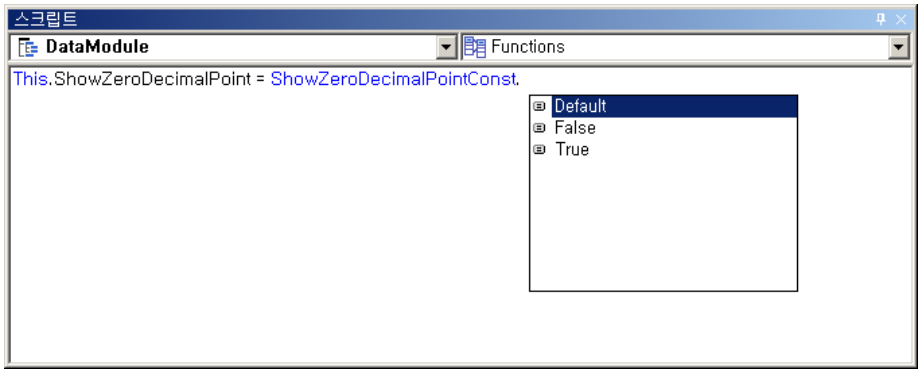
- 오즈 데이터 객체 별 함수 보기

  오즈 데이터 객체명을 입력하고 "."를 입력하면 해당 컴포넌트가 가질 수 있는 함수를 아이템으로 나타낸 스크립트 팝업 헬프 창이 자동으로 표시됩니다.




- 상수 값 보기

  상수명을 입력하고 "."를 입력하면 해당 상수가 가질 수 있는 값을 아이템으로 나타낸 스크립트 팝업 헬프창이 자동으로 표시됩니다.




- 스크립트 자동 완성 규칙

  스크립트 편집 창에 스크립트 입력 시 입력한 문자열에 따라 함수, 속성, 상수가 자동 완성되어 표시됩니다. 함수, 속성, 상수가 자동 완성되는 규칙은 다음 표와 같습니다.

| 동작 | 자동 완성 규칙 |
|---|---|
| 컴포넌트 이름 없이 <Ctrl + Space> 키를 눌렀을 경우 | 모든 함수, 속성, 상수가 팝업 헬프 창에 표시됩니다. |
| 컴포넌트 이름 없이 "*문자열*" 입력 후 <Ctrl + Space> 키를 눌렀을 경우 | 입력한 문자열로 시작되는 모든 함수, 속성, 상수가 팝업 헬프 창에 자동으로 선택되어 표시됩니다. |
| "*컴포넌트명*." 입력 후 <Ctrl + Space> 키를 눌렀을 경우 | 해당 컴포넌트에서 사용할 수 있는 함수 또는 속성이 팝업 헬프 창에 표시됩니다. |

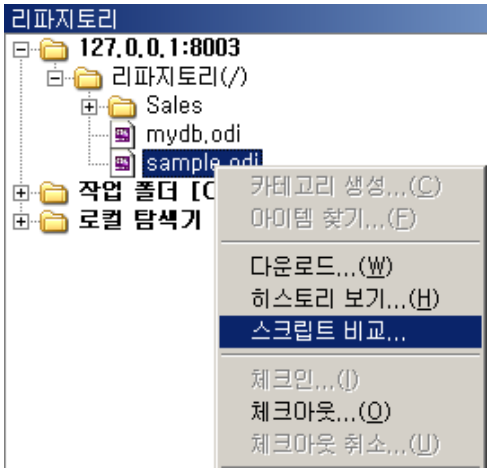
| | |
|---|---|
| "*컴포넌트명.문자열*" 입력 후 <Ctrl + Space> 키를 눌렀을 경우 | 해당 컴포넌트에서 사용할 수 있는 함수 또는 속성 중 입력한 문자열로 시작되는 함수 또는 속성이 팝업 헬프 창에 자동으로 선택되어 표시됩니다.<br>만일, 입력한 문자열로 시작되는 함수 또는 속성이 유일한 경우에는 팝업 헬프 창이 뜨지 않고 함수 또는 속성의 전체 이름으로 입력됩니다. |
| "*상수명.*" 입력 후 <Ctrl + Space> 키를 눌렀을 경우 | 해당 상수 값이 팝업 헬프 창에 표시됩니다. |
| "*상수명.문자열*" 입력 후 <Ctrl + Space> 키를 눌렀을 경우 | 해당 상수 값 중 입력한 문자열로 시작되는 상수 값이 팝업 헬프 창에 자동으로 선택되어 표시됩니다.<br>만일, 입력한 문자열로 시작되는 상수 값이 유일한 경우에는 팝업 헬프 창이 뜨지 않고 상수 값의 전체 이름으로 입력됩니다. |

## 스크립트 비교

ODI 파일에 입력된 스크립트를 비교하는 방법에 대해 설명합니다.

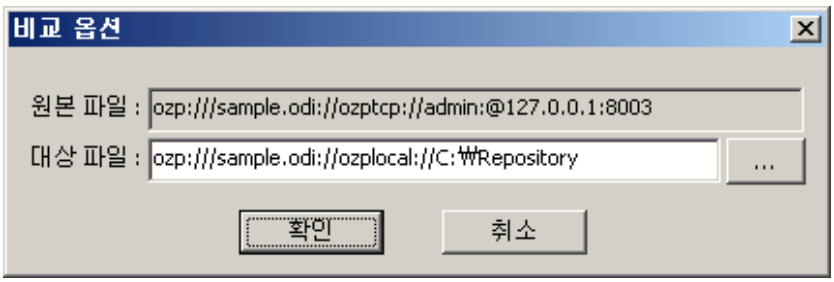
### ■ 스크립트 비교 – 서버 파일과 로컬 파일 비교

리파지토리창에서 스크립트를 비교할 ODI 파일을 마우스 오른쪽 버튼을 클릭하여 나타나는 팝업 메뉴에서 [스크립트 비교] 메뉴를 클릭합니다.



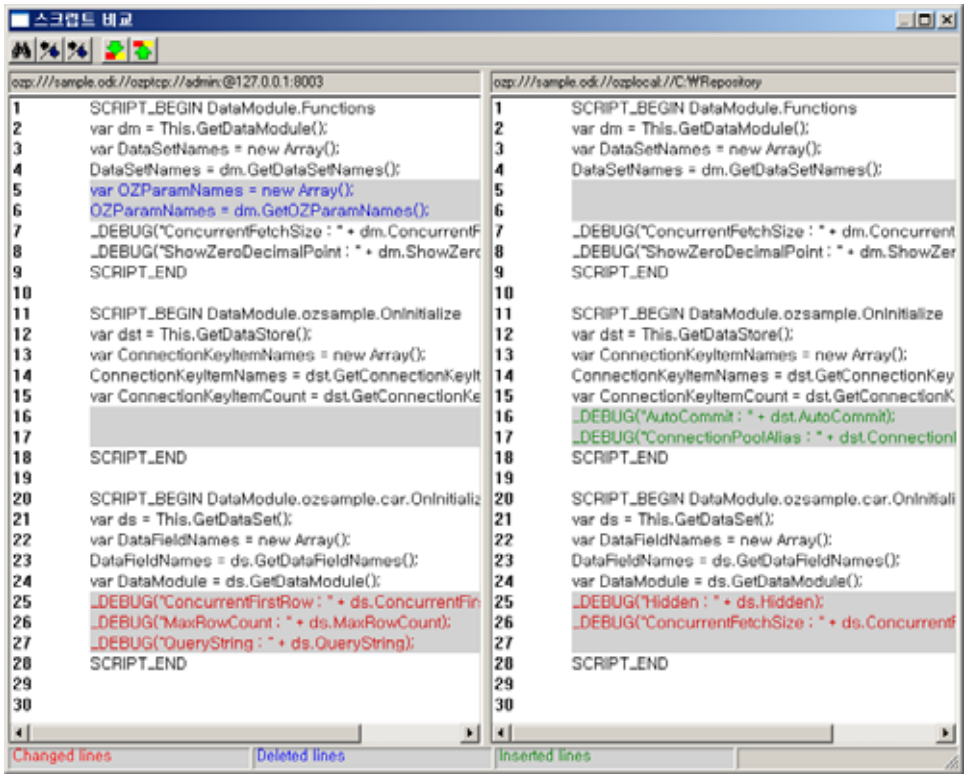

[스크립트 비교] 메뉴를 클릭하면 원본 파일과 대상 파일을 선택할 수 있는 "비교 옵션 다

이얼로그"가 나타납니다.




원본 파일에는 비교할 원본 파일의 전체 경로를 리파지토리 위치를 포함하여 나타내고, 대상 파일에는 비교할 대상 파일의 전체 경로를 리파지토리 위치를 포함하여 나타냅니다.
리파지토리 서버에 연결한 후 로컬 작업 폴더에 있는 파일을 선택하고 [스크립트 비교] 메뉴를 클릭하면 서버 리파지토리에서 파일을 찾아 자동으로 대상 파일에 표시해주고, 서버 리파지토리에서 [스크립트 비교] 메뉴를 클릭하면 로컬 작업 폴더에서 파일을 찾아 자동으로 대상 파일에 표시해줍니다. 파일을 찾지 못했을 경우 대상 파일은 빈 항목으로 표시됩니다.
대상 파일은 [...] 버튼을 클릭하여 변경할 수 있으며, 원본 파일과 같은 확장자의 파일만 선택할 수 있습니다.

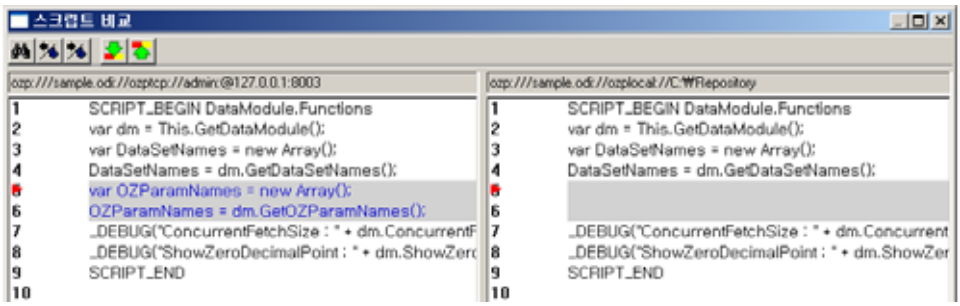
스크립트 비교 옵션을 선택한 후 [확인] 버튼을 클릭하면 스크립트 비교 다이얼로그가 실행됩니다.




스크립트 비교 시 원본 파일에만 스크립트가 있을 경우 파란색으로, 대상 파일에만 스크립트가 있을 경우 초록색으로, 같은 라인이 다를 경우 빨간색 글씨로 표시됩니다.

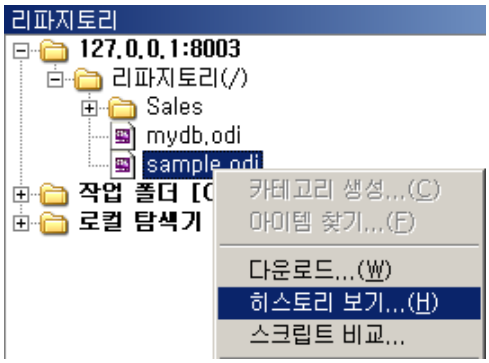
찾기(), 이전 찾기(), 다음 찾기() 아이콘을 클릭하여 텍스트를 검색할 수 있으며, 다음 차이점 이동(), 이전 차이점 이동() 아이콘을 클릭하여 스크립트가 다른 라인으로 커서를 이동할 수 있습니다.

차이점을 이동하면 해당 라인에 빨간색 화살표가 표시됩니다.

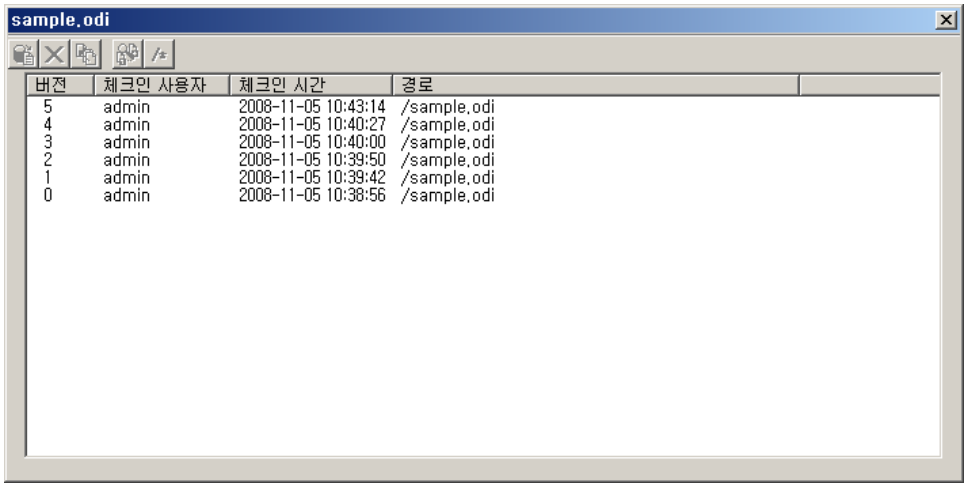


## ■ 스크립트 비교 – 서버 파일의 버전 간 비교

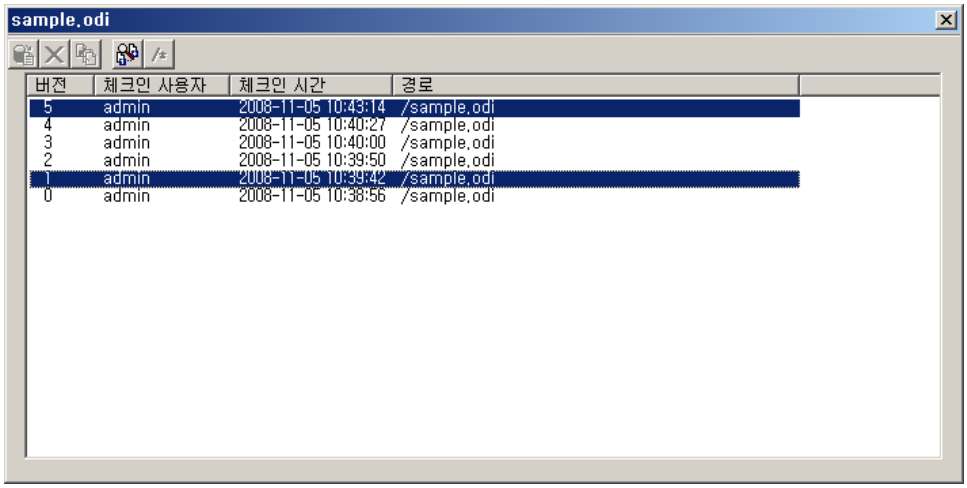
리파지토리창의 서버 리파지토리에서 버전 간 스크립트를 비교할 ODI 파일을 마우스 오른쪽 버튼을 클릭하여 나타나는 팝업 메뉴에서 [히스토리 보기] 메뉴를 클릭합니다.



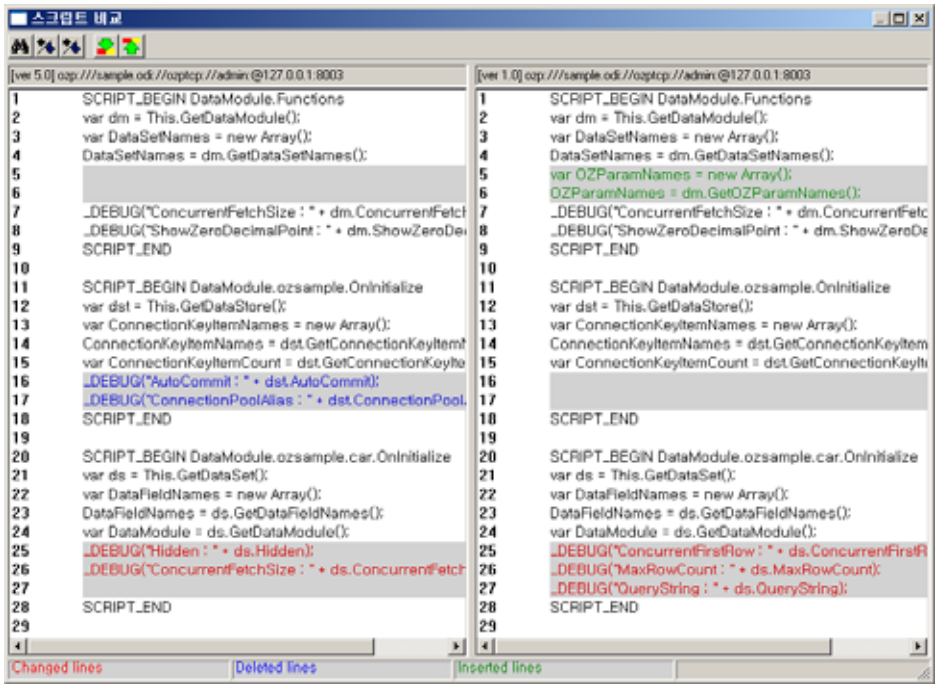

[히스토리 보기] 메뉴를 클릭하면 선택한 아이템의 히스토리를 나타낸 다이얼로그가 나타납니다.




스크립트를 비교할 버전을 선택한 후 스크립트 비교 아이콘()을 클릭합니다.




스크립트 비교 다이얼로그가 실행되면 버전별로 스크립트를 비교할 수 있습니다.




버전은 화면 상단 타이틀을 확인하시기 바랍니다.

※ 참고사항

- 특정 버전과 로컬에 있는 파일 또는 작업 중인 파일과 비교하려면 해당 버전을 선택한 후 스크립트 비교 아이콘( )을 클릭하시기 바랍니다.
- 두 개의 버전을 선택하면 툴바의 다운로드, 삭제, 롤백, 주석 아이콘은 비활성화되

고, 세 개 이상의 버전을 선택하면 모든 아이콘이 비활성화됩니다.

■ **스크립트 비교 관련 참고사항**

- 확장자가 ODI인 파일끼리만 스크립트 비교를 할 수 있습니다.
- 스크립트 비교 다이얼로그는 단순히 비교만 하는 다이얼로그로, 스크립트 편집은 할 수 없습니다.

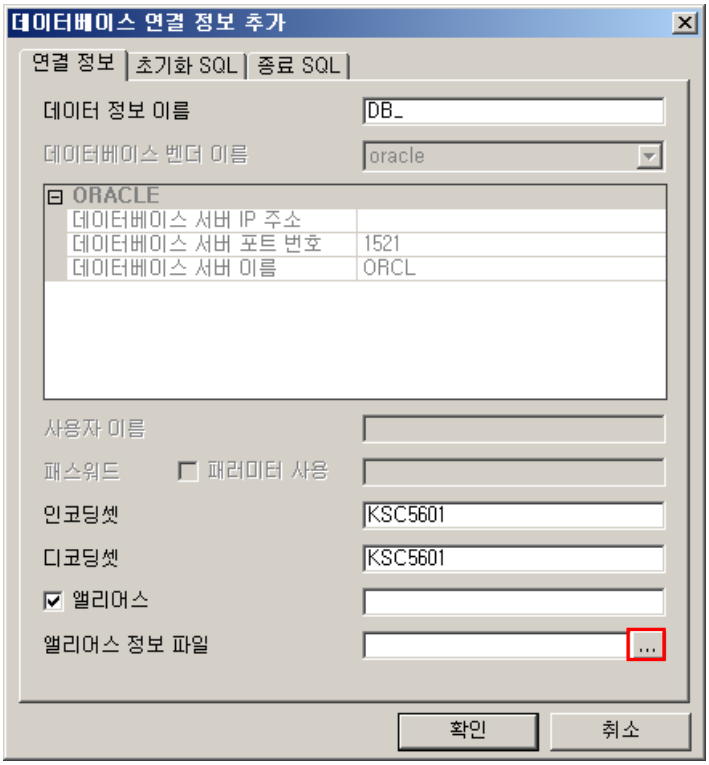
# 파일 경로 설정

오즈 쿼리 디자이너에서 스토어 추가 시 앨리어스 정보 파일(db.properties, datasource.properties), HTTP/파일 스토어의 XML, CSV 파일 등의 경로를 설정하는 방법에 대해 설명합니다.

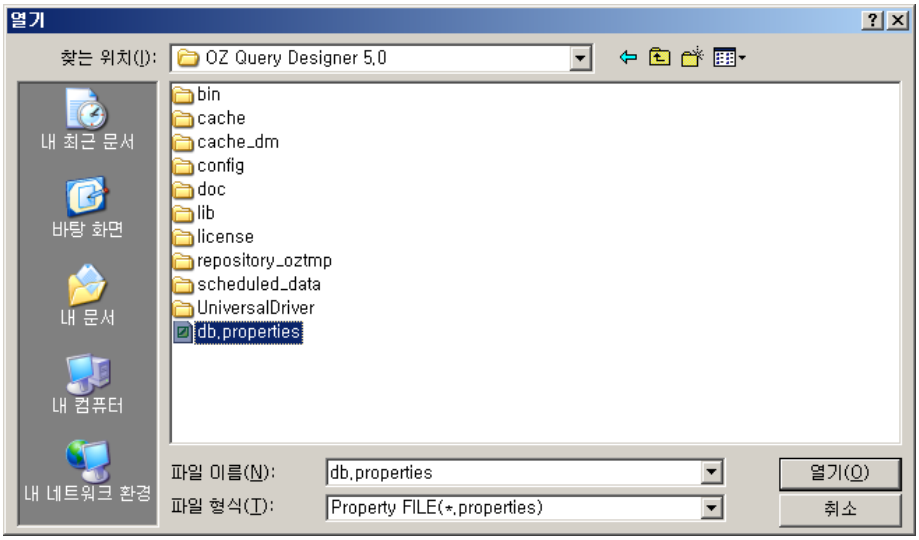
## 다이얼로그에서 설정하기

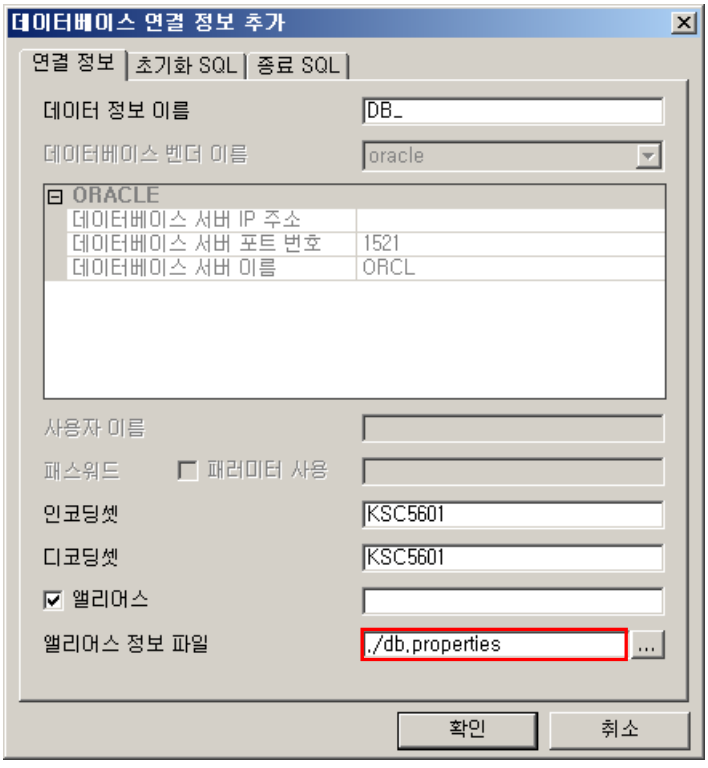
파일 경로 선택 버튼을 클릭하여 표시되는 "열기 다이얼로그"에서 파일을 선택합니다.

예를 들어 데이터베이스 연결 정보의 앨리어스 정보 파일을 설정할 경우 파일 경로 선택 버튼()을 클릭하면 "열기 다이얼로그"가 표시됩니다.




"열기 다이얼로그"에서 파일을 선택한 후 [확인] 버튼을 클릭하면 파일 경로가 설정됩니다.




"열기 다이얼로그"에서 선택한 드라이브(ex. C:\)가 디자이너가 설치된 드라이브(ex. C:\)와 동일한 경우 자동으로 상대 경로로 표시되며, 다른 드라이브일 경우에는 절대 경로로 표시됩니다.

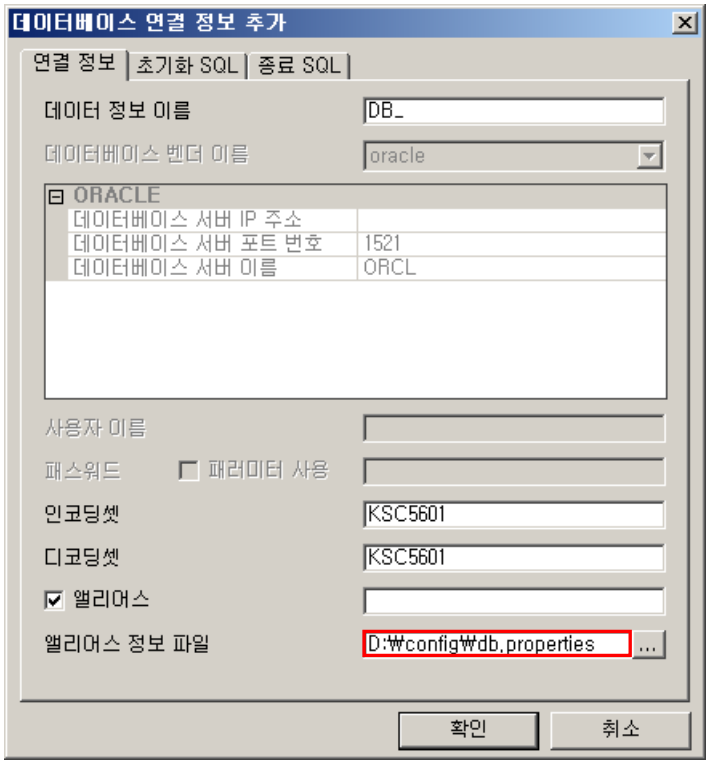
## 직접 입력하여 설정하기

파일 경로를 절대 경로 또는 상대 경로로 사용자가 직접 입력하여 설정하는 방법에 대해 설명합니다.

### ■ 절대 경로

파일의 전체 경로를 사용자가 직접 입력하여 설정합니다.

예를 들어 데이터베이스 연결 정보의 앨리어스 정보 파일을 설정할 경우, 아래와 같이 파일의 절대 경로를 입력하여 설정합니다.




### ■ 상대 경로

쿼리 디자이너가 실행되는 경로를 기준으로 설정된 파일의 상대 경로를 사용자가 직접 입력하여 설정합니다.

상대 경로가 지원되는 다이얼로그는 다음과 같습니다.

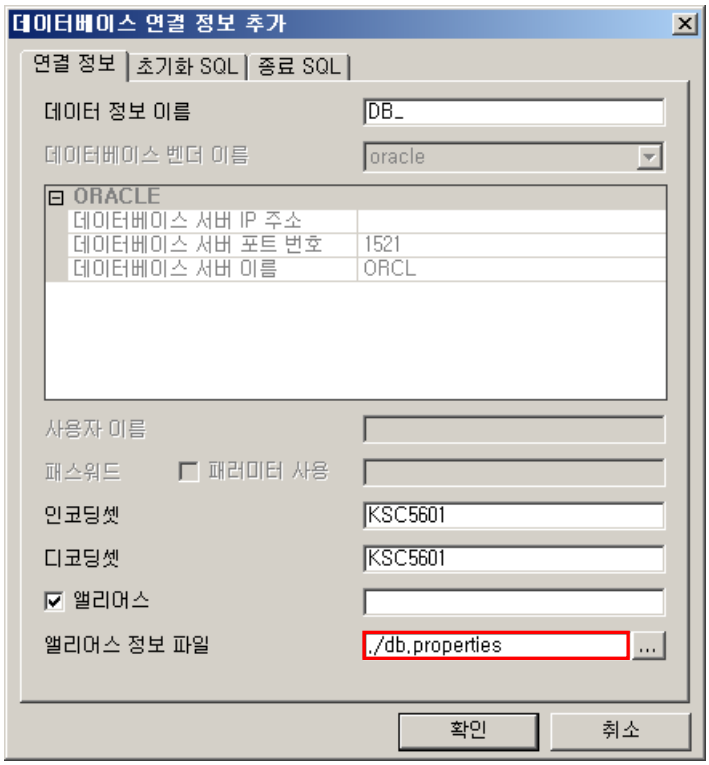
1) 모든 데이터 스토어, 데이터 셋 : 앨리어스 정보 파일
2) 파일/HTTP 스토어 : XML 데이터 셋의 XML 파일 경로, CSV 데이터 셋의 CSV 파일

경로

3) XML 스토어 : XML, DTD, XSD 파일 경로

4) SOAP 스토어 : WSDL 위치

예를 들어 데이터베이스 연결 정보의 앨리어스 정보 파일을 설정할 경우 쿼리 디자이너가 실행되는 경로가 "C:\Program Files\Forcs\OZ XStudio 5.0\OZ Query Designer 5.0\bin" 이고, 앨리어스 정보 파일의 경로가 "C:\Program Files\Forcs\OZ XStudio 5.0\OZ Query Designer 5.0\db.properties"일 때 아래와 같이 파일의 상대 경로를 입력하여 설정합니다.




※ 주의사항 : 애플리케이션 디자이너 또는 리포트 디자이너에서 파일 경로가 상대 경로로 설정된 ODI를 사용할 경우 애플리케이션/리포트 디자이너가 설치된 경로를 기준으로 상대 경로가 설정되므로, 폼 또는 보고서 개발 시에는 애플리케이션/리포트 디자이너의 상대 경로에 관련 파일을 위치시키고 운영 시에는 ozserver.properties에 설정된 HOME 경로를 기준으로하여 관련 파일을 위치시켜야 합니다.

※ 참고사항

- 파일 경로 설정 시 경로는 "/"로 시작할 수 없으며, "./../aa/../bb"의 형태의 경로는 지원하지 않습니다.

- 앨리어스 정보 파일(db.properties, datasource.properties)이 HOME 경로에 위치한 경우 파일 경로에 파일명만 입력하여 상대 경로로 설정 가능합니다.

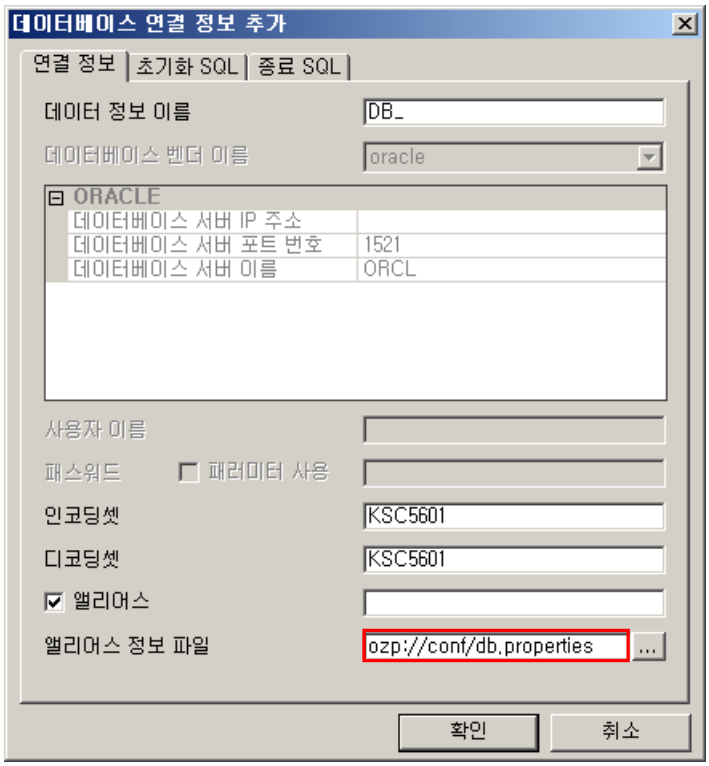
### ■ 오즈 URL

리파지토리를 기준으로 한 오즈 URL(ozp://*카테고리*/*파일명*)로 파일의 경로를 사용자가 직접 입력하여 설정합니다.

오즈 URL이 지원되는 다이얼로그는 다음과 같습니다.

1) 모든 데이터 스토어, 데이터 셋 : 앨리어스 정보 파일
2) 파일/HTTP 스토어 : XML 데이터 셋의 XML 파일 경로, CSV 데이터 셋의 CSV 파일 경로
3) XML 스토어 : XML, DTD, XSD 파일 경로
4) SOAP 스토어 : WSDL 위치

예를 들어 데이터베이스 연결 정보의 앨리어스 정보 파일을 설정할 경우 리파지토리의 conf 폴더에 앨리어스 정보 파일이 위치할 때 아래와 같이 파일의 상대 경로를 입력하여 설정합니다.



※ 참고사항 : "ozp://"로 경로 설정 시 ODI를 서버에서 열 경우에는 서버 리파지토리 기준으로, 로컬에서 열 경우에는 작업 폴더를 기준으로 파일 경로가 설정됩니다.

# Ⅴ. 데이터 소스와 데이터 셋

- 데이터 소스 개요
- 데이터 셋 개요
- 데이터 소스 종류
- 데이터 소스별 속성
- 데이터베이스 연결
- 동적 데이터베이스 연결

**본 장에서는 오즈 쿼리 디자이너에서 지원되는 데이터 소스의 종류와 각 속성에 대한 개요, 데이터 소스별 연결 방법 및 각 데이터 소스별로 데이터 셋을 생성하는 방법 등을 설명합니다.**

## 데이터 소스 개요

데이터 소스란 보고서 및 애플리케이션 작성에 사용되는 데이터의 제공자를 말합니다. 오즈 쿼리 디자이너에서는 상용 데이터베이스 뿐만 아니라 파일이나 HTTP 출력, XML 등 다양한 데이터 소스를 지원합니다.

오즈 쿼리 디자이너는 다양한 사용자 데이터 소스에 대해서 일관성 있는 접근 방법을 부여함으로써 개발자가 데이터 소스의 종류에 상관없이 일관되고 유연한 보고서 및 애플리케이션을 작성할 수 있도록 지원합니다.

## 데이터 셋 개요

데이터 셋이란 데이터 소스로부터 필요한 데이터를 선택적으로 수집한 정보의 집합을 말합니다.

데이터 셋은 데이터 소스의 종류에 따라 각기 다른 방법으로 생성되지만 데이터 소스와 상관없이 항상 동일한 형태의 데이터 셋이 생성되며, 보고서 및 애플리케이션에 동일한 방법으로 적용됩니다.

# 데이터 소스 종류

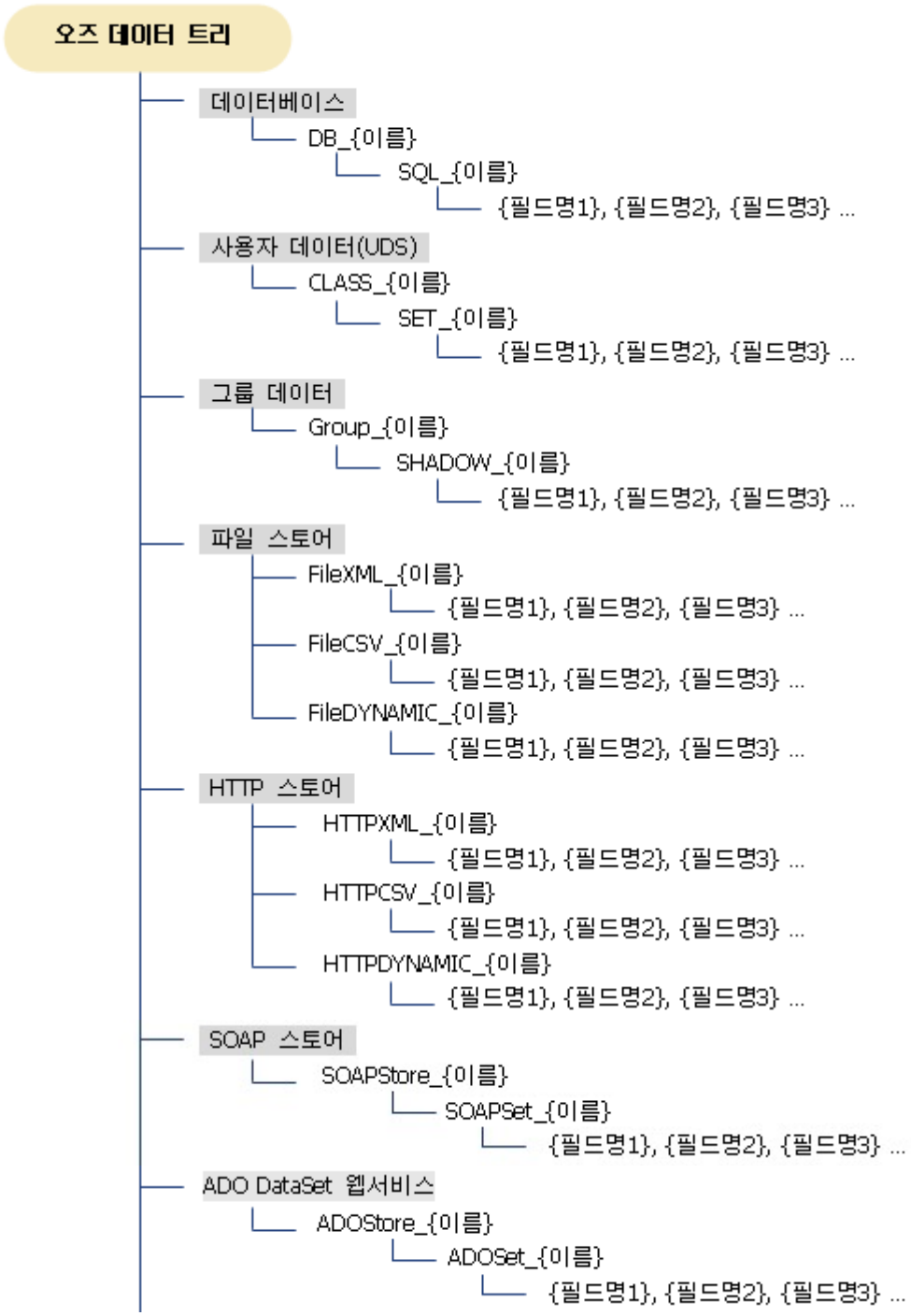
## 데이터 소스 종류

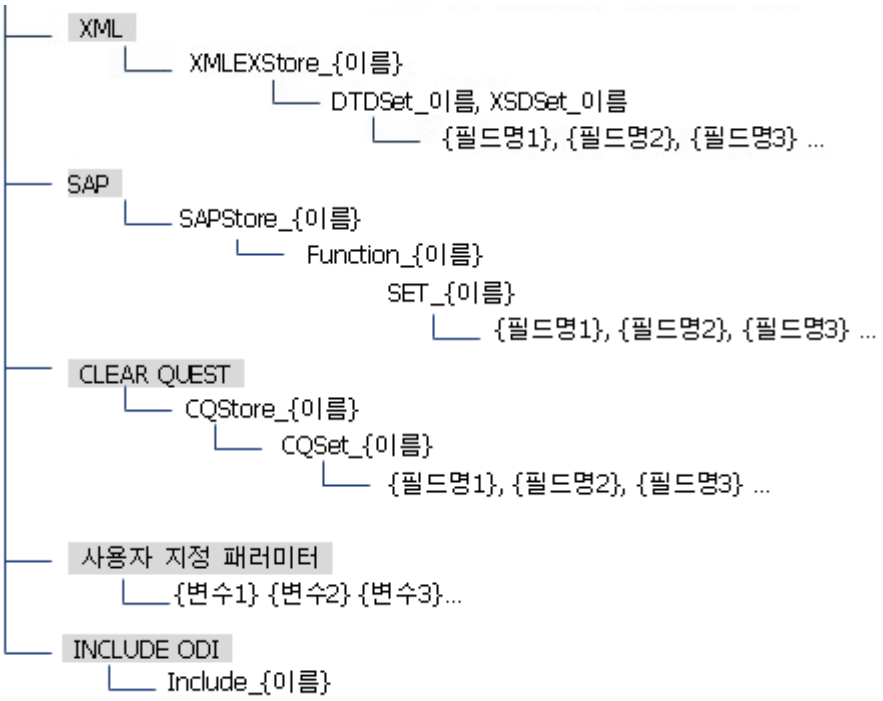
오즈 쿼리 디자이너가 지원하는 데이터 소스의 종류는 다음과 같습니다.

| | |
|---|---|
| **데이터베이스** | JDBC 또는 ODBC를 지원하는 데이터베이스로부터 SQL 쿼리를 이용해 데이터 셋을 생성하는 기능을 제공합니다. |
| **사용자 데이터(UDS)** | 일반 파일 데이터(텍스트 파일, CSV, XML) 및 사용자 프로그램 데이터(EJB, Servlet, ASP, JSP, Stored Procedure)와 같은 다양한 형태의 외부 데이터 소스로부터 데이터 셋을 생성하기 위해 정의된 Java Interface입니다. |
| **그룹 데이터** | 데이터베이스, UDS 등 타 데이터 소스를 통해 생성한 데이터 셋을 특정 필드들을 기준으로 분할하여 새로운 데이터 셋을 생성하는 기능을 제공합니다.<br>일반적으로 그룹 데이터 스토어는 하나의 데이터 셋을 마스터/디테일 데이터 셋으로 재구성하고자 할 때 사용됩니다. |
| **파일 스토어** | 로컬에 있는 XML(DTD가 없는 일정한 형식의 XML 파일), CSV 형식의 파일 데이터에 대한 표준화된 연동 방법입니다. |
| **HTTP 스토어** | 원격 시스템에 있는 XML(DTD가 없는 일정한 형식의 XML 파일), CSV 형식의 파일 데이터에 대한 표준화된 연동 방법입니다. |
| **SOAP 스토어** | 웹 서비스를 통하여 얻어온 데이터를 데이터 소스로 사용하여 데이터 셋을 생성하는 기능을 제공합니다. |
| **ADO DataSet 웹서비스** | 닷넷 웹 서비스에서 ADO DataSet 객체를 연동하여 데이터 셋을 생성하는 기능을 제공합니다. |
| **XML** | DTD/XSD를 이용하는 XML 형식의 파일 데이터에 대한 표준화된 연동 방법입니다. |
| **SAP** | SAP R/3에 정의된 데이터를 활용하여 다양한 보고서 및 애플리케이션을 작성하기 위한 SAP 연결을 지원하여 별도의 ABAP 코딩 없이 데이터 셋을 생성하는 기능을 제공합니다. |
| **CLEAR QUEST** | 형상 관리 툴인 IBM Rational Clear Quest 프로그램에서 생성한 DB를 연동하여 데이터 셋을 생성하는 기능을 제공합니다. |

## 오즈 데이터 트리

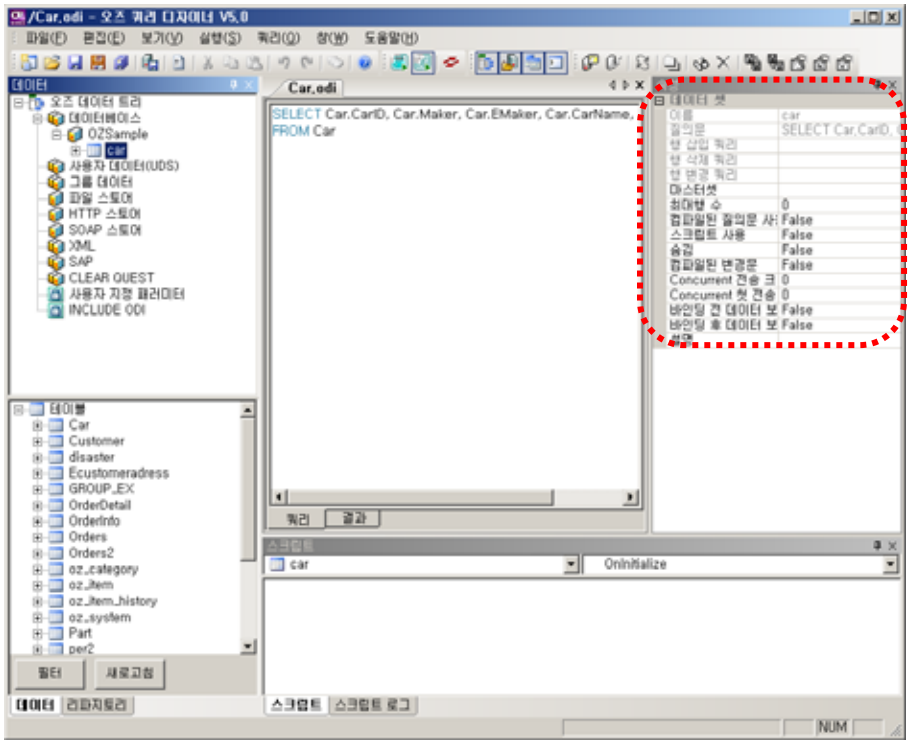
오즈 데이터 트리는 오즈 쿼리 디자이너에서 연결할 수 있는 데이터 스토어, 데이터 셋, 사용자 지정 패러미터 등을 나타낸 것으로, 오즈 쿼리 디자이너의 데이터 창에 트리 형태로 표시됩니다. 오즈 데이터 트리의 구성은 다음과 같습니다.




XML
XMLEXStore_{이름}
DTDSet_이름, XSDSet_이름
{필드명1}, {필드명2}, {필드명3} ...
SAP
SAPStore_{이름}
Function_{이름}
SET_{이름}
{필드명1}, {필드명2}, {필드명3} ...
CLEAR QUEST
CQStore_{이름}
CQSet_{이름}
{필드명1}, {필드명2}, {필드명3} ...
사용자 지정 패러미터
{변수1} {변수2} {변수3}...
INCLUDE ODI
Include_{이름}

# 데이터 소스별 속성

데이터 스토어, 데이터 셋, 데이터 필드 등의 속성 값은 오즈 쿼리 디자이너 화면 오른쪽의 속성 창에 표시되며, 속성 창을 이용하여 속성 값을 변경합니다.




다음으로 각 데이터 소스별로 데이터 스토어, 데이터 셋, 데이터 필드의 속성에 대해 설명합니다.

## 오즈 데이터 트리

| 속성명 | 설명 | |
|---|---|---|
| 널 데이터 무시 | True | 문자형일 경우는 ""(빈 문자)로, 숫자형일 경우는 0으로 처리합니다. |

<table>
<tr><td></td><td>False</td><td>데이터 값이 Null인 경우 Null Data(데이터 값이 없음을 나타냄)로 취급하며, 스크립트에서 "null" 상수와의 비교를 통해 Null Data를 판단할 수 있습니다. 오즈 리포트 디자이너에서 바인딩 시 확인할 수 있습니다.</td></tr>
<tr><td>사용자 지정 패러미터 구분자</td><td colspan="2">패러미터를 구분하는 구분자를 지정하는 속성으로 디폴트 값은 '#'입니다. 예를 들어 구분자가 '#'으로 지정된 경우 패러미터는 #OZParam.패러미터이름# 형식으로 설정됩니다.<br>※ 주의사항<br>▪ 디자인 모드가 매뉴얼이고 데이터 셋의 속성 값 중 '스크립트 사용'이 'True'인 경우 구분자로 디폴트 값인 '#'만 사용할 수 있습니다.<br>▪ 사용자 지정 패러미터는 한 글자만 입력하여야 정상적으로 동작합니다.<br>▪ 알파벳, 숫자 또는 1byte 이상의 문자는 구분자로 사용할 수 없으며 RDB일 경우 ^ 또는 DB 연결 정보에 포함되어 있는 문자열도 구분자로 사용할 수 없습니다.</td></tr>
<tr><td>Concurrent 전송 크기</td><td colspan="2">클라이언트에서 데이터 요청 시 한꺼번에 가져올 결과 데이터 셋의 크기를 Byte 단위로 설정합니다. 설정된 값만큼의 버퍼 단위로 데이터를 내보냅니다.<br>※ 참고사항 : 속성 값을 '0'으로 설정하면 서버에 설정된 값이 적용됩니다.<br>※ 주의사항 : 이 속성 값은 쿼리 디자이너에서 쿼리문을 실행할 때에는 적용되지 않으며, 클라이언트 즉, 애플리케이션 뷰어 또는 리포트 뷰어에서 데이터를 요청할 경우에만 적용되는 속성입니다.</td></tr>
<tr><td rowspan="4">서버 연결 위치</td><td colspan="2">데이터 바인딩 시 연결할 서버를 설정합니다.<br>※ 주의사항 : 서버 연결 위치는 오즈 데이터 트리의 "서버 연결 위치" 속성 값과 쿼리 디자이너의 환경 설정 옵션 중 "서버 연결 위치" 옵션 값에 따라 다르게 설정될 수 있으며, 자세한 내용은 아래 "바인딩 시 연결되는 서버 위치"를 참조하시기 바랍니다.</td></tr>
<tr><td>Default</td><td>쿼리 디자이너 환경 설정 중 "서버 연결 위치" 옵션 값에 따라 연결할 서버를 설정합니다. (기본 값)</td></tr>
<tr><td>OZServer</td><td>현재 연결된 리파지토리 서버로 서버를 설정합니다.</td></tr>
<tr><td>Local</td><td>로컬 서버로 설정합니다.</td></tr>
</table>

| | | |
|---|---|---|
| **소수점 앞에 0 표시** | 데이터 값이 1 보다 작을 경우 소수점 앞에 0을 표시할지 여부를 설정합니다.<br>※ 주의사항 : 해당 기능은 데이터 타입이 "Numeric" 또는"Decimal"인 데이터에서만 지원됩니다. | |
| | Default | DB 서버에 설정된 속성 값에 따라 소수점 앞에 0을 표시합니다. (기본 값) |
| | True | DB 서버에 설정된 속성 값에 상관없이 소수점 앞에 무조건 0을 표시합니다. |
| | False | DB 서버에 설정된 속성 값에 상관없이 소수점 앞에 무조건 0을 표시하지 않습니다. |
| **스크립트 에러 무시** | 쿼리문 실행 시 스크립트 에러가 발생할 경우 에러 메시지 표시 여부를 설정합니다. (기본 값:False)<br>※ 참고사항 : 이 옵션을 'True'로 설정했을 때 스크립트 에러 발생 시 에러 메시지가 표시되지 않습니다. | |

※ 참고사항 : 바인딩 시 연결되는 서버 위치

쿼리 디자이너에서 쿼리문을 실행할 때 연결되는 서버는 쿼리 디자이너 환경 설정의 "서버 연결 위치" 옵션 값과 오즈 데이터 트리의 "서버 연결 위치" 속성 값 및 리파지토리 서버 연결 여부에 따라 각각 다르게 선택됩니다.

| 환경 설정 옵션 값 | 오즈 데이터 트리 속성 값 | 설명 |
|---|---|---|
| **OpenType** | Default | ODI 파일을 오픈한 위치가 로컬인지 리파지토리 서버인지에 따라 바인딩되는 서버를 선택합니다. 즉, 로컬에서 오픈한 경우에는 로컬 서버로, 서버 리파지토리에서 오픈한 경우에는 리파지토리 서버로 선택됩니다. |
| | OZServer | 연결되어 있는 리파지토리 서버로 선택됩니다.<br>※ 주의사항 : 리파지토리 서버가 연결되지 않았을 경우에는 자동으로 로컬 서버로 선택됩니다. |
| | Local | 무조건 로컬 서버로 선택됩니다. |
| **Local** | Default | 무조건 로컬 서버로 선택됩니다. |
| | OZServer | 연결되어 있는 리파지토리 서버로 선택됩니다.<br>※ 주의사항 : 리파지토리 서버가 연결되지 않았을 경우에는 자동으로 로컬 서버로 선택됩니다. |

| | | |
|---|---|---|
| | Local | 무조건 로컬 서버로 선택됩니다. |
| **OZServer** | Default | 연결되어 있는 리파지토리 서버로 선택됩니다.<br>※ 주의사항 : 리파지토리 서버가 연결되지 않았을 경우에는 자동으로 로컬 서버로 선택됩니다. |
| | OZServer | 연결되어 있는 리파지토리 서버로 선택됩니다.<br>※ 주의사항 : 리파지토리 서버가 연결되지 않았을 경우에는 자동으로 로컬 서버로 선택됩니다. |
| | Local | 무조건 로컬 서버로 선택됩니다. |
| **ForceLocal** | 무조건 로컬 서버로 선택됩니다. | |
| **ForceOZServer** | 연결되어 있는 리파지토리 서버로 선택됩니다.<br>※ 주의사항 : 리파지토리 서버가 연결되지 않았을 경우에는 자동으로 로컬 서버로 선택됩니다. | |
| **ForceOpenType** | ODI 파일을 오픈한 위치가 로컬인지 리파지토리 서버인지에 따라 바인딩되는 서버를 선택합니다. 즉, 로컬에서 오픈한 경우에는 로컬 서버로, 서버 리파지토리에서 오픈한 경우에는 리파지토리 서버로 선택됩니다. | |

## 데이터베이스

JDBC 또는 ODBC를 통한 데이터베이스 연결 및 데이터 소스 정보를 관리합니다. [데이터베이스] 항목은 다수의 데이터베이스 연결 정보를 설정할 수 있으며, [데이터베이스 스토어] - [데이터베이스 데이터 셋] - [필드]의 계층 구조를 가지고 있습니다.

■ **데이터베이스 스토어 (DB_이름)**

데이터베이스 스토어 정보를 관리합니다. 하위에 다수의 쿼리 데이터 셋을 가질 수 있습니다.

| 속성명 | 설명 |
|---|---|
| **이름** | 데이터베이스 스토어 이름입니다. |
| **사용자 이름** | 데이터베이스 사용자 이름입니다. |
| **JDBC 드라이버** | 데이터베이스 스토어 연결을 위한 ODBC/JDBC 드라이버 이름을 설정합니다. |
| **앨리어스 정보파일 사용** | 데이터베이스 스토어 연결 정보를 별도의 연결 정보 파일에서 가져올 것인지 여부를 설정합니다. |

| | |
|---|---|
| **앨리어스 이름** | 데이터베이스 연결 정보 파일에서 원하는 앨리어스 이름을 설정합니다.<br>데이터베이스 앨리어스는 데이터베이스 연결 정보 파일에서 설정한 여러 개의 데이터베이스 연결을 구분해 주는 이름입니다. 오즈 서버에서 실행하기 위해서는 해당 앨리어스 이름이 서버의 db.properties 파일안에 명시되어 있어야 합니다. |
| **앨리어스 정보파일 이름** | 데이터베이스 연결 정보 파일 이름입니다. |
| **자동 커밋** | 자동 커밋(Commit)할 것인지 여부를 설정합니다. |
| **벤더** | 데이터베이스 벤더 종류를 표시합니다. |
| **데이터 액션 클래스** | 데이터 액션 시 사용할 클래스명을 표시합니다.<br>※ 참고사항 : 이 기능의 자세한 사용법은 오즈 API 활용 매뉴얼의 "Ⅱ장. RDB Store DataAction"을 참조하시기 바랍니다. |
| **데이터 액션 초기화 패러미터** | 데이터 액션 초기화 패러미터를 표시합니다.<br>※ 참고사항 : 이 기능의 자세한 사용법은 오즈 API 활용 매뉴얼의 "Ⅱ장. RDB Store DataAction"을 참조하시기 바랍니다. |
| **행 변경문 허용** | 데이터베이스 스토어의 데이터 액션 수행 시 데이터 셋에 설정한 행 삽입/삭제/변경 쿼리문을 동시에 수행할지 여부를 설정합니다. |

※ 주의사항 : 자동 커밋 사용은 기본적으로 일반적인 JDBC에서 지원하고 있는 기능입니다. 그러나 사용자 환경에 따라 사용에 제약이 있을 수 있습니다. 예를 들어 Oracle에서 dblink를 이용하여 DB2에 연결하여 사용할 경우 제약 사항이 있으므로 아래와 같이 설정하여 사용하면 됩니다. 자세한 사항은 DB 벤더별 JDBC 문서에서 아래의 항목을 참조하시기 바랍니다.

| JDBC | OZ |
|---|---|
| com.setAutoCommit(false) | 자동 커밋=false |
| com.preparedStatement(sqlstr) | 컴파일된 질의문 사용=true |

■ **데이터베이스 데이터 셋 (SQL_이름)**

대상 데이터베이스에서 특정 질의문으로 가져온 데이터 셋 정보를 관리합니다. 하위에 다수의 필드 정보를 가집니다.

| 속성명 | 설명 |
|---|---|
| **이름** | 쿼리 데이터 셋 이름입니다. |
| **질의문** | 필요한 쿼리문을 설정합니다. |

| | |
|---|---|
| **행 삽입 쿼리** | 오즈 애플리케이션 디자이너에서 DataAction 중 행 삽입을 실행하고자 할 때 행 삽입 쿼리문을 작성합니다.<br>※ 참고사항 : 이 기능은 오즈 애플리케이션 디자이너에서만 사용됩니다. 자세한 작성법은 본 매뉴얼의 "Ⅴ장. 쿼리 데이터 셋 디자인의 행 삽입/삭제/변경 쿼리 디자인하기"를 참조하시기 바랍니다. |
| **행 삭제 쿼리** | 오즈 애플리케이션 디자이너에서 DataAction 중 행 삭제를 실행하고자 할 때 행 삭제 쿼리문을 작성합니다.<br>※ 참고사항 : 이 기능은 오즈 애플리케이션 디자이너에서만 사용됩니다. 자세한 작성법은 본 매뉴얼의 "Ⅴ장. 쿼리 데이터 셋 디자인의 행 삽입/삭제/변경 쿼리 디자인하기"를 참조하시기 바랍니다. |
| **행 변경 쿼리** | 오즈 애플리케이션 디자이너에서 DataAction 중 행 변경을 실행하고자 할 때 행 변경 쿼리문을 작성합니다.<br>※ 참고사항 : 이 기능은 오즈 애플리케이션 디자이너에서만 사용됩니다. 자세한 작성법은 본 매뉴얼의 "Ⅴ장. 쿼리 데이터 셋 디자인의 행 삽입/삭제/변경 쿼리 디자인하기"를 참조하시기 바랍니다. |
| **마스터셋** | 해당 데이터 셋의 마스터 데이터 셋을 설정합니다. |
| **최대행 수** | 바인딩시 해당 데이터 셋의 최대 행(Row)수를 설정합니다.<br>※ 주의사항 : 0 ~ 2147483647 이외의 값은 최대행 수로 설정할 수 없습니다. |
| **컴파일된 질의문 사용** | 컴파일된 질의문의 사용 여부를 설정합니다. 'True'로 설정하면 이미 만들어진 질의문이 실행되므로, 같은 질의문이 여러 번 반복 실행될 경우 이 속성 값을 'True'로 설정하면 서버의 성능을 높여줍니다. 컴파일된 질의문은 각 데이터베이스에서 제공하는 컴파일된 질의문을 사용하여 쿼리를 수행합니다. (JDBC에서는 쿼리 실행 시 preparedstatement를 사용합니다.)<br>※ 참고사항 : 이 옵션을 'True'로 설정했을 때 질의문 작성법은 본 매뉴얼의 "Ⅴ장. 컴파일된 질의문 디자인하기"를 참조하시기 바랍니다. |
| **스크립트 사용** | 스크립트를 이용한 쿼리문 작성이 가능하도록 할지 여부를 설정합니다.<br>※ 참고사항 : 이 옵션을 'True'로 설정했을 때 질의문 작성법은 본 매뉴얼의 "Ⅴ장. 컴파일된 질의문 디자인하기"를 참조하시기 바랍니다. |
| **숨김** | 서버에서 해당 데이터 셋을 사용자에게 전송할지 여부를 설정합니다. 값이 'True'인 경우 전송하지 않습니다. 그룹 데이터 스토어를 사용할 경우 원본셋을 숨김으로 설정하면 원본셋은 서버에서 전송하지 않습니다. |

| | |
|---|---|
| **컴파일된 변경문** | 컴파일된 행 삽입/삭제/변경 쿼리문의 사용 여부를 설정합니다. 'True'로 설정하면 서버에 이미 만들어진 행 삽입/삭제/변경 쿼리문이 실행되므로, 같은 쿼리문이 여러 번 반복 실행될 경우 성능을 높여줍니다.<br>※ 참고사항 : 이 옵션을 'True'로 설정했을 때 행 삽입/삭제/변경 쿼리문 작성법은 본 매뉴얼의 "Ⅴ장. 쿼리 데이터 셋 디자인의 행 삽입/삭제/변경 쿼리 디자인하기"를 참조하시기 바랍니다. |
| **Concurrent 전송 크기** | 클라이언트에서 데이터 요청 시 한꺼번에 가져올 결과 데이터 셋의 크기를 Byte 단위로 설정합니다. 설정된 값만큼의 버퍼 단위로 데이터를 내보냅니다.<br>※ 참고사항 : 속성 값을 '0'으로 설정하면 서버에 설정된 값이 적용됩니다.<br>※ 주의사항 : 이 속성 값은 쿼리 디자이너에서 쿼리문을 실행할 때에는 적용되지 않으며, 클라이언트 즉, 애플리케이션 뷰어 또는 리포트 뷰어에서 데이터를 요청할 경우에만 적용되는 속성입니다. 이 속성을 이용하려면 뷰어의 패치 단위 설정 패러미터를 데이터 셋 단위로 변경하시기 바랍니다. |
| **Concurrent 첫 전송 Row** | 클라이언트에서 데이터 요청 시 우선적으로 먼저 전송할 데이터의 행수를 설정합니다. 대용량 데이터일 경우 첫 데이터를 빠르게 받아 응답속도를 향상시키고자 할 때 유용하게 사용될 수 있습니다.<br>※ 참고사항 : 속성 값을 '0'으로 설정하면 서버에 설정된 값이 적용됩니다.<br>※ 주의사항 : 이 속성 값은 쿼리 디자이너에서 쿼리문을 실행할 때에는 적용되지 않으며, 클라이언트 즉, 애플리케이션 뷰어 또는 리포트 뷰어에서 데이터를 요청할 경우에만 적용되는 속성입니다. 이 속성을 이용하려면 뷰어의 패치 단위 설정 패러미터를 데이터 셋 단위로 변경하시기 바랍니다. |
| **바인딩 전 데이터 보내기** | 데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부를 설정합니다. 'False'로 설정하면 바인딩 전 데이터 보내기 기능을 사용하지 않습니다.<br>※ 주의사항 : 이 기능은 패치 타입이 concurrent이고 패치 단위가 데이터 모듈인 경우에만 적용됩니다. |
| **바인딩 후 데이터 보내기** | 데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부를 설정합니다. 'False'로 설정하면 바인딩 후 데이터 보내기 기능을 사용하지 않습니다.<br>※ 주의사항 : 이 기능은 패치 타입이 concurrent이고 패치 단위가 데이터 모듈인 경우에만 적용됩니다. |
| **설명** | 데이터 셋에 대한 설명을 입력합니다. |

■ **다이나믹 데이터 셋 (DYNAMIC_)**

| 속성명 | 설명 |
|---|---|

| | |
|---|---|
| **이름** | 쿼리 데이터 셋 이름입니다. |
| **마스터셋** | 해당 데이터 셋의 마스터 데이터 셋을 설정합니다. |
| **최대행 수** | 바인딩 시 해당 데이터 셋의 최대 행(Row)수를 설정합니다.<br>※ 주의사항 : 0 ~ 2147483647 이외의 값은 최대행 수로 설정할 수 없습니다. |
| **시작 행 번호** | 다이나믹 데이터 셋 생성 시 ROW INDEX에 처음 시작되는 번호입니다. 기본 값은 '0'이며 각 ROW INDEX는 0부터 차례로 번호가 부여됩니다. 사용자가 0 이상의 값을 설정할 수 있습니다.<br>※ 주의사항 : 0 ~ 2147483647 이외의 값은 시작 행 번호로 설정할 수 없습니다. |
| **숨김** | 서버에서 해당 데이터 셋을 사용자에게 전송할지 여부를 설정합니다. 값이 'True'인 경우 전송하지 않습니다. 그룹 데이터 스토어를 사용할 경우 원본셋을 숨김으로 설정하면 원본셋은 서버에서 전송하지 않습니다. |
| **Concurrent 전송 크기** | 클라이언트에서 데이터 요청 시 한꺼번에 가져올 결과 데이터 셋의 크기를 Byte 단위로 설정합니다. 설정된 값만큼의 버퍼 단위로 데이터를 내보냅니다.<br>※ 참고사항 : 속성 값을 '0'으로 설정하면 서버에 설정된 값이 적용됩니다.<br>※ 주의사항 : 이 속성 값은 쿼리 디자이너에서 쿼리문을 실행할 때에는 적용되지 않으며, 클라이언트 즉, 애플리케이션 뷰어 또는 리포트 뷰어에서 데이터를 요청할 경우에만 적용되는 속성입니다. 이 속성을 이용하려면 뷰어의 패치 단위 설정 패러미터를 데이터 셋 단위로 변경하시기 바랍니다. |
| **Concurrent 첫 전송 Row** | 클라이언트에서 데이터 요청 시 우선적으로 먼저 전송할 데이터의 행수를 설정합니다. 대용량 데이터일 경우 첫 데이터를 빠르게 받아 응답속도를 향상시키고자 할 때 유용하게 사용될 수 있습니다.<br>※ 참고사항 : 속성 값을 '0'으로 설정하면 서버에 설정된 값이 적용됩니다.<br>※ 주의사항 : 이 속성 값은 쿼리 디자이너에서 쿼리문을 실행할 때에는 적용되지 않으며, 클라이언트 즉, 애플리케이션 뷰어 또는 리포트 뷰어에서 데이터를 요청할 경우에만 적용되는 속성입니다. 이 속성을 이용하려면 뷰어의 패치 단위 설정 패러미터를 데이터 셋 단위로 변경하시기 바랍니다. |

| | |
|---|---|
| **컴파일된 잘의문 사용 (원본 셋)** | 컴파일된 질의문의 사용 여부를 설정합니다. 'True'로 설정하면 이미 만들어진 질의문이 실행되므로, 같은 질의문이 여러 번 반복 실행될 경우 이 속성 값을 'True'로 설정하면 서버의 성능을 높여줍니다. 컴파일된 질의문은 각 데이터베이스에서 제공하는 컴파일된 질의문을 사용하여 쿼리를 수행합니다. (JDBC에서는 쿼리 실행 시 preparedstatement를 사용합니다.)<br>※ 참고사항 : 이 옵션을 'True'로 설정했을 때 질의문 작성법은 본 매뉴얼의 "Ⅴ장. 컴파일된 질의문 디자인하기"를 참조하시기 바랍니다. |
| **스크립트 사용 (원본 셋)** | 스크립트를 이용한 쿼리문 작성이 가능하도록 할지 여부를 설정합니다.<br>※ 참고사항 : 이 옵션을 'True'로 설정했을 때 질의문 작성법은 본 매뉴얼의 "Ⅴ장. 컴파일된 질의문 디자인하기"를 참조하시기 바랍니다. |
| **바인딩 전 데이터 보내기** | 데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부를 설정합니다. 'False'로 설정하면 바인딩 전 데이터 보내기 기능을 사용하지 않습니다.<br>※ 주의사항 : 이 기능은 패치 타입이 concurrent이고 패치 단위가 데이터 모듈인 경우에만 적용됩니다. |
| **바인딩 후 데이터 보내기** | 데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부를 설정합니다. 'False'로 설정하면 바인딩 후 데이터 보내기 기능을 사용하지 않습니다.<br>※ 주의사항 : 이 기능은 패치 타입이 concurrent이고 패치 단위가 데이터 모듈인 경우에만 적용됩니다. |
| **설명** | 데이터 셋에 대한 설명을 입력합니다. |

■ **멀티 데이터 셋 (SQL_이름)**

| 속성명 | 설명 |
|---|---|
| **이름** | 쿼리 데이터 셋 이름입니다. |
| **질의문** | 필요한 쿼리문을 설정합니다. |
| **마스터셋** | 해당 데이터 셋의 마스터 데이터 셋을 설정합니다. |
| **컴파일된 질의문 사용** | 컴파일된 질의문의 사용 여부를 설정합니다. 'True'로 설정하면 이미 만들어진 질의문이 실행되므로, 같은 질의문이 여러 번 반복 실행될 경우 이 속성 값을 'True'로 설정하면 서버의 성능을 높여줍니다. 컴파일된 질의문은 각 데이터베이스에서 제공하는 컴파일된 질의문을 사용하여 쿼리를 수행합니다. (JDBC에서는 쿼리 실행 시 preparedstatement를 사용합니다.)<br>※ 참고사항 : 이 옵션을 'True'로 설정했을 때 질의문 작성법은 본 매뉴얼의 "Ⅴ장. 컴파일된 질의문 디자인하기"를 참조하시기 바랍니다. |

| | |
|---|---|
| **스크립트 사용** | 스크립트를 이용한 쿼리문 작성이 가능하도록 할지 여부를 설정합니다.<br>※ 참고사항 : 이 옵션을 'True'로 설정했을 때 질의문 작성법은 본 매뉴얼의 "Ⅴ장. 컴파일된 질의문 디자인하기"를 참조하시기 바랍니다. |

■ **프로시저 데이터 셋 (MULTISET)**

| 속성명 | 설명 |
|---|---|
| **이름** | 쿼리 데이터 셋 이름입니다. |
| **프로시저 이름** | 프로시저 이름입니다. |
| **프로시저 타입** | 프로시저 타입입니다. (프로시저 또는 함수로 설정 가능) |
| **타겟 패러미터** | 실행된 프로시저의 결과 값을 가져올 패러미터 이름입니다.<br>패러미터의 데이터 타입이 Cursor일 경우에만 표시되며, 그 외 타입의 필드를 참조할 경우에는 빈 값이 표시됩니다. |
| **마스터셋** | 해당 데이터 셋의 마스터 데이터 셋을 설정합니다. |

■ **데이터 셋 필드**

쿼리해온 데이터 셋의 필드 정보를 관리합니다.

| 속성명 | 설명 |
|---|---|
| **이름** | 쿼리 결과 데이터 필드명입니다. |
| **타입** | 쿼리 결과 데이터 필드 타입입니다. |
| **설명** | 데이터 필드의 설명을 입력합니다. |

## 사용자 데이터(UDS)

사용자 데이터(UDS:사용자 데이터 스토어)는 데이터베이스가 아닌 EJB, 파일, 서블릿 등의 사용자 애플리케이션 데이터를 이용하여 보고서를 제작해야 하는 경우 사용하는 Non-DB 데이터 연동 방식입니다. UDS의 작성에 대한 자세한 사항은 "API 활용 매뉴얼의 Ⅲ. 사용자 데이터 스토어"를 참조하시기 바랍니다.

■ **사용자 데이터 스토어 (CLASS_이름)**

| 속성명 | 설명 |
|---|---|
| **이름** | UDS 연결에 대한 이름입니다. |
| **클래스 이름** | UDS 클래스 파일명입니다. |

| 필드 정의 사용 | 오즈 리포트 디자이너에 필드와 타입을 삽입할 수 있도록 설정합니다. |
|---|---|

■ 사용자 데이터 셋 (SET_이름)

| 속성명 | 설명 |
|---|---|
| 이름 | UDS 데이터 셋 이름입니다. |
| 실행문 | UDS 클래스를 통해 데이터를 가져오기 위한 실행문으로 UDS 클래스에 아규먼트로 전달됩니다. |
| 행 삽입 쿼리 | 오즈 애플리케이션 디자이너에서 DataAction 중 행 삽입을 실행하고자 할 때 행 삽입 쿼리문을 작성합니다.<br>※ 참고사항 : 이 기능은 오즈 애플리케이션 디자이너에서만 사용됩니다. 자세한 작성법은 본 매뉴얼의 "Ⅴ장. 쿼리 데이터 셋 디자인의 행 삽입/삭제/변경 쿼리 디자인하기"를 참조하시기 바랍니다. |
| 행 삭제 쿼리 | 오즈 애플리케이션 디자이너에서 DataAction 중 행 삭제를 실행하고자 할 때 행 삭제 쿼리문을 작성합니다.<br>※ 참고사항 : 이 기능은 오즈 애플리케이션 디자이너에서만 사용됩니다. 자세한 작성법은 본 매뉴얼의 "Ⅴ장. 쿼리 데이터 셋 디자인의 행 삽입/삭제/변경 쿼리 디자인하기"를 참조하시기 바랍니다. |
| 행 변경 쿼리 | 오즈 애플리케이션 디자이너에서 DataAction 중 행 변경을 실행하고자 할 때 행 변경 쿼리문을 작성합니다.<br>※ 참고사항 : 이 기능은 오즈 애플리케이션 디자이너에서만 사용됩니다. 자세한 작성법은 본 매뉴얼의 "Ⅴ장. 쿼리 데이터 셋 디자인의 행 삽입/삭제/변경 쿼리 디자인하기"를 참조하시기 바랍니다. |
| 마스터셋 | 디테일 데이터 셋으로 사용할 경우 마스터 데이터 셋을 설정합니다. |
| 최대행 수 | 바인딩시 해당 데이터 셋의 최대 행(Row)수를 설정합니다. |
| 숨김 | 서버에서 해당 데이터 셋을 사용자에게 전송할지 여부를 설정합니다. 값이 'True'인 경우 전송하지 않습니다. |
| 스크립트 사용 | 스크립트를 이용한 쿼리문 작성이 가능하도록 할지 여부를 설정합니다.<br>※ 참고사항 : 이 옵션을 'True'로 설정했을 때 질의문 작성법은 본 매뉴얼의 "Ⅴ장. 컴파일된 질의문 디자인하기"를 참조하시기 바랍니다. |

| | |
|---|---|
| **Concurrent 전송 크기** | 클라이언트에서 데이터 요청 시 한꺼번에 가져올 결과 데이터 셋의 크기를 Byte 단위로 설정합니다. 설정된 값만큼의 버퍼 단위로 데이터를 내보냅니다.<br>※ 참고사항 : 속성 값을 '0'으로 설정하면 서버에 설정된 값이 적용됩니다.<br>※ 주의사항 : 이 속성 값은 쿼리 디자이너에서 쿼리문을 실행할 때에는 적용되지 않으며, 클라이언트 즉, 애플리케이션 뷰어 또는 리포트 뷰어에서 데이터를 요청할 경우에만 적용되는 속성입니다. |
| **Concurrent 첫 전송 Row** | 클라이언트에서 데이터 요청 시 우선적으로 먼저 전송할 데이터의 행수를 설정합니다. 대용량 데이터일 경우 첫 데이터를 빠르게 받아 응답 속도를 향상시키고자 할 때 유용하게 사용될 수 있습니다.<br>※ 참고사항 : 속성 값을 '0'으로 설정하면 서버에 설정된 값이 적용됩니다.<br>※ 주의사항 : 이 속성 값은 쿼리 디자이너에서 쿼리문을 실행할 때에는 적용되지 않으며, 클라이언트 즉, 애플리케이션 뷰어 또는 리포트 뷰어에서 데이터를 요청할 경우에만 적용되는 속성으로, 클라이언트에서 패치 단위를 데이터 셋 단위로 요청할 경우에만 적용됩니다. |
| **바인딩 전 데이터 보내기** | 데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부를 설정합니다. 'False'로 설정하면 바인딩 전 데이터 보내기 기능을 사용하지 않습니다.<br>※ 주의사항 : 이 기능은 패치 타입이 concurrent이고 패치 단위가 데이터 모듈인 경우에만 적용됩니다. |
| **바인딩 후 데이터 보내기** | 데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부를 설정합니다. 'False'로 설정하면 바인딩 후 데이터 보내기 기능을 사용하지 않습니다.<br>※ 주의사항 : 이 기능은 패치 타입이 concurrent이고 패치 단위가 데이터 모듈인 경우에만 적용됩니다. |
| **설명** | 데이터 셋의 설명을 입력합니다. |

■ **데이터 셋 필드**

| 속성명 | 설명 |
|---|---|
| **이름** | UDS 데이터 필드명입니다. |
| **타입** | UDS 데이터 필드 타입입니다. |
| **설명** | 데이터 필드의 설명을 입력합니다. |

## 그룹 데이터

데이터베이스, 사용자 데이터 등에서 생성된 데이터 소스로부터 여러 개의 새로운 그룹 데이터 소스를 생성하여 관리합니다. 하나의 데이터 소스에서 다수의 그룹 데이터를 생성하여 마스터-디테일 관계를 설정할 수 있습니다.

■ **그룹 데이터 스토어 (GROUP_이름)**

| 속성명 | 설명 |
|---|---|
| 이름 | 그룹 데이터 스토어 이름입니다. |
| 원본셋 | 그룹 데이터 스토어에서 사용할 데이터 메타 혹은 그룹 데이터 스토어의 원본 데이터 셋 이름입니다. |

■ **그룹 데이터 셋 (Shadow_이름)**

| 속성명 | 설명 |
|---|---|
| 이름 | 그룹 데이터 셋 이름입니다. |
| 실행문 | 그룹 데이터를 생성하기 위한 조건문입니다. |
| 마스터셋 | 디테일 데이터 셋으로 사용할 경우 마스터 데이터 셋을 설정합니다. |
| Consume Row | Comsum Row는 그룹 데이터 스토어에서 원본셋의 데이터를 여러 개의 데이터 셋으로 분할할 때 사용합니다. 한 행의 데이터를 모두 분할하고 다음행을 가져오는 시점에 'True'를 설정하여야 그룹 데이터 스토어의 바인딩이 정상적으로 작동하게 됩니다. 일반적으로 하나의 그룹 데이터 스토어 내의 데이터 셋들 중 가장 하위의 데이터 셋만 'True'로 설정합니다. |
| 최대행 수 | 바인딩시 해당 데이터 셋의 최대 행(Row)수를 설정합니다. |
| 숨김 | 서버에서 해당 데이터 셋을 사용자에게 전송할지 여부를 설정합니다. 값이 'True'인 경우 전송하지 않습니다. |
| Concurrent 전송 크기 | 클라이언트에서 데이터 요청 시 한꺼번에 가져올 결과 데이터 셋의 크기를 Byte 단위로 설정합니다. 설정된 값만큼의 버퍼 단위로 데이터를 내보냅니다.<br>※ 참고사항 : 속성 값을 '0'으로 설정하면 서버에 설정된 값이 적용됩니다.<br>※ 주의사항 : 이 속성 값은 쿼리 디자이너에서 쿼리문을 실행할 때에는 적용되지 않으며, 클라이언트 즉, 애플리케이션 뷰어 또는 리포트 뷰어에서 데이터를 요청할 경우에만 적용되는 속성입니다. |

| | |
|---|---|
| **Concurrent 첫 전송 Row** | 클라이언트에서 데이터 요청 시 우선적으로 먼저 전송할 데이터의 행수를 설정합니다. 대용량 데이터일 경우 첫 데이터를 빠르게 받아 응답 속도를 향상시키고자 할 때 유용하게 사용될 수 있습니다.<br>※ 참고사항 : 속성 값을 '0'으로 설정하면 서버에 설정된 값이 적용됩니다.<br>※ 주의사항 : 이 속성 값은 쿼리 디자이너에서 쿼리문을 실행할 때에는 적용되지 않으며, 클라이언트 즉, 애플리케이션 뷰어 또는 리포트 뷰어에서 데이터를 요청할 경우에만 적용되는 속성으로, 클라이언트에서 패치 단위를 데이터 셋 단위로 요청할 경우에만 적용됩니다. |
| **바인딩 전 데이터 보내기** | 데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부를 설정합니다. 'False'로 설정하면 바인딩 전 데이터 보내기 기능을 사용하지 않습니다.<br>※ 주의사항 : 이 기능은 패치 타입이 concurrent이고 패치 단위가 데이터 모듈인 경우에만 적용됩니다. |
| **바인딩 후 데이터 보내기** | 데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부를 설정합니다. 'False'로 설정하면 바인딩 후 데이터 보내기 기능을 사용하지 않습니다.<br>※ 주의사항 : 이 기능은 패치 타입이 concurrent이고 패치 단위가 데이터 모듈인 경우에만 적용됩니다. |
| **설명** | 데이터 셋의 설명을 입력합니다. |

■ **데이터 셋 필드**

| 속성명 | 설명 |
|---|---|
| **이름** | 그룹 데이터 셋의 필드명입니다. |
| **타입** | 그룹 데이터 셋의 필드 타입입니다. |
| **설명** | 데이터 필드의 설명을 입력합니다. |

## 파일 스토어

사용자 애플리케이션에서 생성한 파일이나 스트림 형식의 텍스트 데이터를 데이터 소스로 사용합니다.

■ **파일 스토어 XML 데이터 셋 (FileXML_이름)**

| 속성명 | 설명 |
|---|---|
| **이름** | 파일 스토어 데이터 셋 이름입니다. |

| | |
|---|---|
| **마스터셋** | 디테일 데이터 셋으로 사용할 경우 마스터 데이터 셋을 설정합니다. |
| **최대행 수** | 바인딩시 해당 데이터 셋의 최대 행(Row)수를 설정합니다. |
| **데이터 셋 노드 이름** | XML 파일의 데이터 셋 노드 이름을 설정합니다. |
| **레코드셋 노드 이름** | XML 파일의 레코드셋 노드 이름을 설정합니다. |
| **숨김** | 서버에서 해당 데이터 셋을 사용자에게 전송할지 여부를 설정합니다. 값이 'True'인 경우 전송하지 않습니다. |
| **Concurrent 전송 크기** | 클라이언트에서 데이터 요청 시 한꺼번에 가져올 결과 데이터 셋의 크기를 Byte 단위로 설정합니다. 설정된 값만큼의 버퍼 단위로 데이터를 내보냅니다.<br>※ 참고사항 : 속성 값을 '0'으로 설정하면 서버에 설정된 값이 적용됩니다.<br>※ 주의사항 : 이 속성 값은 쿼리 디자이너에서 쿼리문을 실행할 때에는 적용되지 않으며, 클라이언트 즉, 애플리케이션 뷰어 또는 리포트 뷰어에서 데이터를 요청할 경우에만 적용되는 속성입니다. |
| **Concurrent 첫 전송 Row** | 클라이언트에서 데이터 요청 시 우선적으로 먼저 전송할 데이터의 행수를 설정합니다. 대용량 데이터일 경우 첫 데이터를 빠르게 받아 응답 속도를 향상시키고자 할 때 유용하게 사용될 수 있습니다.<br>※ 참고사항 : 속성 값을 '0'으로 설정하면 서버에 설정된 값이 적용됩니다.<br>※ 주의사항 : 이 속성 값은 쿼리 디자이너에서 쿼리문을 실행할 때에는 적용되지 않으며, 클라이언트 즉, 애플리케이션 뷰어 또는 리포트 뷰어에서 데이터를 요청할 경우에만 적용되는 속성으로, 클라이언트에서 패치 단위를 데이터 셋 단위로 요청할 경우에만 적용됩니다. |
| **바인딩 전 데이터 보내기** | 데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부를 설정합니다. 'False'로 설정하면 바인딩 전 데이터 보내기 기능을 사용하지 않습니다.<br>※ 주의사항 : 이 기능은 패치 타입이 concurrent이고 패치 단위가 데이터 모듈인 경우에만 적용됩니다. |
| **바인딩 후 데이터 보내기** | 데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부를 설정합니다. 'False'로 설정하면 바인딩 후 데이터 보내기 기능을 사용하지 않습니다.<br>※ 주의사항 : 이 기능은 패치 타입이 concurrent이고 패치 단위가 데이터 모듈인 경우에만 적용됩니다. |
| **설명** | 데이터 셋의 설명을 입력합니다. |

■ **파일 스토어 CSV 데이터 셋 (FileCSV_이름)**

| 속성명 | 설명 |
|---|---|

| | |
|---|---|
| **이름** | 파일 스토어 데이터 셋 이름입니다. |
| **마스터셋** | 디테일 데이터 셋으로 사용할 경우 마스터 데이터 셋을 설정합니다. |
| **최대행 수** | 바인딩시 해당 데이터 셋의 최대 행(Row)수를 설정합니다. |
| **필드 구분자** | 필드의 구분자를 지정하는 문자를 나타냅니다 |
| **라인 구분자** | 라인의 구분자를 지정하는 문자를 나타냅니다. (기본 값 : LINEFEED)<br>※ 참고사항<br>▪ 라인 구분자는 영문 대소문자를 구분합니다.<br>▪ 속성 값이 "LINEFEED"일 경우 라인 구분자는 개행 문자(\n)로 설정됩니다.<br>▪ 기존 버전의 쿼리 디자이너에서 저장된 ODI의 CSV 데이터 셋 라인 구분자는 기본 값인 "LINEFEED"로 설정됩니다. |
| **문자셋** | 해당 파일의 문자셋을 나타냅니다. |
| **널 마크** | 데이터가 널인 경우 표시할 값을 나타냅니다. |
| **숨김** | 서버에서 해당 데이터 셋을 사용자에게 전송할지 여부를 설정합니다. 값이 'True'인 경우 전송하지 않습니다. |
| **Concurrent 전송 크기** | 클라이언트에서 데이터 요청 시 한꺼번에 가져올 결과 데이터 셋의 크기를 Byte 단위로 설정합니다. 설정된 값만큼의 버퍼 단위로 데이터를 내보냅니다.<br>※ 참고사항 : 속성 값을 '0'으로 설정하면 서버에 설정된 값이 적용됩니다.<br>※ 주의사항 : 이 속성 값은 쿼리 디자이너에서 쿼리문을 실행할 때에는 적용되지 않으며, 클라이언트 즉, 애플리케이션 뷰어 또는 리포트 뷰어에서 데이터를 요청할 경우에만 적용되는 속성입니다. |
| **Concurrent 첫 전송 Row** | 클라이언트에서 데이터 요청 시 우선적으로 먼저 전송할 데이터의 행수를 설정합니다. 대용량 데이터일 경우 첫 데이터를 빠르게 받아 응답 속도를 향상시키고자 할 때 유용하게 사용될 수 있습니다.<br>※ 참고사항 : 속성 값을 '0'으로 설정하면 서버에 설정된 값이 적용됩니다.<br>※ 주의사항 : 이 속성 값은 쿼리 디자이너에서 쿼리문을 실행할 때에는 적용되지 않으며, 클라이언트 즉, 애플리케이션 뷰어 또는 리포트 뷰어에서 데이터를 요청할 경우에만 적용되는 속성으로, 클라이언트에서 패치 단위를 데이터 셋 단위로 요청할 경우에만 적용됩니다. |

| | |
|---|---|
| **바인딩 전 데이터 보내기** | 데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부를 설정합니다. 'False'로 설정하면 바인딩 전 데이터 보내기 기능을 사용하지 않습니다.<br>※ 주의사항 : 이 기능은 패치 타입이 concurrent이고 패치 단위가 데이터 모듈인 경우에만 적용됩니다. |
| **바인딩 후 데이터 보내기** | 데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부를 설정합니다. 'False'로 설정하면 바인딩 후 데이터 보내기 기능을 사용하지 않습니다.<br>※ 주의사항 : 이 기능은 패치 타입이 concurrent이고 패치 단위가 데이터 모듈인 경우에만 적용됩니다. |
| **설명** | 데이터 셋의 설명을 입력합니다. |

※ 주의사항 : PGP로 암호화된 CSV 파일의 경우에는 UDS를 사용해야 합니다.

■ **데이터 셋 필드**

| 속성명 | 설명 |
|---|---|
| **이름** | 필드명입니다. |
| **타입** | 필드 타입입니다. |
| **설명** | 데이터 필드의 설명을 입력합니다. |

## HTTP 스토어

사용자 애플리케이션에서 생성한 파일이나 스트림 형식의 텍스트 데이터를 데이터 소스로 사용합니다.

■ **HTTP 스토어 XML 데이터 셋 (HTTPXML_이름)**

| 속성명 | 설명 |
|---|---|
| **이름** | Http 스토어 데이터 셋 이름입니다. |
| **마스터셋** | 디테일 데이터 셋으로 사용할 경우 마스터 데이터 셋을 설정합니다. |
| **최대행 수** | 바인딩시 해당 데이터 셋의 최대 행(Row)수를 설정합니다. |
| **데이터 셋 노드 이름** | XML 파일의 데이터 셋 노드 이름을 설정합니다. |
| **레코드셋 노드 이름** | XML 파일의 레코드셋 노드 이름을 설정합니다. |
| **숨김** | 서버에서 해당 데이터 셋을 사용자에게 전송할지 여부를 설정합니다. 값이 'True'인 경우 전송하지 않습니다. |

| | |
|---|---|
| **Concurrent 전송 크기** | 클라이언트에서 데이터 요청 시 한꺼번에 가져올 결과 데이터 셋의 크기를 Byte 단위로 설정합니다. 설정된 값만큼의 버퍼 단위로 데이터를 내보냅니다.<br>※ 참고사항 : 속성 값을 '0'으로 설정하면 서버에 설정된 값이 적용됩니다.<br>※ 주의사항 : 이 속성 값은 쿼리 디자이너에서 쿼리문을 실행할 때에는 적용되지 않으며, 클라이언트 즉, 애플리케이션 뷰어 또는 리포트 뷰어에서 데이터를 요청할 경우에만 적용되는 속성입니다. |
| **Concurrent 첫 전송 Row** | 클라이언트에서 데이터 요청 시 우선적으로 먼저 전송할 데이터의 행수를 설정합니다. 대용량 데이터일 경우 첫 데이터를 빠르게 받아 응답 속도를 향상시키고자 할 때 유용하게 사용될 수 있습니다.<br>※ 참고사항 : 속성 값을 '0'으로 설정하면 서버에 설정된 값이 적용됩니다.<br>※ 주의사항 : 이 속성 값은 쿼리 디자이너에서 쿼리문을 실행할 때에는 적용되지 않으며, 클라이언트 즉, 애플리케이션 뷰어 또는 리포트 뷰어에서 데이터를 요청할 경우에만 적용되는 속성으로, 클라이언트에서 패치 단위를 데이터 셋 단위로 요청할 경우에만 적용됩니다. |
| **바인딩 전 데이터 보내기** | 데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부를 설정합니다. 'False'로 설정하면 바인딩 전 데이터 보내기 기능을 사용하지 않습니다.<br>※ 주의사항 : 이 기능은 패치 타입이 concurrent이고 패치 단위가 데이터 모듈인 경우에만 적용됩니다. |
| **바인딩 후 데이터 보내기** | 데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부를 설정합니다. 'False'로 설정하면 바인딩 후 데이터 보내기 기능을 사용하지 않습니다.<br>※ 주의사항 : 이 기능은 패치 타입이 concurrent이고 패치 단위가 데이터 모듈인 경우에만 적용됩니다. |
| **설명** | 데이터 셋의 설명을 입력합니다. |

■ **HTTP 스토어 CSV 데이터 셋 (HTTPCSV_이름)**

| 속성명 | 설명 |
|---|---|
| **이름** | Http 스토어 데이터 셋 이름입니다. |
| **마스터셋** | 디테일 데이터 셋으로 사용할 경우 마스터 데이터 셋을 설정합니다. |
| **최대행 수** | 바인딩시 해당 데이터 셋의 최대 행(Row)수를 설정합니다. |
| **필드 구분자** | 필드의 구분자를 지정하는 문자를 나타냅니다 |

| | |
|---|---|
| **라인 구분자** | 라인의 구분자를 지정하는 문자를 나타냅니다. (기본 값 : LINEFEED)<br>※ 참고사항<br>▪ 라인 구분자는 영문 대소문자를 구분합니다.<br>▪ 속성 값이 "LINEFEED"일 경우 라인 구분자는 개행 문자(\n)로 설정됩니다.<br>▪ 기존 버전의 쿼리 디자이너에서 저장된 ODI의 CSV 데이터 셋 라인 구분자는 기본 값인 "LINEFEED"로 설정됩니다. |
| **문자셋** | 해당 파일의 문자셋을 나타냅니다. |
| **널 마크** | 데이터가 널인 경우 표시할 값을 나타냅니다. |
| **숨김** | 서버에서 해당 데이터 셋을 사용자에게 전송할지 여부를 설정합니다. 값이 'True'인 경우 전송하지 않습니다. |
| **Concurrent 전송 크기** | 클라이언트에서 데이터 요청 시 한꺼번에 가져올 결과 데이터 셋의 크기를 Byte 단위로 설정합니다. 설정된 값만큼의 버퍼 단위로 데이터를 내보냅니다.<br>※ 참고사항 : 속성 값을 '0'으로 설정하면 서버에 설정된 값이 적용됩니다.<br>※ 주의사항 : 이 속성 값은 쿼리 디자이너에서 쿼리문을 실행할 때에는 적용되지 않으며, 클라이언트 즉, 애플리케이션 뷰어 또는 리포트 뷰어에서 데이터를 요청할 경우에만 적용되는 속성입니다. |
| **Concurrent 첫 전송 Row** | 클라이언트에서 데이터 요청 시 우선적으로 먼저 전송할 데이터의 행수를 설정합니다. 대용량 데이터일 경우 첫 데이터를 빠르게 받아 응답 속도를 향상시키고자 할 때 유용하게 사용될 수 있습니다.<br>※ 참고사항 : 속성 값을 '0'으로 설정하면 서버에 설정된 값이 적용됩니다.<br>※ 주의사항 : 이 속성 값은 쿼리 디자이너에서 쿼리문을 실행할 때에는 적용되지 않으며, 클라이언트 즉, 애플리케이션 뷰어 또는 리포트 뷰어에서 데이터를 요청할 경우에만 적용되는 속성으로, 클라이언트에서 패치 단위를 데이터 셋 단위로 요청할 경우에만 적용됩니다. |
| **바인딩 전 데이터 보내기** | 데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부를 설정합니다. 'False'로 설정하면 바인딩 전 데이터 보내기 기능을 사용하지 않습니다.<br>※ 주의사항 : 이 기능은 패치 타입이 concurrent이고 패치 단위가 데이터 모듈인 경우에만 적용됩니다. |
| **바인딩 후 데이터 보내기** | 데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부를 설정합니다. 'False'로 설정하면 바인딩 후 데이터 보내기 기능을 사용하지 않습니다.<br>※ 주의사항 : 이 기능은 패치 타입이 concurrent이고 패치 단위가 데이터 모듈인 경우에만 적용됩니다. |

| 설명 | 데이터 셋의 설명을 입력합니다. |
|---|---|

■ 데이터 셋 필드

| 속성명 | 설명 |
|---|---|
| 이름 | 필드명입니다. |
| 타입 | 필드 타입입니다. |
| 설명 | 데이터 필드의 설명을 입력합니다. |

## SOAP 스토어

웹 서비스를 통하여 얻어온 데이터를 데이터 소스로 사용합니다.

■ SOAP 스토어(SOAPStore_이름)

| 속성명 | 설명 |
|---|---|
| 이름 | SOAP 스토어 이름입니다. |

■ SOAP 데이터 셋(SOAPSet_이름)

| 속성명 | 설명 |
|---|---|
| 이름 | SOAP 스토어 데이터 셋 이름입니다. |
| 마스터셋 | 데이터 셋이 디테일 데이터 셋일 경우 해당 데이터 셋의 마스터 데이터 셋 이름입니다. |
| 숨김 | 서버에서 해당 데이터 셋을 사용자에게 전송할지 여부를 설정합니다. 값이 'True'인 경우 전송하지 않습니다. |
| Concurrent 전송 크기 | 클라이언트에서 데이터 요청 시 한꺼번에 가져올 결과 데이터 셋의 크기를 Byte 단위로 설정합니다. 설정된 값만큼의 버퍼 단위로 데이터를 내보냅니다.<br>※ 참고사항 : 속성 값을 '0'으로 설정하면 서버에 설정된 값이 적용됩니다.<br>※ 주의사항 : 이 속성 값은 쿼리 디자이너에서 쿼리문을 실행할 때에는 적용되지 않으며, 클라이언트 즉, 애플리케이션 뷰어 또는 리포트 뷰어에서 데이터를 요청할 경우에만 적용되는 속성입니다. |

| | |
|---|---|
| **Concurrent 첫 전송 Row** | 클라이언트에서 데이터 요청 시 우선적으로 먼저 전송할 데이터의 행수를 설정합니다. 대용량 데이터일 경우 첫 데이터를 빠르게 받아 응답 속도를 향상시키고자 할 때 유용하게 사용될 수 있습니다.<br>※ 참고사항 : 속성 값을 '0'으로 설정하면 서버에 설정된 값이 적용됩니다.<br>※ 주의사항 : 이 속성 값은 쿼리 디자이너에서 쿼리문을 실행할 때에는 적용되지 않으며, 클라이언트 즉, 애플리케이션 뷰어 또는 리포트 뷰어에서 데이터를 요청할 경우에만 적용되는 속성으로, 클라이언트에서 패치 단위를 데이터 셋 단위로 요청할 경우에만 적용됩니다. |
| **바인딩 전 데이터 보내기** | 데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부를 설정합니다. 'False'로 설정하면 바인딩 전 데이터 보내기 기능을 사용하지 않습니다.<br>※ 주의사항 : 이 기능은 패치 타입이 concurrent이고 패치 단위가 데이터 모듈인 경우에만 적용됩니다. |
| **바인딩 후 데이터 보내기** | 데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부를 설정합니다. 'False'로 설정하면 바인딩 후 데이터 보내기 기능을 사용하지 않습니다.<br>※ 주의사항 : 이 기능은 패치 타입이 concurrent이고 패치 단위가 데이터 모듈인 경우에만 적용됩니다. |
| **기준 XML 경로** | 기준 XML 경로입니다. |
| **설명** | 데이터 셋의 설명을 입력합니다. |

■ **SOAP 데이터 필드**

| 속성명 | 설명 |
|---|---|
| **이름** | 필드명입니다. |
| **타입** | 필드 타입입니다. |
| **XPath** | 필드의 기준 XPath입니다. |
| **설명** | 데이터 필드의 설명을 입력합니다. |

## ADO DataSet 웹서비스

닷넷 웹 서비스에서 ADO DataSet 객체를 연동하여 데이터 셋을 생성합니다.

■ **ADO DataSet 웹서비스 스토어(ADOStore_이름)**

| 속성명 | 설명 |
|---|---|

| 이름 | ADO DataSet 웹서비스 스토어 이름입니다. |
|---|---|

■ **ADO DataSet 웹서비스 데이터 셋(ADOSet_이름)**

| 속성명 | 설명 |
|---|---|
| 이름 | ADO DataSet 웹서비스 스토어 데이터 셋 이름입니다. |
| 마스터셋 | 데이터 셋이 디테일 데이터 셋일 경우 해당 데이터 셋의 마스터 데이터 셋 이름입니다. |
| EndPoint | EndPoint입니다. |
| SOAPAction | SOAPAction입니다. |
| 최대행 수 | 바인딩시 해당 데이터 셋의 최대 행(Row)수를 설정합니다. |
| 숨김 | 서버에서 해당 데이터 셋을 사용자에게 전송할지 여부를 설정합니다. 값이 'True'인 경우 전송하지 않습니다. |
| Concurrent 전송 크기 | 클라이언트에서 데이터 요청 시 한꺼번에 가져올 결과 데이터 셋의 크기를 Byte 단위로 설정합니다. 설정된 값만큼의 버퍼 단위로 데이터를 내보냅니다.<br>※ 참고사항 : 속성 값을 '0'으로 설정하면 서버에 설정된 값이 적용됩니다.<br>※ 주의사항 : 이 속성 값은 쿼리 디자이너에서 쿼리문을 실행할 때에는 적용되지 않으며, 클라이언트 즉, 애플리케이션 뷰어 또는 리포트 뷰어에서 데이터를 요청할 경우에만 적용되는 속성입니다. |
| Concurrent 첫 전송 Row | 클라이언트에서 데이터 요청 시 우선적으로 먼저 전송할 데이터의 행수를 설정합니다. 대용량 데이터일 경우 첫 데이터를 빠르게 받아 응답 속도를 향상시키고자 할 때 유용하게 사용될 수 있습니다.<br>※ 참고사항 : 속성 값을 '0'으로 설정하면 서버에 설정된 값이 적용됩니다.<br>※ 주의사항 : 이 속성 값은 쿼리 디자이너에서 쿼리문을 실행할 때에는 적용되지 않으며, 클라이언트 즉, 애플리케이션 뷰어 또는 리포트 뷰어에서 데이터를 요청할 경우에만 적용되는 속성으로, 클라이언트에서 패치 단위를 데이터 셋 단위로 요청할 경우에만 적용됩니다. |
| 바인딩 전 데이터 보내기 | 데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부를 설정합니다. 'False'로 설정하면 바인딩 전 데이터 보내기 기능을 사용하지 않습니다.<br>※ 주의사항 : 이 기능은 패치 타입이 concurrent이고 패치 단위가 데이터 모듈인 경우에만 적용됩니다. |

| | |
|---|---|
| **바인딩 후 데이터 보내기** | 데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부를 설정합니다. 'False'로 설정하면 바인딩 후 데이터 보내기 기능을 사용하지 않습니다.<br>※ 주의사항 : 이 기능은 패치 타입이 concurrent이고 패치 단위가 데이터 모듈인 경우에만 적용됩니다. |
| **설명** | 데이터 셋의 설명을 입력합니다. |

■ **ADO DataSet 웹서비스 데이터 필드**

| 속성명 | 설명 |
|---|---|
| **이름** | 필드명입니다. |
| **타입** | 필드 타입입니다. |
| **설명** | 데이터 필드의 설명을 입력합니다. |

## XML

DTD/XSD를 이용하는 XML 형식의 데이터를 데이터 소스로 사용합니다.

■ **XML DTD/XSD 스토어(XMLEXStore_이름)**

| 속성명 | 설명 |
|---|---|
| **이름** | XML 스토어 이름입니다. |

■ **XML DTD/XSD 스토어 데이터 셋 (DTDSet_이름, XSDSet_이름)**

| 속성명 | 설명 |
|---|---|
| **이름** | XML 스토어 데이터 셋 이름입니다. |
| **마스터셋** | 디테일 데이터 셋으로 사용할 경우 마스터 데이터 셋을 설정합니다. |
| **숨김** | 서버에서 해당 데이터 셋을 사용자에게 전송할지 여부를 설정합니다. 값이 'True'인 경우 전송하지 않습니다. |

| | |
|---|---|
| **Concurrent 전송 크기** | 클라이언트에서 데이터 요청 시 한꺼번에 가져올 결과 데이터 셋의 크기를 Byte 단위로 설정합니다. 설정된 값만큼의 버퍼 단위로 데이터를 내보냅니다.<br>※ 참고사항 : 속성 값을 '0'으로 설정하면 서버에 설정된 값이 적용됩니다.<br>※ 주의사항 : 이 속성 값은 쿼리 디자이너에서 쿼리문을 실행할 때에는 적용되지 않으며, 클라이언트 즉, 애플리케이션 뷰어 또는 리포트 뷰어에서 데이터를 요청할 경우에만 적용되는 속성입니다. |
| **Concurrent 첫 전송 Row** | 클라이언트에서 데이터 요청 시 우선적으로 먼저 전송할 데이터의 행수를 설정합니다. 대용량 데이터일 경우 첫 데이터를 빠르게 받아 응답 속도를 향상시키고자 할 때 유용하게 사용될 수 있습니다.<br>※ 참고사항 : 속성 값을 '0'으로 설정하면 서버에 설정된 값이 적용됩니다.<br>※ 주의사항 : 이 속성 값은 쿼리 디자이너에서 쿼리문을 실행할 때에는 적용되지 않으며, 클라이언트 즉, 애플리케이션 뷰어 또는 리포트 뷰어에서 데이터를 요청할 경우에만 적용되는 속성으로, 클라이언트에서 패치 단위를 데이터 셋 단위로 요청할 경우에만 적용됩니다. |
| **바인딩 전 데이터 보내기** | 데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부를 설정합니다. 'False'로 설정하면 바인딩 전 데이터 보내기 기능을 사용하지 않습니다.<br>※ 주의사항 : 이 기능은 패치 타입이 concurrent이고 패치 단위가 데이터 모듈인 경우에만 적용됩니다. |
| **바인딩 후 데이터 보내기** | 데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부를 설정합니다. 'False'로 설정하면 바인딩 후 데이터 보내기 기능을 사용하지 않습니다.<br>※ 주의사항 : 이 기능은 패치 타입이 concurrent이고 패치 단위가 데이터 모듈인 경우에만 적용됩니다. |
| **설명** | 데이터 셋의 설명을 입력합니다. |

■ **데이터 셋 필드**

| 속성명 | 설명 |
|---|---|
| **이름** | 필드명입니다. |
| **필드 고유 이름** | 상위 노드명을 포함한 필드명(Full Name)입니다. |
| **노드 타입** | XML 노드의 타입입니다. |
| **노드 깊이** | XML 노드의 깊이입니다. |
| **타입** | 필드 타입입니다. |
| **설명** | 데이터 필드의 설명을 입력합니다. |

## SAP

SAP R/3에 정의된 RFC(Remote Function Call) 실행 결과를 데이터 소스로 사용합니다.

■ **SAP 데이터 스토어**

| 속성명 | 설명 |
|---|---|
| 클라이언트 이름 | 로그인할 클라이언트 번호를 설정합니다. |
| 언어 | 로그인시 사용할 언어를 설정합니다. |
| 호스트 이름(주소) | SAP 서버의 호스트 이름을 설정합니다. |
| 시스템 번호 | SAP 서버의 시스템 번호를 설정합니다. |

■ **SAP 함수**

| 속성명 | 설명 |
|---|---|
| 이름 | SAP 함수 이름입니다. |

■ **SAP 데이터 셋**

| 속성명 | 설명 |
|---|---|
| 이름 | SAP 스토어 데이터 셋 이름입니다. |
| 데이터 셋 | 디테일 데이터 셋으로 사용할 경우 마스터 데이터 셋을 설정합니다. |
| Concurrent 전송 크기 | 클라이언트에서 데이터 요청 시 한꺼번에 가져올 결과 데이터 셋의 크기를 Byte 단위로 설정합니다. 설정된 값만큼의 버퍼 단위로 데이터를 내보냅니다.<br>※ 참고사항 : 속성 값을 '0'으로 설정하면 서버에 설정된 값이 적용됩니다.<br>※ 주의사항 : 이 속성 값은 쿼리 디자이너에서 쿼리문을 실행할 때에는 적용되지 않으며, 클라이언트 즉, 애플리케이션 뷰어 또는 리포트 뷰어에서 데이터를 요청할 경우에만 적용되는 속성입니다. |

| | |
|---|---|
| **Concurrent 첫 전송 Row** | 클라이언트에서 데이터 요청 시 우선적으로 먼저 전송할 데이터의 행수를 설정합니다. 대용량 데이터일 경우 첫 데이터를 빠르게 받아 응답 속도를 향상시키고자 할 때 유용하게 사용될 수 있습니다.<br>※ 참고사항 : 속성 값을 '0'으로 설정하면 서버에 설정된 값이 적용됩니다.<br>※ 주의사항 : 이 속성 값은 쿼리 디자이너에서 쿼리문을 실행할 때에는 적용되지 않으며, 클라이언트 즉, 애플리케이션 뷰어 또는 리포트 뷰어에서 데이터를 요청할 경우에만 적용되는 속성으로, 클라이언트에서 패치 단위를 데이터 셋 단위로 요청할 경우에만 적용됩니다. |
| **바인딩 전 데이터 보내기** | 데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부를 설정합니다. 'False'로 설정하면 바인딩 전 데이터 보내기 기능을 사용하지 않습니다.<br>※ 주의사항 : 이 기능은 패치 타입이 concurrent이고 패치 단위가 데이터 모듈인 경우에만 적용됩니다. |
| **바인딩 후 데이터 보내기** | 데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부를 설정합니다. 'False'로 설정하면 바인딩 후 데이터 보내기 기능을 사용하지 않습니다.<br>※ 주의사항 : 이 기능은 패치 타입이 concurrent이고 패치 단위가 데이터 모듈인 경우에만 적용됩니다. |
| **설명** | 데이터 셋의 설명을 입력합니다. |

■ **데이터 셋 필드**

| 속성명 | 설명 |
|---|---|
| **이름** | 필드명입니다. |
| **타입** | 필드 타입입니다. |
| **설명** | 데이터 필드의 설명을 입력합니다. |

## CLEAR QUEST 스토어

형상 관리 툴인 IBM Rational Clear Quest 프로그램에서 생성한 DB를 통하여 얻어온 데이터를 데이터 소스로 사용합니다.

■ **CLEAR QUEST 스토어(CQStore_ 이름)**

| 속성명 | 설명 |
|---|---|
| **이름** | CLEAR QUEST 스토어 이름입니다. |

■ **CLEAR QUEST 데이터 셋(CQSet_이름)**

| 속성명 | 설명 |
|---|---|
| 이름 | CLEAR QUEST 스토어 데이터 셋 이름입니다. |
| 타입 | CLEAR QUEST 스토어 데이터 타입입니다. |
| 마스터셋 | 데이터 셋이 디테일 데이터 셋일 경우 해당 데이터 셋의 마스터 데이터 셋 이름입니다. |
| 최대행 수 | 바인딩시 해당 데이터셋의 최대 행(Row)수를 설정합니다.<br>※ 주의사항 : 0 ~ 2147483647 이외의 값은 최대행 수로 설정할 수 없습니다. |
| 스크립트 사용 | 스크립트를 이용한 쿼리문 작성이 가능하도록 할지 여부를 설정합니다.<br>※ 참고사항 : 이 옵션을 'True'로 설정했을 때 질의문 작성법은 본 매뉴얼의 "Ⅴ장. 컴파일된 질의문 디자인하기"를 참조하시기 바랍니다. |
| 숨김 | 서버에서 해당 데이터 셋을 사용자에게 전송할지 여부를 설정합니다. 값이 'True'인 경우 전송하지 않습니다. |
| Concurrent 전송 크기 | 클라이언트에서 데이터 요청 시 한꺼번에 가져올 결과 데이터 셋의 크기를 Byte 단위로 설정합니다. 설정된 값만큼의 버퍼 단위로 데이터를 내보냅니다.<br>※ 참고사항 : 속성 값을 '0'으로 설정하면 서버에 설정된 값이 적용됩니다.<br>※ 주의사항 : 이 속성 값은 쿼리 디자이너에서 쿼리문을 실행할 때에는 적용되지 않으며, 클라이언트 즉, 애플리케이션 뷰어 또는 리포트 뷰어에서 데이터를 요청할 경우에만 적용되는 속성입니다. 이 속성을 이용하려면 뷰어의 패치 단위 설정 패러미터를 데이터 셋 단위로 변경하시기 바랍니다. |
| Concurrent 첫 전송 Row | 클라이언트에서 데이터 요청 시 우선적으로 먼저 전송할 데이터의 행수를 설정합니다. 대용량 데이터일 경우 첫 데이터를 빠르게 받아 응답 속도를 향상시키고자 할 때 유용하게 사용될 수 있습니다.<br>※ 참고사항 : 속성 값을 '0'으로 설정하면 서버에 설정된 값이 적용됩니다.<br>※ 주의사항 : 이 속성 값은 쿼리 디자이너에서 쿼리문을 실행할 때에는 적용되지 않으며, 클라이언트 즉, 애플리케이션 뷰어 또는 리포트 뷰어에서 데이터를 요청할 경우에만 적용되는 속성입니다. 이 속성을 이용하려면 뷰어의 패치 단위 설정 패러미터를 데이터 셋 단위로 변경하시기 바랍니다. |

| | |
|---|---|
| **바인딩 전 데이터 보내기** | 데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부를 설정합니다. 'False'로 설정하면 바인딩 전 데이터 보내기 기능을 사용하지 않습니다.<br>※ 주의사항 : 이 기능은 패치 타입이 concurrent이고 패치 단위가 데이터 모듈인 경우에만 적용됩니다. |
| **바인딩 후 데이터 보내기** | 데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부를 설정합니다. 'False'로 설정하면 바인딩 후 데이터 보내기 기능을 사용하지 않습니다.<br>※ 주의사항 : 이 기능은 패치 타입이 concurrent이고 패치 단위가 데이터 모듈인 경우에만 적용됩니다. |
| **설명** | 데이터 셋의 설명을 입력합니다. |

■ **CLEAR QUEST 데이터 필드**

| 속성명 | 설명 |
|---|---|
| **이름** | 필드명입니다. |
| **타입** | 필드 타입입니다. |
| **설명** | 데이터 필드의 설명을 입력합니다. |

## 사용자 지정 패러미터

사용자가 임의로 정의하여 사용하는 패러미터 변수 목록을 관리합니다.

■ **패러미터 항목 속성**

| 속성명 | 설명 |
|---|---|
| **이름** | 사용자 정의 패러미터 변수 이름입니다. |
| **타입** | 사용자 정의 패러미터 변수 타입입니다. |
| **값** | 패러미터 값의 초기값으로 디자인시 보고서의 테스트를 위해 사용자가 임의로 설정할 수 있습니다. |
| **세션 키** | 추가할 패러미터의 값을 설정할 세션 키를 입력합니다.<br>※ 참고사항 : 추가할 패러미터의 값을 오즈 뷰어 패러미터에서 설정하는 대신 서버 세션으로 설정하고자 할 때 사용합니다. |
| **설명** | 패러미터 변수에 대한 설명을 입력합니다. |

## Include ODI

기존 작성되어 서버 리파지토리 혹은 LocalExplorer에 있는 ODI 파일을 Include한 스토어를 그대로 사용합니다.

■ **Include 필드**

| 속성명 | 설명 |
|---|---|
| **이름** | Include된 필드 이름입니다. |
| **OZURL** | 카테고리와 파일 이름을 URL 형태로 표시합니다.<br>설정된 ODI 파일에 대한 경로인 OZ URL은 서버 리파지토리에 연결되어 있는 경우에는 [Repository Root]의 경로를 의미하며, 서버 리파지토리에 연결되어 있지 않은 경우에는 [LocalExplorer]의 경로를 의미합니다.<br>예) ozp:///카테고리명/파일명 |

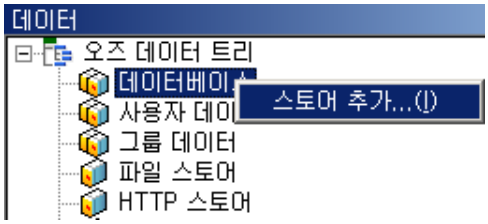
# 데이터베이스 연결

ODBC/JDBC 드라이버를 이용하여 다양한 데이터베이스 소스에 연결하는 방법에 대해 설명합니다.

## 데이터베이스 추가

다음은 데이터베이스 스토어를 추가하는 방법을 순차적으로 설명합니다.

**Step 1** 오즈 데이터 트리의 [데이터베이스]에서 마우스 오른쪽 버튼을 클릭하여 [스토어 추가] 메뉴를 클릭합니다.




**Step 2** '데이터베이스 연결 정보 추가 다이얼로그'가 나타나면 원하는 데이터베이스의 연결 정보를 설정합니다.

**Step 3** [확인] 버튼을 클릭하면 아래와 같이 "데이터베이스" 아래에 해당된 데이터베이스 정보가 나타납니다.

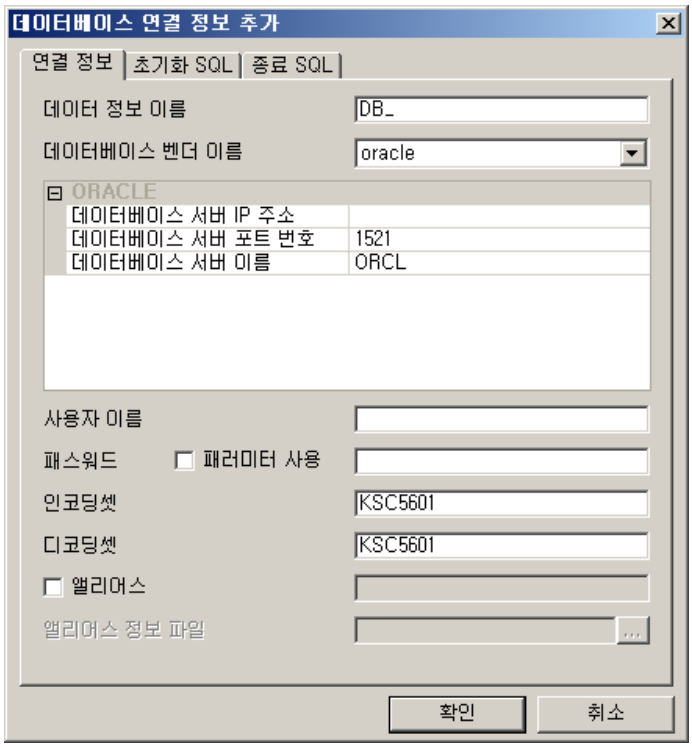



## 데이터베이스 연결 정보 추가 다이얼로그

데이터 트리에서 데이터베이스 연결 정보를 입력받는 다이얼로그로 '연결 정보', '초기화 SQL', '종료 SQL'을 설정할 수 있는 탭으로 구성됩니다.

■ **연결 정보**

JDBC, ODBC, OLE DB 등을 통하여 데이터베이스에 접속하기 위한 기본 정보를 설정하는 다이얼로그로 다음과 같은 항목으로 구성되어 있습니다.




| 항목 | 설명 |
|---|---|
| **데이터 정보 이름** | 해당 데이터베이스 정보를 가리키는 이름입니다. |
| **데이터베이스 벤더 이름** | 사용할 데이터베이스의 벤더를 선택합니다.<br>사용할 데이터베이스의 벤더를 선택한 후 해당 벤더의 IP, Port 등의 정보를 입력합니다. |
| **사용자 이름** | 데이터베이스 사용자 ID를 설정합니다. |
| **패스워드** | 데이터베이스 사용자 패스워드를 설정합니다. |
| **패러미터 사용** | 데이터베이스 사용자 암호를 사용자 지정 패러미터 값으로 설정합니다. |
| **인코딩셋** | 데이터베이스의 문자셋을 설정합니다. |
| **디코딩셋** | 가져올 데이터의 문자셋을 설정합니다. |
| **DB 앨리어스** | 오즈 서버의 db.properties 파일에 정의한 DB 앨리어스 이름을 설정합니다. |

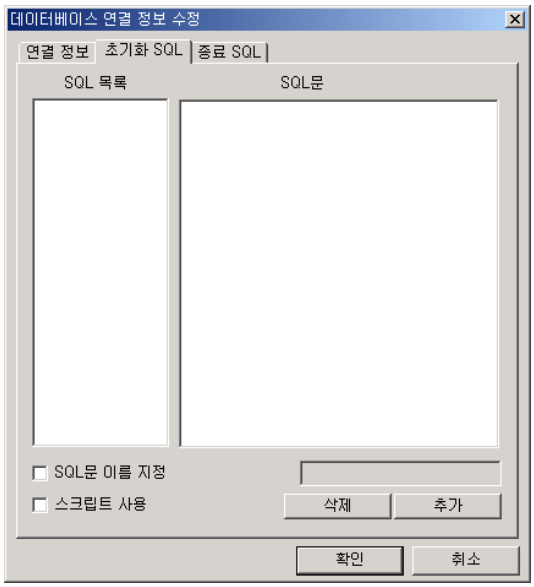
| 앨리어스 정보 파일 | DB 앨리어스를 사용할 경우 사용하며, 디자인 시 사용할 DB 앨리어스 정보가 있는 DB 연결 정보 파일명을 설정합니다. 오즈 서버의 db.properties 파일을 로컬 시스템에 복사하여 사용할 수 있습니다. |
|---|---|

※ 참고사항

- 인코딩/디코딩 셋에 지정할 문자셋명에 대해서는 "Appendix. 문자셋과 유효한 앨리어스"를 참조하시기 바랍니다.
- OLE DB 관련 데이터베이스 벤더는 연결되는 서버가 닷넷일 경우에만 지원됩니다.

■ **초기화 SQL / 종료 SQL**

데이터베이스 연결 후 바로 실행할 SQL문과 데이터베이스 연결 해제 전에 실행할 SQL문을 설정합니다.




| 항목 | 설명 |
|---|---|
| **SQL 목록** | 수행할 SQL문 목록을 보여줍니다. |
| **SQL 문** | SQL 목록에서 선택한 SQL문의 내용을 보여줍니다. SQL문은 JDBC에서 실행 가능한 표준 SQL문으로 설정하여야 하며 결과 셋에 대한 처리를 하지 않으므로 결과 셋이 지원되지 않는 SQL을 사용합니다.<br>예) INSERT, UPDATE, DELETE 등 |
| **SQL문 이름 지정** | SQL문의 이름을 지정할 지의 여부와 SQL문의 이름을 직접 입력합니다. |

| 스크립트 사용 | 스크립트를 사용하여 SQL을 작성할지 여부를 설정합니다. |
|---|---|
| 삭제 | 선택한 SQL문을 삭제합니다. |
| 추가 | 새로운 SQL문을 등록합니다. |

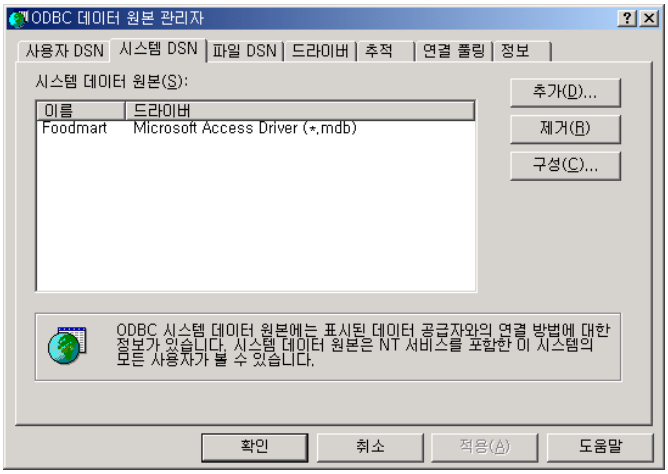
※ 참고사항 : SQL문 목록은 설정한 순서대로 실행됩니다.

## 데이터베이스별 연결 설정

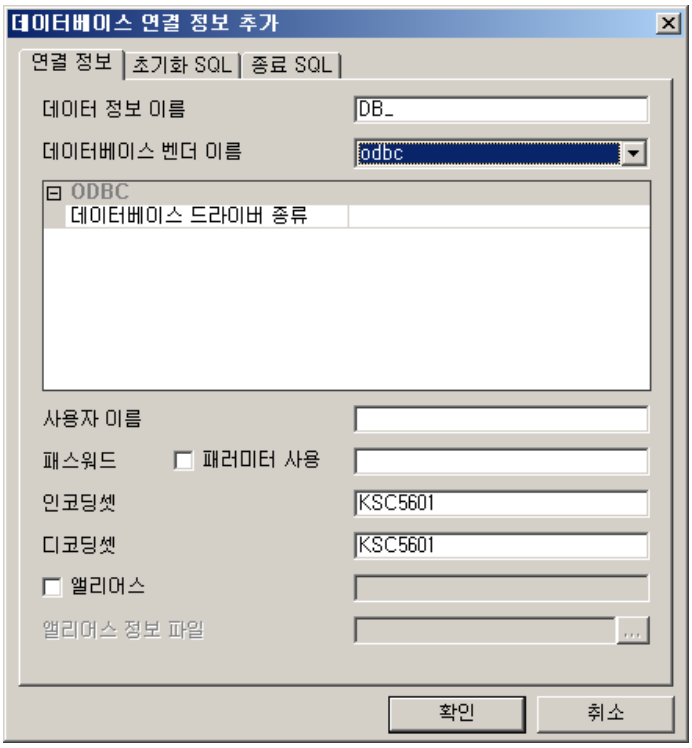
ODBC 또는 JDBC 드라이버를 이용해 원하는 데이터베이스와의 연결을 설정하며, 각 데이터베이스별 연결 설정 방법은 다음과 같습니다.

■ **ODBC를 이용한 연결**

ODBC를 이용한 연결은 ODBC 드라이버를 지원하는 MS Access, dBase, Foxpro 등의 데이터베이스와 연결할 때 사용할 수 있습니다. ODBC를 사용하기 위해서는 해당 드라이버가 아래의 그림과 같이 윈도우즈의 시스템 데이터 원본으로 등록되어 있어야 합니다.




ODBC를 이용하여 데이터베이스에 연결하기 위해서는 다음과 같이 데이터베이스 연결 정보를 설정합니다.



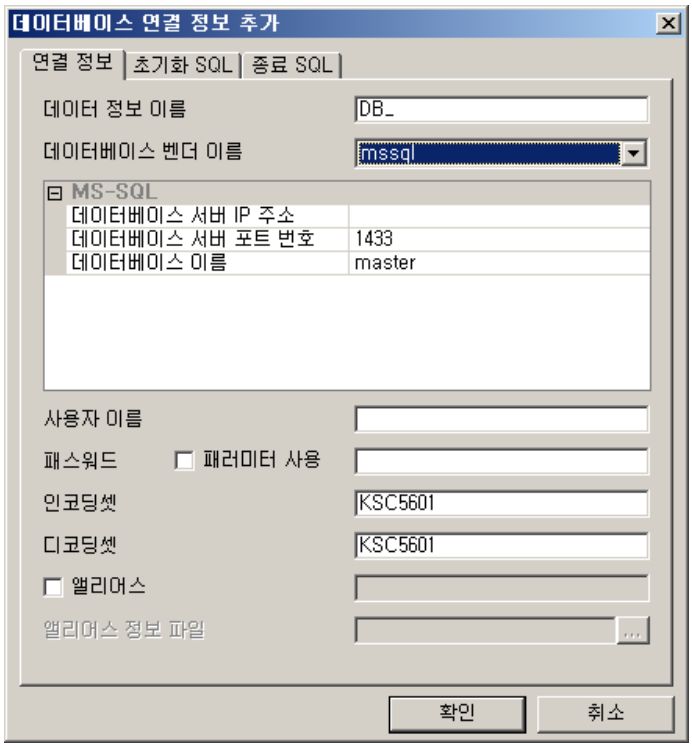

- "데이터 정보 이름"은 오즈 쿼리 디자이너가 "데이터베이스 연결 정보"를 구분하는 고유 이름으로 개발자가 임의로 부여합니다.
- "데이터베이스 벤더 이름"을 "odbc"로 선택합니다.
- "데이터베이스 드라이버 종류"에 윈도우즈에 등록된 "ODBC 데이터 원본" 이름을 입력합니다.
- "사용자 이름", "패스워드"에 연결할 데이터베이스의 사용자 이름과 패스워드를 입력합니다. "패러미터 사용" 옵션을 체크하면 사용자 패스워드가 사용자 지정 패러미터 값으로 설정됩니다.

■ **JDBC를 이용한 연결**

JDBC 드라이버를 이용한 연결은 JDBC 드라이버를 지원하는 MS-SQL, Oracle, Sybase 등의 데이터베이스와 연결할 때 사용할 수 있습니다.

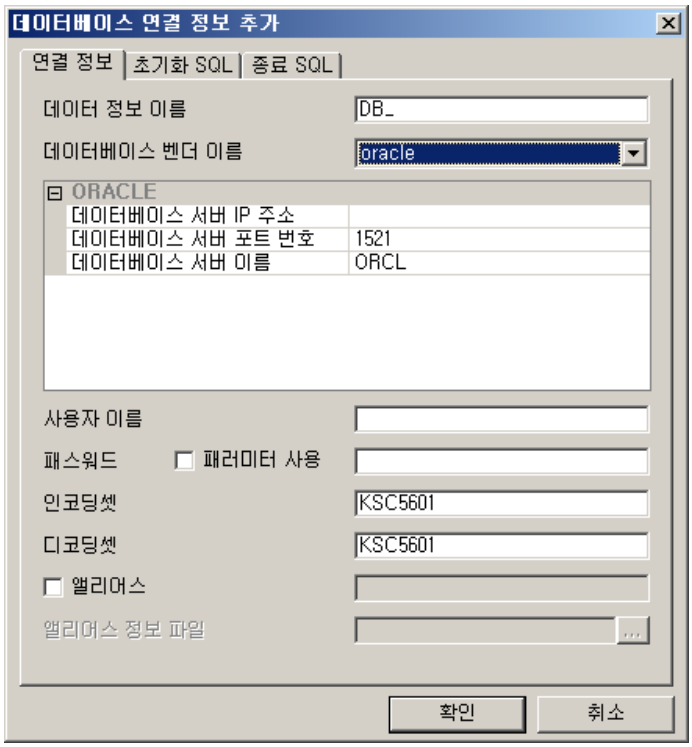
JDBC 드라이버를 이용해 데이터베이스에 연결하기 위해서는 먼저 "데이터 벤더 이름" 옵션을 선택한 후 각 데이터 벤더 별로 연결 정보를 입력하여야 합니다. 오즈가 지원하는 데이터 벤더 별 입력 예는 다음과 같습니다.

- **MS-SQL 연결 예**



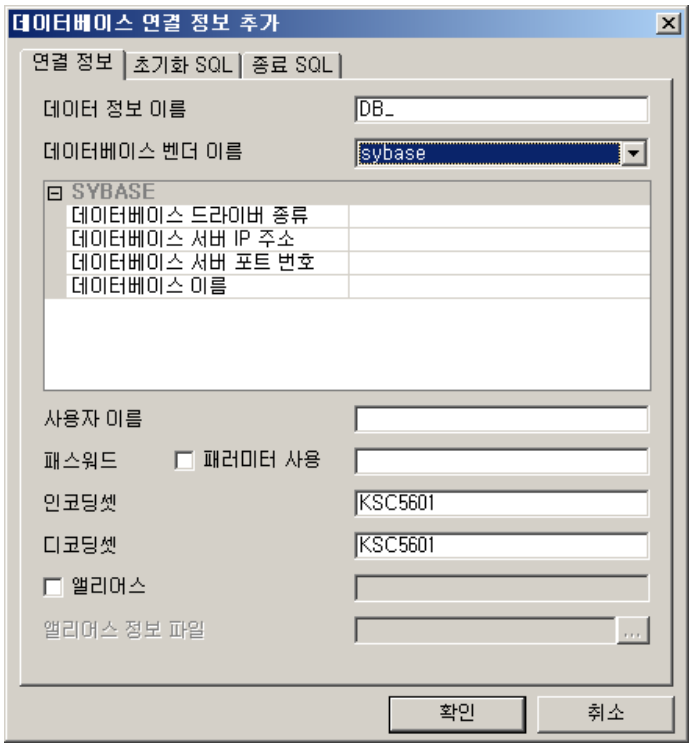

- "데이터 정보 이름"은 오즈 쿼리 디자이너가 "데이터베이스 연결 정보"를 구분하는 고유 이름으로 개발자가 임의로 부여합니다.
- "데이터베이스 벤더 이름"을 "mssql"로 선택합니다.
- "데이터베이스 서버 IP 주소", "데이터베이스 서버 포트 번호", "데이터베이스 이름"에 연결할 데이터베이스 서버의 IP, 포트 번호, 데이터베이스명을 입력합니다.
- "사용자 이름", "패스워드"에 연결할 데이터베이스의 사용자 이름과 패스워드를 입력합니다. "패러미터 사용" 옵션을 체크하면 사용자 패스워드가 사용자 지정 패러미터 값으로 설정됩니다.

- **Oracle 연결 예**



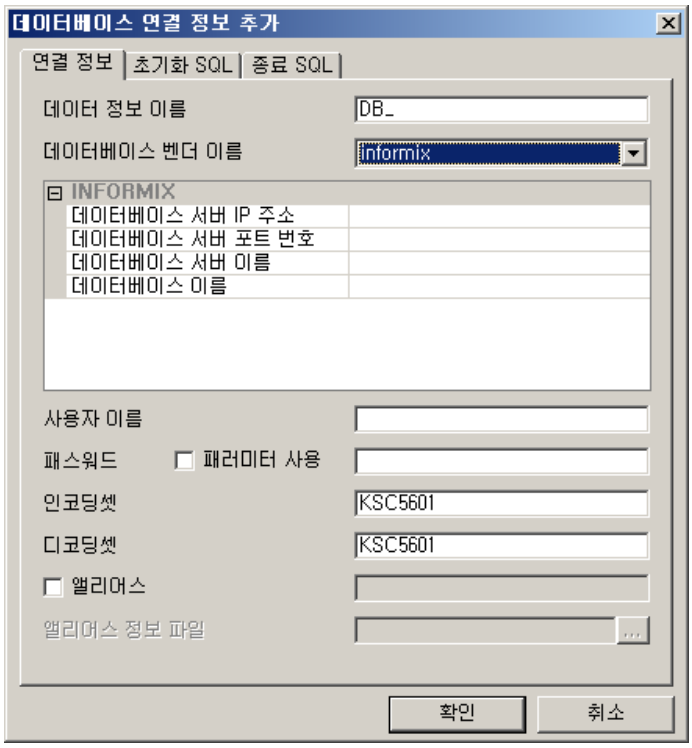

- "데이터 정보 이름"은 오즈 쿼리 디자이너가 "데이터베이스 연결 정보"를 구분하는 고유 이름으로 개발자가 임의로 부여합니다.
- "데이터베이스 벤더 이름"을 "oracle"로 선택합니다.
- "데이터베이스 서버 IP 주소", "데이터베이스 서버 포트 번호", "데이터베이스 서버 이름"에 연결할 데이터베이스 서버의 IP, 포트 번호, SID를 입력합니다.
- "사용자 이름", "패스워드"에 연결할 데이터베이스의 사용자 이름과 패스워드를 입력합니다. "패러미터 사용" 옵션을 체크하면 사용자 패스워드가 사용자 지정 패러미터 값으로 설정됩니다.

- **Sybase 연결 예**



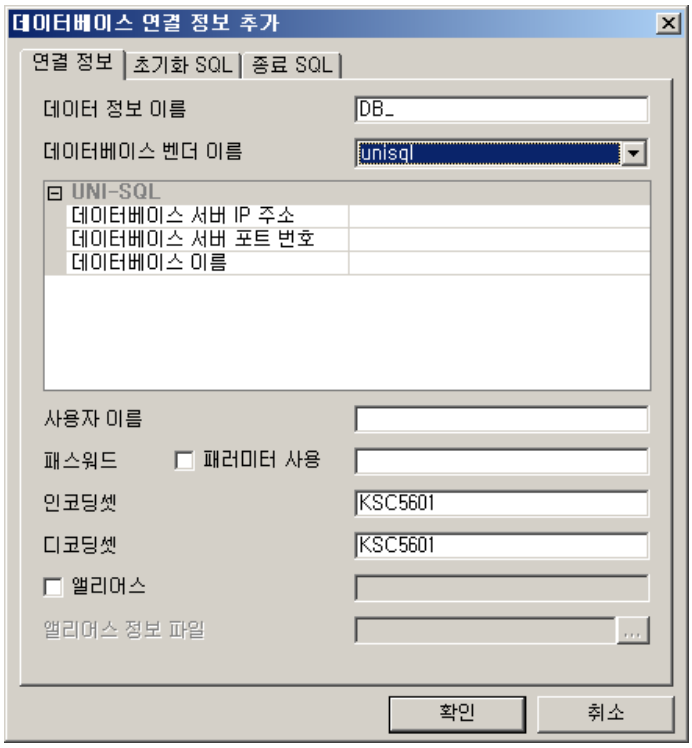

- "데이터 정보 이름"은 오즈 쿼리 디자이너가 "데이터베이스 연결 정보"를 구분하는 고유 이름으로 개발자가 임의로 부여합니다.
- "데이터베이스 벤더 이름"을 "sysbase"로 선택합니다.
- "데이터베이스 서버 드라이버 종류"에는 Sybase의 JDBC 드라이버인 jConnect를 사용하기 위해 'Tds'를 설정합니다. Tds(Tabular Data Stream)는 Adative Server를 위한 Sybase의 네이티브 통신 프로토콜입니다.
- "데이터베이스 서버 IP 주소", "데이터베이스 서버 포트 번호", "데이터베이스 이름"에 연결할 데이터베이스 서버의 IP, 포트 번호, 데이터베이스명을 입력합니다.
- "사용자 이름", "패스워드"에 연결할 데이터베이스의 사용자 이름과 패스워드를 입력합니다. "패러미터 사용" 옵션을 체크하면 사용자 패스워드가 사용자 지정 패러미터 값으로 설정됩니다.

- **Informix 연결 예**



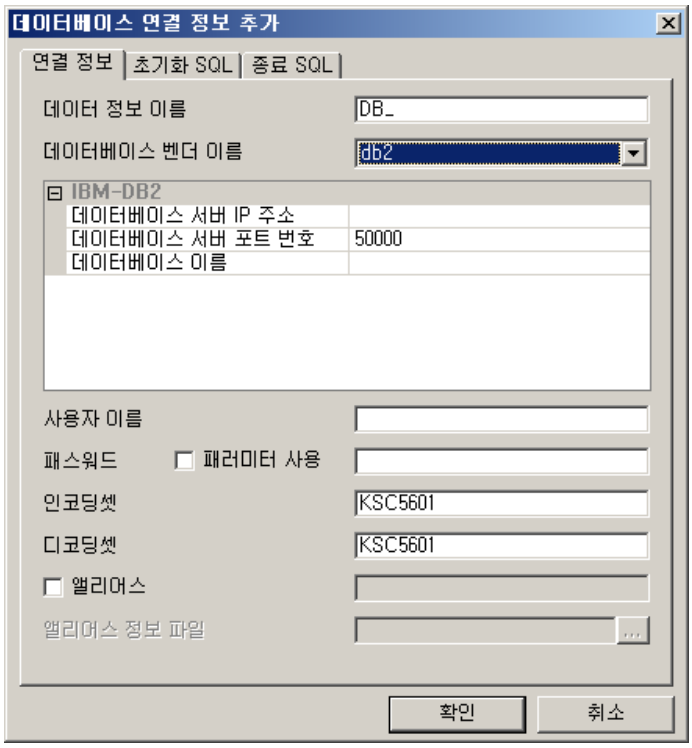

- "데이터 정보 이름"은 오즈 쿼리 디자이너가 "데이터베이스 연결 정보"를 구분하는 고유 이름으로 개발자가 임의로 부여합니다.
- "데이터베이스 벤더 이름"을 "informix"로 선택합니다.
- "데이터베이스 서버 IP 주소", "데이터베이스 서버 포트 번호", "데이터베이스 서버 이름", "데이터베이스 이름"에 연결할 데이터베이스 서버의 IP, 포트 번호, 데이터베이스명, 데이터베이스 서버명을 입력합니다.
- "사용자 이름", "패스워드"에 연결할 데이터베이스의 사용자 이름과 패스워드를 입력합니다. "패러미터 사용" 옵션을 체크하면 사용자 패스워드가 사용자 지정 패러미터 값으로 설정됩니다.

- **Uni-SQL 연결 예**




- "데이터 정보 이름"은 오즈 쿼리 디자이너가 "데이터베이스 연결 정보"를 구분하는 고유 이름으로 개발자가 임의로 부여합니다.
- "데이터베이스 벤더 이름"을 "unisql"로 선택합니다.
- "데이터베이스 서버 IP 주소", "데이터베이스 서버 포트 번호", "데이터베이스 이름"에 연결할 데이터베이스 서버의 IP, 포트 번호, 데이터베이스명, 데이터베이스 서버명을 입력합니다.
- "사용자 이름", "패스워드"에 연결할 데이터베이스의 사용자 이름과 패스워드를 입력합니다. "패러미터 사용" 옵션을 체크하면 사용자 패스워드가 사용자 지정 패러미터 값으로 설정됩니다.

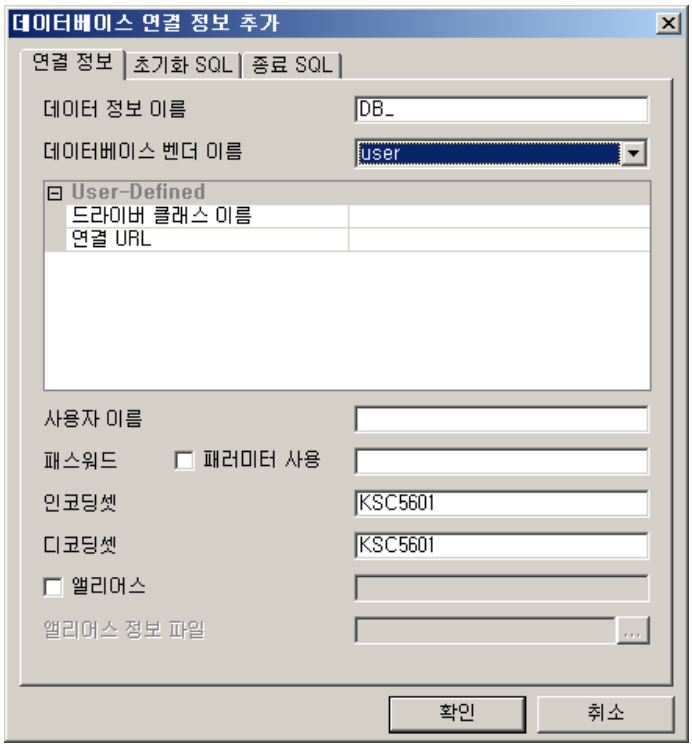
- **DB2 연결 예**




- "데이터 정보 이름"은 오즈 쿼리 디자이너가 "데이터베이스 연결 정보"를 구분하는 고유 이름으로 개발자가 임의로 부여합니다.
- "데이터베이스 벤더 이름"을 "db2"로 선택합니다.
- "데이터베이스 서버 IP 주소", "데이터베이스 서버 포트 번호", "데이터베이스 이름"에 연결할 데이터베이스 서버의 IP, 포트 번호, 데이터베이스명, 데이터베이스 서버명을 입력합니다.
- "사용자 이름", "패스워드"에 연결할 데이터베이스의 사용자 이름과 패스워드를 입력합니다. "패러미터 사용" 옵션을 체크하면 사용자 패스워드가 사용자 지정 패러미터 값으로 설정됩니다.

- **User-Definded 연결 예**

User-Defined 연결은 오즈에서 기본으로 제공되지 않는 데이터베이스인 경우 각각의 JDBC드라이버 정보를 사용자가 임의로 설정하여 연결하는 것으로 JDBC의 드라이버 정보와 Connection URL을 직접 입력합니다.



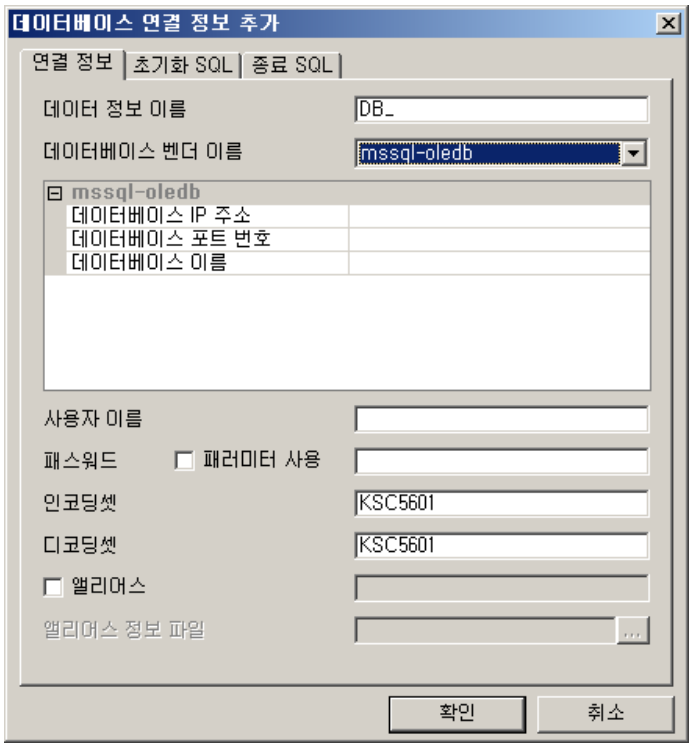

- "데이터 정보 이름"은 오즈 쿼리 디자이너가 "데이터베이스 연결 정보"를 구분하는 고유 이름으로 개발자가 임의로 부여합니다.
- "데이터베이스 벤더 이름"을 "user"로 선택합니다.
- "드라이버 클래스 이름", "연결 URL"에 연결할 데이터베이스의 JDBC의 드라이버 정보와 Connection URL 정보를 입력합니다.
- "사용자 이름", "패스워드"에 연결할 데이터베이스의 사용자 이름과 패스워드를 입력합니다. "패러미터 사용" 옵션을 체크하면 사용자 패스워드가 사용자 지정 패러미터 값으로 설정됩니다.

■ **OLE DB를 이용한 연결**

OLE DB를 이용한 연결은 데이터를 바인딩하는 서버 타입이 닷넷일 경우에만 지원되며 OLE DB를 지원하는 MS-SQL, Oracle, Sybase 등의 데이터베이스와 연결할 때 사용할 수 있습니다.

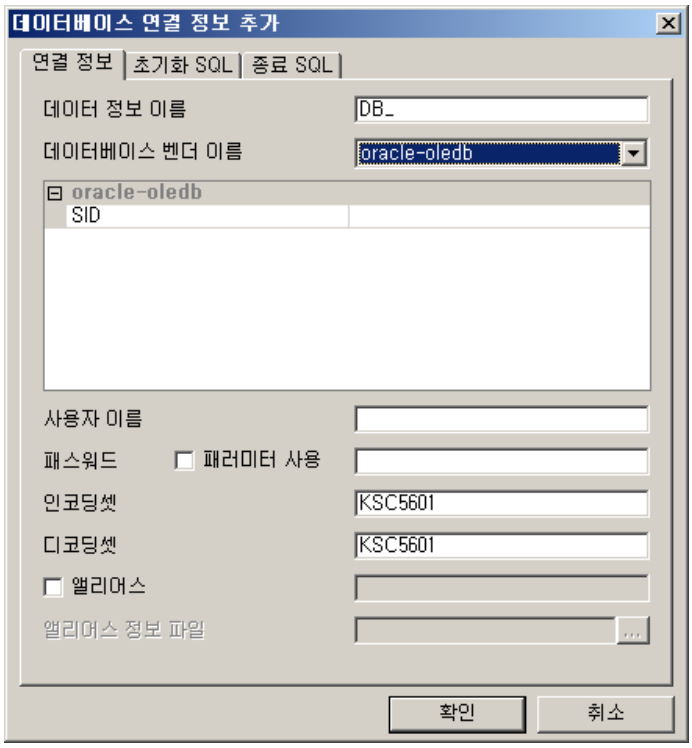
OLE DB를 이용해 데이터베이스에 연결하기 위해서는 먼저 "데이터 벤더 이름" 옵션을 선택한 후 각 데이터 벤더 별로 연결 정보를 입력하여야 합니다. 오즈가 지원하는 데이터 벤더 별 입력 예는 다음과 같습니다.

- **MSSQL-OLEDB 연결 예**



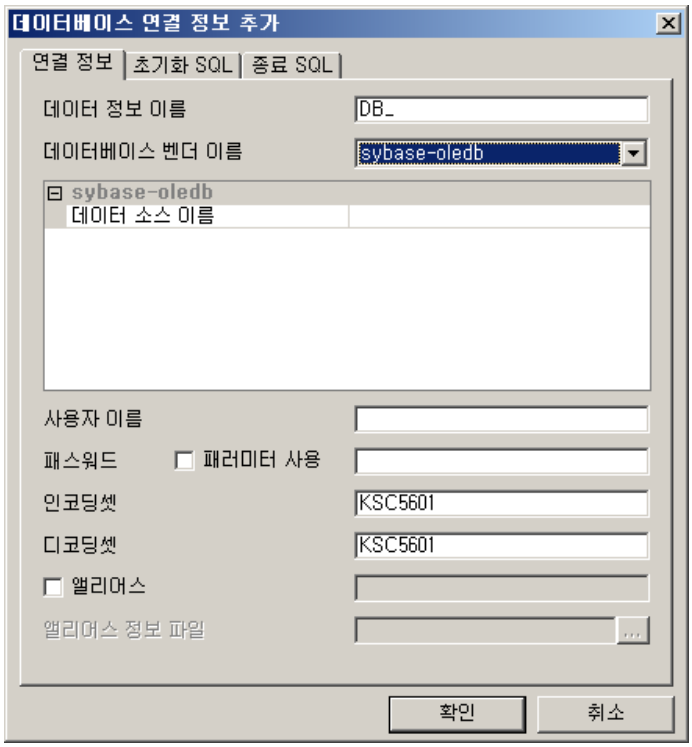

- "데이터 정보 이름"은 오즈 쿼리 디자이너가 "데이터베이스 연결 정보"를 구분하는 고유 이름으로 개발자가 임의로 부여합니다.
- "데이터베이스 벤더 이름"을 "MSSQL-OLEDB"로 선택합니다.
- "데이터베이스 서버 IP 주소", "데이터베이스 서버 포트 번호", "데이터베이스 이름"에 연결할 데이터베이스 서버의 IP, 포트 번호, 데이터베이스명을 입력합니다.
- "사용자 이름", "패스워드"에 연결할 데이터베이스의 사용자 이름과 패스워드를 입력합니다. "패러미터 사용" 옵션을 체크하면 사용자 패스워드가 사용자 지정 패러미터 값으로 설정됩니다.

- **ORACLE-OLEDB 연결 예**



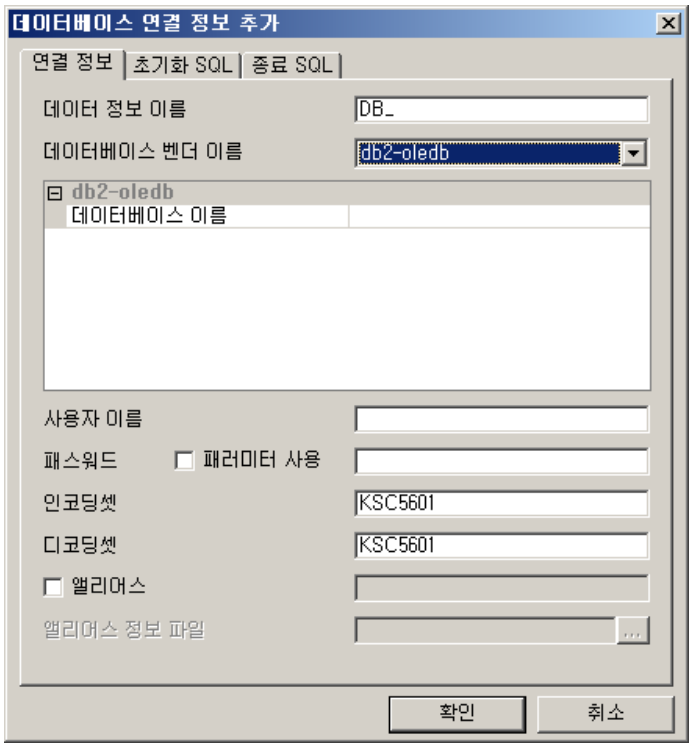

- "데이터 정보 이름"은 오즈 쿼리 디자이너가 "데이터베이스 연결 정보"를 구분하는 고유 이름으로 개발자가 임의로 부여합니다.
- "데이터베이스 벤더 이름"을 "ORACLE-OLEDB"로 선택합니다.
- "SID"에 연결할 데이터베이스 서버의 SID를 입력합니다.

- **SYBASE-OLEDB 연결 예**



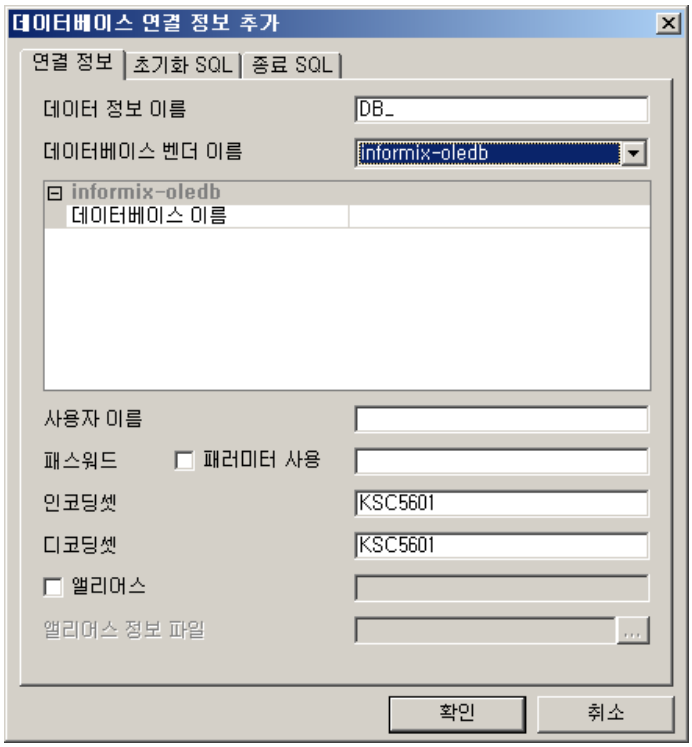

- "데이터 정보 이름"은 오즈 쿼리 디자이너가 "데이터베이스 연결 정보"를 구분하는 고유 이름으로 개발자가 임의로 부여합니다.
- "데이터베이스 벤더 이름"을 "SYBASE-OLEDB"로 선택합니다.
- "데이터 소스 이름"에 연결할 데이터베이스 서버의 데이터 소스 이름을 입력합니다.

- **DB2-OLEDB 연결 예**




- "데이터 정보 이름"은 오즈 쿼리 디자이너가 "데이터베이스 연결 정보"를 구분하는 고유 이름으로 개발자가 임의로 부여합니다.
- "데이터베이스 벤더 이름"을 "DB2-OLEDB"로 선택합니다.
- "데이터베이스 이름"에 연결할 데이터베이스 서버명을 입력합니다.

- **INFORMIX-OLEDB 연결 예**

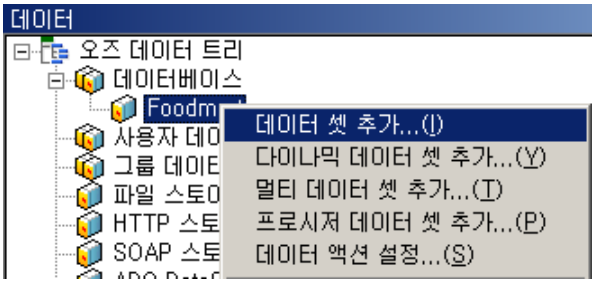



- "데이터 정보 이름"은 오즈 쿼리 디자이너가 "데이터베이스 연결 정보"를 구분하는 고유 이름으로 개발자가 임의로 부여합니다.
- "데이터베이스 벤더 이름"을 "INFORMIX-OLEDB"로 선택합니다.
- "데이터베이스 이름"에 연결할 데이터베이스 서버명을 입력합니다.

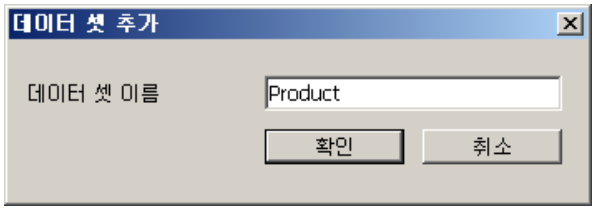
## 데이터 셋 생성

다음은 연결된 데이터베이스로부터 데이터 셋을 생성하는 과정을 설명합니다.

**Step 1** 'Foodmart'라는 데이터베이스 소스에서 마우스 오른쪽 버튼을 클릭하여 [데이터 셋 추가] 메뉴를 클릭합니다.



**Step 2** "데이터 셋 이름"에 생성할 데이터 셋 이름을 입력합니다.




**Step 3** [확인] 버튼을 누르면 아래와 같이 데이터베이스 항목 아래에 해당 데이터 셋 정보가 나타납니다.




**Step 4** 데이터 셋의 내용(쿼리문)을 작성합니다. 데이터 셋 작성에 대한 상세 내용은 "Ⅴ. 쿼리 데이터 셋 디자인"을 참조하시기 바랍니다.

# 동적 데이터베이스 연결

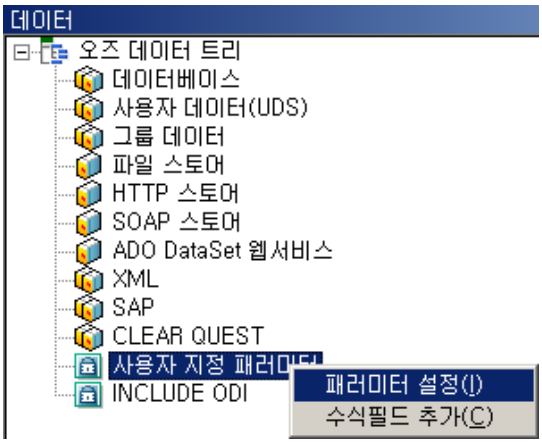
다음은 오즈 패러미터와 앨리어스를 이용하여 데이터베이스 소스에 동적으로 연결하고, 연결된 데이터베이스 소스로부터 SQL을 이용해 데이터 셋을 생성하는 과정을 설명합니다.

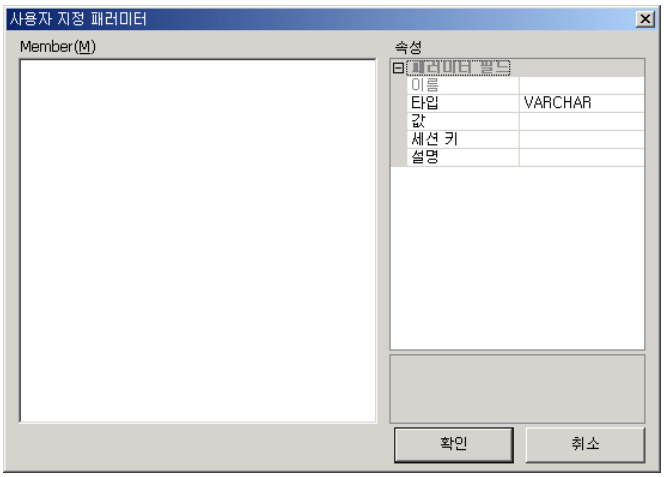
## 오즈 패러미터를 이용한 동적 데이터베이스 연결

데이터베이스에 대한 연결 정보가 보고서 디자인시가 아니라 운영 시점에 사용자 로그인 정보 등에 의해 결정될 경우 오즈 패러미터를 이용하여 동적으로 데이터베이스에 연결할 수 있습니다.

**Step 1**

우선 데이터베이스 서버 IP, 포트 번호, 사용자 계정, 패스워드 등 동적으로 변하는 정보를 위해 오즈 패러미터를 등록합니다. 패러미터 등록은 오즈 데이터 트리에서 '사용자 지정 패러미터'를 마우스 오른쪽 버튼을 클릭하여 나타나는 팝업 메뉴에서 [패러미터 설정] 메뉴를 클릭하여 '사용자 지정 패러미터 다이얼로그'를 띄웁니다.

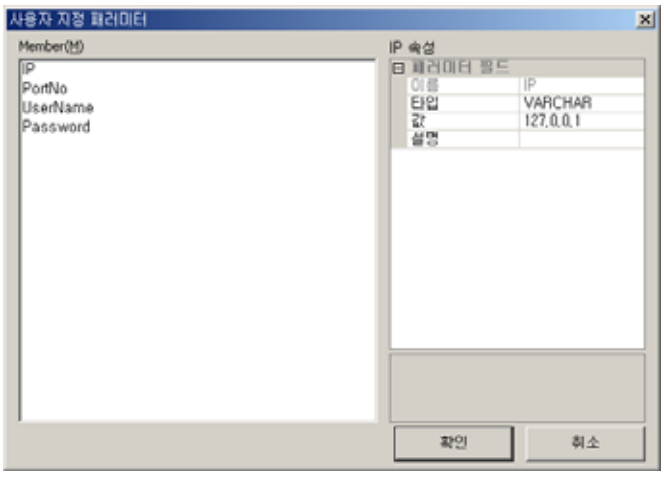




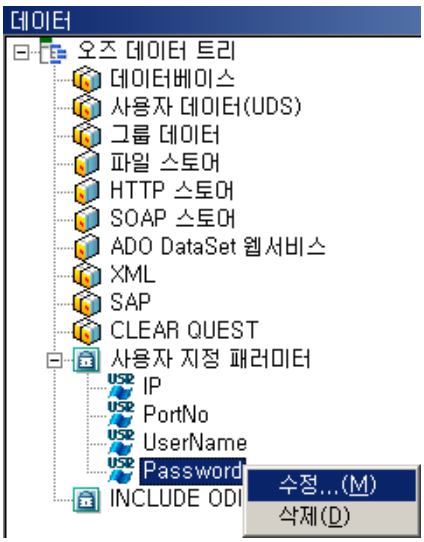

| 항목 | 설명 |
|---|---|
| **이름** | 추가할 패러미터 이름을 입력합니다. |
| **타입** | 추가할 패러미터의 타입을 설정합니다. |
| **값** | 추가할 패러미터의 기본 값(Default)을 입력합니다. |
| **세션 키** | 추가할 패러미터의 값을 설정할 세션 키를 입력합니다.<br>※ 참고사항 : 추가할 패러미터의 값을 오즈 뷰어 패러미터에서 설정하는 대신 서버 세션으로 설정하고자 할 때 사용합니다. |
| **설명** | 추가할 패러미터에 대한 설명을 입력합니다. |

**Step 2**

'사용자 지정 패러미터 다이얼로그'에서 패러미터 이름과 필드 타입, 패러미터 값 등을 설정한 후 [확인] 버튼을 입력하여 패러미터를 추가합니다. 본 매뉴얼에서는 'IP', 'PortNo', 'UserName', 'Password' 등 네 개의 패러미터를 추가하고, 각 패러미터의 값도 연결할 데이터베이스 서버의 연결 정보에 맞추어 입력하였습니다.



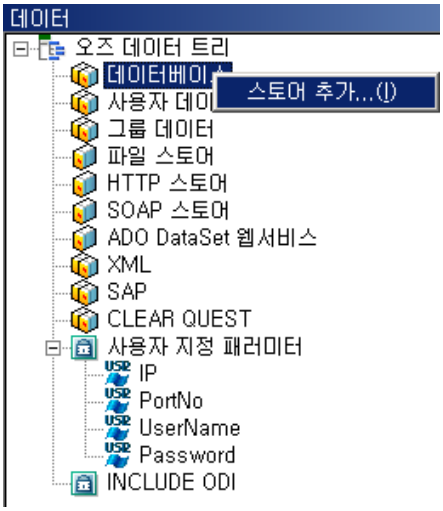
입력한 패러미터 값을 수정하기 위해서는 해당 패러미터를 마우스 오른쪽 버튼을 클릭하여 나타나는 팝업 메뉴에서 [수정] 메뉴를 클릭하거나 해당 패러미터의 속성 중 '값'을 변경하여 수정합니다.




**Step 3**

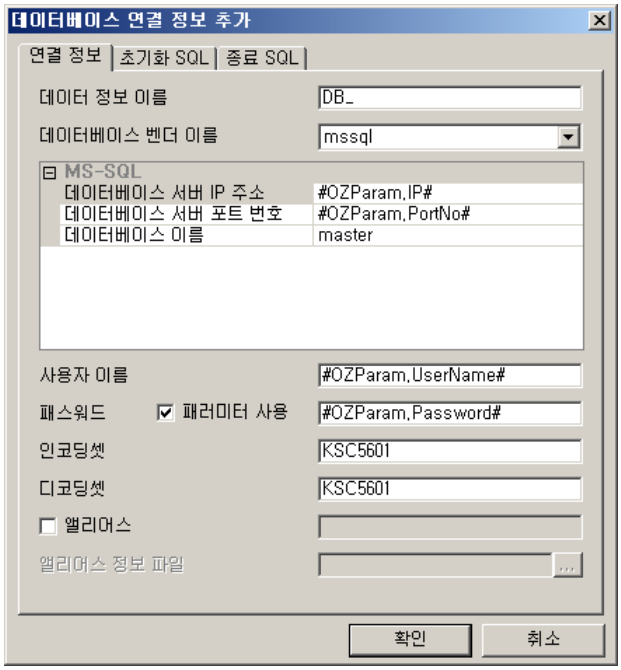
추가한 패러미터를 데이터베이스 스토어 연결 정보에 설정합니다.

오즈 데이터 트리에서 '데이터베이스'를 선택한 후 마우스 오른쪽 버튼을 클릭하여 나타나는 팝업 메뉴에서 [스토어 추가] 메뉴를 클릭합니다.




**Step 4**

데이터베이스 연결 정보에서 동적으로 변하는 항목에 대해 다음과 같이 해당 패러미터를 설정합니다.




※ 참고사항

- 패스워드를 사용자 지정 패러미터로 설정할 경우에는 "패러미터 사용" 옵션을 체크하여야합니다.
- 각 패러미터에 값이 제대로 설정되었다면 정상적으로 데이터베이스에 연결됩니다.
- 각 패러미터 값은 운영 시점에 사용자 애플리케이션으로부터 전달되며, 전달된 정보에 따라 동적으로 데이터베이스에 연결됩니다.

## 앨리어스(Alias)를 이용한 동적 데이터베이스 연결

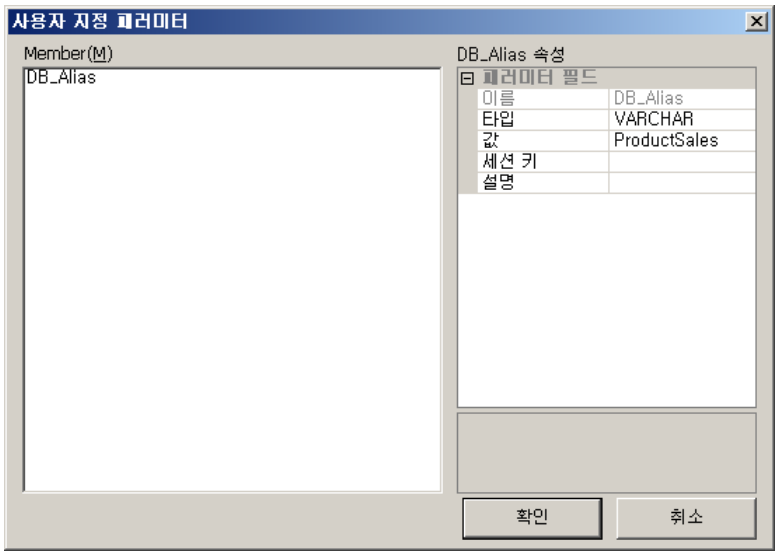
패러미터를 이용한 동적 데이터베이스 연결의 경우 중요한 데이터베이스 연결 정보가 유출될 우려가 있으므로 오즈 서버의 db.properties 파일에 등록되어 있는 데이터베이스 앨리어스명을 패러미터로 전달하여 데이터베이스에 동적으로 연결하는 방법을 제공합니다.
오즈 서버의 [/conf] 디렉토리에 있는 db.properties 파일에는 연결할 데이터베이스에 대한 정보가 등록되어 있습니다.

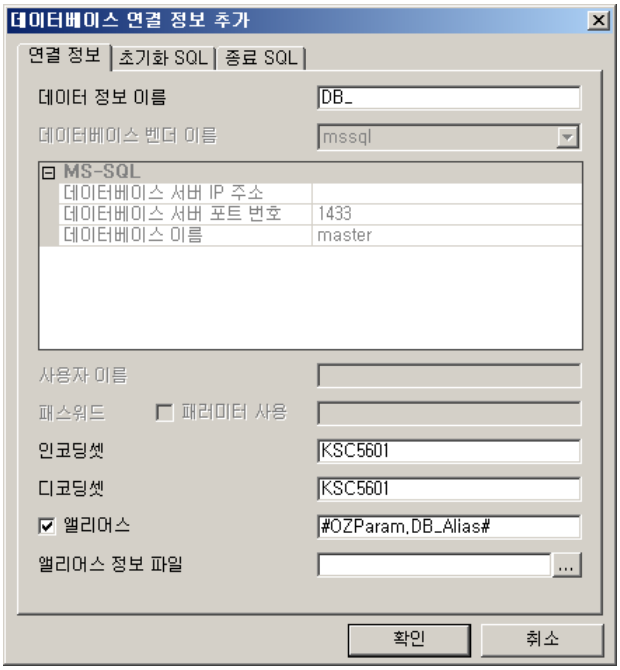
예를 들어 연결할 데이터베이스 벤더는 MS_SQL이고, 앨리어스를 "ProductSales"로 설정할 경우에 db.properties 파일은 아래와 같이 구성됩니다.

```
# setting for MS-SQL with Alias ProductSales

  ProductSales.vendor=mssql
  ProductSales.serverAddress=127.0.0.1
  ProductSales.portNo=1433
  ProductSales.dbName=marketing
  ProductSales.user=sa
  ProductSales.password=
  ProductSales.maxconns=20
  ProductSales.initconns=5
  ProductSales.timeout=5
```

ODI 파일에 앨리어스를 나타낼 사용자 지정 패러미터(패러미터 이름:DB_Alias, 패러미터 값:ProductSales)를 추가한 후 데이터베이스 연결 정보를 아래 그림과 같이 설정하면 운영 시에 동적 데이터베이스 연결이 가능합니다. 일반적으로 보고서 폼 작성 시 권장하는 방법입니다.





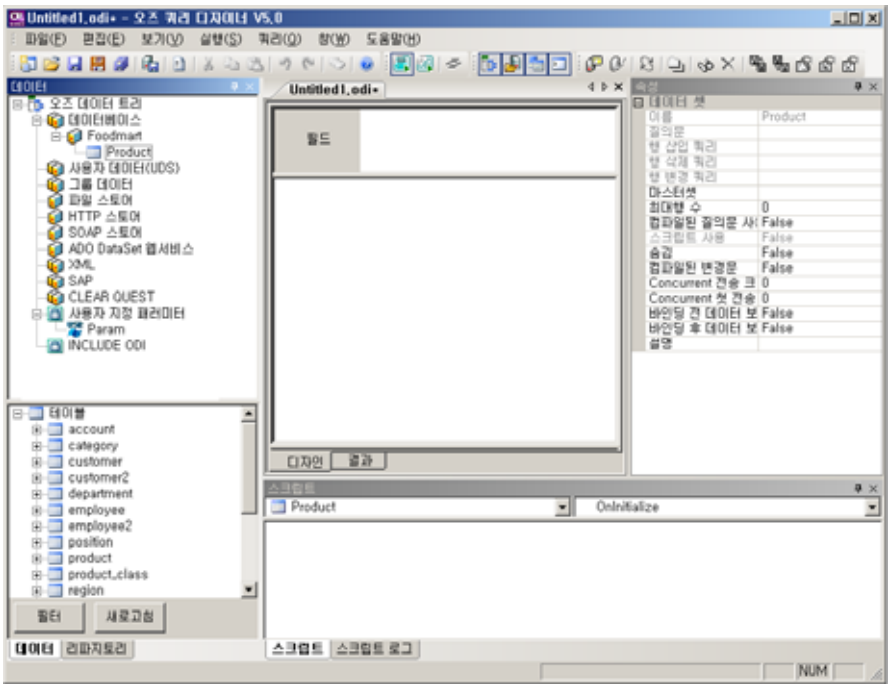

※ 주의사항 : DB 앨리어스를 사용하더라도 데이터베이스의 인코딩 및 디코딩 셋 설정은 UI에서 입력하여야합니다.

# Ⅵ. 쿼리 데이터 셋 디자인

- 매뉴얼 모드로 쿼리 디자인하기
- 디자인뷰 모드로 쿼리 디자인하기
- 사용자 지정 패러미터 추가하기
- 패러미터 마법사 사용하기
- 쿼리 마법사 사용하기
- 수식필드 추가하기
- INCLUDE ODI를 통한 필드 추가하기
- 마스터/디테일 셋 디자인하기
- 멀티 데이터 셋 디자인하기
- 다이나믹 데이터 셋 디자인하기
- 프로시저 데이터 셋 디자인하기
- 컴파일된 질의문 디자인하기
- 행 삽입/삭제/변경 쿼리 디자인하기

**본 장에서는 쿼리 데이터 셋을 생성하고 매뉴얼 모드 및 디자인뷰 모드를 이용해 쿼리문을 작성하는 방법에 대해 설명합니다.**

RDB 데이터 소스에 쿼리 데이터 셋을 생성하면 다음과 같이 쿼리 창이 나타납니다.




쿼리 작성은 디자인 뷰 모드와 직접 쿼리문을 입력할 수 있는 매뉴얼 모드로 나눌 수 있습니다. 두 모드의 차이점은 다음과 같습니다.

| 구분 | 디자인 뷰 모드(기본모드) | 매뉴얼 모드 |
|---|---|---|
| **뷰** | 디자인 뷰/결과 뷰 | 쿼리 뷰/결과 뷰 |
| **쿼리문 입력** | 불가 | 가능 |
| **쿼리 마법사 사용** | 가능 | 불가 |
| **스크립트 사용** | 불가 | 가능 |

※ 참고사항

- [쿼리] 메뉴의 [디자인 모드] 메뉴를 이용하여 모드를 변경할 수 있습니다.
- 매뉴얼 모드에서 디자인뷰 모드로 변경하면 매뉴얼 모드에서 작성한 쿼리문이나 스크립트는 삭제될 수 있습니다. 디자인 뷰 모드는 SQL문을 정확하게 모르는 사용자들도 위저드 방식을 통해 손쉽게 쿼리문을 작성할 수 있는 장점이 있습니다. 그러나, 마법사 방식에서 쿼리문 작동 생성에는 한계가 있기 때문에 데이터 추출 로직이 아주 복잡한 경우나 SQL 성능 튜닝이 필요한 경우에는 매뉴얼 모드로 변환하여 사용자가 직접 추가 작업을 수행하는 것을 권장합니다.

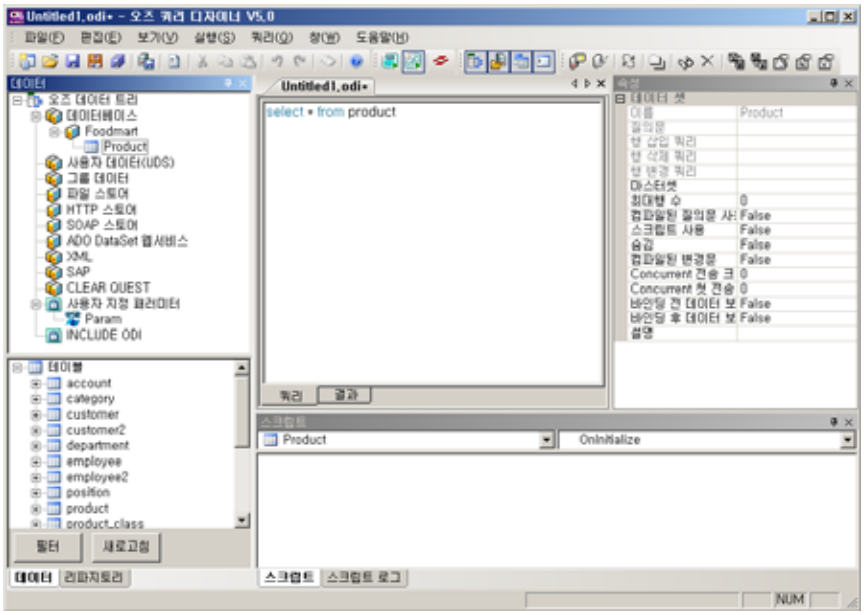
# 매뉴얼 모드로 쿼리 디자인하기

데이터 창에서 데이터 셋을 선택한 후 메뉴바에서 [쿼리] - [디자인모드] - [매뉴얼] 메뉴를 클릭하거나 툴바에서 매뉴얼 모드로 변경 아이콘()을 클릭하면 쿼리 창이 매뉴얼 모드 쿼리 창으로 변경됩니다. 쿼리 창에 쿼리문을 직접 입력하여 쿼리를 디자인합니다.

**Step 1**

**쿼리문 작성**

쿼리 디자인 모드를 매뉴얼 모드로 변경한 후 [쿼리] 탭을 선택하고 쿼리문을 직접 입력합니다.

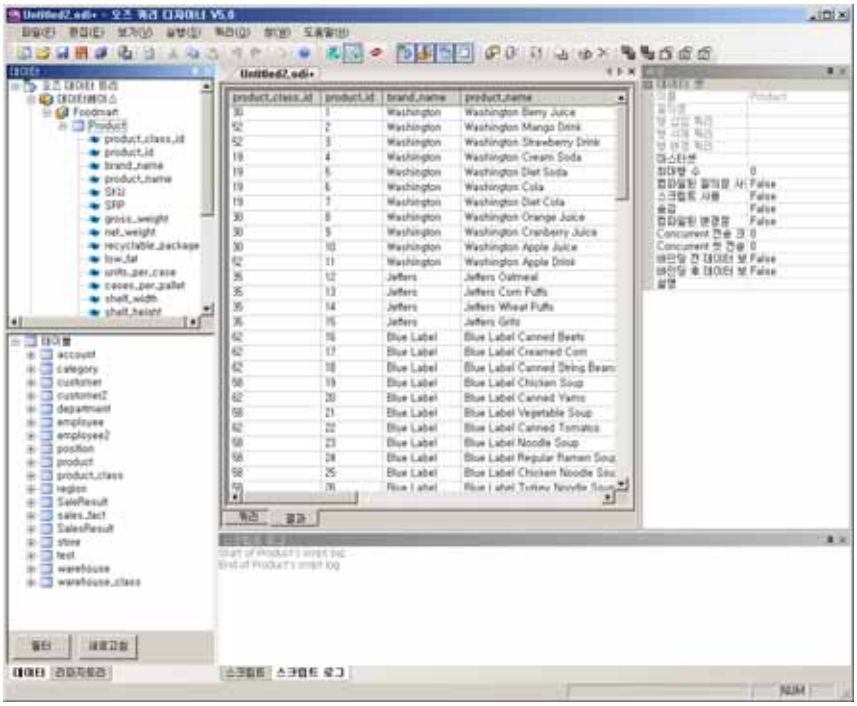



※ 주의사항 : 쿼리문에 주석을 사용한 경우 주석 내용에 사용자 지정 패러미터, 사용자 지정 패러미터 구분자, 마스터 셋의 필드 값이 포함되어 있으면 쿼리문이 정상 동작하지 않을 수 있으니 사용 시 주의 바랍니다.

**Step 2**

**쿼리문 실행**

쿼리문을 입력한 후 메뉴바에서 [실행] - [쿼리문 실행] 메뉴를 클릭하거나 툴바에서 쿼리문 실행()아이콘을 클릭하거나 또는 키보드의 <F5> 키를 눌러 쿼리문을 실행합니다. 쿼리문에 에러가 없을 경우에는 쿼리 창이 [결과] 탭으로 선택되면서 결과 데이터가 표시됩니다.

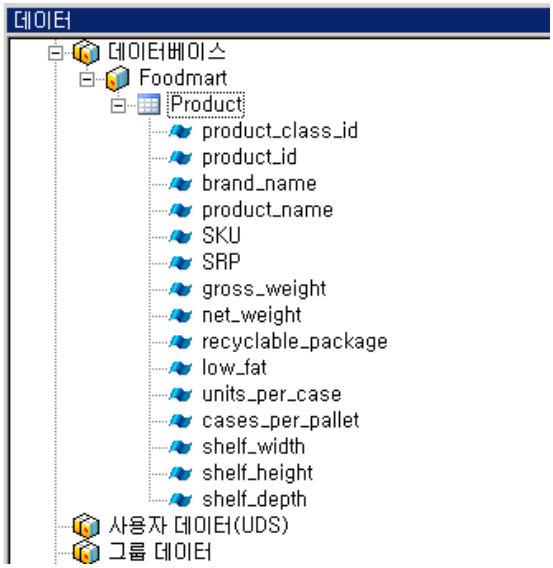


※ 참고사항 : 쿼리문에 에러가 있을 경우에는 쿼리문 실행이 되지 않으며 에러 창이 나타납니다. 에러를 수정하고 다시 실행합니다.

**Step 3**

**결과 필드 확인**

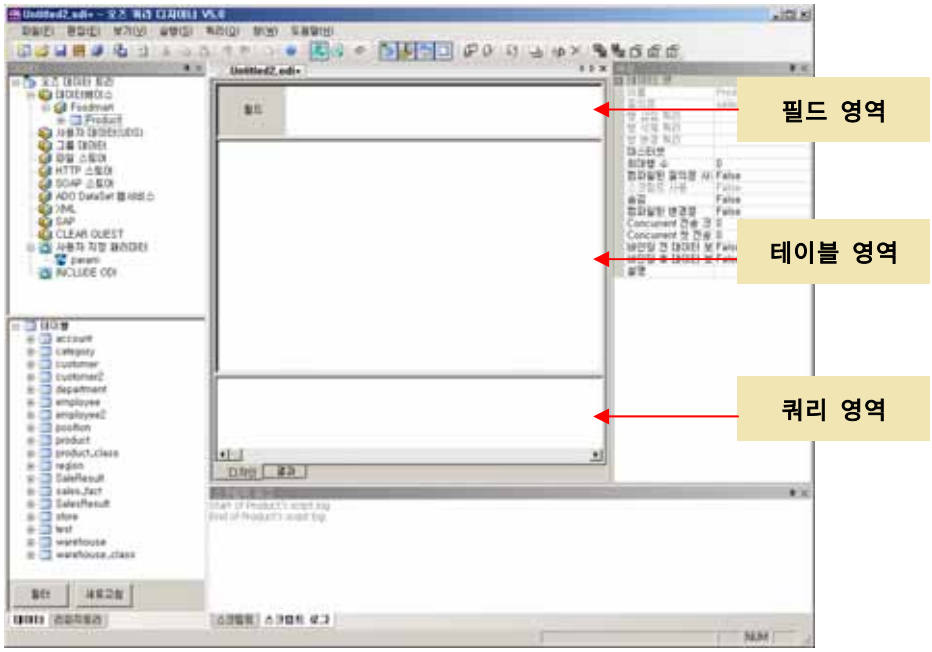
쿼리가 제대로 실행이 되면 해당 데이터 셋 아래에 결과 필드가 표시됩니다.



# 디자인뷰 모드로 쿼리 디자인하기

데이터에서 데이터 셋을 선택한 후 메뉴바에서 [쿼리] - [디자인모드] - [디자인 뷰] 메뉴를 클릭하거나 툴바에서 디자인뷰 모드로 변경 아이콘( )을 클릭하면 쿼리 창이 디자인뷰 모드 쿼리 창으로 변경됩니다.

디자인뷰 모드의 [디자인] 탭은 화면 상단에 선택한 필드를 나타내는 필드 영역, 화면 중앙에 추가한 테이블을 나타내는 테이블 영역, 화면 하단에 쿼리 마법사를 이용하여 작성된 쿼리문을 표시하는 쿼리 영역으로 구성되어 있습니다.

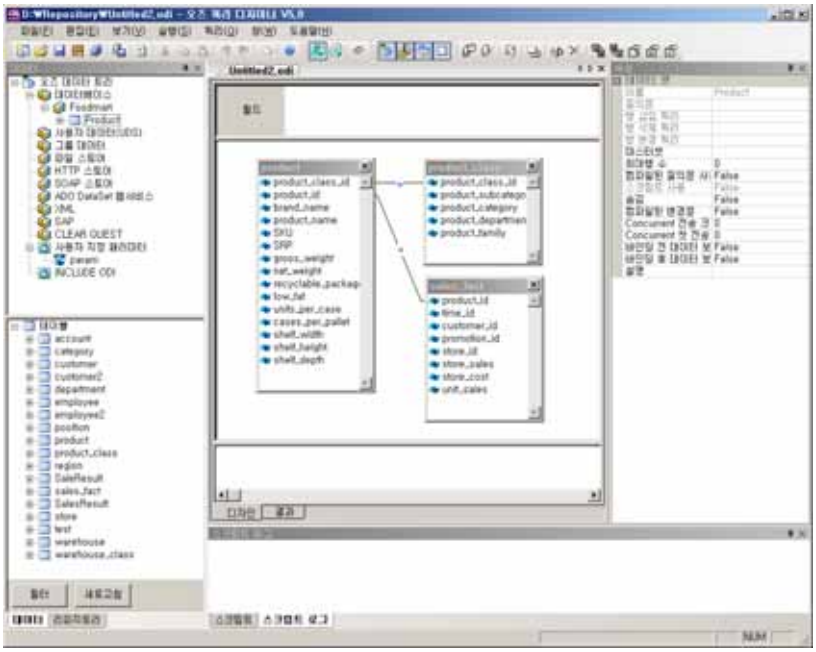



다음은 디자인뷰 모드에서 쿼리를 디자인하는 방법입니다.

**Step 1**

**테이블 선택**

[디자인] 탭을 선택한 후 쿼리에 포함시키고자 하는 테이블을 데이터 창에 표시된 테이블 목록에서 선택하여 쿼리 창의 테이블 영역에 드래그&드롭합니다. 원하는 테이블을 반복해서 가져다 놓습니다. 이때 선택한 테이블들은 쿼리 창의 테이블 영역에 표시되며, 쿼리문의 From절에 삽입됩니다.



추가된 테이블을 삭제할 경우에는 테이블 다이얼로그에서 닫기 아이콘()을 클릭합니다.

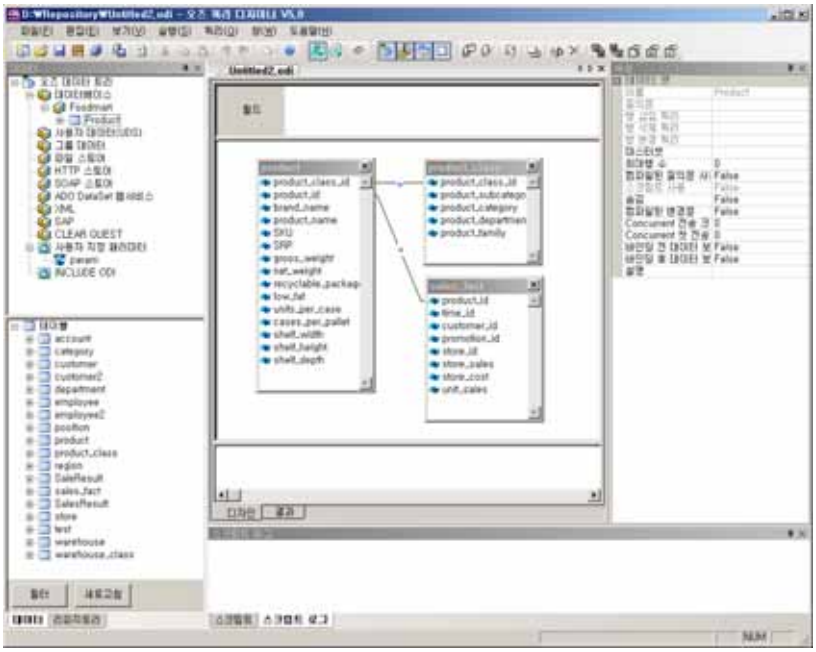
**Step 2**

**조인 생성**

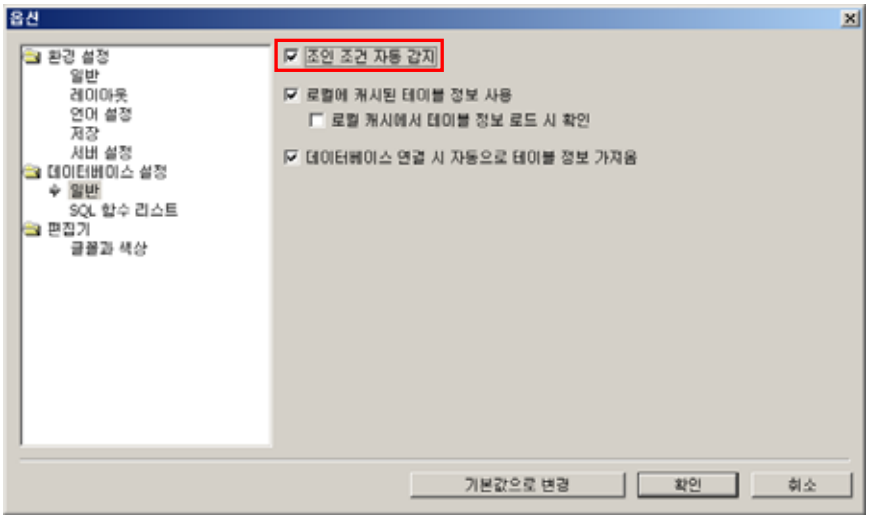
테이블이 두 개 이상이면 조인 조건을 생성할 수 있습니다. 조인 조건은 자동으로 감지되어 생성되기도 하며, 수동으로 조인 조건을 추가할 수 있습니다.

쿼리 디자이너의 옵션 중 '조인 조건 자동 감지' 기능이 사용함으로 설정되어 있을 경우에 한하여 다음과 같은 경우에는 조인 조건이 자동으로 생성됩니다.

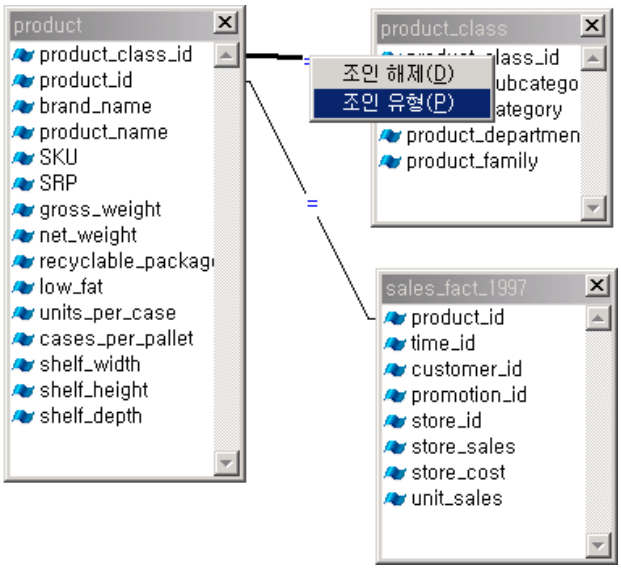
- DB에서 Primary, Foreign Key 조합을 JDBC 함수를 통해 가져올 수 있는 경우(Oracle, MS-SQL 등) : 두 테이블의 필드가 Primay, Foreign Key로 연결되어 있는 경우 조인 조건 자동 생성
- DB에서 Primary, Foreign Key 조합을 JDBC 함수를 통해 가져올 수 없는 경우(ODBC 등) : 두 테이블의 필드명과 필드 타입이 같은 경우 조인 조건 자동 생성



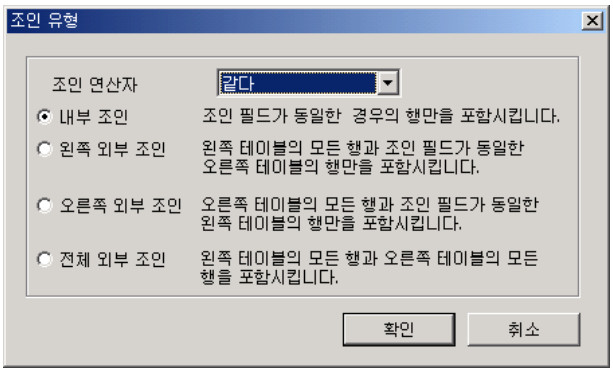
'조인 조건 자동 감지' 기능의 사용 여부는 메뉴바의 [파일] - [옵션] 메뉴를 클릭하여 나타나는 '옵션 다이얼로그'의 [데이터베이스 설정] – [일반] 탭에서 확인할 수 있습니다.




수동으로 조인 조건을 추가하기 위해서는 테이블에서 원하는 필드를 선택한 다음 다른 테이블의 조인하고자 하는 필드로 드래그&드롭하여 연결합니다. 조인 유형을 바꾸기 위해서는 해당 조인 연결선을 선택하여 더블 클릭하거나 마우스 오른쪽 클릭을 클릭하여 나타나는 팝업 메뉴에서 [조인 유형] 메뉴를 클릭합니다.

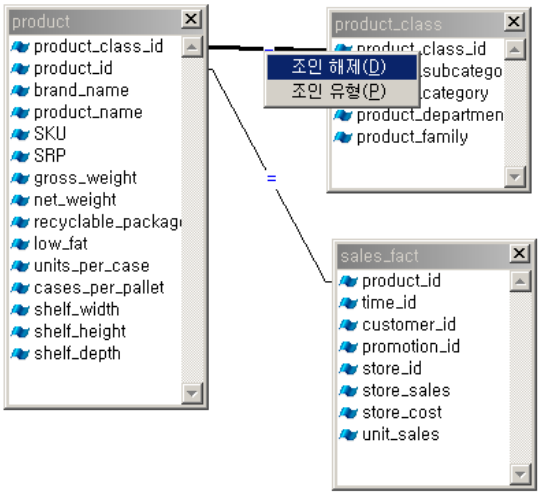



다음과 같이 '조인 유형 다이얼로그'를 이용하여 원하는 조인 유형으로 변경할 수 있습니다.




※ 주의사항 : 조인 유형에서 왼쪽, 오른쪽 외부 조인이나 전체 외부 조인은 현재 사용하는 데이터베이스의 해당 버전에서 지원하는지 먼저 체크하여야 합니다. 데이터베이스 벤더에서 지원하지 않는 조인 유형을 사용할 경우 쿼리문 실행 시 에러가 발생합니다. 예를 들어 오라클의 경우 전체 외부 조인은 9i 이상 버전에서만 지원합니다.

생성된 조인 조건을 삭제하기 위해서는 해당 조인 연결선을 선택한 후 키보드의 <Delete>키를 누르거나 조인 연결선에서 마우스 오른쪽 버튼을 클릭하여 나타나는 팝업 메뉴에서 [조인 해제] 메뉴를 클릭합니다.

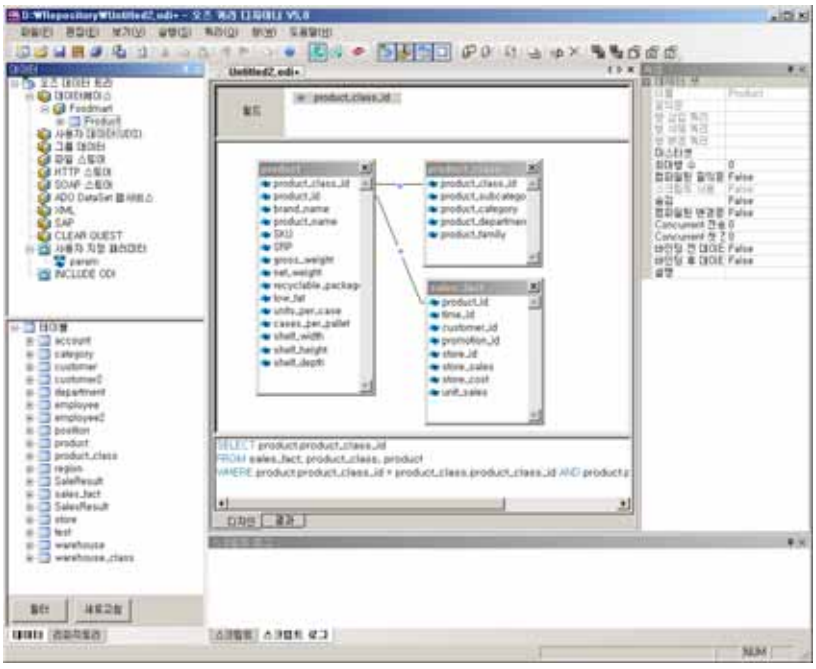



조인 조건이 없는 두 테이블은 크로스 조인으로 연결됩니다.

**Step 3**

**필드 설정**

데이터 셋에 포함하고자 하는 필드를 테이블 영역에 추가된 테이블 다이얼로그에서 선택한 후 필드 영역으로 드래그&드롭하거나 더블 클릭하여 해당 데이터 셋에 추가합니다.



<Shift> 또는 <Ctrl> 키를 이용하여 여러 개의 필드를 한꺼번에 추가할 수 있으며, 추가된 필드를 삭제하려면 필드 영역에서 필드를 선택한 후 키보드의

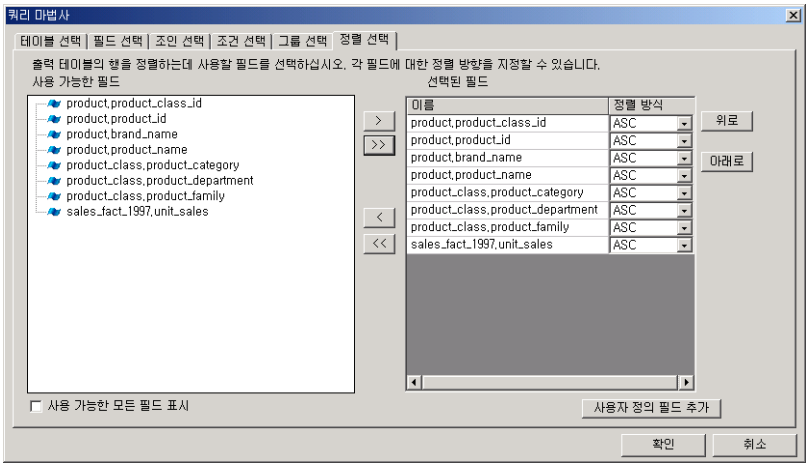
<Delete>키를 누르면 필드가 삭제됩니다.

**Step 4**

**정렬 선택**

쿼리 마법사를 이용하여 각 필드의 정렬 방식을 선택합니다. 메뉴바의 [쿼리] - [쿼리 마법사] 메뉴를 클릭하거나 쿼리 창에서 마우스 오른쪽 버튼을 클릭하여 나타나는 팝업 메뉴에서 [쿼리 마법사] 메뉴를 클릭합니다.

쿼리 마법사에 [정렬 선택] 탭을 선택하여 각 필드의 정렬 방식을 설정합니다. 사용 가능한 필드 목록에서 정렬 기준이 되는 필드를 선택하여 추가하고, 오름차순 또는 내림차순으로 정렬 방식을 설정하고, 정렬 순서도 변경합니다. 자세한 사용법은 본 장의 '쿼리 마법사' 부분을 참조하시기 바랍니다.




**Step 5**

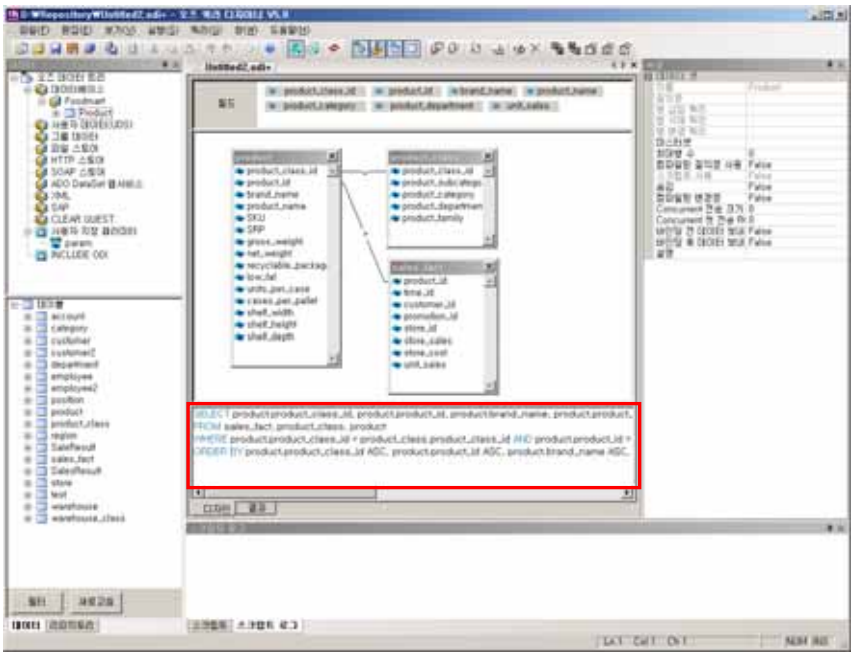
**추가 조건 설정, 그룹 선택**

조인 조건 외에 추가 조건 설정이나 그룹 선택 등을 쿼리 마법사를 이용하여 설정합니다.

**Step 6**

**쿼리문 확인**

작성된 쿼리는 쿼리 창의 쿼리 영역에 반영되어 보여집니다.



※ 참고사항 : 연결한 데이터 소스가 MDB 또는 MS-SQL일 때 다음과 같은 경우에는 쿼리가 "[테이블].필드명" 형태로 작성됩니다.

① 테이블명에 공백이 있는 경우

ex) Car Sales → [Car Sales].필드명

② 테이블명 앞에 숫자가 있는 경우

ex) 9Sales → [9Sales].필드명

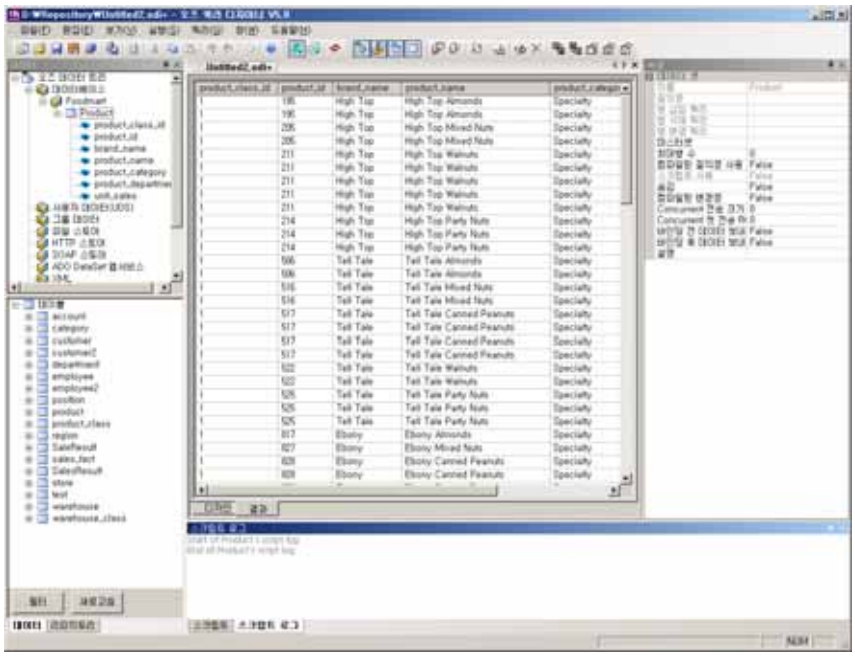
③ 테이블명과 필드명이 같은 경우

ex) Sales.Sales → [Sales].Sales

**Step 7**

**쿼리문 실행**

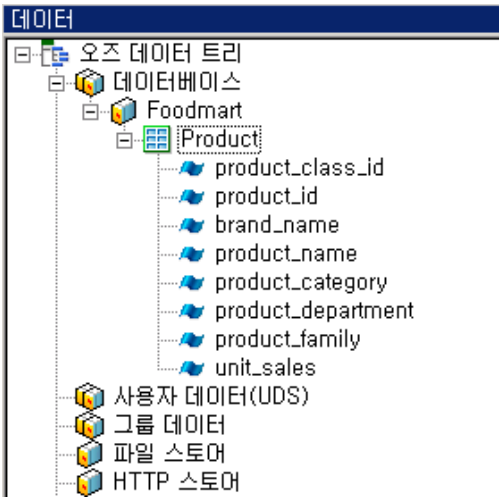
쿼리 디자인을 마친 후에는 메뉴바의 [실행] - [쿼리문 실행] 메뉴를 클릭하거나 툴바에서 쿼리문 실행 아이콘( )을 클릭하거나 또는 키보드의 <F5>키를 눌러 쿼리문을 실행합니다. 쿼리 창이 [결과] 탭으로 선택되면서 결과 데이터가 표시됩니다.



**Step 8**

**결과 필드 확인**

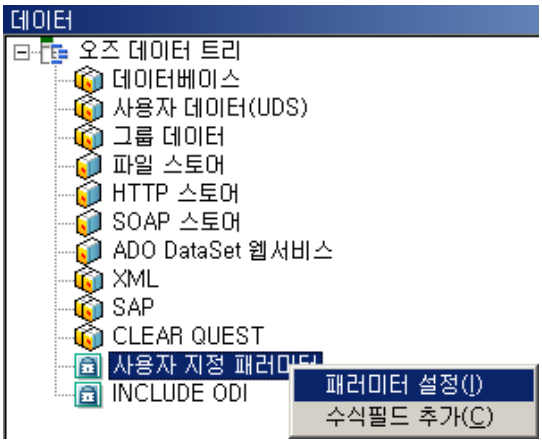
해당 데이터 셋 아래에 결과 필드가 표시됩니다.



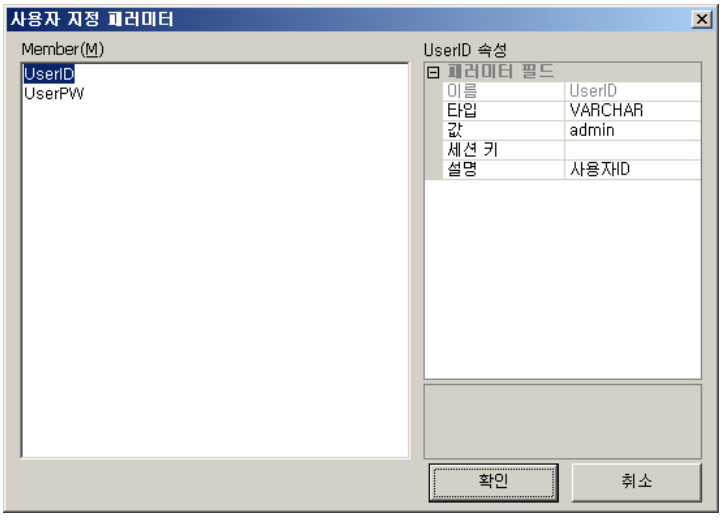
# 사용자 지정 패러미터 추가하기

사용자 지정 패러미터에서 사용자가 임의로 정의하여 사용하는 패러미터 변수를 추가하여 사용할 수 있습니다.

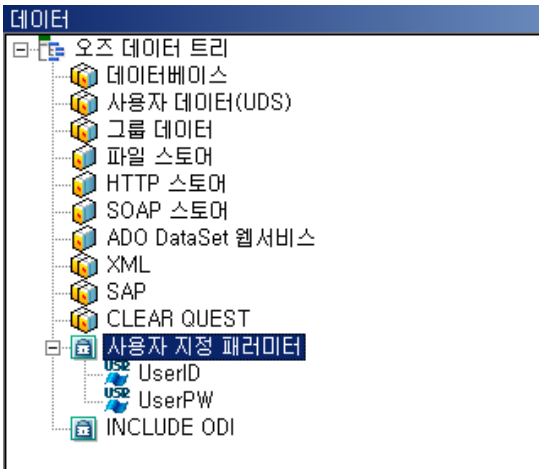
**Step 1** 오즈 쿼리 디자이너의 데이터 창에서 사용자 지정 패러미터를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [패러미터 설정] 메뉴를 클릭합니다.




**Step 2** [Member]에 패러미터 이름을 입력하고 속성 창에 패러미터 속성을 설정합니다.




**Step 3** [확인] 버튼을 클릭하면 사용자 지정 패러미터에 사용자 지정 패러미터가 추가됩니다.

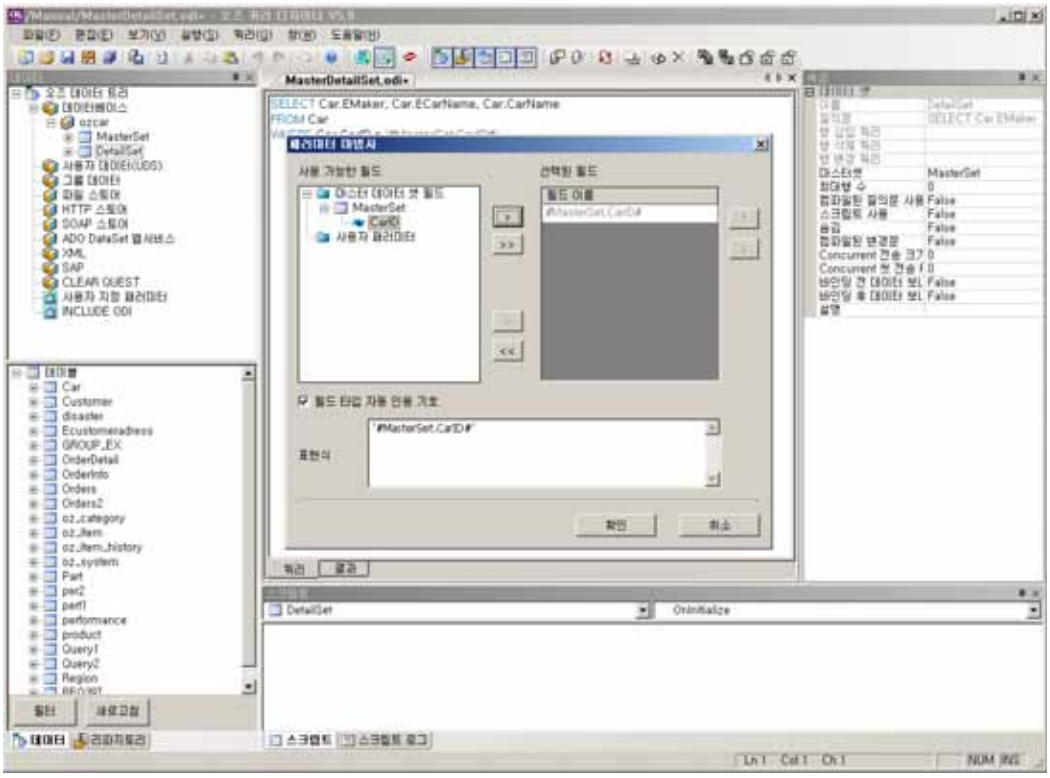



※ 참고사항 : 쿼리문에서 사용자 지정 패러미터를 이용하여 조건문을 설정할 경우 쿼리문 실행 시 '패러미터 입력 다이얼로그'가 표시되며 '사용자 지정 패러미터 다이얼로그'에서 해당 패러미터 값이 설정어 있으면 '패러미터 입력 다이얼로그'에서 설정한 패러미터 값이 표시되고 설정되어 있지 않으면 빈 값으로 표시됩니다.

# 패러미터 마법사 사용하기

패러미터 마법사는 쿼리의 Where절에서 마스터 데이터 셋 필드나 사용자 지정 패러미터를 이용하여 조건을 설정하기 위한 마법사로 마스터 데이터 셋의 필드나 사용자 패러미터를 리스트 형태로 표시하고 선택할 수 있게 합니다.

매뉴얼 모드에서 쿼리문을 작성할 때 쿼리 창에서 마우스 오른쪽 버튼을 클릭하여 나타나는 팝업 메뉴에서 [패러미터 마법사] 메뉴를 클릭하면 패러미터 마법사가 나타납니다



| 항목 | 설명 |
|---|---|
| **사용 가능한 필드** | 마스터 데이터 셋 필드, 사용자 패러미터를 표시합니다.<br>※ 참고사항 : 마스터 데이터 셋 필드는 데이터셋에 마스터 셋이지정된 경우에만 표시됩니다. |
| **선택된 필드** | 표현식에 추가된 필드 리스트를 나타냅니다. |
| **>** | '사용 가능한 필드'에 선택된 필드를 '선택된 필드'에 추가합니다. |
| **>>** | '사용 가능한 필드'에 추가된 모든 필드를 '선택된 필드'에 추가합니다. |

| | |
|---|---|
| < | '선택된 필드'에서 선택한 필드를 삭제합니다. |
| << | '선택된 필드'에 추가된 모든 필드를 삭제합니다. |
| ↑, ↓ | '선택된 필드'에서 선택한 필드의 위치를 위 또는 아래로 이동하여 표현식의 필드 순서를 변경합니다. |
| **필드 타입 자동 인용 기호** | 필드 타입이 CHAR, VARCHAR, LONGVARCHAR, DATE, TIME, TIMESTAMP일 경우 표현식에 패러미터 값을 따옴표로 묶어서 표시합니다.<br>※ 참고사항 : 데이터셋의 "컴파일된 변경문" 속성 값이 True일 경우 해당 옵션이 비활성화됩니다. |
| **표현식** | 선택된 필드를 패러미터 형식으로 표시합니다. 표현식에 추가된 패러미터는 ','를 구분자로 구분됩니다.<br>※ 참고사항<br>▪ 표현식에 추가된 패러미터는 선택된 필드에 추가된 필드와 동기화되어 표시됩니다.<br>▪ 행 삽입/삭제/변경 쿼리 다이얼로그에서 행 변경문을 드래그한 후 [패러미터 추가] 버튼을 클릭하면 드래그된 내용이 패러미터 마법사의 표현식에 추가되고, 선택된 필드와 동기화됩니다. |

# 쿼리 마법사 사용하기

쿼리 마법사는 디자인뷰 모드에서 사용할 수 있는 쿼리문 제작 도구로서 데이터베이스 밴더별로 SQL 문법을 정확히 모르는 사용자도 단계별로 손쉽게 쿼리문을 작성할 수 있도록 도와줍니다.

## 쿼리 마법사 설정 창

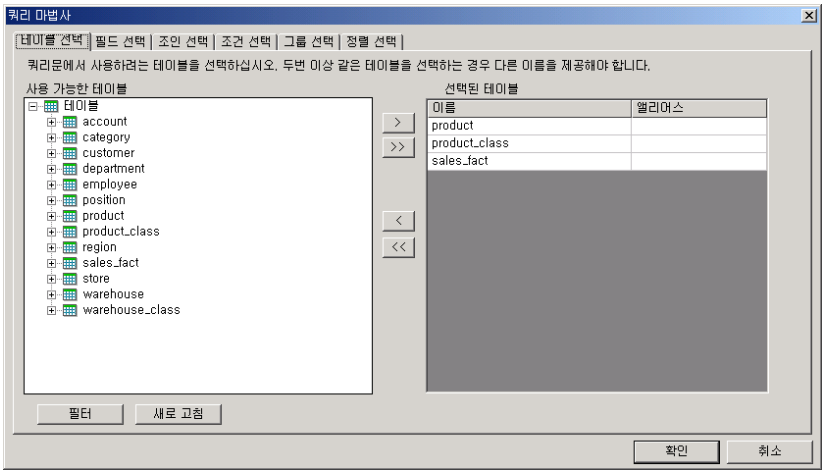
쿼리 마법사는 표준 SQL SELECT 문법에 대응하는 각각의 UI를 하나의 마법사 다이얼로그에 탭 형태로 제공합니다.

| 탭 | 설명 |
|---|---|
| **테이블 선택** | 데이터를 가져올 테이블을 선택합니다. 선택된 테이블들은 쿼리문의 From절에 추가됩니다. |
| **필드 선택** | 원하는 결과 필드들을 선택합니다. 해당 필드들은 Select절에 추가됩니다. |
| **조인 선택** | 테이블간의 조인 조건을 설정합니다. 해당 조건은 Where절에 추가됩니다. |
| **조건 선택** | 데이터를 가져오기 위한 추가 조건을 선택합니다. 해당 조건은 Where절에 추가됩니다. |
| **그룹 선택** | 데이터를 그룹핑하기 위한 기준 필드들을 선택하고 나머지 필드들에 대한 집합 함수를 선택합니다. 기준 필드들은 Group by절에 추가되며, Select절의 나머지 필드들은 집합 함수가 적용됩니다. |
| **정렬 선택** | 데이터를 정렬하기 위한 기준 필드들을 선택합니다. 해당 필드들은 Order by절에 추가됩니다. |

쿼리 마법사의 각 탭에서는 <Shift> 또는 <Ctrl>키를 이용하여 여러 개의 테이블 또는 데이터 필드를 선택할 수 있으며, 여러 개를 한꺼번에 선택할 때에는 테이블은 테이블끼리만 데이터 필드는 데이터 필드끼리만 선택할 수 있습니다.

- **테이블 선택**

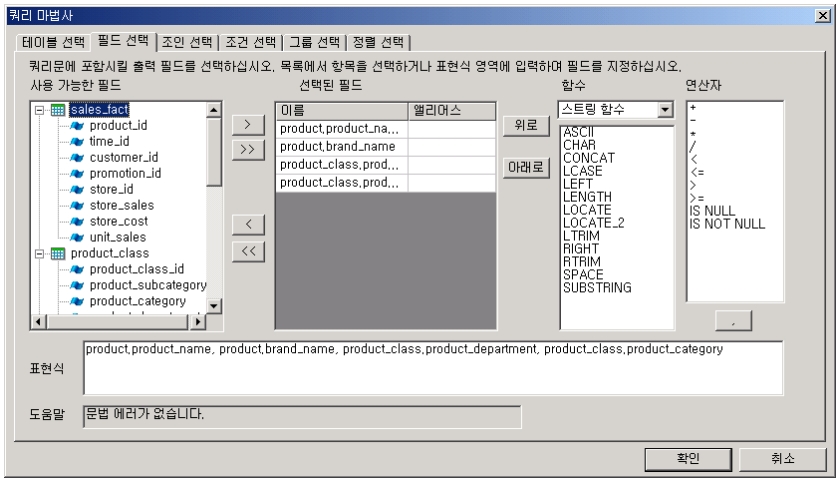
  현재 연결된 데이터베이스의 테이블을 리스트하고 쿼리에 사용할 테이블을 선택하는 탭입니다. 하나 이상의 테이블을 선택할 수 있습니다. 선택된 테이블명은 쿼리문의 From절에 추가됩니다.




| UI | 설명 |
|---|---|
| **사용 가능한 테이블** | 현재 연결된 데이터베이스 소스가 갖고 있는 테이블 리스트를 나타냅니다. |
| **필터** | 테이블 필터 창을 이용해 '사용 가능한 테이블' 리스트에 표시할 테이블을 필터링합니다.<br>※ 참고사항 : 필터 버튼을 클릭하면 '테이블 필터 다이얼로그'가 표시됩니다.<br>'테이블 필터 다이얼로그'에 대한 자세한 내용은 본 매뉴얼의 "III. 오즈 쿼리 디자이너 윈도우의 데이터 창, 리파지토리 창"을 참조하시기 바랍니다. |
| **새로 고침** | '사용 가능한 테이블' 창에 표시된 테이블 리스트을 갱신합니다. |
| **선택된 테이블** | 쿼리문에 사용될 테이블 리스트를 나타냅니다. '앨리어스'는 선택된 테이블에 별명을 붙일 때 사용하며, 선택된 테이블들을 구분할 수 있는 고유한 이름으로 입력합니다. |
| **> , >>** | '사용 가능한 테이블'에서 선택된 테이블을 '선택된 테이블'로 추가하거나 '사용 가능한 테이블'에 표시되어 있는 모든 테이블을 추가합니다. |
| **< , <<** | '선택된 테이블'에서 선택한 테이블을 삭제하거나 '선택된 테이블'에 추가된 모든 테이블을 삭제합니다. |

■ **필드 선택**

데이터 셋에 포함되는 데이터 필드를 지정하는 탭으로써 하나 이상의 필드를 쿼리문에 추가하거나 삭제할 수 있으며, 각종 함수 및 연산자를 이용해 사용자 정의 필드를 생성할 수 있습니다.



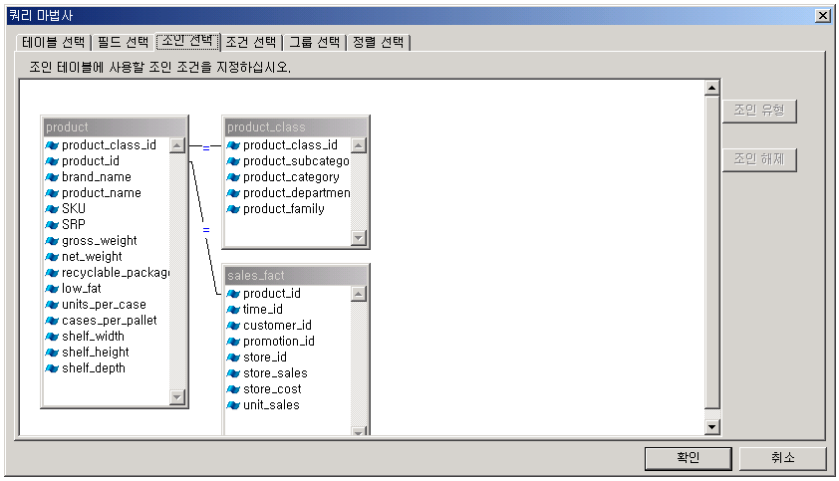

| UI | 설명 |
|---|---|
| **사용 가능한 필드** | 현재 쿼리문에 추가된 모든 테이블과 필드, 사용자 정의 필드, 사용자 패러미터, 다른 데이터 셋 필드 등을 표시합니다. |
| **선택된 필드** | 쿼리문에 추가된 필드 리스트를 나타냅니다. [앨리어스]는 선택된 필드에 별명을 붙일 때 사용합니다. 앨리어스를 설정할 경우 결과 필드명은 실제 필드명 대신 앨리어스명을 사용합니다. |
| **>** | '사용 가능한 필드'에 선택되어있는 필드를 '선택된 필드'에 추가합니다. |
| **> >** | '사용 가능한 필드'에 추가되어있는 모든 테이블의 모든 필드를 '선택된 필드'에 추가합니다. |
| **<** | '선택된 필드'에 선택되어있는 필드를 삭제합니다. |
| **< <** | '선택된 필드'에 추가되어있는 모든 필드를 삭제합니다. |
| **위로, 아래로** | '선택된 필드'에서 선택한 필드의 위치를 위 또는 아래로 이동하여 데이터 셋의 필드 순서를 변경할 수 있습니다. |
| **함수** | 사용자 정의 필드를 구성하기 위해 사용할 수 있는 함수 리스트를 나타냅니다. 옵션 설정에서 지정한 함수 리스트도 포함됩니다. 기본적으로 각 데이터베이스 벤더별로 JDBC 드라이버에서 리턴하는 표준 SQL 함수(스트링 함수, 수학 함수, 시스템 함수, 시간 함수, 집합 함수)를 보여줍니다. |
| **연산자** | 사용자 정의 필드를 구성하기 위해 사용할 수 있는 연산자 리스트를 나타냅니다. |
| **표현식** | 선택된 필드, 연산자 및 함수식을 포함한 텍스트를 표시합니다. |
| **,** | 표현식에 ','를 추가합니다. |
| **도움말** | 표현식의 문법 에러를 표시합니다. |

■ **조인 선택**

쿼리문에 사용될 테이블 간의 조인 조건을 설정하는 탭으로써, 드래그&드롭을 이용한 직관적인 조인 설정 기능을 지원합니다.

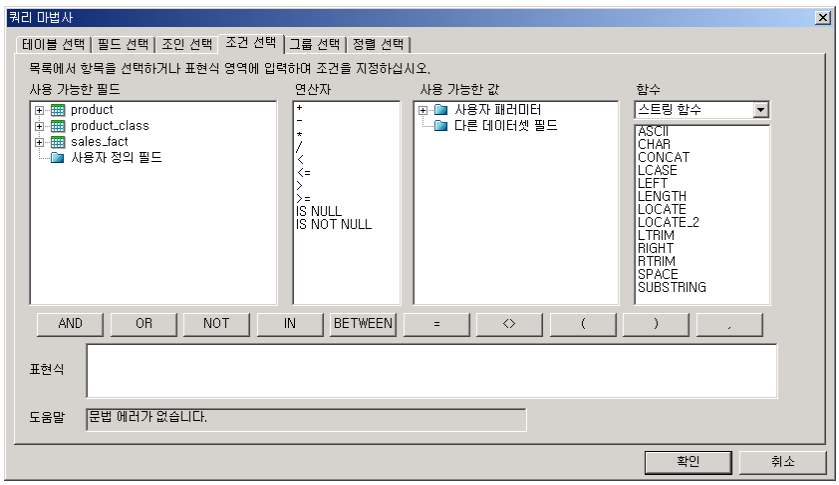
조인 설정은 쿼리문에 사용될 테이블이 두개 이상일 경우에만 적용되며, 테이블 간의 조인 조건을 설정하지 않을 경우 크로스 조인으로 연결됩니다.

조인 선택 탭의 자세한 사용 방법에 대해서는 "디자인뷰 모드로 쿼리 디자인하기"를 참고하시기 바랍니다.




■ **조건 선택**

쿼리 데이터 셋을 구성하기 위한 검색 조건을 지정하는 탭으로써 SQL의 Where절에 해당하는 조건문을 생성하는 기능을 제공합니다.

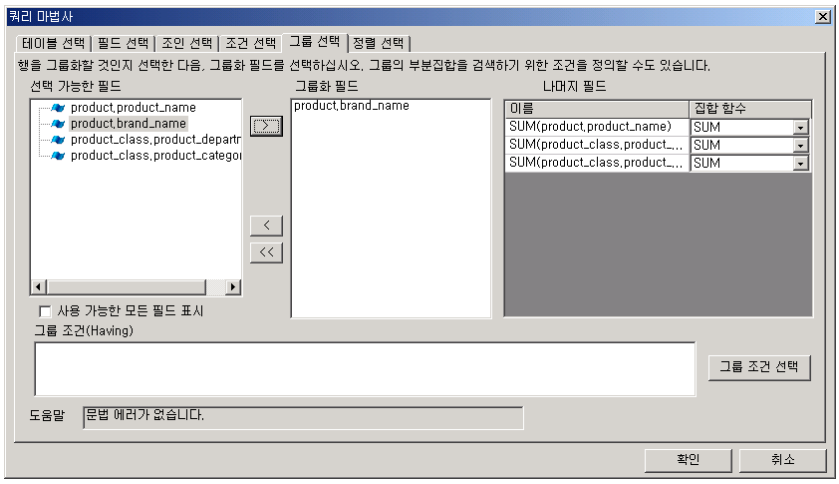



| UI | 설명 |
|---|---|
| **사용 가능한 필드** | 현재 쿼리문에 추가된 테이블의 필드 리스트를 나타냅니다. |

| | |
|---|---|
| **연산자** | 조건문을 구성하는데 사용할 수 있는 연산자 리스트를 나타냅니다. |
| **사용 가능한 값** | 조건문에 사용할 수 있는 사용자 패러미터 및 데이터 셋 리스트를 나타냅니다. |
| **함수** | 조건문을 구성하는데 사용할 수 있는 함수 리스트를 나타냅니다. 기본적으로 각 데이터베이스 벤더별로 JDBC 드라이버에서 리턴하는 표준 SQL 함수(스트링 함수, 수학 함수, 시스템 함수, 시간 함수, 집합 함수)를 보여줍니다. |
| **AND / OR /NOT/ IN /BETWEEN/ = /< >/ ( / ) / ,** | 조건문을 구성하는데 사용할 수 있는 논리 연산자를 나타냅니다. |
| **표현식** | 선택된 필드, 연산자 및 함수식을 포함한 텍스트를 표시합니다. |
| **도움말** | 표현식의 문법 에러를 표시합니다. |

■ **그룹 선택**

데이터 셋의 그룹조건을 지정하는 탭으로써 SQL의 Group by절에 해당하는 조건을 생성하는 기능을 제공합니다.

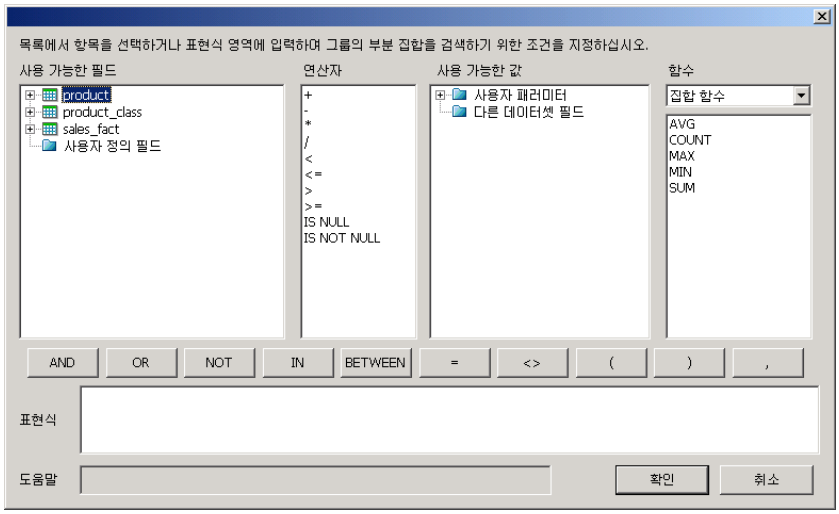



| UI | 설명 |
|---|---|
| **선택 가능한 필드** | 그룹 조건에 사용될 수 있는 필드 리스트를 나타냅니다. |
| **사용 가능한 모든 필드 표시** | 그룹 조건에 사용될 수 있는 필드 리스트를 나타내는 창인 '선택 가능한 필드' 창에 선택된 테이블에 대한 모든 필드를 표시할지 또는 데이터 셋에 추가된 필드만 표시할지 여부를 설정합니다. |

| | |
|---|---|
| **>** | '사용 가능한 필드' 창에서 선택한 필드를 '선택된 필드'창에 추가합니다. |
| **<, < <** | '선택된 필드' 창에서 선택한 필드를 삭제합니다. |
| **그룹화 필드** | 그룹 조건에 추가된 필드 리스트를 나타냅니다. |
| **나머지 필드** | 그룹 조건에 추가되지 않은 나머지 필드의 집합 함수를 설정합니다(AVG, COUNT, MAX, MIN, SUM) |
| **그룹 조건(Having)** | 그룹 조건 입력 및 편집 창입니다. |
| **그룹 조건 선택** | 조건 선택 창을 활성화하기 위한 버튼입니다. 조건 선택 창에서 만들어진 조건은 그룹 조건 편집 창으로 자동 추가됩니다. |
| **도움말** | 표현식의 문법 에러를 표시합니다. |

- **그룹 조건 선택**

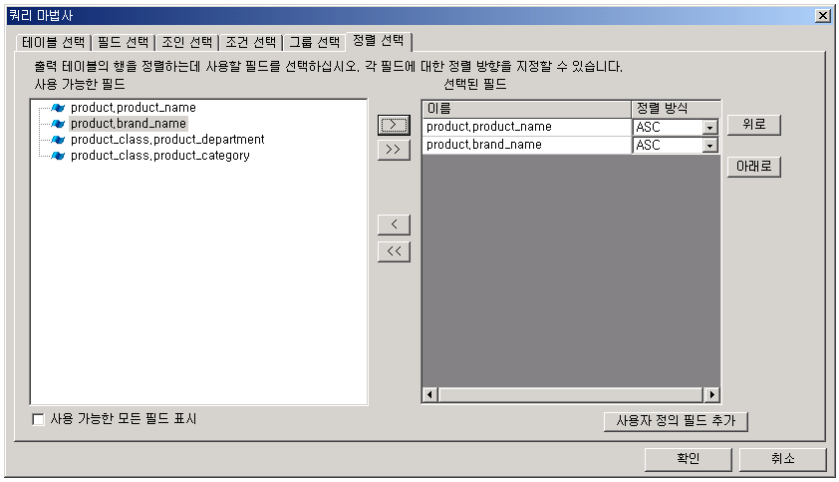
  [그룹 조건 선택] 버튼을 클릭하면 아래와 같은 다이얼로그가 표시됩니다.




  각 설명은 조건 선택 항목과 동일하며, 함수 부분에 집합 함수가 표시됩니다.

■ **정렬 선택**

쿼리 데이터 셋의 정렬조건을 지정하는 탭으로써 SQL의 Order by절에 해당하는 조건을 생성하는 기능을 제공합니다.




| UI | 설명 |
|---|---|
| **사용 가능한 필드** | 정렬 조건에 사용될 수 있는 필드 리스트를 나타냅니다. |
| **사용 가능한 모든 필드 표시** | 그룹 조건에 사용될 수 있는 필드 리스트를 나타내는 창인 '선택 가능한 필드' 창에 선택된 테이블에 대한 모든 필드를 표시할지 또는 데이터 셋에 추가된 필드만 표시할지 여부를 설정합니다. |
| **> , >>** | '사용 가능한 필드' 창에서 선택한 필드를 '선택된 필드' 창에 추가합니다. |
| **< , <<** | '선택된 필드' 창에서 선택한 필드를 삭제합니다. |
| **선택된 필드** | 정렬 조건에 추가된 필드 리스트를 나타냅니다. 각 필드별 오름차순/내림차순 정렬을 설정할 수 있으며, 각 필드간의 정렬 우선순위는 리스트의 아래로 갈수록 낮아집니다. |
| **위로, 아래로** | '선택된 필드' 창에서 선택한 필드의 위치를 위 또는 아래로 이동하여 필드의 우선순위를 변경할 수 있습니다. |
| **사용자 정의 필드** | 고급 표현식 창을 이용하여 새로운 정렬 필드를 추가합니다. '사용자 정의 필드 추가' 창에 대한 자세한 설명은 앞의 '조건 선택 탭 설명'부분을 참조하시기 바랍니다. |

# 수식필드 추가하기

생성된 데이터 셋에 수식필드를 추가할 수 있습니다. 쿼리 데이터 셋, UDS 데이터 셋, XML 데이터 셋, CSV 데이터 셋, 다이나믹 데이터 셋에서 추가 가능합니다.

## 수식 필드 추가

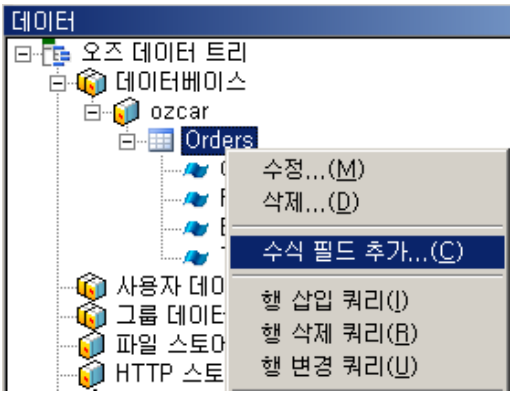
수식 필드는 사용할 스크립트 타입에 따라 자바 스크립트 타입 또는 오즈 스크립트 타입으로 추가할 수 있습니다.
본 장에서는 두 개의 데이터 필드 값을 연산하여 수식 필드에 값에 설정하는 방법을 예를 들어 살펴보겠습니다.

※ 주의사항 : 수식 필드 생성 시 수식 필드에 사용할 스크립트 타입을 먼저 선택하여야 하며, 스크립트 타입은 수식 필드를 추가한 후에는 변경할 수 없습니다.

※ 참고사항 : 쿼리문에서 수식 필드를 이용하여 조건문을 설정할 경우 쿼리문 실행 시 '패러미터 입력 다이얼로그'에서 수식 필드 값이 항상 빈 값으로 표시됩니다.

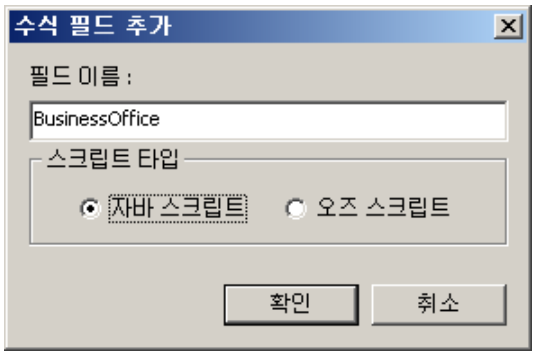
■ 자바 스크립트 타입

➢ 오즈 데이터 트리에서 데이터셋을 선택한 후 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [수식 필드 추가] 메뉴를 클릭합니다.

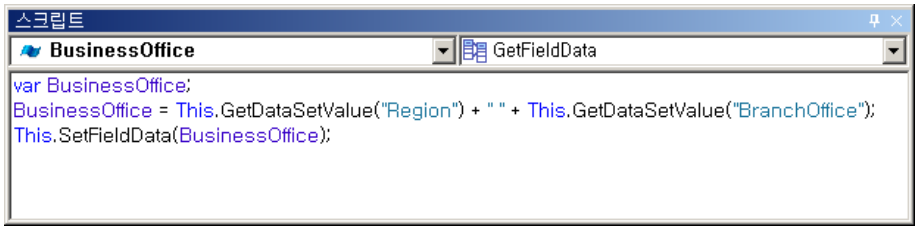



➢ [수식 필드 추가] 메뉴를 클릭하여 나타난 '수식 필드 추가 다이얼로그'에서 필드 이름을 입력하고 "스크립트 타입"을 "자바 스크립트"로 선택한 후 [확인] 버튼을 클릭하여

수식 필드를 추가합니다.

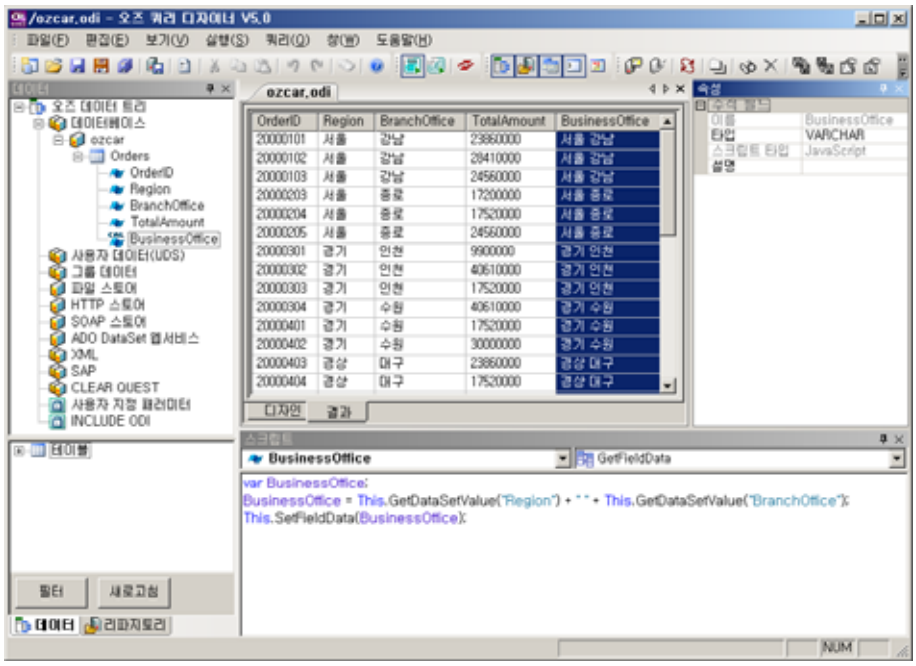



- 추가된 수식 필드를 선택하면 표시되는 스크립트 편집 창에 데이터 필드 중 Region과 BranchOffice를 더하여 수식 필드의 값으로 설정는 자바 스크립트를 아래와 같이 입력합니다.




```
var BusinessOffice;
BusinessOffice = This.GetDataSetValue("Region") + " " +
This.GetDataSetValue("BranchOffice");
This.SetFieldData(BusinessOffice);
```

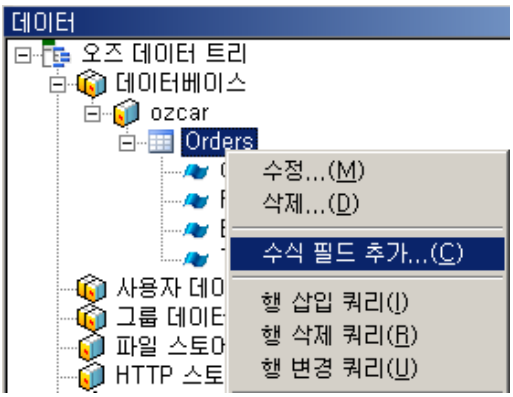
- 데이터셋의 쿼리문을 실행하면 수식 필드에 자바 스크립트로 설정한 값이 표시됩니다.



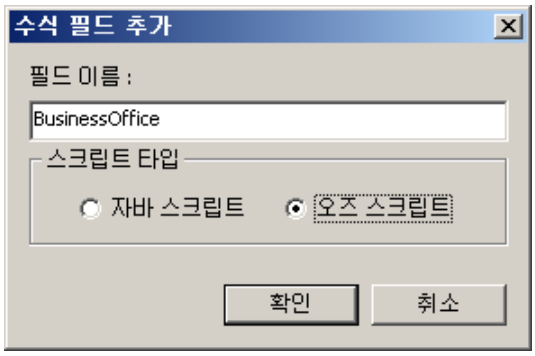
■ 오즈 스크립트 타입

수식 필드에 작성할 오즈 스크립트의 문법 및 함수는 'OZ Script Guide'를 참조하시기 바랍니다.

➢ 오즈 데이터 트리에서 데이터셋을 선택한 후 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [수식 필드 추가] 메뉴를 클릭합니다.

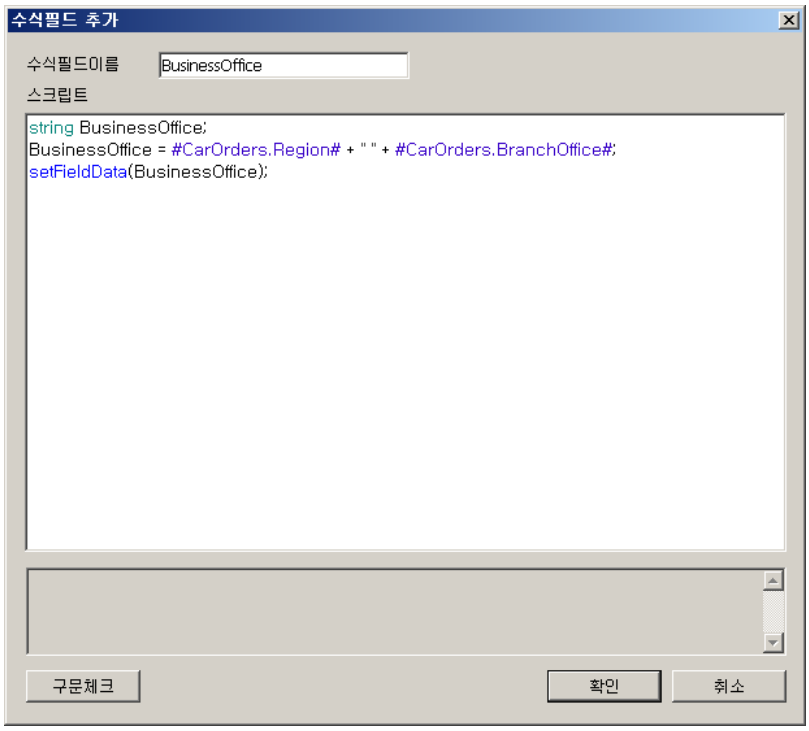



➢ [수식 필드 추가] 메뉴를 클릭하여 나타난 '수식 필드 추가 다이얼로그'에서 필드 이름을 입력하고 "스크립트 타입"을 "오즈 스크립트"로 선택한 후 [확인] 버튼을 클릭하면



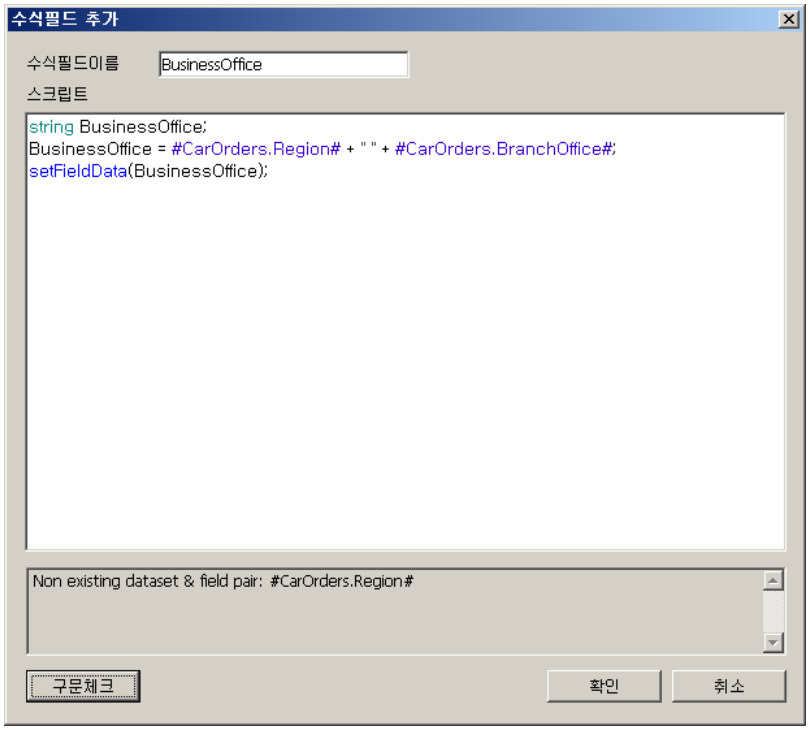

'수식필드 추가 다이얼로그'가 나타납니다.

스크립트 입력 창에 데이터 필드 중 Region과 BranchOffice를 더하여 수식 필드의 값으로 설정하는 오즈 스크립트를 아래와 같이 입력합니다.



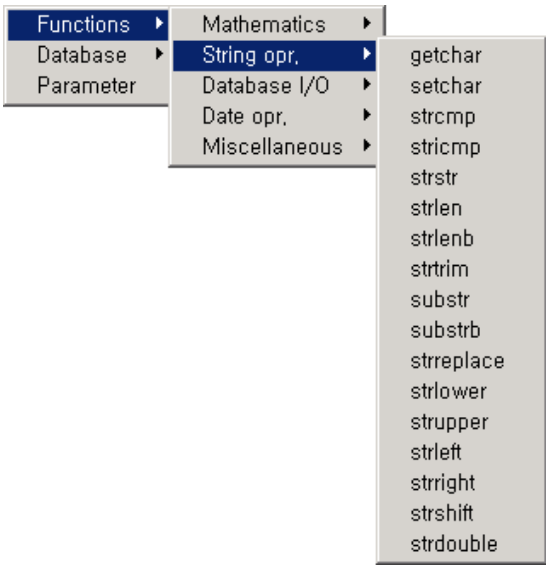

```
string BusinessOffice;
BusinessOffice = #CarOrders.Region# + " " + #CarOrders.BranchOffice#;
setFieldData(BusinessOffice);
```

- 스크립트 입력 창에 스크립트를 입력하고 [구문체크] 버튼을 클릭하면 적절한 스크립트인지를 확인할 수 있습니다.



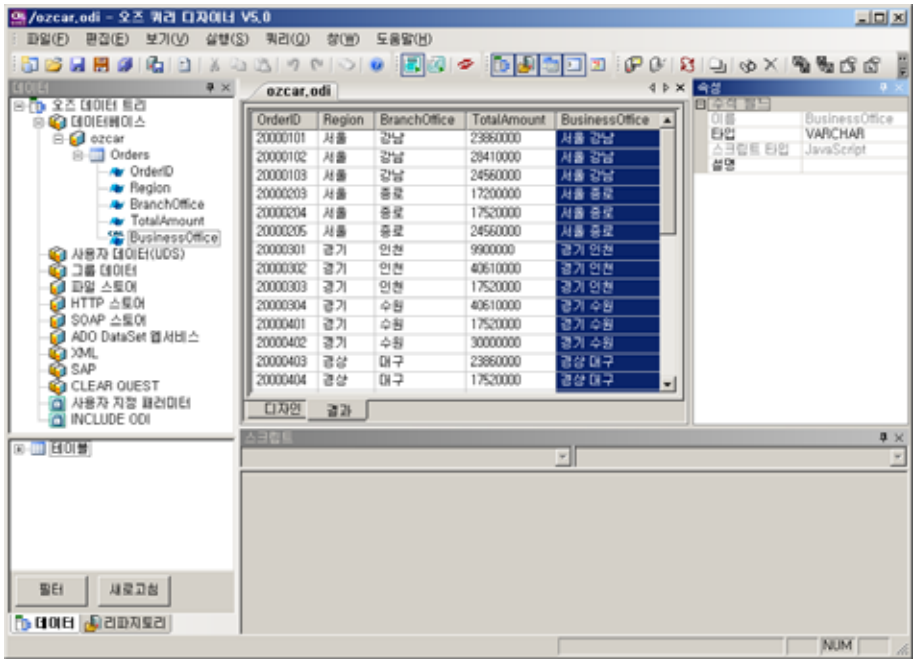

| UI | 설명 |
|---|---|
| **수식필드이름** | 추가할 수식 필드의 이름을 설정합니다. |
| **스크립트** | 오즈 스크립트를 작성합니다.<br>스크립트에 대한 자세한 내용은 "스크립트 가이드"를 참조하시기 바랍니다.<br>※ 주의사항 : 수식필드에서는 setFieldData(변수이름) 스크립트를 반드시 사용하여야 하며, setQueryString(변수이름)을 사용할 경우에는 문법 에러는 발생하지 않으나, 쿼리 수행시 에러가 발생합니다. |
| **구문체크** | 작성한 스크립트의 문법이 맞는지 여부를 체크합니다. |

스크립트 창에서 마우스 오른쪽 버튼을 클릭하면 스크립트 작성에 도움이 되는 함수/명령어 등이 나타납니다. 원하는 정보나 함수 등을 선택하면 해당 함수가 표현되며, 디버그 창에 그 이용에 대한 설명이 나타납니다.




- Functions : Mathematics, Sting opr., Database I/O, Date opr., Miscellaneous에 사용되는 함수들을 보여줍니다.
- Database : 데이터 셋 리스트와 데이터 셋별 필드 정보를 보여줍니다.
- Parameter : 보고서에서 사용자가 정의한 패러미터 변수들을 보여줍니다.

➢ 데이터셋의 쿼리문을 실행하면 수식 필드에 오즈 스크립트로 설정한 값이 표시됩니다.

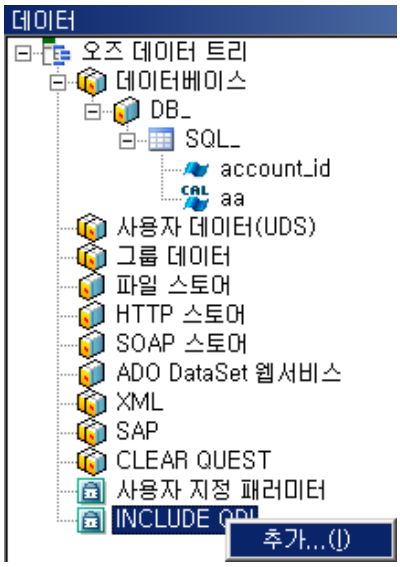


# INCLUDE ODI를 통한 필드 추가하기

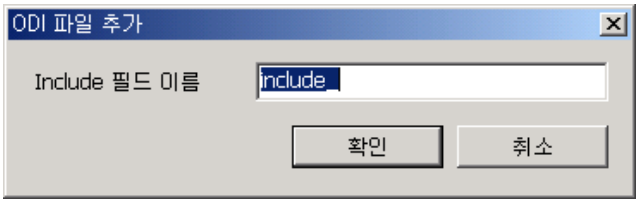
이미 작성하여 서버 리파지토리 혹은 로컬 작업 폴더에 저장된 ODI 파일을 가져와서 현재 편집 중이 ODI 파일에 추가하여 사용할 수 있습니다.

## Include 필드 추가

**Step 1** 데이터 창의 [INCLUDE ODI]에서 마우스 오른쪽 버튼을 클릭하여 나타나는 팝업 메뉴에서 [추가] 메뉴를 클릭합니다.



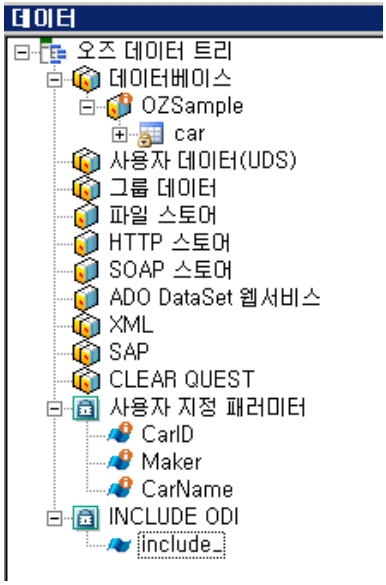

**Step 2** 'Include 필드 이름'을 입력하는 다이얼로그가 표시됩니다.




'Include 필드 이름'을 입력한 후 확인 버튼을 클릭하면 ODI 파일이 추가됩니다.

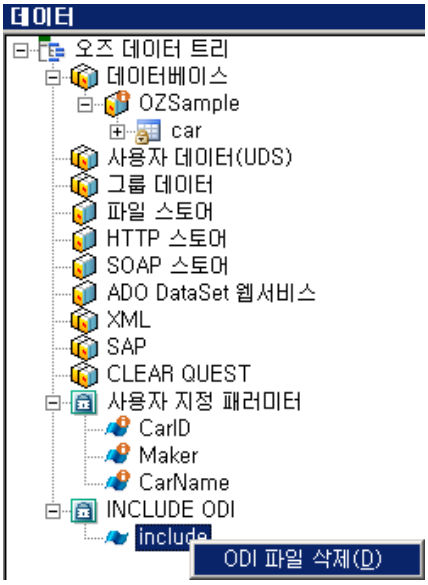
**Step 3**

삽입되는 ODI 파일에 생성되어 있는 스토어 및 데이터 셋, 필드, 사용자 지정 패러미터 등이 Include 필드로 추가됩니다. Include 필드에 의해 추가된 스토어 및 데이터 셋, 필드 등은 수정할 수 없으며, [ , , ] 아이콘으로 구분됩니다.




## Include 필드 삭제

삭제하고자 하는 Include 필드에서 마우스 오른쪽 버튼을 클릭하여 나타나는 팝업 메뉴에서 [ODI 파일 삭제] 메뉴를 클릭합니다.




해당 Include 필드를 삭제할 것인지 확인하는 창이 나타나고, [예] 버튼을 클릭하면 Include 필드로 추가되었던 스토어 및 데이터 셋, 필드, 사용자 지정 패러미터가 모두 삭제됩니다.

## Include 필드 추가 제약 사항

"Include ODI"를 이용하여 현재 편집 중인 ODI 파일에 새로운 ODI 파일을 추가할 경우 다음과 같은 제약 사항이 있습니다.

- 현재 편집 중인 ODI와 같은 데이터 소스를 사용하는 ODI는 추가할 수 없습니다.
- 현재 편집 중인 ODI와 추가하려는 ODI의 스토어명, 데이터 셋명, 필드명이 중복되는 경우에는 추가할 수 없습니다.
- ODI 추가는 한 단계로만 가능합니다. 즉, A.odi 파일이 추가된 B.odi 파일을 C.odi에 추가할 경우 C.odi에는 A.odi의 데이터 셋 정보는 추가되지 않고, B.odi의 고유한 데이터 셋 정보만 추가됩니다.

# 마스터/디테일 셋 디자인하기

하나 이상의 쿼리 데이터 셋은 마스터/디테일 관계로 연결될 수 있습니다.

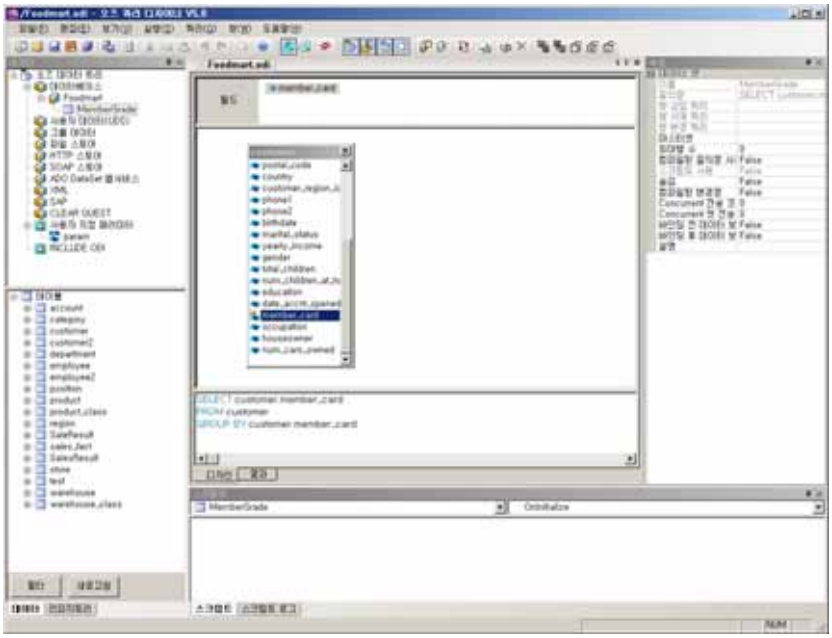
마스터 셋 : 고객 등급(MemberGrade)

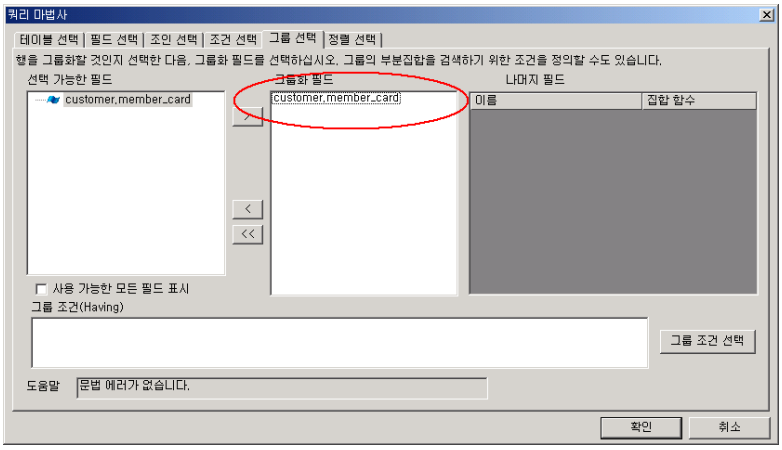
디테일 셋 : 고객 정보(Customer)

**Step 1**

**마스터 셋 작성**

쿼리 마법사를 이용하여 고객 등급를 그룹핑한 결과 데이터 셋을 생성합니다.





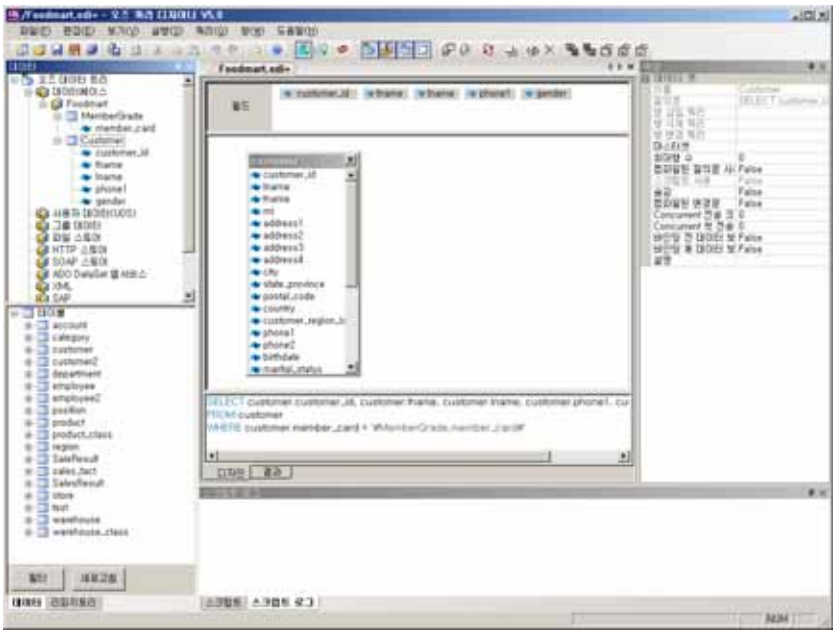
작성된 쿼리문은 아래와 같습니다.

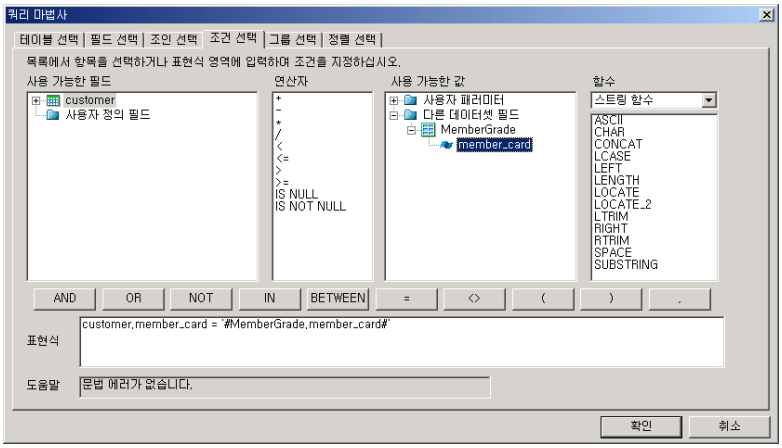
```
SELECT customer.member_card FROM customer
GROUP BY customer.member_card
```

**Step 2**

### 디테일 셋 작성

쿼리 마법사를 이용하여 고객 등급에 맞게 고객 정보 데이터를 조회하는 데이터 셋을 생성합니다.





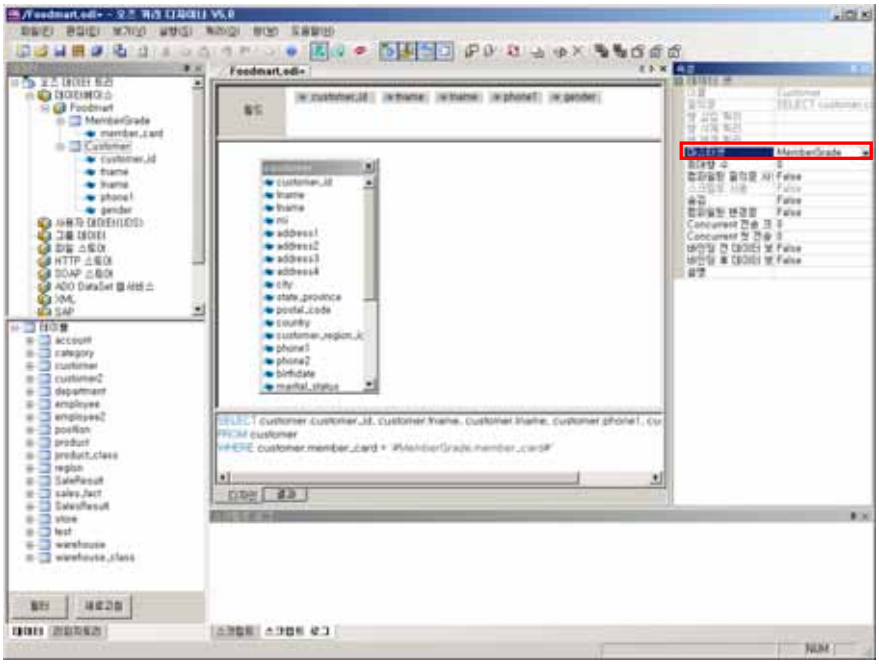

작성된 쿼리문은 아래와 같습니다.

```
SELECT customer.customer_id, customer.fname, customer.lname,
customer.phone1, customer.gender FROM customer
WHERE customer.member_card = '#MemberGrade.member_card#'
```

**Step 3**

**마스터 셋 관계 설정**

데이터 창에서 디테일 셋으로 설정할 데이터 셋(Customer)을 선택한 후 속성 창에서 마스터셋 속성 값을 마스터셋으로 설정할 데이터 셋 이름(MemberGrade)으로 설정합니다.



# 멀티 데이터 셋 디자인하기

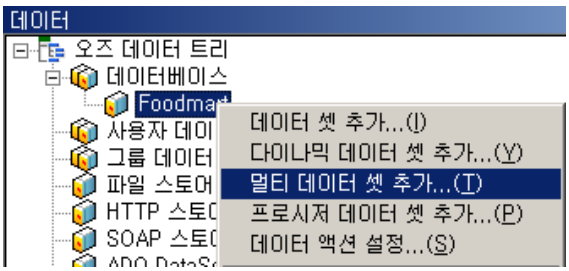
데이터 셋에서 프로시저나 여러 개의 쿼리문을 입력하여 복수 개의 결과 셋을 생성할 수 있습니다.

※ 주의사항

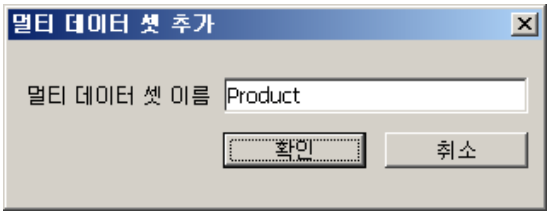
- 멀티 데이터 셋 기능은 MSSQL, SYBASE만 지원합니다.
- 멀티 데이터 셋 기능은 매뉴얼 모드만 지원합니다.
- 하위 데이터 셋의 마스터셋 속성 값은 따로 설정할 수 없으며 멀티 데이터 셋의 마스터 셋 속성 값을 따릅니다

## 멀티 데이터 셋 작성

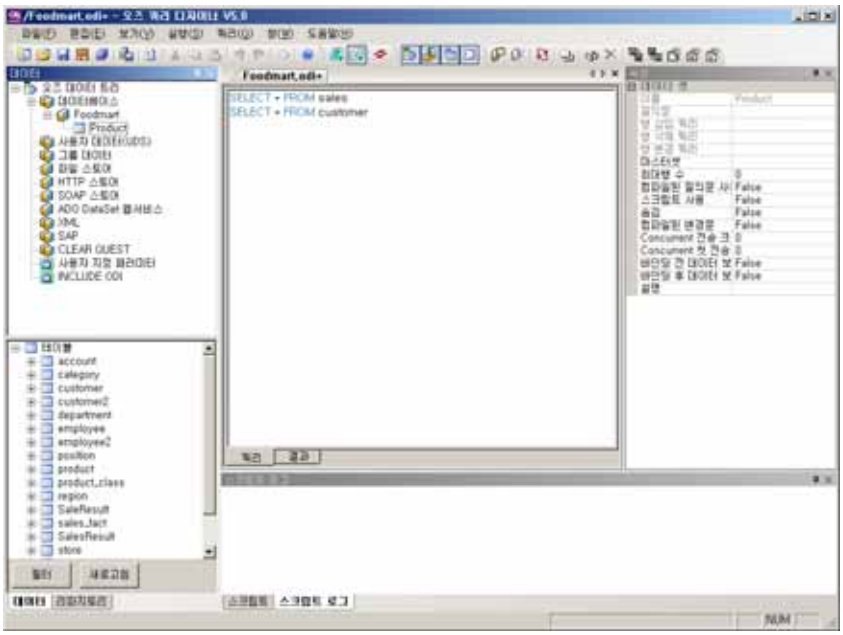
**Step 1** 데이터 창에서 "데이터베이스 스토어"를 추가한 후 "데이터베이스 스토어"를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [멀티 데이터 셋 추가] 메뉴를 클릭합니다.




**Step 2** '멀티 데이터 셋 추가 다이얼로그'에서 멀티 데이터 셋 이름을 입력합니다.



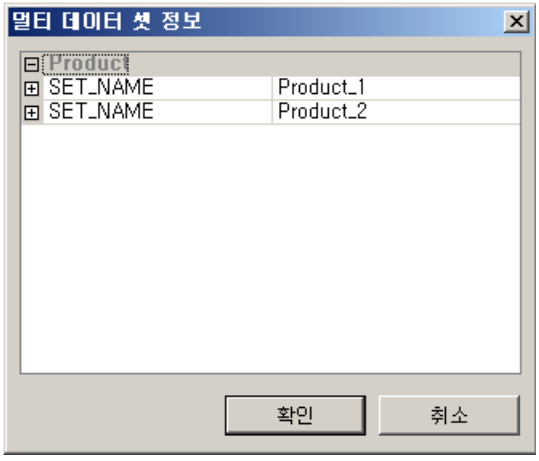
**Step 3** 추가한 멀티 데이터 셋의 쿼리 창에 프로시저나 여러 개의 쿼리문을 입력합니다.



작성된 쿼리문은 아래와 같습니다.

```
SELECT * FROM sales
SELECT * FROM customer
```

**Step 3** [실행] 메뉴의 [쿼리문 실행] 메뉴를 클릭하거나 툴바에서 쿼리문 실행( )아이콘을 클릭하거나 또는 키보드의 <F5> 키를 눌러 쿼리문을 실행하면 '멀티 데이터 셋 정보 다이얼로그'가 표시되며 해당 다이얼로그에 결과 셋의 개수만큼 데이터 셋의 정보가 표시됩니다.

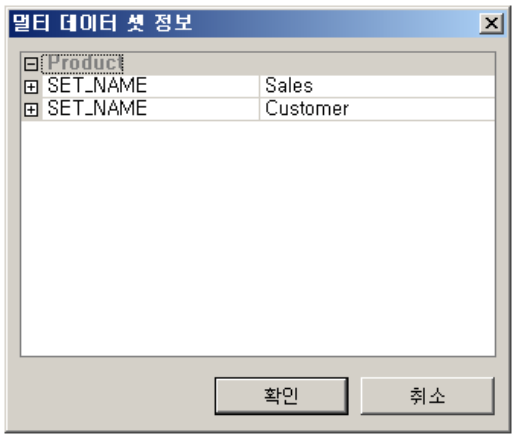


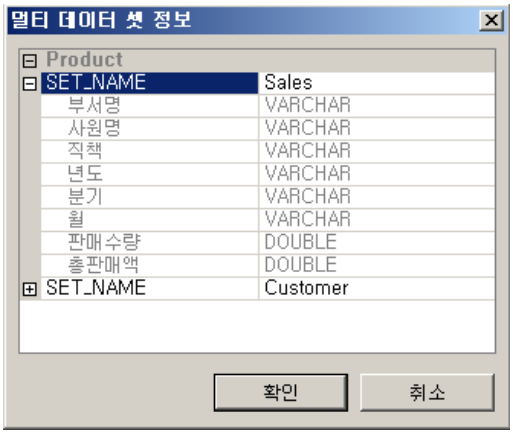
※ 참고사항

- '멀티 데이터 셋 정보 다이얼로그'는 입력한 쿼리문을 최초로 실행하거나 기존의 쿼리문을 실행하여 생성되는 결과 셋의 개수가 변경되었을 경우에만 표시됩니다.
- 입력한 쿼리문을 최초로 실행하여 '멀티 데이터 셋 정보 다이얼로그'가 표시될 경우 데이터 셋의 기본 이름은 "*멀티 데이터 셋 이름*_#"의 형태로 표시됩니다.
- 결과 셋의 개수가 추가되어 '멀티 데이터 셋 정보 다이얼로그'가 표시될 경우 데이터 셋의 이름은 기존에 생성된 데이터 셋 이름이 먼저 표시되고 추가된 데이터 셋의 이름은 "멀티 데이터 셋 이름_#"의 형태로 표시됩니다.

**Step 3**

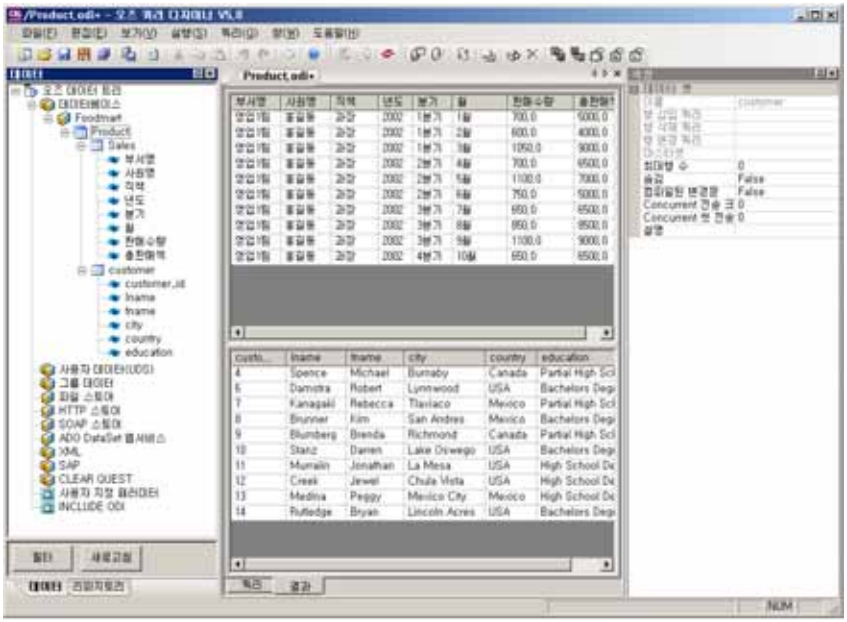
'멀티 데이터 셋 정보 다이얼로그'에서 "SET_NAME" 값을 변경하여 멀티 데이터 셋의 하위 데이터 셋 이름을 변경할 수 있으며, 각 데이터 셋의 트리를 펼치면 해당 데이터 셋의 필드명 및 필드 타입이 표시됩니다.



**Step 3**

'멀티 데이터 셋 정보 다이얼로그'에서 [확인] 버튼을 클릭하면 데이터 창에 생성된 데이터 셋과 필드가 표시되며 결과 창에 아래 그림과 같이 데이터 셋 별로 데이터가 표시됩니다.



## 멀티 데이터 셋 쿼리문 입력 방법

멀티 데이터 셋의 쿼리 창에 프로시저나 여러 개의 쿼리문을 입력하여 멀티 데이터 셋을 생성합니다. 여러 개의 쿼리문을 입력하는 방법은 다음과 같습니다.

- 입력 방법 1 : 하나의 쿼리문을 입력한 후 공백으로 구분하여 다음 쿼리문을 입력합니다.

```
SELECT * FROM sales SELECT * FROM customer
```

- 입력 방법 2 : 하나의 쿼리문을 입력한 후 줄바꿈하여 다음 쿼리문을 입력합니다.

```
SELECT * FROM sales
SELECT * FROM customer
```

# 다이나믹 데이터 셋 디자인하기

## 다이나믹 쿼리 개요

다이나믹 쿼리란 데이터에 대한 필드 정보 등이 운영 시점에 동적으로 결정되도록 하는 쿼리를 말합니다. 이는 동적으로 변하는 필드 정보로 인해 디자인 시점에 필드 정보를 설정할 수 없는 경우에 다이나믹 쿼리를 이용해 구현할 수 있습니다. 따라서 사용자가 표시할 필드나 조건까지 선택해야 하는 경우 다이나믹 쿼리를 사용할 수 있습니다.

## 다이나믹 데이터 셋 개요

다이나믹 데이터 셋이란 데이터베이스, 사용자 데이터, 파일/HTTP 스토어, XML DTD 스토어를 통해 가져온 데이터를 다이나믹 쿼리 리포트를 구성하기 위한 형태로 변경해 주는 2차 데이터 셋입니다.
다이나믹 데이터 셋은 쿼리를 통해서 만들어지는 동적인 필드 정보를 저장하는 대신 모든 쿼리 결과를 ROWIDX, COLUMN, VALUE 세가지 고정 항목으로 저장하고, 실행 시점에 이들 세가지 고정 항목으로부터 동적인 필드 정보를 찾아내 재구성하는 기능을 제공합니다.
따라서 오즈 개발자는 다이나믹 데이터 셋의 세가지 고정 필드를 이용해 보고서나 오즈 애플리케이션을 디자인하고, 보고서나 오즈 애플리케이션이 실행될 때 동적으로 생성되는 쿼리 결과로 보고서나 오즈 애플리케이션을 자동 확장하도록 할 수 있습니다.

다이나믹 데이터 셋이 생성하는 세가지 항목의 의미는 다음과 같습니다.

- ROWIDX : 실제 데이터 셋의 레코드 번호
- COLUMN : 실제 데이터 셋의 필드 이름
- VALUE : 해당 필드의 데이터 값

예를 들어 테이블 1과 같은 데이터 셋에 다이나믹 데이터 셋을 적용하면 테이블 2와 같이 ROWIDX, COLUMN, VALUE 필드를 갖는 새로운 데이터 셋이 생성되는 것을 알 수 있습니다.
다시 말해, 필드와 해당 정보를 나열하는 일반 데이터 셋에 다이나믹 데이터 셋을 적용하면, Row Index별로 필드 이름과 그 값을 순서대로 저장하는 형태의 새로운 데이터 셋이 생성됩니다.

■ 테이블 1 – 원본

| Region | Sales2000 | Sales2001 |
|---|---|---|
| 캐나다 | 157000 | 890235 |
| 동부 | 5234991 | 5432123 |
| 중서부 | 1234789 | 1123454 |
| 남부 | 4563300 | 4992100 |
| 서부 | 3551090 | 3901234 |

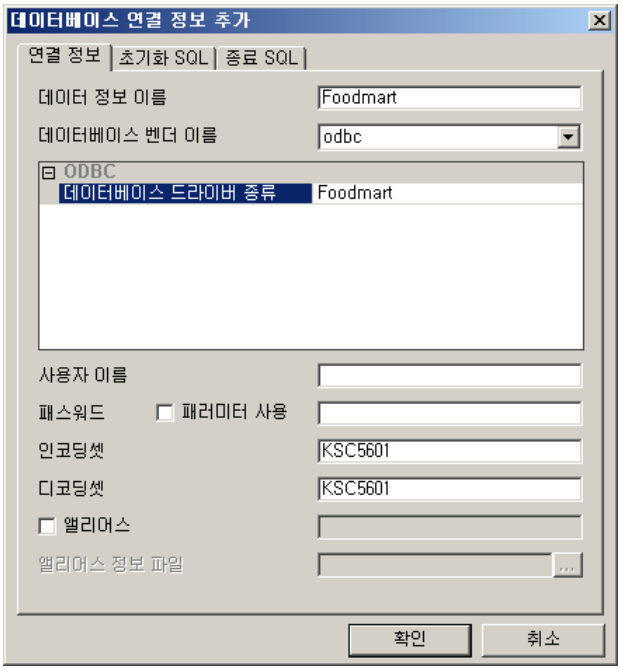
■ 테이블 2 – 다이나믹 데이터 셋을 이용하여 만든 테이블

| ROWIDX | COLUMN | VALUE |
|---|---|---|
| 0 | Region | 캐나다 |
| 0 | Sales2000 | 257000 |
| 0 | Sales2001 | 890235 |
| 1 | Region | 동부 |
| 1 | Sales2000 | 5234991 |
| 1 | Sales2001 | 5432123 |
| 2 | Region | 중서부 |
| 2 | Sales2000 | 1234789 |
| 2 | Sales2001 | 1123454 |
| 3 | Region | 남부 |
| 3 | Sales2000 | 4563300 |
| 3 | Sales2001 | 4992100 |
| 4 | Region | 서부 |
| 4 | Sales2000 | 3551090 |
| 4 | Sales2002 | 3901234 |

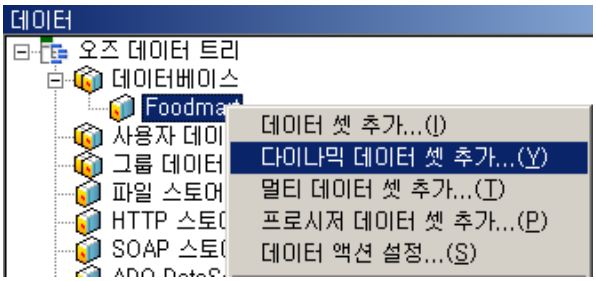
## 다이나믹 데이터 셋 작성

다음은 데이터베이스 소스를 이용하여 다이나믹 데이터 셋을 만드는 과정을 설명합니다.

**Step 1** 데이터 창의 [데이터베이스]를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [스토어 추가] 메뉴를 클릭하여 아래 그림처럼 데이터베이스 연결 정보를 설정하여 데이터베이스 스토어를 추가합니다.



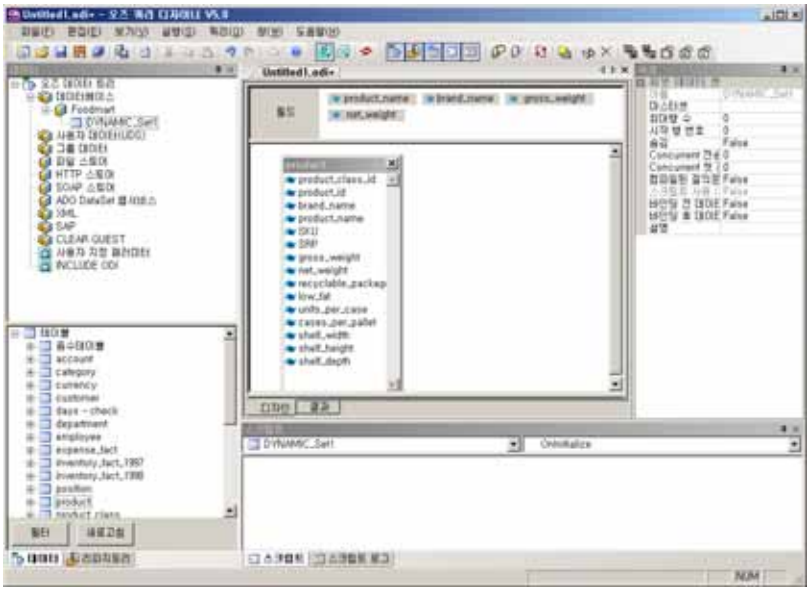

**Step 2** 추가된 데이터베이스 스토어를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [다이나믹 데이터 셋 추가] 메뉴를 클릭합니다.



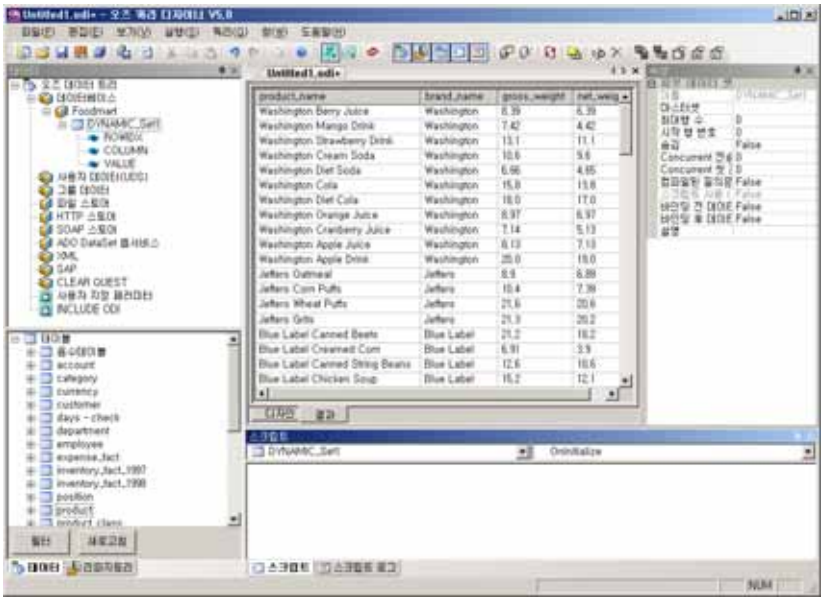

**Step 3** '다이나믹 데이터 셋 추가 다이얼로그'에서 데이터 셋 이름과 시작 행 번호를 입력한 후 [확인] 버튼을 클릭하여 다이나믹 데이터 셋을 추가합니다.




**Step 4** 쿼리 디자이너의 쿼리 창을 디자인 뷰 모드 또는 매뉴얼 모드로 변경하여 쿼리문을 디자인합니다.



**Step 5** 쿼리문 실행 아이콘( )을 클릭하여 쿼리를 실행하고 다이나믹 데이터 셋에 추가된 필드 정보를 확인합니다. 다이나믹 데이터 셋의 필드는 실제 테이블의 필드명이 아닌 ROWIDX, COLUMN, VALUE로 추가됩니다.



# 프로시저 데이터 셋 디자인하기

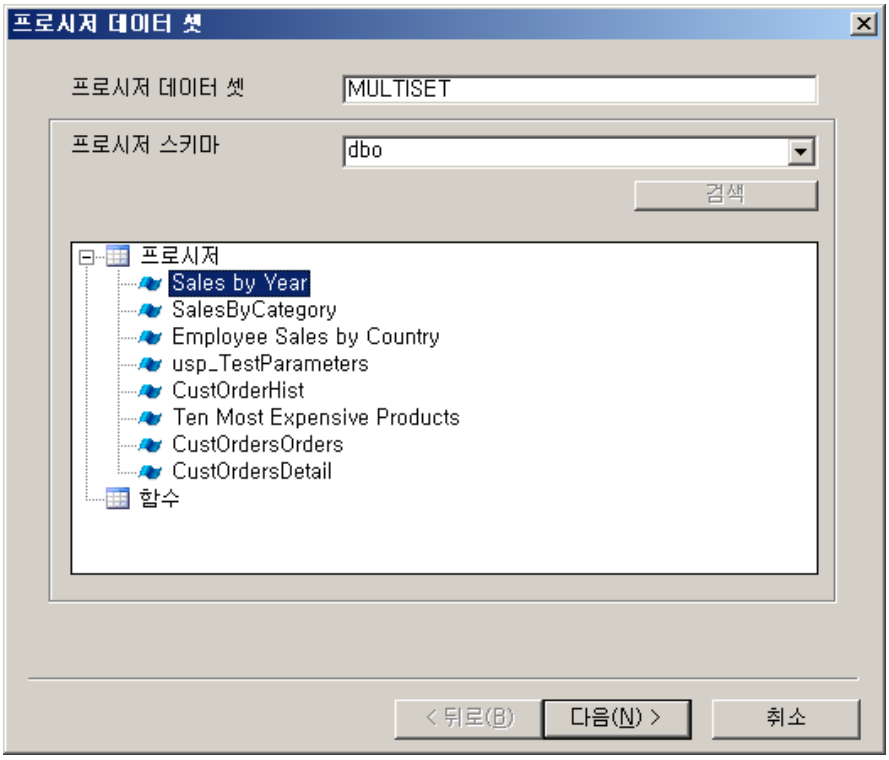
데이터 셋에서 프로시저를 설정하여 복수 개의 결과 셋을 생성할 수 있습니다.

※ 주의사항

- 프로시저 데이터 셋 기능은 로컬 서버 타입이 자바일 경우에는 JDBC에서 프로시저를 지원하는 Vendor에서만 지원하며, 닷넷일 경우에는 MSSQL, ORACLE에서만 지원합니다.
- 프로시저 기능이 지원되지 않는 DB 벤더를 사용하여 프로시저 데이터 셋을 추가할 경우 '프로시저 데이터 셋 다이얼로그'에서 [검색] 버튼을 클릭하면 에러 메시지가 표시됩니다.

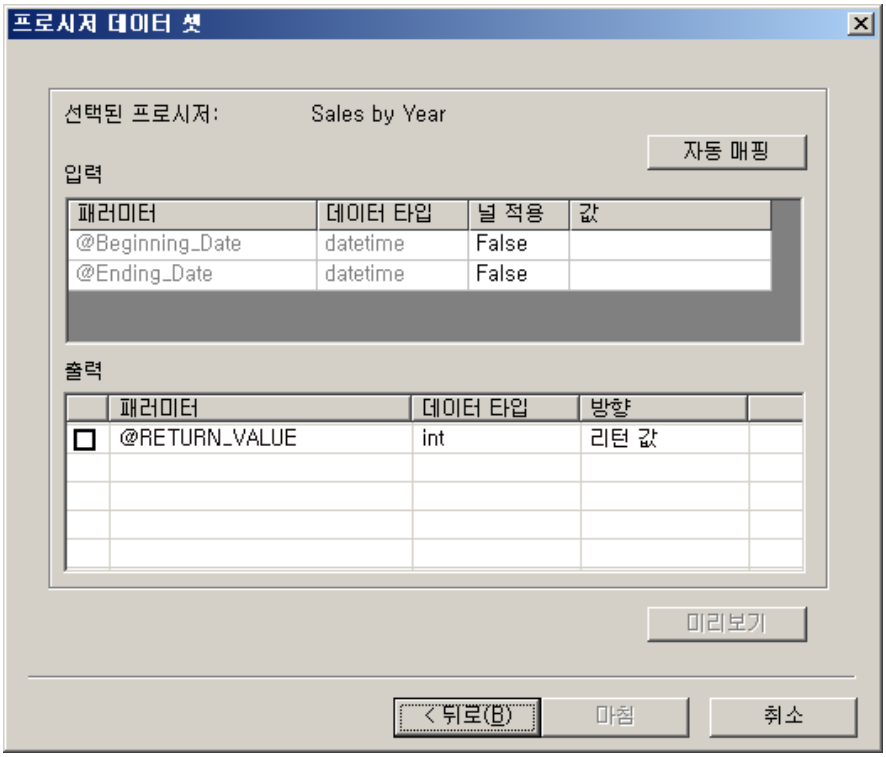
## 프로시저 데이터 셋 설정 다이얼로그

프로시저를 데이터 셋으로 설정할 때 사용하는 다이얼로그입니다.



| 항목 | 설명 |
|---|---|
| **프로시저 데이터 셋** | 프로시저 데이터 셋의 이름으로 사용자가 임의로 설정합니다. |
| **프로시저 스키마** | 프로시저를 선택할 스키마 목록을 표시합니다.<br>기본 값으로 데이터베이스 스토어에 연결된 사용자의 스키마가 표시되며, 사용자가 스키마 이름을 직접 입력하여 설정할 수도 있습니다. |
| **검색** | 선택한 스키마의 프로시저를 가져와 트리 형태로 표시합니다. |

프로시저를 선택한 후 [다음] 버튼을 클릭합니다.

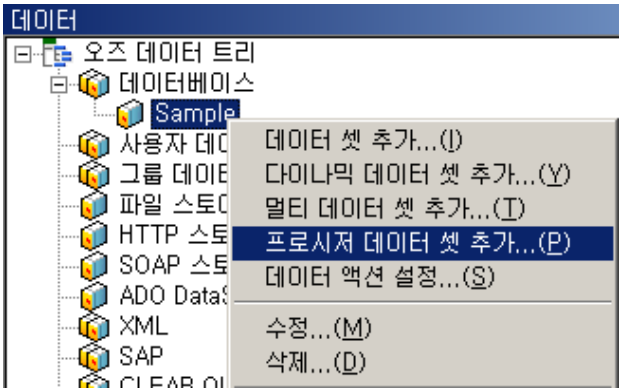



| 항목 | 설명 | |
|---|---|---|
| **선택된 프로시저** | 선택한 프로시저 이름이 표시됩니다. | |
| **자동 매핑** | "#OZParam.*패러미터명*#" 형태로 입력 패러미터 값을 매핑합니다. | |
| **입력** | 실행할 프로시저의 패러미터를 설정합니다. | |
| | **패러미터** | 실행할 프로시저의 패러미터 이름을 입력합니다. |
| | **데이터 타입** | 패러미터 데이터 타입입니다. |
| | **널 적용** | 값을 null로 설정할지 여부를 설정합니다. (기본 값 : False) |

| | | |
|---|---|---|
| | **값** | 패러미터 값을 입력합니다.<br>사용자가 값을 직접 입력하거나 "OZParam" 값을 설정할 수 있습니다.<br>※ 참고사항 : "널 적용" 속성 값이 True일 경우 값을 입력할 수 없습니다. |
| **출력** | 실행된 프로시저의 결과 값을 가져올 패러미터를 선택합니다.<br>결과 값을 가져올 패러미터가 존재할 경우 반드시 한 개 이상의 패러미터를 선택하여야하며, 패러미터가 존재하지 않을 경우에는 패러미터 선택없이 설정이 가능합니다.<br>※ 참고사항 : 데이터 타입이 Cursor인 패러미터와 그 외 타입인 패러미터를 동시에 선택할 수 없으며, 방향이 다른 패러미터를 동시에 선택할 수 없습니다. 즉, 방향이 출력과 입출력인 패러미터는 동시 선택이 가능하나 리턴 값인 패러미터는 선택할 수 없습니다. | |
| | **패러미터** | 프로시저의 결과 값을 가져올 패러미터 이름을 입력합니다. |
| | **데이터 타입** | 패러미터 데이터 타입입니다. |
| | **방향** | 패러미터 방향입니다. (입력/출력/입출력/리턴값) |
| **미리보기** | 설정한 프로시저를 멀티 데이터 셋의 형태로 미리보기합니다. | |

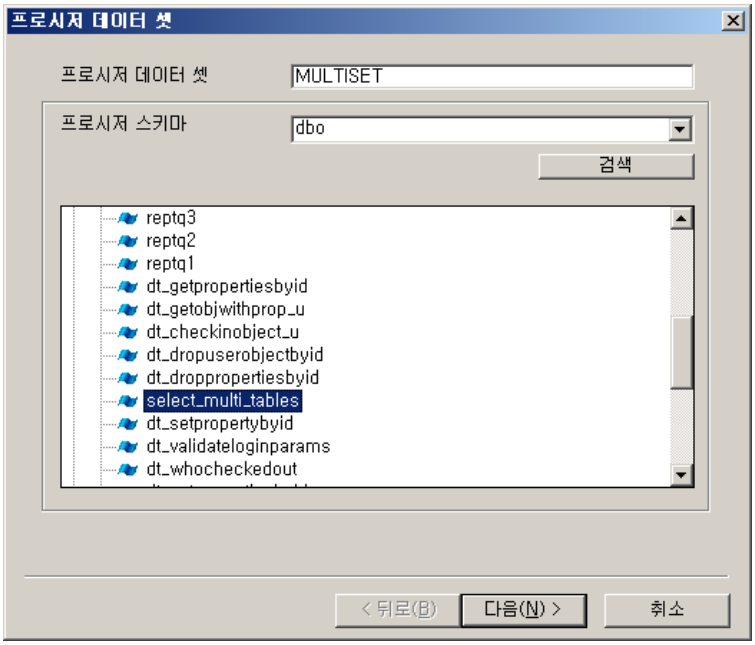
## 프로시저 데이터 셋 작성

**Step 1** 데이터 창에서 "데이터베이스 스토어"를 추가한 후 "데이터베이스 스토어"를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [프로시저 데이터 셋 추가] 메뉴를 클릭합니다.



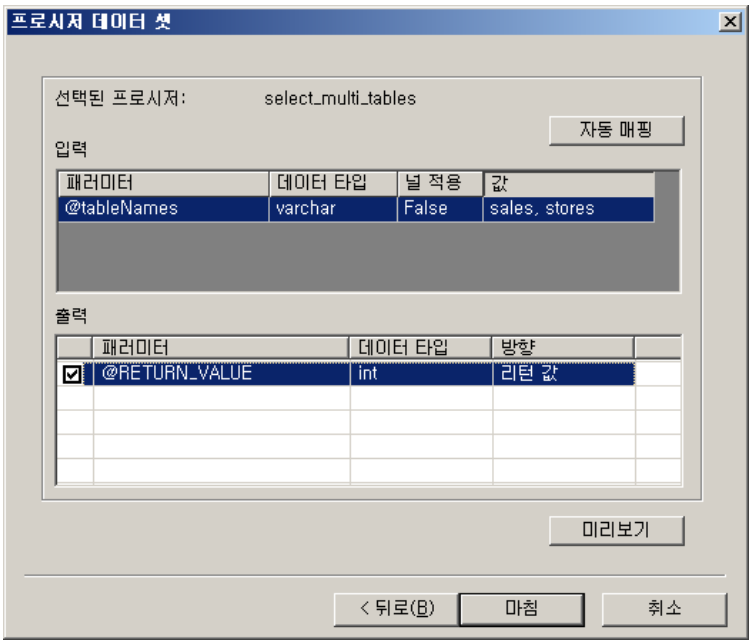
**Step 3**

'프로시저 데이터 셋 다이얼로그'에서 프로시저 데이터 셋 이름을 입력하고 프로시저 스키마를 설정한 후 [검색] 버튼을 클릭한 후 프로시저가 표시된 다이얼로그에서 프로시저를 선택하고 [다음] 버튼을 클릭합니다.

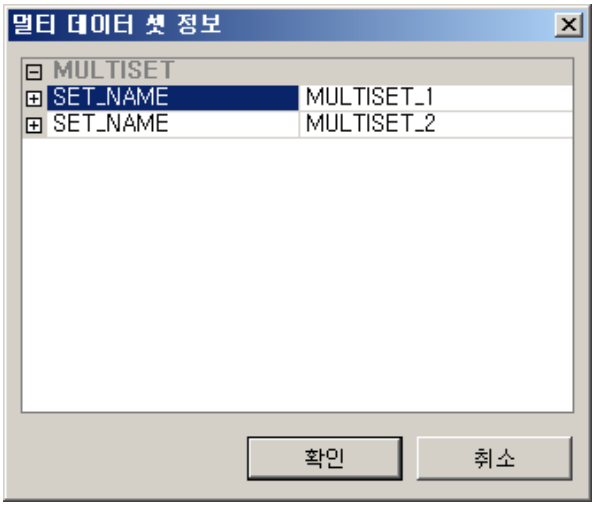



**Step 4**

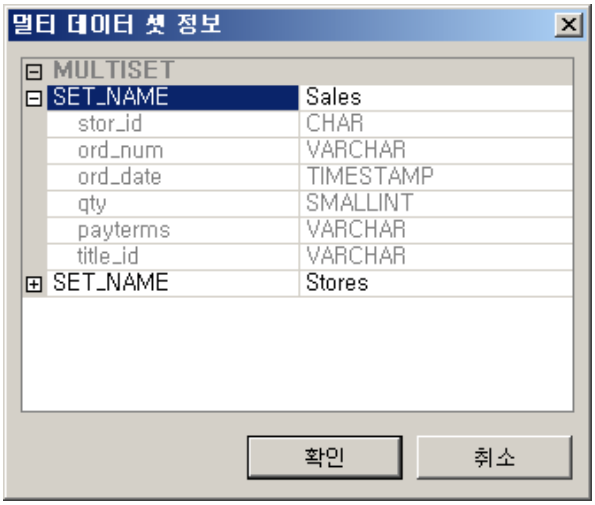
입력 창에서 값을 입력하고 출력할 패러미터를 선택합니다.



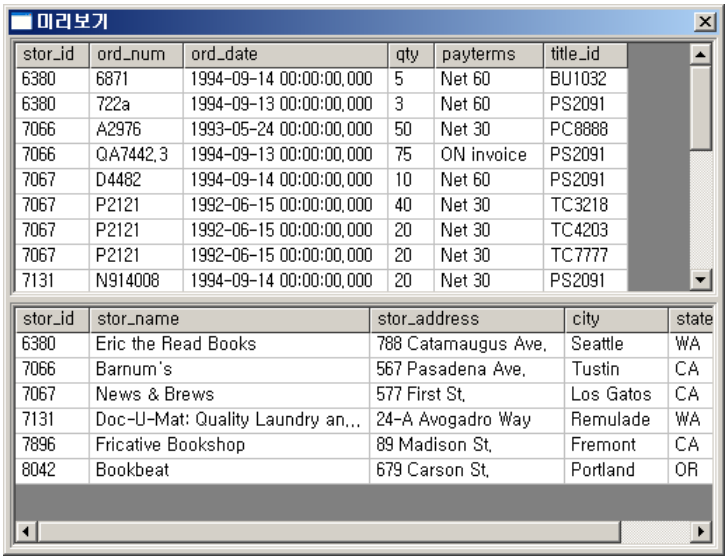
**Step 5** [미리보기] 버튼을 클릭하면 '멀티 데이터 셋 정보 다이얼로그'가 표시되며 해당 다이얼로그에 결과 셋의 개수만큼 데이터 셋의 정보가 표시됩니다.




**Step 6** '멀티 데이터 셋 정보 다이얼로그'에서 "SET_NAME" 값을 변경하여 멀티 데이터 셋의 하위 데이터 셋 이름을 변경할 수 있으며, 각 데이터 셋의 트리를 펼치면 해당 데이터 셋의 필드명 및 필드 타입이 표시됩니다.

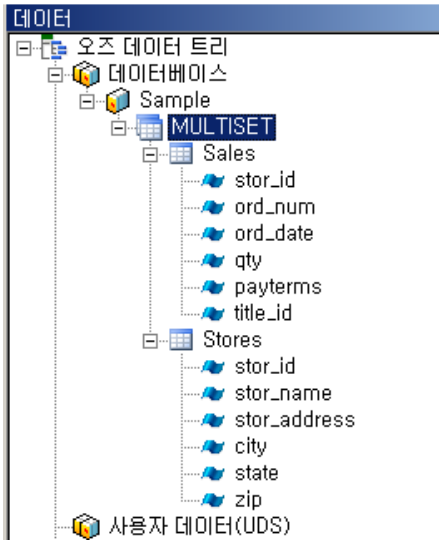



**Step 7** '멀티 데이터 셋 정보 다이얼로그'에서 [확인] 버튼을 클릭하면 미리보기 창에 아래 그림과 같이 데이터 셋 별로 데이터가 표시됩니다.




**Step 8**

미리보기한 후 '프로시저 데이터 셋 다이얼로그'에서 [마침] 버튼을 클릭하면 미리보기 시와 동일하게 '멀티 데이터 셋 정보 다이얼로그'가 표시되며 해당 다이얼로그에서 데이터 셋 정보를 설정한 후 [확인] 버튼을 클릭하면 데이터 창에 생성된 데이터 셋과 필드가 표시됩니다.



# 컴파일된 질의문 디자인하기

컴파일된 질의문은 JDBC에서 쿼리 실행 시 "PreparedStatement"를 사용한다는 개념으로, 해당 데이터 셋의 속성 중 "컴파일된 질의문 사용" 속성 값을 "True"로 설정한 경우에 적용됩니다. 컴파일된 질의문을 사용할 경우 한 번 컴파일된 질의문을 재사용하여 실행하므로, 같은 질의문이 여러 번 반복하여 실행될 경우 처리 속도를 높여줍니다. 컴파일된 질의문은 각 데이터베이스에서 제공하는 컴파일된 질의문을 사용하여 쿼리를 수행합니다.

컴파일된 질의문을 디자인 시에 오즈 패러미터를 이용하여 쿼리문을 작성할 경우에는 데이터 셋의 "스크립트 사용"와 "컴파일된 질의문 사용" 속성 값에 따라 각각 다르게 디자인하여야 합니다.

■ **스크립트 사용 = "False", 컴파일된 질의문 사용 = "True"/"False"인 경우**

※ #OZParam.패러미터명# 또는 '#OZParam.패러미터명#' 와 같은 오즈 패러미터 예약어를 독립적으로 사용합니다.

- 오즈 패러미터가 숫자형인 경우

```
SELECT * FROM 테이블명
WHERE 조건필드명 = #OZParam.패러미터명#
```

- 오즈 패러미터가 숫자형이 아닌 경우

```
SELECT * FROM 테이블명
WHERE 조건필드명 = '#OZParam.패러미터명#'
```

또는

```
SELECT * FROM 테이블명
WHERE 조건필드명 LIKE '%#OZParam.패러미터명#%'
```

■ **스크립트 사용 = "True", 컴파일된 질의문 사용 = "True"인 경우**

※ #OZParam.패러미터명# 또는 '#OZParam.패러미터명#' 와 같은 오즈 패러미터 예약어를 독립적으로 사용할 수 없고, 스크립트 문자열 내에서만 사용 가능합니다.

- 오즈 패러미터가 숫자형인 경우

```
string sql;
sql = "SELECT * FROM 테이블명 WHERE 조건필드명 = #OZParam.패러미터명#";
setQueryString(sql);
```

- 오즈 패러미터가 숫자형이 아닌 경우

```
string sql;
sql = "SELECT * FROM 테이블명 WHERE 조건필드명 = '#OZParam.패러미터명#'
";
setQueryString(sql);
```

또는

```
string sql;
sql = "SELECT * FROM 테이블명 WHERE 조건필드명 LIKE '%#OZParam.패러미터
명#%' ";
setQueryString(sql);
```

■ **스크립트 사용 = "True", 컴파일된 변경문 = "False"인 경우**

※ #OZParam.패러미터명# 또는 '#OZParam.패러미터명#' 와 같은 오즈 패러미터 예약어를 독립적으로 사용하여야 하며, 이때 숫자형이 아닌 오즈 패러미터를 사용할 경우 홀따옴표는 이중따옴표로 묶어야 제대로 동작됩니다.

ex) '#OZParam.패러미터명#' → "'" + #OZParam.패러미터명# + "'"

- 오즈 패러미터가 숫자형인 경우

```
string sql;
sql = "SELECT * FROM 테이블명 WHERE 조건필드명 = " + #OZParam.패러미터명
#;
setQueryString(sql);
```

- 오즈 패러미터가 숫자형이 아닌 경우

```
string sql;
sql = "SELECT * FROM 테이블명 WHERE 조건필드명 = " + "'" + #OZParam.패
러미터명# + "'";
setQueryString(sql);
```

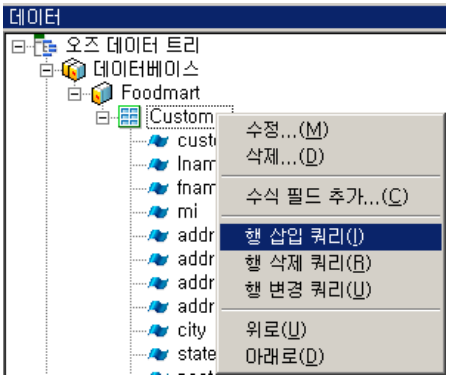
또는

```
string sql;
sql = "SELECT * FROM 테이블명 WHERE 조건필드명 LIKE " + " '%" +
#OZParam.패러미터명# + "%' ";
setQueryString(sql);
```

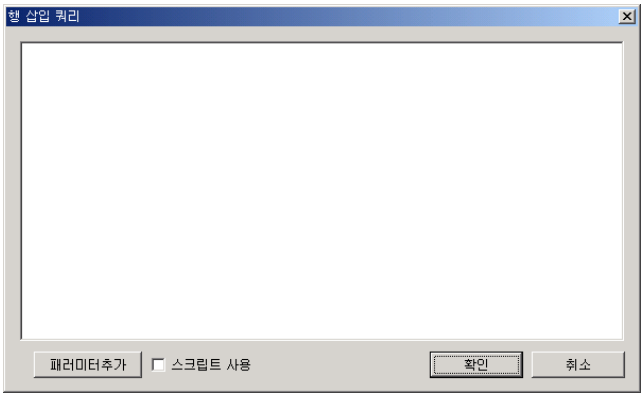
# 행 삽입/삭제/변경 쿼리 디자인하기

"행 삽입/삭제/변경 쿼리"는 오즈 애플리케이션에서 DataAction(Insert, Delete, Update)을 실행하기 위해 생성하는 쿼리문으로, 데이터베이스 데이터 셋 또는 사용자 데이터(UDS) 데이터 셋에서만 입력할 수 있습니다.
또한 "행 삽입/삭제/변경 쿼리"는 오즈 쿼리 디자이너에서는 실행 결과를 확인할 수 없으며, 오즈 애플리케이션에서 행 삽입/삭제/변경 쿼리를 호출하여 실행할 때에만 실행 결과를 확인할 수 있습니다.

행 삽입/삭제/변경 쿼리를 입력하기 위해서는 해당 쿼리를 추가할 데이터 셋을 마우스 오른쪽 버튼을 클릭하여 나타나는 팝업 메뉴에서 [행 삽입 쿼리], [행 삭제 쿼리], [행 변경 쿼리] 메뉴를 클릭합니다.



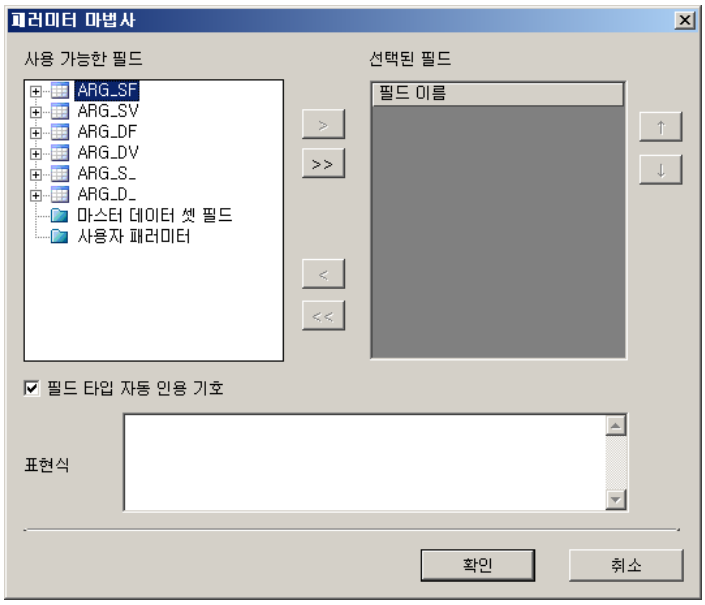

[행 삽입/삭제/변경 쿼리] 메뉴를 클릭하면 아래 그림과 같이 쿼리를 입력할 수 있는 다이얼로그가 표시됩니다.



※ 주의사항 : 행 삽입/삭제/변경 쿼리문에 주석을 사용한 경우 주석 내용에 사용 가능한 필드 (Source Field, Source Value, Destination Field, Destination Value, 마스터 데이터셋 필드, 사용자 패러미터), 사용자 지정 패러미터 구분자가 포함되어 있으면 쿼리문이 정상 동작하지 않을 수 있으니 사용 시 주의 바랍니다.

- 패러미터 추가

[패러미터 추가] 버튼을 클릭하면 '패러미터 마법사'를 이용하여 패러미터를 추가할 수 있도록 '패러미터 마법사'가 화면에 표시됩니다.



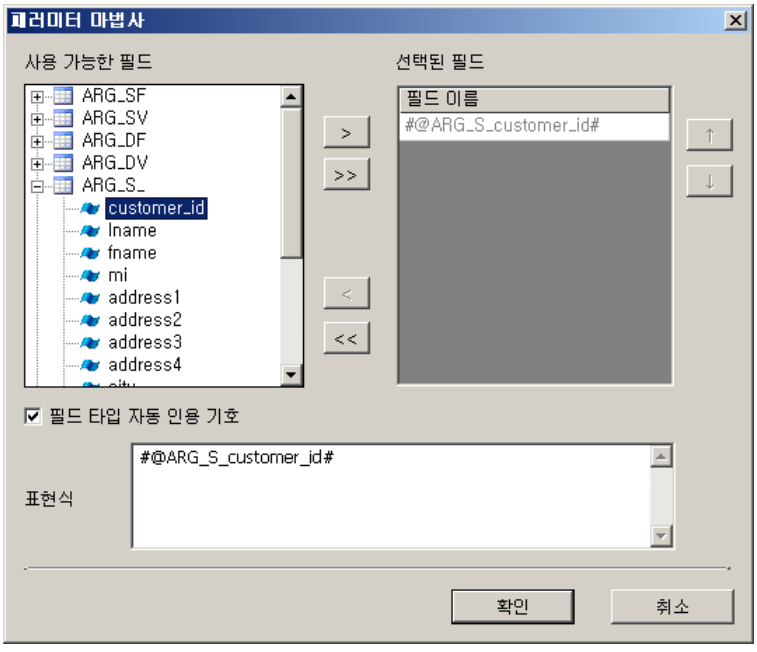

| UI | 설명 |
|---|---|
| 사용 가능한 필드 | Destination Field, Source Field, 마스터 데이터 셋 필드, 사용자 패러미터를 표시합니다.<br>※ 참고사항 : 마스터 데이터 셋 필드는 데이터셋에 마스터셋이지정된 경우에만 표시됩니다. |
| 선택된 필드 | 표현식에 추가된 필드 리스트를 나타냅니다. |
| > | '사용 가능한 필드'에 선택된 필드를 '선택된 필드'에 추가합니다. |
| >> | '사용 가능한 필드'에 추가된 모든 필드를 '선택된 필드'에 추가합니다. |
| < | '선택된 필드'에서 선택한 필드를 삭제합니다. |
| << | '선택된 필드'에 추가된 모든 필드를 삭제합니다. |
| ↑, ↓ | '선택된 필드'에서 선택한 필드의 위치를 위 또는 아래로 이동하여 표현식의 필드 순서를 변경합니다. |

| | |
|---|---|
| **필드 타입 자동 인용 기호** | 필드 타입이 CHAR, VARCHAR, LONGVARCHAR, DATE, TIME, TIMESTAMP일 경우 표현식에 패러미터 값을 따옴표로 묶어서 표시합니다.<br>※ 참고사항 : 데이터셋의 "컴파일된 변경문" 속성 값이 True일 경우 해당 옵션이 비활성화됩니다. |
| **표현식** | 선택된 필드를 패러미터 형식으로 표시합니다. 표현식에 추가된 패러미터는 ','를 구분자로 구분됩니다.<br>※ 참고사항<br>▪ 표현식에 추가된 패러미터는 선택된 필드에 추가된 필드와 동기화되어 표시됩니다.<br>▪ 행 삽입/삭제/변경 쿼리 다이얼로그에서 행 변경문을 드래그한 후 [패러미터 추가] 버튼을 클릭하면 드래그된 내용이 패러미터 마법사의 표현식에 추가되고, 선택된 필드와 동기화됩니다. |

'사용 가능한 필드'에서 사용할 필드를 선택한 후 필드 이동 버튼( > , >> )을 클릭하거나 '선택된 필드'로 드래그 & 드롭하여 필드를 선택한 후 [확인] 버튼을 클릭하면 쿼리 입력 다이얼로그에 패러미터가 추가됩니다.

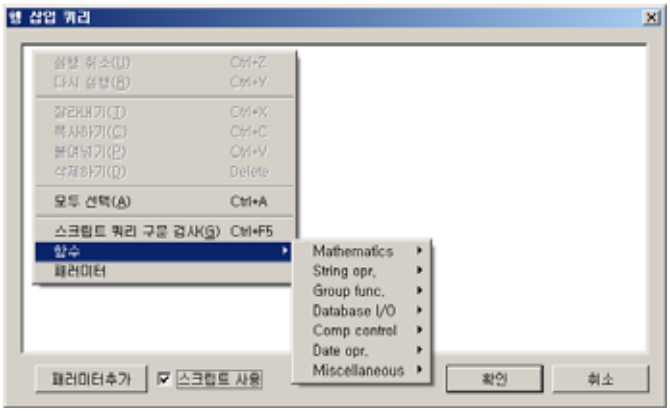
※ 참고사항

- 질의 인수 중 "ARG_SF", "ARG_SV", "ARG_DF", "ARG_DV"를 선택하면 사용 가능한 필드에는 필드 번호가 표시됩니다.
- 질의 인수 중 "ARG_S_", "ARG_D_"을 선택하면 사용 가능한 필드에는 데이터셋에 추가된 필드 이름이 표시됩니다.



- 스크립트 사용

  스크립트를 사용하여 쿼리를 입력할 경우에는 "스크립트 사용" 옵션을 체크하십시오. "스크립트 사용" 옵션을 체크한 후 스크립트 편집 창에서 마우스 오른쪽 버튼을 클릭하면 사용 가능한 스크립트를 나타내는 팝업 메뉴가 나타납니다.




  원하는 스크립트 구문을 입력한 후 [스크립트 쿼리 구문 검사] 메뉴를 이용하여 구문 검사를 실행하고 [확인] 버튼을 클릭하여 스크립트 쿼리문을 행 삽입/삭제/변경 쿼리에 적용합니다.

## 행 삽입 / 삭제 / 변경 쿼리 디자인

행 삽입/삭제/변경 쿼리는 각 쿼리문 입력 다이얼로그의 "스크립트 사용" 옵션 값과 해당 데이터 셋의 "컴파일된 변경문" 속성 값에 따라 각각 다르게 디자인하여야 합니다.

### ■ 행 삽입 / 삭제 / 변경 쿼리 작성시 주의 사항

- 추가 또는 변경되는 필드의 개수와 해당 값의 개수는 동일하여야 합니다.
- 삭제 시 삭제 조건이 되는 필드의 개수와 해당 조건 값의 개수는 동일하여야 합니다.
- 만일 추가/변경되는 필드 또는 삭제 시 조건 필드의 데이터 형식이 숫자형인 경우에는 해당 값은 따옴표로 묶지 않고, 그외 경우에는 홀따옴표로 묶어야 합니다.

  ex) 만일 4개의 필드 값을 추가하는데 첫 번째 필드만 숫자형인 경우

  #@ARG_SV1#, '#@ARG_SV2#', '#@ARG_SV3#', '#@ARG_SV4#'
- 데이터 셋의 "컴파일된 변경문" 속성 값이 "False"로 설정된 상태에서 필드 타입이 문자형이 아닌 필드에 설정되는 값이 "" 일 경우에는 "null" 로 변경됩니다. 이때 해당 필드가 null 값을 허용할 수 있도록 설정되어 있어야 합니다. 만일 필드 타입을 얻을 수

없을 경우에는 "null"로 치환되지 않고, 원래 값인 ""이 그대로 적용됩니다.

- "null"로 치환되는 필드 타입

  TINYINT, SMALLINT, INTEGER, BIGINT, REAL, FLOAT, DOUBLE, BIT, NUMERIC, DECIMAL, DATE, TIME, TIMESTAMP, BINARY, VARBINARY, LONGVARBINARY, BLOB, CLOB

- "null"로 치환되지 않는 필드 타입

  CHAR, VARCHAR, LONGVARCHAR

■ **스크립트 사용 = "False", 컴파일된 변경문 = "True" / "False"인 경우**

※ #@ARG_SF# 또는 '#@ARG_SF#' 와 같은 질의 인수를 독립적으로 사용합니다.

- 행 삽입 쿼리

  - 필드 번호로 쓴 경우(첫 번째 삽입 필드만 숫자형인 경우)

```
INSERT INTO 테이블명
(#@ARG_SF1#, #@ARG_SF2#,  ..... , #@ARG_SFn#)
VALUES
(#@ARG_SV1#, '#@ARG_SV2#',  ..... , '#@ARG_SVn#')
```

  - 필드 이름으로 쓴 경우(첫 번째 삽입 필드만 숫자형인 경우)

```
INSERT INTO 테이블명
(필드1이름, 필드2이름,  ..... , 필드n이름)
VALUES
(#@ARG_S_필드1이름#, '#@ARG_S_필드2이름#',  ..... , '#@ARG_S_필드n
이름#')
```

- 행 삭제 쿼리

  - 필드 번호로 쓴 경우(첫 번째 조건 필드만 숫자형인 경우)

```
DELETE FROM 테이블명
WHERE #@ARG_DF1# = #@ARG_DV1#
AND #@ARG_DF2# = '#@ARG_DV2#'
......
AND #@ARG_DFn# = '#@ARG_DVn#'
```

  - 필드 이름으로 쓴 경우(첫 번째 조건 필드만 숫자형인 경우)

```
DELETE FROM 테이블명
WHERE 필드1이름 = #@ARG_D_필드1이름#
AND 필드2이름 = '#@ARG_D_필드2이름#'
......
AND 필드n이름 = '#@ARG_D_필드n이름#'
```

- 행 변경 쿼리
  - 필드 번호로 쓴 경우(첫 번째 변경/조건 필드만 숫자형인 경우)

```
UPDATE 테이블명
SET  #@ARG_SF1# = #@ARG_SV1#,
     #@ARG_SF2# = '#@ARG_SV2#',
......
#@ARG_SFn# = '#@ARG_SVn#'
WHERE #@ARG_DF1# = #@ARG_DV1#
AND #@ARG_DF2# = '#@ARG_DV2#'
......
AND #@ARG_DFn# = '#@ARG_DVn#'
```

  - 필드 이름으로 쓴 경우(첫 번째 변경/조건 필드만 숫자형인 경우)

```
UPDATE 테이블명
SET  필드1이름 = #@ARG_S_필드1이름#,
     필드2이름 = '#@ARG_S_필드2이름#',
......
필드n이름 = '#@ARG_S_필드n이름#'
WHERE 필드1이름 = #@ARG_D_필드1이름#
AND 필드2이름 = '#@ARG_D_필드2이름#'
......
AND 필드n이름 = '#@ARG_D_필드n이름#'
```

■ **스크립트 사용 = "True", 컴파일된 변경문 = "True"인 경우**

※ #@ARG_SF# 또는 '#@ARG_SF#' 와 같은 질의 인수를 독립적으로 사용할 수 없고, 스크립트 문자열 내에서만 사용 가능합니다.

- 행 삽입 쿼리
  - 필드 번호로 쓴 경우(첫 번째 삽입 필드만 숫자형인 경우)

```
string sql;

sql = " INSERT INTO 테이블명 (#@ARG_SF1#, #@ARG_SF2#,  ....... ,
#@ARG_SFn#)  VALUES  (#@ARG_SV1#,  '#@ARG_SV2#',    ......  ,
'#@ARG_SVn#') ";

setQueryString(sql);
```

  - 필드 이름으로 쓴 경우(첫 번째 삽입 필드만 숫자형인 경우)

```
string sql;

sql = " INSERT INTO 테이블명 (필드1이름, 필드2이름,  ....... , 필드n
이름) VALUES (#@ARG_S_필드1이름#, '#@ARG_S_필드2이름#',  ...... ,
```

```
'#@ARG_S_필드n이름#') ";

setQueryString(sql);
```

- 행 삭제 쿼리
  - 필드 번호로 쓴 경우(첫 번째 조건 필드만 숫자형인 경우)

```
string sql;

sql = " DELETE FROM 테이블명 WHERE #@ARG_DF1# = #@ARG_DV1# AND
#@ARG_DF2# = '#@ARG_DV2#' ..... AND #@ARG_DFn# = '#@ARG_DVn#'
";

setQueryString(sql);
```

  - 필드 이름으로 쓴 경우(첫 번째 조건 필드만 숫자형인 경우)

```
string sql;

sql = " DELETE FROM 테이블명 WHERE 필드1이름 = #@ARG_D_필드1이름#
AND 필드2이름 = '#@ARG_D_필드2이름#' ..... AND 필드n이름 =
'#@ARG_D_필드n이름#' ";

setQueryString(sql);
```

- 행 변경 쿼리
  - 필드 번호로 쓴 경우(첫 번째 변경/조건 필드만 숫자형인 경우)

```
string sql;

sql = " UPDATE 테이블명 SET #@ARG_SF1# = #@ARG_SV1#, #@ARG_SF2#
= '#@ARG_SV2#', ..... , #@ARG_SFn# = '#@ARG_SVn#' WHERE
#@ARG_DF1# = #@ARG_DV1# AND #@ARG_DF2# = '#@ARG_DV2#' .....
AND #@ARG_DFn# = '#@ARG_DVn#' ";

setQueryString(sql);
```

  - 필드 이름으로 쓴 경우(첫 번째 변경/조건 필드만 숫자형인 경우)

```
string sql;

sql = " UPDATE 테이블명 SET 필드1이름 = #@ARG_S_필드1이름#, 필드2이름
= '#@ARG_S_필드2이름#', ..... , 필드n이름 = '#@ARG_S_필드n이름#'
WHERE 필드1이름 = #@ARG_D_필드1이름# AND 필드2이름 = '#@ARG_D_필드2이
름#' ..... AND 필드n이름 = '#@ARG_D_필드n이름#' ";
```

```
setQueryString(sql);
```

■ **스크립트 사용 = "True", 컴파일된 변경문 = "False"인 경우**

※ #@ARG_SF1# 또는 '#@ARG_SF1#' 와 같은 질의 인수를 스크립트 문자열 내에서 사용하거나 독립적으로 사용할 수 있습니다.

※ 주의사항 : 질의 인수를 독립적으로 사용할 경우에 홀따옴표는 이중 따옴표로 묶어야 제대로 동작됩니다.

```
ex) '#@ARG_SF1#' → "'" + #@ARG_SF1# + "'"
```

- 행 삽입 쿼리

➢ 필드 번호로 쓴 경우(첫 번째 삽입 필드만 숫자형인 경우)

```
string sql;

sql = " INSERT INTO 테이블명 (#@ARG_SF1#, #@ARG_SF2#, ....... ,
#@ARG_SFn#) VALUES (#@ARG_SV1#, '#@ARG_SV2#', ..... ,
'#@ARG_SVn#') ";

setQueryString(sql);
```

또는

```
string sql;

sql = " INSERT INTO 테이블명 (#@ARG_SF1#, #@ARG_SF2#, ....... ,
#@ARG_SFn#) VALUES (" + #@ARG_SV1# + ", " + "'" + #@ARG_SV2# +
"'" + " , ..... , " + "'" + #@ARG_SVn# + "'" + ") ";

setQueryString(sql);
```

➢ 필드 이름으로 쓴 경우(첫 번째 삽입 필드만 숫자형인 경우)

```
string sql;

sql = " INSERT INTO 테이블명 (필드1이름, 필드2이름, ....... , 필드n
이름) VALUES (#@ARG_S_필드1이름#, '#@ARG_S_필드2이름#', ..... ,
'#@ARG_S_필드n이름#') ";

setQueryString(sql);
```

또는

```
string sql;

sql = " INSERT INTO 테이블명 (필드1이름, 필드2이름,  ..... , 필드n
이름) VALUES (" + #@ARG_S_필드1이름# + ", " + "'" + #@ARG_S_필드2
이름# + "'" + " ,  ..... , " + "'" + #@ARG_S_필드n이름# + "'" +
") ";

setQueryString(sql);
```

- 행 삭제 쿼리
  - 필드 번호로 쓴 경우(첫 번째 조건 필드만 숫자형인 경우)

```
string sql;

sql = " DELETE FROM 테이블명 WHERE #@ARG_DF1# = #@ARG_DV1# AND
#@ARG_DF2# = '#@ARG_DV2#' ..... AND #@ARG_DFn# = '#@ARG_DVn#'
";

setQueryString(sql);
```

또는

```
string sql;

sql = " DELETE FROM 테이블명 WHERE #@ARG_DF1# = " + #@ARG_DV1# +
" AND #@ARG_DF2# = " + "'" + #@ARG_DV2# + "'" + " ..... AND
#@ARG_DFn# = " + "'" + #@ARG_DVn# + "'";

setQueryString(sql);
```

  - 필드 이름으로 쓴 경우(첫 번째 조건 필드만 숫자형인 경우)

```
string sql;

sql = " DELETE FROM 테이블명 WHERE 필드1이름 = #@ARG_D_필드1이름#
AND  필드2이름  =  '#@ARG_D_필드2이름#'  .....  AND  필드n이름  =
'#@ARG_D_필드n이름#' ";

setQueryString(sql);
```

또는

```
string sql;
```

```
sql = " DELETE FROM 테이블명 WHERE 필드1이름 = " +  #@ARG_D_필드1이름# + " AND 필드2이름 = " + "'" + #@ARG_D_필드2이름# + "'" + " ..... AND 필드n이름 = " + "'" + #@ARG_D_필드n이름# + "'";

setQueryString(sql);
```

- 행 변경 쿼리
  - 필드 번호로 쓴 경우(첫 번째 변경/조건 필드만 숫자형인 경우)

```
string sql;

sql = " UPDATE 테이블명 SET  #@ARG_SF1# = #@ARG_SV1#,  #@ARG_SF2# = '#@ARG_SV2#',  ......  ,  #@ARG_SFn#  =  '#@ARG_SVn#'  WHERE #@ARG_DF1#  =  #@ARG_DV1#  AND  #@ARG_DF2#  =  '#@ARG_DV2#'  ..... AND #@ARG_DFn# = '#@ARG_DVn#' ";

setQueryString(sql);
```

또는

```
string sql;

sql = " UPDATE 테이블명 SET  #@ARG_SF1# = " + #@ARG_SV1# + ", #@ARG_SF2# = " + "'" + #@ARG_SV2# + "'" + ", ..... #@ARG_SFn# = " + "'" + #@ARG_SVn# + "'" + " WHERE #@ARG_DF1# = " + #@ARG_DV1# + " AND #@ARG_DF2# = " + "'" + #@ARG_DV2# + "'" + " ..... AND #@ARG_DFn# = " + "'" + #@ARG_DVn# + "'";

setQueryString(sql);
```

  - 필드 이름으로 쓴 경우(첫 번째 변경/조건 필드만 숫자형인 경우)

```
string sql;

sql = " UPDATE 테이블명 SET  필드1이름 = #@ARG_S_필드1이름#, 필드2이름 = '#@ARG_S_필드2이름#', ..... , 필드n이름 = '#@ARG_S_필드n이름#' WHERE 필드1이름 = #@ARG_D_필드1이름# AND 필드2이름 = '#@ARG_D_필드2이름#' ..... AND 필드n이름 = '#@ARG_D_필드n이름#' ";

setQueryString(sql);
```

또는

```
string sql;
```

```
sql = " UPDATE 테이블명 SET  필드1이름 = " + #@ARG_S_필드1이름# + ",
_필드2이름 = " + "'" + #@ARG_S_필드2이름# +"'" + ", ..... , 필드n
이름 = " + "'" + #@ARG_S_필드n이름# + "'" + " WHERE 필드1이름 =  "
+ #@ARG_D_필드1이름# + " AND 필드2이름 = " + "'" + #@ARG_D_필드2이름
# + "'" + " ..... AND 필드n이름 = " + "'" + #@ARG_D_필드n이름# +
"'";

setQueryString(sql);
```

## 마스터셋을 이용한 행 삽입/삭제/변경 쿼리 디자인

마스터-디테일 구조의 ODI인 경우에는 디테일셋은 마스터셋의 필드를 이용하여 행 삽입/삭제/변경 쿼리를 디자인할 수 있습니다. 디테일셋에서 사용할 마스터셋의 필드를 숫자형인 경우에는 #<마스터셋명>.<필드명># 형태로, 문자형인 경우에는 '#<마스터셋명>.<필드명>#'로 입력합니다. 마스터셋을 이용한 행 삽입/삭제/변경 쿼리의 경우 스크립트 사용 또는 컴파일된 변경문 사용 여부에 따라 작성하는 법이 다르며, 작성법 또는 일반적인 주의사항은 일반적인 행 삽입/삭제/변경 쿼리 디자인시 주의사항을 참조하시기 바랍니다.

본 매뉴얼에서는 마스터셋을 이용한 행 삽입/삭제/변경 쿼리를 디자인하는 예로 스크립트 사용 = "False", 컴파일된 변경문 = "False"이고 필드 번호를 이용하여 작성한 경우만 살펴보도록 하겠습니다. 그외의 경우에 대한 작성법은 일반적인 행 삽입/삭제/변경 쿼리를 디자인 방식을 참조하시기 바랍니다.

■ **스크립트 사용 = "False", 컴파일된 변경문 = "True"인 경우**

아래는 마스터셋명은 "MDSet", 필드명은 "지역", 필드 타입이 문자형이고, 디테일셋의 세번째 필드에 마스터셋을 적용할 경우의 예입니다.

- 행 삽입 쿼리
  - ➢ 필드 번호로 쓴 경우(첫 번째 삽입 필드만 숫자형인 경우)

```
INSERT INTO 테이블명
(#@ARG_SF1#, #@ARG_SF2#, #@ARG_SF3#, ....... , #@ARG_SFn#)
VALUES
(#@ARG_SV1#, '#@ARG_SV2#', '#MDSet.지역#'...... , '#@ARG_SVn#')
```

- 행 삭제 쿼리
  - ➢ 필드 번호로 쓴 경우(첫 번째 조건 필드만 숫자형인 경우)

```
DELETE FROM 테이블명
```

```
WHERE #@ARG_DF1# = #@ARG_DV1#
AND #@ARG_DF2# = '#@ARG_DV2#'
AND #@ARG_DF2# = '#MDSet.지역#'
......
AND #@ARG_DFn# = '#@ARG_DVn#'
```

- 행 변경 쿼리
  - ➢ 필드 번호로 쓴 경우(첫 번째 변경/조건 필드만 숫자형인 경우)

```
UPDATE 테이블명
SET  #@ARG_SF1# = #@ARG_SV1#,
     #@ARG_SF2# = '#@ARG_SV2#',
     #@ARG_SF3# = '#MDSet.지역#',
......
#@ARG_SFn# = '#@ARG_SVn#'
WHERE #@ARG_DF1# = #@ARG_DV1#
AND #@ARG_DF2# = '#@ARG_DV2#'
AND #@ARG_DF2# = '#MDSet.지역#'
......
AND #@ARG_DFn# = '#@ARG_DVn#'
```

## 행 삽입 / 삭제 / 변경 쿼리 작성 예

■ **스크립트 사용 = "False", 컴파일된 변경문 = "True" / "False"인 경우**

테이블명이 "Customer"이고, 첫 번째 삽입/변경/조건 필드만 숫자형인 경우

- 행 삽입 쿼리

```
INSERT INTO Customer
(#@ARG_SF1#,#@ARG_SF2#,#@ARG_SF3#,#@ARG_SF4#,#@ARG_SF5#,#@ARG_SF6#,
#@ARG_SF7#,#@ARG_SF8#)
VALUES(#@ARG_SV1#,'#@ARG_SV2#','#@ARG_SV3#','#@ARG_SV4#','#@ARG_SV5
#','#@ARG_SV6#','#@ARG_SV7#','#@ARG_SV8#')
```

- 행 삭제 쿼리

```
DELETE FROM Customer
WHERE #@ARG_DF1# = #@ARG_DV1# AND #@ARG_DF2# = '#@ARG_DV2#' AND
#@ARG_DF3# = '#@ARG_DV3#'
```

- 행 변경 쿼리

```
UPDATE Customer SET #@ARG_SF1# = #@ARG_SV1#, #@ARG_SF2# =
```

```
'#@ARG_SV2#', #@ARG_SF3# = '#@ARG_SV3#', #@ARG_SF4# = '#@ARG_SV4#',
#@ARG_SF5# = '#@ARG_SV5#', #@ARG_SF6# = '#@ARG_SV6#', #@ARG_SF7# =
'#@ARG_SV7#', #@ARG_SF8# = '#@ARG_SV8#'
WHERE #@ARG_DF1# = #@ARG_DV1# AND #@ARG_DF2# = '#@ARG_DV2#' AND
#@ARG_DF3# = '#@ARG_DV3#'
```

■ **스크립트 사용 = "True", 컴파일된 변경문 = "True"인 경우**

테이블명이 "Customer"이고, 첫 번째 삽입/변경/조건 필드만 숫자형인 경우

- 행 삽입 쿼리

```
string sql;

sql = "INSERT INTO Customer
(#@ARG_SF1#,#@ARG_SF2#,#@ARG_SF3#,#@ARG_SF4#,#@ARG_SF5#,#@ARG_SF6#,
#@ARG_SF7#,#@ARG_SF8#)
VALUES(#@ARG_SV1#,'#@ARG_SV2#','#@ARG_SV3#','#@ARG_SV4#','#@ARG_SV5
#','#@ARG_SV6#','#@ARG_SV7#','#@ARG_SV8#')";

setQueryString(sql);
```

- 행 삭제 쿼리

```
string sql;

sql = "DELETE FROM Customer WHERE #@ARG_DF1# = #@ARG_DV1# AND
#@ARG_DF2# = '#@ARG_DV2#' AND #@ARG_DF3# = '#@ARG_DV3#' ";

setQueryString(sql);
```

- 행 변경 쿼리

```
string sql;

sql = "UPDATE Customer SET #@ARG_SF1# = #@ARG_SV1#, #@ARG_SF2# =
'#@ARG_SV2#', #@ARG_SF3# = '#@ARG_SV3#', #@ARG_SF4# = '#@ARG_SV4#',
#@ARG_SF5# = '#@ARG_SV5#', #@ARG_SF6# = '#@ARG_SV6#', #@ARG_SF7# =
'#@ARG_SV7#', #@ARG_SF8# = '#@ARG_SV8#' WHERE #@ARG_DF1# =
#@ARG_DV1# AND #@ARG_DF2# = '#@ARG_DV2#' AND #@ARG_DF3# =
'#@ARG_DV3#' ";

setQueryString(sql);
```

# Ⅶ. 사용자 데이터 스토어

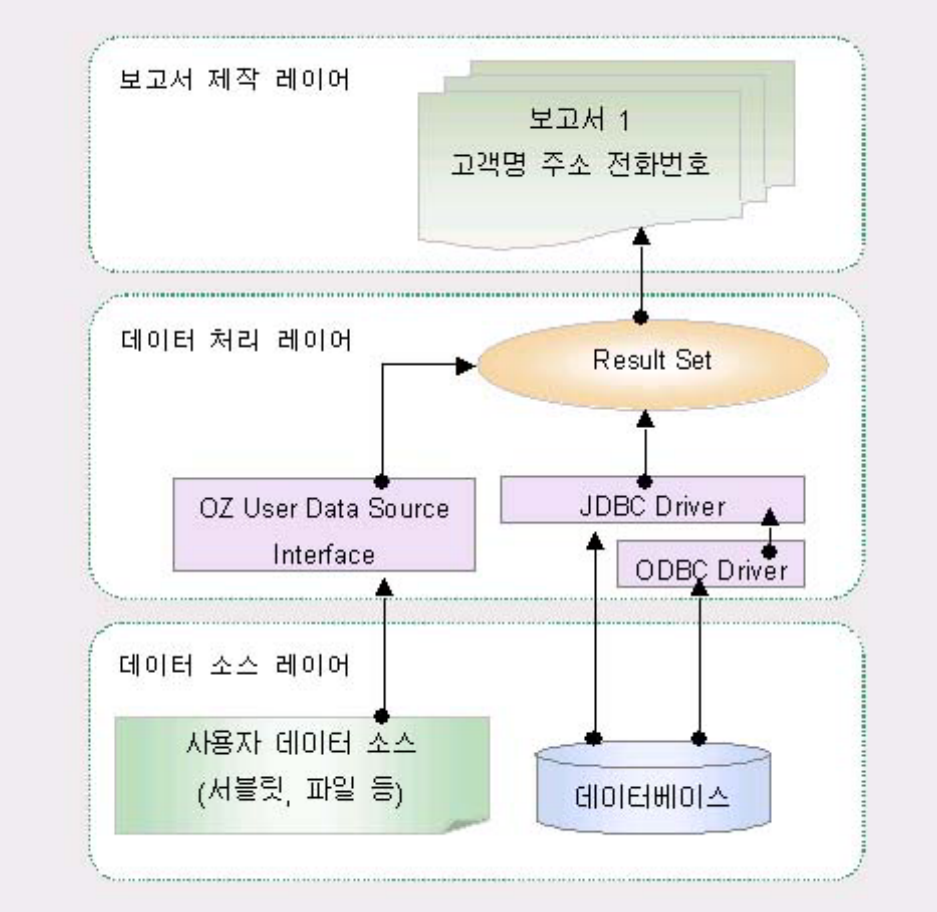
- 사용자 데이터 스토어 개요
- 사용자 데이터 스토어 이용

# 사용자 데이터 스토어 개요

## 사용자 데이터 스토어 개요

사용자 데이터 스토어 즉 UDS(User Data Store)는 일반 파일 데이터(텍스트 파일, CSV, XML) 및 사용자 프로그램 데이터(EJB, Servlet, ASP, JSP, Applet)와 같은 다양한 형태의 외부 데이터 소스로부터 데이터 셋을 생성하는 Java Interface 규격입니다.

따라서 UDS는 JDBC/ODBC를 통해 데이터베이스로부터 데이터를 가져오는 것과 마찬가지로 사용자 애플리케이션에서 파일이나 XML, 벡터, 스트림, EJB 객체 등의 형태로 데이터를 받아서 오즈의 데이터 소스로 변환해 주는 기능을 담당합니다.




UDS 개발자는 UDS Java Interface에 따라 다양한 형태의 사용자 데이터를 오즈가 인식할 수

있는 ResultSet 객체 형태로 가공하여 전달하는 방법을 구현해야 합니다. 이렇게 구현된 자바 사용자 프로그램은 클래스 파일 형태로 오즈 보고서와 연동되며, UDS를 통해 생성되는 UDS 데이터 셋은 일반 쿼리 데이터 셋과 동일하게 사용할 수 있습니다.

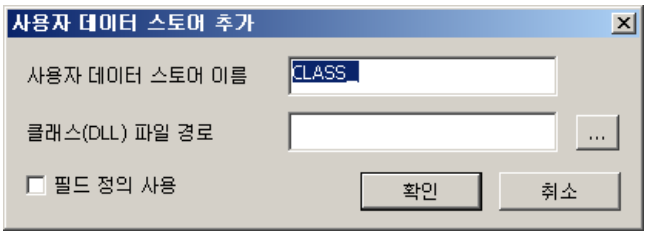
오즈는 사용자 클래스로 패러미터 스트링을 전달할 수 있으며 패러미터 값에 따라 다이나믹하게 변화하는 데이터를 전달 받을 수도 있습니다.

UDS 클래스 개발을 위한 자세한 사항은 "API 활용 매뉴얼"을 참조하시기 바랍니다.

# 사용자 데이터 스토어 이용

다음은 사용자가 개발한 UDS 클래스를 통해 사용자 데이터베이스 소스에 연결하고, 연결된 사용자 데이터베이스 소스로부터 데이터 셋을 생성하는 과정을 설명합니다.

## 사용자 데이터 정보 추가 다이얼로그

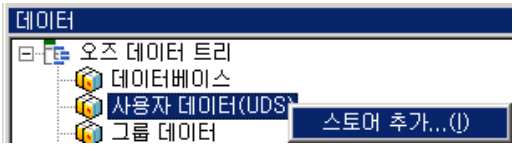



| 항목 | 설명 |
|---|---|
| **사용자 데이터 스토어 이름** | 해당 데이터 셋을 가리키는 이름입니다. |
| **클래스(DLL) 파일 경로** | UDS 클래스 파일의 경로를 설정합니다. |
| **필드 정의 사용** | 해당 UDS 클래스에서 필드 정보를 가져올 수 없는 경우 사용자가 필드 정보를 직접 정의할 것인지 여부를 설정합니다. |
| **[...]** | 클래스 파일이 위치한 경로를 설정합니다. |

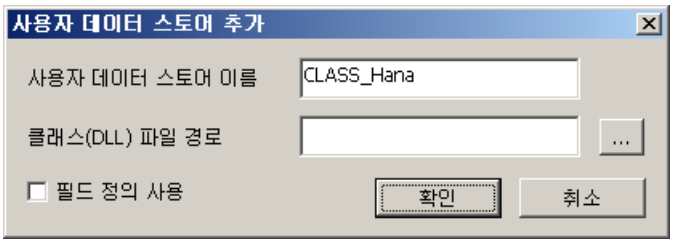
## UDS 연결

다음에서는 사용자 클래스를 이용해 오즈 오즈 쿼리 디자이너와 사용자 데이터베이스를 연결하고 데이터 셋을 생성하는 방법을 순차적으로 설명합니다.

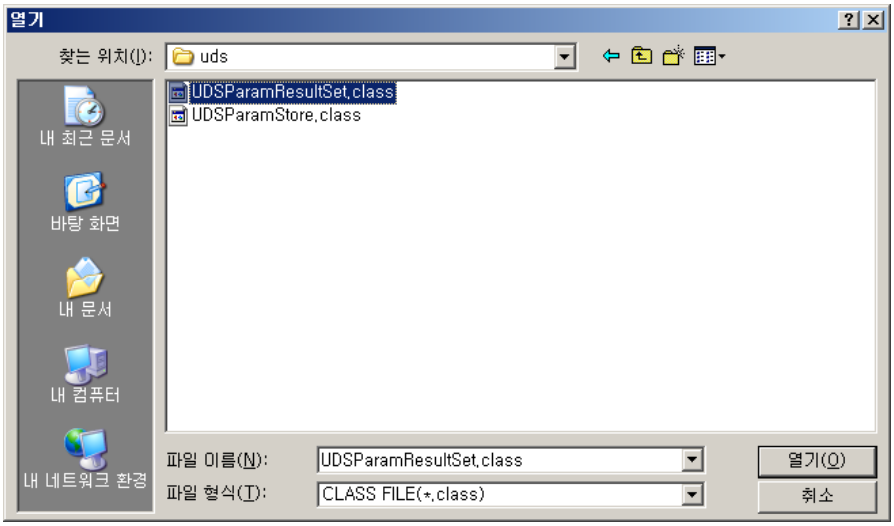
**Step 1** 데이터 창의 [사용자 데이터(UDS)]를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [스토어 추가] 메뉴를 클릭합니다.



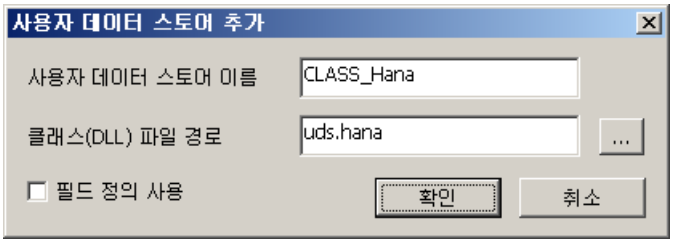
**Step 2** '사용자 데이터 스토어 추가 다이어로그'에서 사용자 데이터 스토어 정보를 설정하고 […] 버튼을 클릭하면 클래스 파일을 설정할 수 있는 열기 다이얼로그가 표시됩니다.



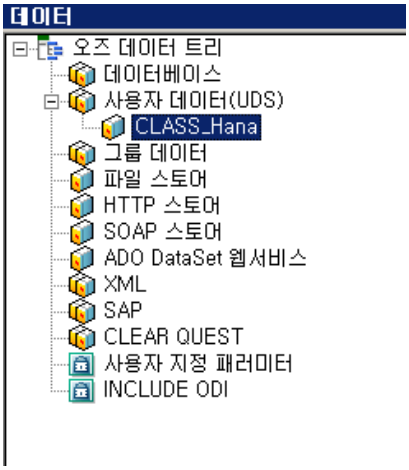

**Step 3** 열기 다이얼로그에서 추가할 클래스 파일을 선택한 후 더블 클릭하거나 [열기] 버튼을 클릭하여 클래스 파일 경로를 설정합니다.



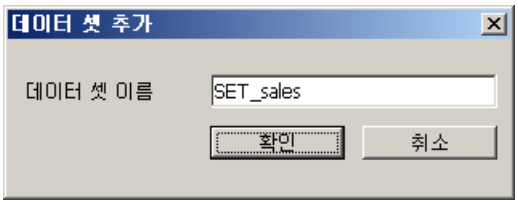

**Step 2** 설정한 클래스 파일 경로를 확인한 후 [확인] 버튼을 클릭합니다.



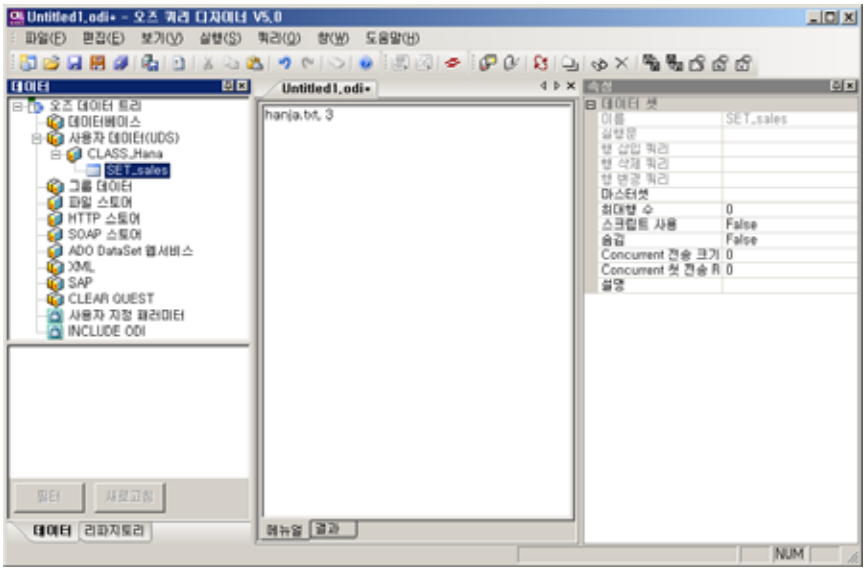

**Step 3** [확인] 버튼을 누르면 아래와 같이 [사용자 데이터]에 추가한 사용자 데이터 스토어 정보가 나타납니다.



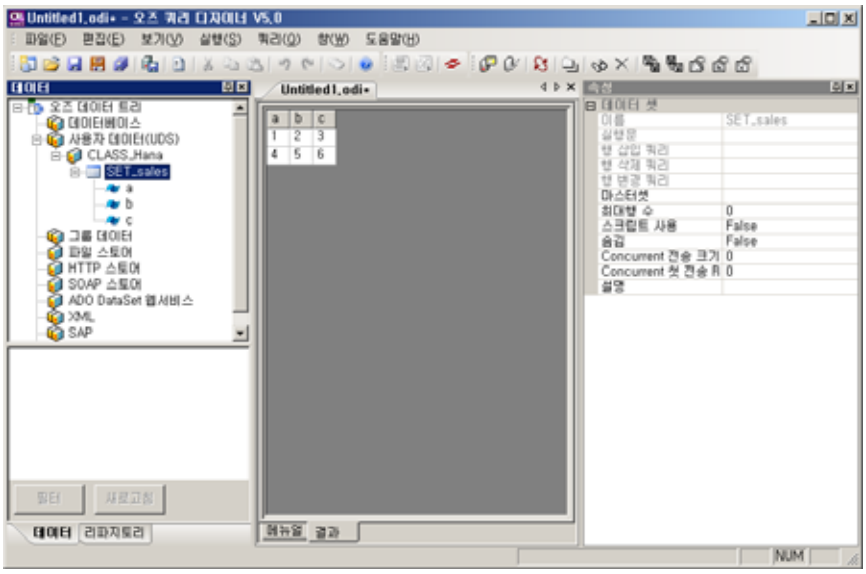

**Step 4** 데이터 창에 생성된 사용자 데이터 스토어를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [데이터 셋 추가] 메뉴를 클릭하면 데이터 셋 이름을 입력하는 다이얼로그가 나타납니다.




**Step 5** 데이터 셋 이름을 입력한 후 [확인] 버튼을 클릭하면 쿼리 디자이너의 쿼리 창에 아래 그림처럼 사용자 데이터 스토어 정보를 설정할 수 있도록 변경됩니다.



쿼리 창의 [매뉴얼] 탭을 선택하고 쿼리 창에 실행문을 입력한 후 [실행] 메뉴의 [쿼리문 실행] 메뉴를 클릭하거나 툴바에서 쿼리문 실행( )아이콘을 클릭하거나 또는 키보드의 <F5> 키를 눌러 쿼리문을 실행합니다. 실행문에 에러가 없을 경우에는 쿼리 창이 [결과] 탭으로 선택되면서 결과 데이터가 표시되며 해당 데이터 셋 아래에 결과 필드가 표시됩니다.



※ 주의사항 : 사용자 데이터 스토어의 디자인 모드는 매뉴얼 모드만 지원되며 디자인 뷰 모드는 지원되지 않습니다.

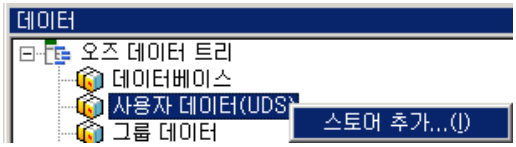
## 사용자 필드 추가

다음은 사용자 데이터 스토어에서 사용자 필드를 추가하는 방법에 대해 설명합니다.

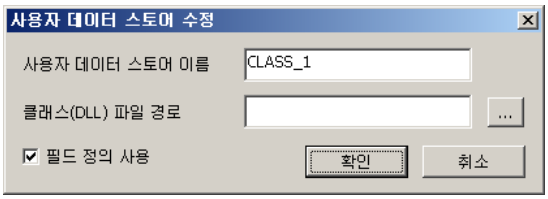
※ 주의사항

- 사용자 필드는 연결할 클래스 파일을 설정하지 않아도 데이터 셋과 필드를 추가할 수 있습니다.
- 만일 클래스 파일이 설정된 사용자 데이터 스토어에 사용자 필드를 추가할 경우 사용자 필드가 우선시되므로, 클래스 파일에 설정된 필드 정보는 더 이상 사용할 수 없습니다.

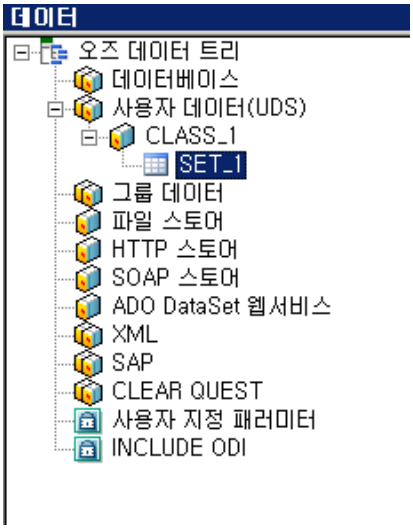
**Step 1** 데이터 창의 [사용자 데이터(UDS)]를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [스토어 추가] 메뉴를 클릭합니다.



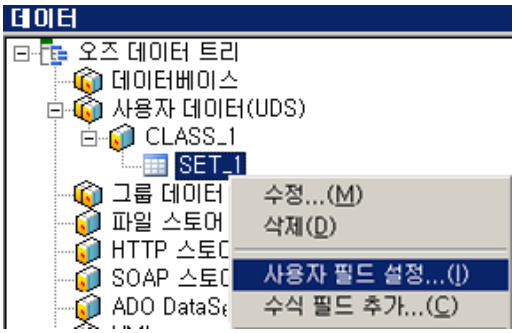

**Step 2** '사용자 데이터 스토어 추가 다이어롤그'에서 "필드 정의 사용" 옵션을 체크하여, 사용자 데이터 스토어에 사용자 필드를 추가할 수 있도록 설정합니다.



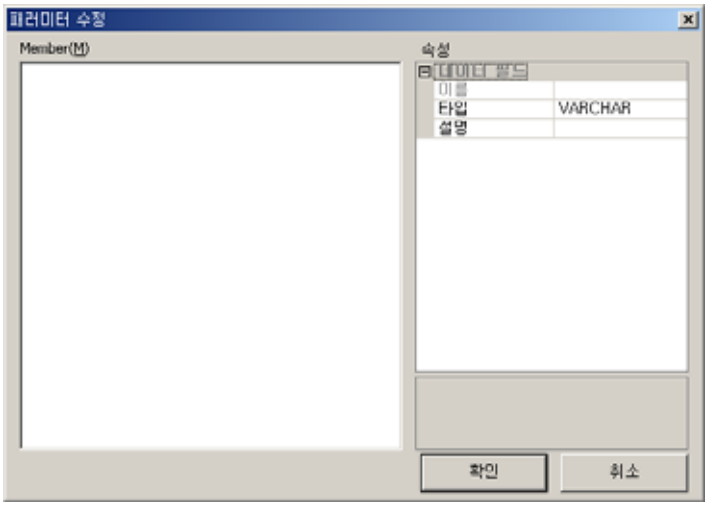

**Step 3** [확인] 버튼을 누르면 아래와 같이 [사용자 데이터]에 추가한 사용자 데이터 스토어 정보가 나타납니다. 추가된 사용자 데이터 스토어를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [데이터 셋 추가] 메뉴를 클릭하여 데이터 셋을 추가합니다.



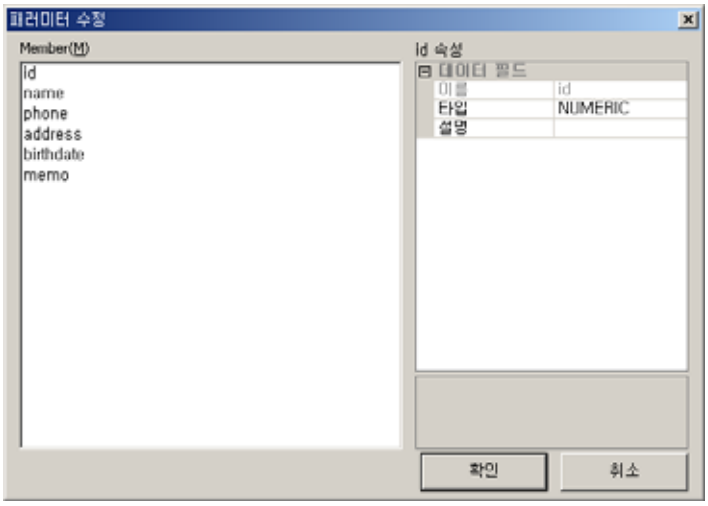

**Step 4** 추가된 데이터 셋을 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [사용자 필드 설정] 메뉴를 클릭합니다.



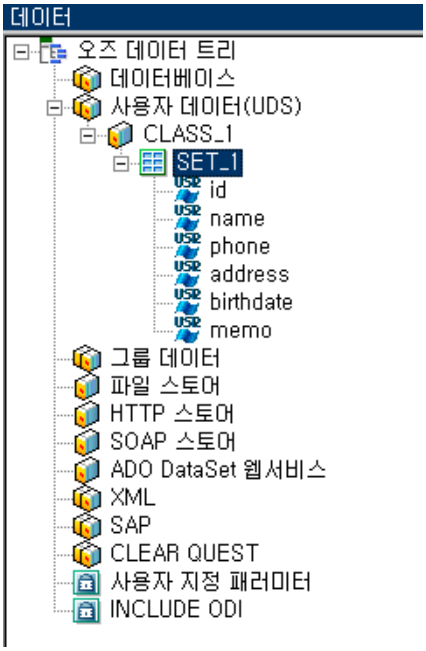

**Step 5** [사용자 필드 설정] 메뉴를 클릭하면 아래 그림과 같이 '사용자 필드 설정 다이얼로그'가 나타납니다.




필드 이름을 입력하고 필드 타입을 설정합니다.



[확인] 버튼을 클릭하면 데이터 셋에 사용자 필드가 추가됩니다.



# Ⅷ. 그룹 데이터 스토어

- 그룹 데이터 스토어 개요
- 그룹 데이터 스토어 이용

# 그룹 데이터 스토어 개요

## 그룹 데이터 스토어 개요

그룹 데이터 스토어는 다른 데이터 소스(데이터베이스, UDS, 파일/HTTP 스토어, XML DTD 스토어)에서 생성된 데이터 셋을 다수의 데이터 셋으로 분할하는 2차 데이터 소스로써, 다수의 데이터 셋이 하나의 데이터 셋에서 파생 가능할 때 이용하는 데이터 소스입니다.

그룹 데이터 스토어는 하나의 데이터 셋을 이용해 다수의 데이터 셋을 파생시키기 때문에 다수의 독립 데이터 셋을 이용하는 것에 비해 쿼리와 통신 횟수가 감소하는 장점이 있습니다.

예를 들어 마스터-디테일 보고서를 작성하기 위해 세 개의 쿼리로 세 개의 데이터 셋을 생성하는 것보다 하나의 쿼리로 하나의 데이터 셋을 생성하여 GDS로 분할하는 것이 데이터베이스 성능을 향상시킬 수 있습니다

- **원본 데이터 셋**

  다중 데이터 셋으로 분할되는 데이터 셋을 원본 데이터 셋이라 하고, 마스터/디테일 데이터 셋을 분할할 경우 가장 상위 마스터 셋으로 사용되는 데이터 셋만을 원본 데이터 셋으로 사용할 수 있습니다.

- **그룹 데이터 셋**

  원본 데이터 셋을 분할해서 만든 데이터 셋을 그룹 데이터 셋이라 합니다. 원본 데이터 셋으로부터 생성된 그룹 데이터 셋은 일반 데이터 셋과 동일하게 사용할 수 있습니다.

## 데이터 셋 분할

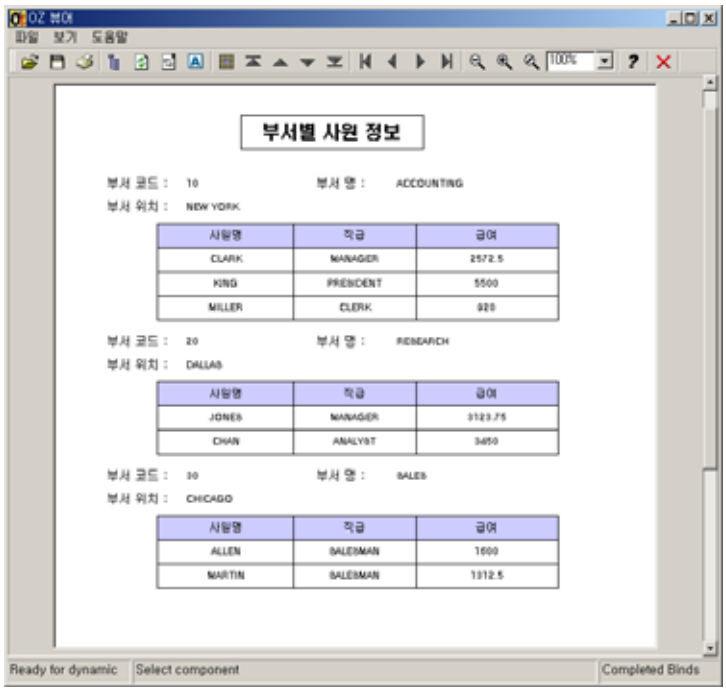
원본 데이터 셋은 분할 조건문에 의해서 여러 개의 그룹 데이터 셋으로 나눠지는데 다음과 같은 형태의 분할을 지원합니다.

- **수직 분할**

  일반적으로 가장 많이 사용되는 형태로 하나의 데이터 셋으로 마스터-디테일 관계의 보고서나 오즈 애플리케이션을 작성하기 위해 원본 셋을 마스터 그룹 데이터 셋과 디테일 그룹 데

이터 셋으로 분할합니다.

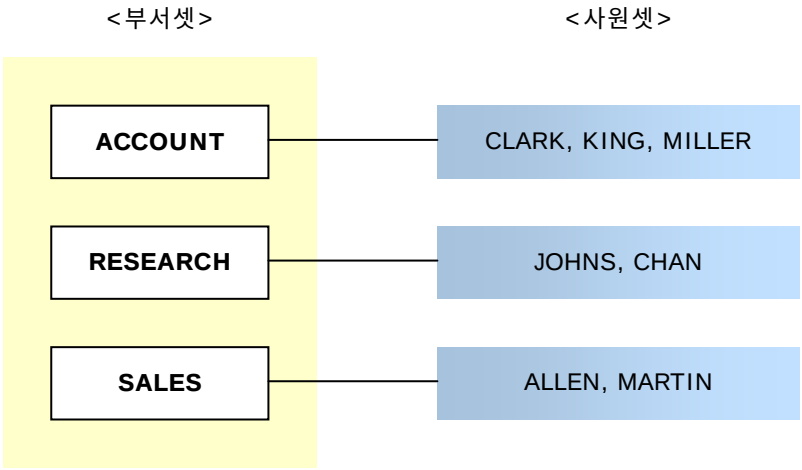
예를 들어 부서별 사원 정보를 나열하는 다음과 같은 보고서가 있을 때,




위와 같은 보고서를 제작하기 위해 부서 정보와 사원 정보를 아래의 표와 같이 하나의 데이터 셋으로 생성합니다.

| 코드 | 부서명 | 부서위치 | 사원명 | 직급 | 급여 |
|---|---|---|---|---|---|
| 10 | ACCOUNTING | NEW YORK | CLARK | MANAGER | 2572.2 |
| 10 | ACCOUNTING | NEW YORK | KING | PRESIDENT | 5500 |
| 10 | ACCOUNTING | NEW YORK | MILLER | CLERK | 920 |
| 20 | RESEARCH | DALLAS | JONES | MANAGER | 3123.75 |
| 20 | RESEARCH | DALLAS | CHAN | ANALYST | 3450 |
| 30 | SALES | CHICAGO | ALLEN | SALESMAN | 1600 |
| 30 | SALES | CHICAGO | MARTIN | SALESMAN | 1312.5 |

위의 원본 데이터 셋을 수직분할하여 두 개의 데이터 그룹 데이터 셋 즉, 부서셋과 사원셋으로 나눌 수 있으며 각 셋의 데이터와 분할 조건은 다음과 같습니다.




조건이 true인 경우 원본 셋에서 해당 셋으로 무조건 데이터를 가져옵니다. 조건식이 있을 경우 해당 조건식이 'true'가 될 경우만 해당 데이터 셋으로 데이터를 가져옵니다.

■ **수평 분할**

하나의 데이터 셋이 서로 독립된 여러 개의 셋으로 분할됩니다.

| 상품코드 | 상품명 | 단가 | ... |
|---|---|---|---|
| A | FLAT TV 19" | 500,000 | |
| A | FLAT TV 21" | 800,000 | |
| A | FLAT TV 25" | 1,500,000 | |
| B | 광폭 타이어 | 100,000 | ... |
| B | 광폭 타이어 V | 180,000 | |
| C | 사과맛 음료수 | 500 | |
| C | 포도맛 음료수 | 500 | |

원본 데이터 셋이 위와 같을 때 각 코드별로 보고서 양식에서 표현하는 형태가 다를 경우 코드별로 데이터 셋을 분할해서 보고서를 작성합니다

- < A 셋 > (조건) #원본셋.코드# == "A"

| 상품코드 | 상품명 | 단가 | ... |
|---|---|---|---|
| A | FLAT TV 19" | 500,000 | ... |
| A | FLAT TV 21" | 800,000 | |

| | | | |
|---|---|---|---|
| A | FLAT TV 25" | 1,500,000 | |

- < B 셋 > (조건) #원본셋.코드# == "B"

| 상품코드 | 상품명 | 단가 | ... |
|---|---|---|---|
| B | 광폭 타이어 | 100,000 | ... |
| B | 광폭 타이어 V | 180,000 | |

- < C 셋 > (조건) #원본셋.코드# == "C"

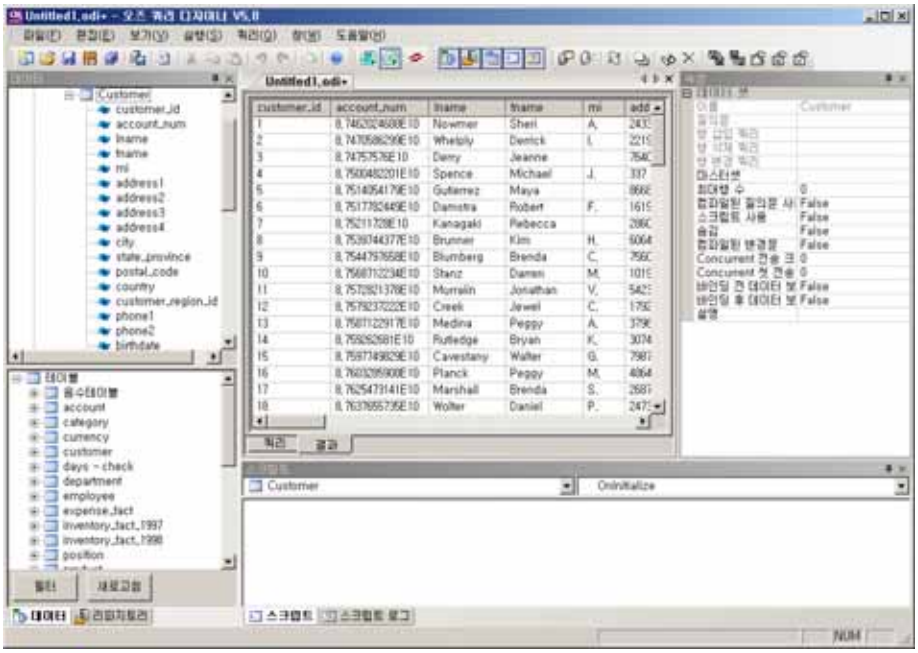
| 상품코드 | 상품명 | 단가 | ... |
|---|---|---|---|
| C | 사과맛 음료수 | 500 | ... |
| C | 포도맛 음료수 | 500 | |

※ 참고사항 : 수직 분할과 수평 분할을 혼합하여 사용할 수도 있습니다.

# 그룹 데이터 스토어 이용

## 원본 데이터 셋 생성

실제 데이터베이스나 UDS 등에서 실제 데이터를 가져오기 위한 원본 데이터 셋을 생성합니다.



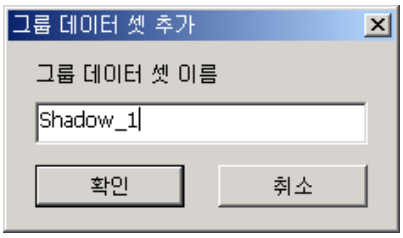
## 그룹 데이터 스토어 추가

데이터 창의 [그룹 데이터]를 오른쪽 마우스으로 클릭하여 나타나는 팝업 메뉴에서 [스토어 추가] 메뉴를 클릭하면 다음과 같이 원본 데이터 셋을 선택하기 위한 그룹 데이터 선택 창이 나타납니다. 그룹 데이터 스토어 이름을 입력하고, 연동할 원본셋 이름을 선택한 후 [확인] 버튼을 클릭하여 그룹 데이터 스토어를 추가합니다.

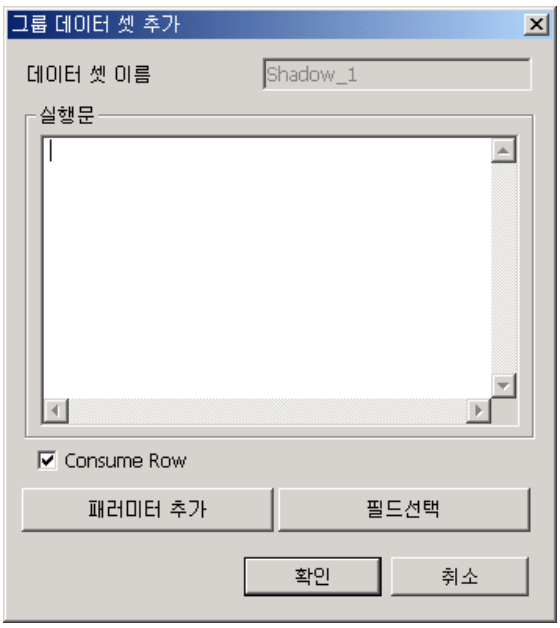


## 그룹 데이터 셋 생성

그룹 데이터 스토어 추가가 완료되면 해당 그룹 데이터 스토어를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [데이터 셋 추가] 메뉴를 클릭하면 '그룹 데이터 셋 추가 다이얼로그'가 나타납니다.



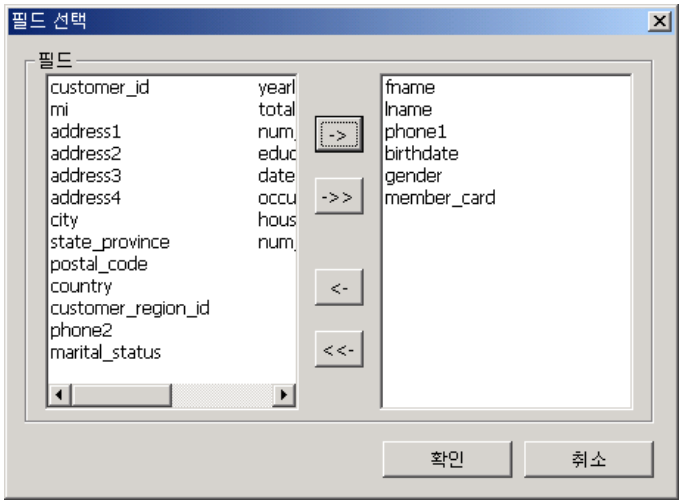

그룹 데이터 셋 이름을 입력한 후 [확인] 버튼을 클릭하면 아래와 같이 '그룹 데이터 셋 추가 다이얼로그'가 표시됩니다.




| 항목 | 설명 |
|---|---|
| **데이터 셋 이름** | 해당 데이터 셋에 대한 이름으로 사용자가 임의로 설정합니다. |
| **실행문** | 그룹 데이터를 생성하기 위한 조건문을 입력합니다. |
| **Consume Row** | Comsum Row는 그룹 데이터 스토어에서 원본셋의 데이터를 여러 개의 데이터 셋으로 분할할 때 사용합니다. 한 행의 데이터를 모두 분할하고 다음 행을 가져오는 시점에 'True'를 설정하여야 그룹 데이터 스토의 바인딩이 정상적으로 작동하게 됩니다. 일반적으로 하나의 그룹 데이터 스토어 내의 데이터 셋들 중 가장 하위의 데이터 셋만 'True'로 설정합니다. |

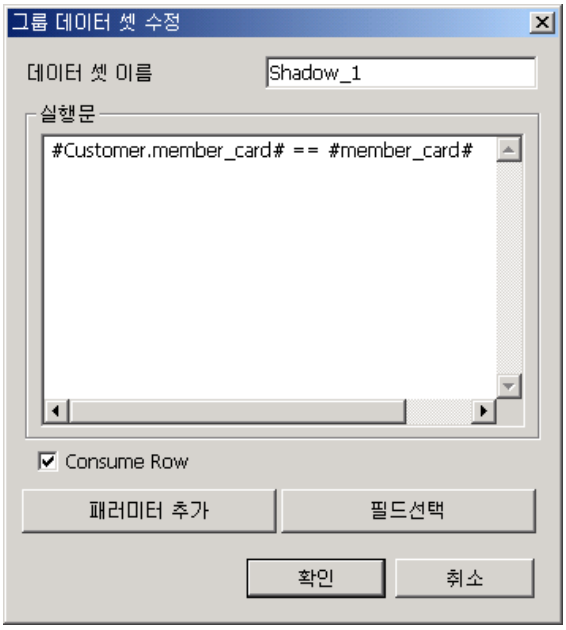
| | |
|---|---|
| **패러미터 추가** | 사용할 패러미터를 패러미터 마법사를 이용하여 추가합니다. 패러미터 마법사에 대한 자세한 내용은 본 매뉴얼 "Ⅴ. 쿼리 데이터 셋 디자인의 패러미터 마법사 이용하기"를 참조하시기 바랍니다. |
| **필드 선택** | 사용할 필드를 선택합니다. |

[필드 선택] 버튼을 클릭하여 표시할 필드를 설정합니다.

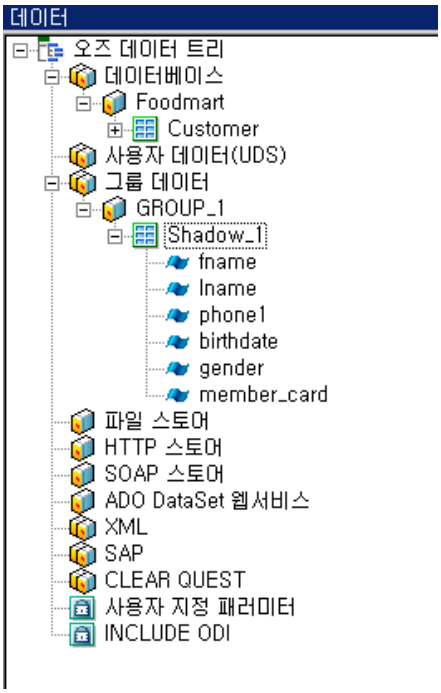



원하는 필드를 선택하여 추가한 후 [확인] 버튼을 클릭합니다.

다음으로 실행문에 원본 셋 분할을 위한 조건문을 입력합니다.



[확인] 버튼을 클릭하여 그룹 데이터 셋을 추가되며, 추가된 그룹 데이터 셋은 데이터 창의 [그룹 데이터] 아래에 표시됩니다.



# Ⅸ. 파일/HTTP 스토어

- 파일/HTTP 스토어 개요
- 파일 스토어 이용
- HTTP 스토어 이용

# 파일/HTTP 스토어 개요

파일/HTTP 스토어는 일반 텍스트 파일이나 HTTP 출력을 데이터 소스로 활용하는 기능으로써, CSV 형식과 DTD나 XSD를 이용하지 않는 표준 XML 형식을 지원합니다.

## CSV(Comma-Separated Value) 파일

CSV 파일은 데이터베이스 간의 데이터 교환을 위해 널리 사용되는 파일 포맷으로, 필드와 레코드를 구분하기 위해 콤마를 이용하는 일반 텍스트 파일입니다. 표준 CSV 파일은 콤마만을 구분자로 이용하지만, 오즈 파일/HTTP 스토어에서는 특정 문자를 구분자로 지정할 수 있습니다.

### ■ CSV 규격

파일/HTTP 스토어가 지원하는 CSV 규격은 다음과 같습니다.

- 필드 간의 구분자는 모두 동일한 문자를 이용하여야 하며, 필드 정보와 데이터 레코드는 새로운 줄에서 시작되어야 합니다.
- 첫 줄과 두 번째 줄에 필드 이름과 필드 타입으로 구성되는 필드 정보가 오고, 필드 정보 다음 줄부터 필드 정보와 일치하는 데이터 레코드가 반복되어야 합니다.
- 필드 정보에서 필드 타입을 나타내는 두 번째 줄은 필요에 따라 생략될 수 있습니다.

### ■ CSV 예제

다음은 ','를 구분자로 이용하는 CSV 파일의 예제입니다.

```
no,jumin_id,name,address
INT,VARCHAR,VARCHAR,VARCHAR
1,7411231932347,kim,seoul
2,6305212983717,lee,pusan
3,7006031837261,park,daegu
4,7271722848281,song,daejeon
```

## XML(Extensible Markup Language) 파일

XML은 웹 상에서 서로 다른 업체의 응용 프로그램 간의 데이터 교환을 위한 데이터 표현 언어로써, The World Wide Web Consortium(W3C)이 정의한 데이터를 응용 프로그램으로부터 분리하는 웹 언어입니다. 예를 들어, HTML이 웹 상에서 표준 UI를 구현하는 웹 언어라면 XML은 웹 상에서의 데이터 교환을 위한 웹 언어라고 할 수 있습니다.

오즈는 XML을 데이터 소스로 이용하기 위해 파일/HTTP 스토어와 XML 스토어를 제공합니다. DTD나 XSD 없이 정해진 형태의 단순한 XML을 데이터 소스로 이용하기 위해서는 파일/HTTP 스토어를 이용하여야 하고, XML 스토어는 DTD나 XSD를 이용한 XML을 데이터 소스로 이용할 수 있습니다.

■ **XML 규격**

파일/HTTP 스토어가 지원하는 XML 파일 규격은 다음과 같습니다.

- DTD가 없는 표준 XML 규격에 맞는 파일을 지원합니다.
- 데이터 셋을 나타내는 노드를 포함하여야 합니다.
- 필드 정보를 나타내는 노드는 다음과 같은 형태로 기술되어야 하며 생략할 수 있습니다.

다음은 XML 파일의 예제입니다.

```
<FIELDINFO>
<FIELD NAME>="ID" TYPE="INT"/>
<FIELD NAME>="NAME" TYPE="CHAR"/>
<FIELD NAME>="ADDRESS" TYPE="VARCHAR"/>
….
</FIELDINFO>
```

※ 레코드 셋을 나타내는 노드를 포함하여야 합니다.

■ **XML 예제**

다음은 데이터 셋, 레코드셋 노드의 이름이 각각 'DATASET', 'RECORD'이고 필드 정보 노드를 포함한 XML 파일의 예제입니다.

```
<?xml version="1.0" encoding="KSC5601"?>
<XMLSET>
<DATASET>
<FIELDINFO>
<FIELD NAME="ID" TYPE="INT"/>
<FIELD NAME="NAME" TYPE="CHAR"/>
<FIELD NAME="ADDRESS" TYPE="VARCHAR"/>
</FIELDINFO>
```

```
<RECORD>
<ID>1</ID>
<NAME>Park Chan Ho</NAME>
<ADDRESS>Texas</ADDRESS>
</RECORD>
<RECORD>
<ID>2</ID>
<NAME>Shin Ki Sung</NAME>
<ADDRESS>Seoul</ADDRESS>
</RECORD>
<RECORD>
<ID>3</ID>
<NAME>Kim Byung Hyun</NAME>
<ADDRESS>Arizona</ADDRESS>
</RECORD>
</DATASET>
</XMLSET>
```

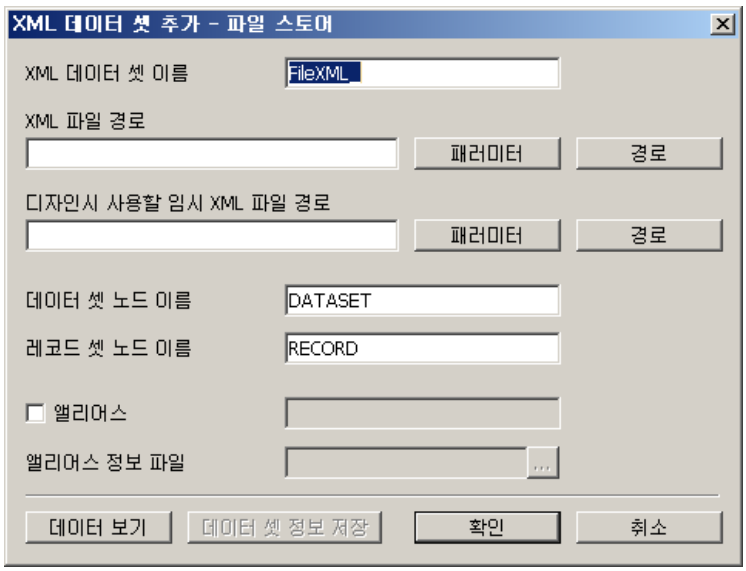
# 파일 스토어 이용

다음은 파일 스토어에 연결하고 데이터 셋을 생성하는 방법에 대해 설명합니다.

## 파일 스토어 설정

XML이나 CSV 타입의 파일을 데이터 소스로 연동할 때 사용하는 창입니다.

■ **XML 데이터 셋 설정**



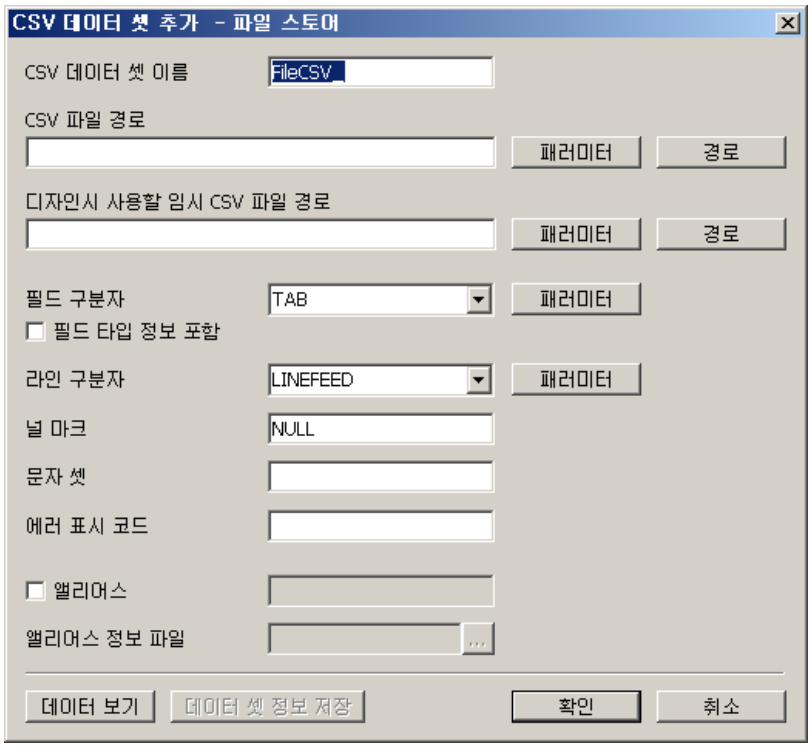

| 항목 | 설명 |
|---|---|
| **XML 데이터 셋 이름** | 해당 데이터 셋에 대한 이름으로 사용자가 임의로 부여합니다. |
| **XML 파일 경로** | 실제 서버에서 이용할 XML 파일의 경로와 이름입니다. |
| **디자인시 사용할 임시 XML 파일 경로** | XML 파일 경로가 오즈 서버와 디자이너의 다른 경우 디자인 시점에만 사용하는 임시 XML 파일의 경로와 이름입니다. |
| **데이터 셋 노드 이름** | 해당 XML 파일에서 데이터 셋을 나타내는 노드 이름입니다. 기본 값은 'DATASET' 입니다. |
| **레코드셋 노드 이름** | 해당 XML 파일에서 레코드 셋을 나타내는 노드 이름입니다. 기본 값은 'RECORD'입니다. |

| | |
|---|---|
| **패러미터** | 사용할 패러미터를 패러미터 마법사를 이용하여 설정합니다. 패러미터 마법사에 대한 자세한 내용은 본 매뉴얼 "Ⅴ. 쿼리 데이터 셋 디자인의 패러미터 마법사 이용하기"를 참조하시기 바랍니다. |
| **경로** | 가져올 XML 파일을 선택합니다. |
| **데이터 보기** | 설정한 파일의 데이터를 미리보기합니다. |
| **데이터 셋 정보 저장** | 해당 데이터 셋의 정보(필드 이름, 필드 타입)를 XML 파일로 저장합니다.<br>※ 참고사항<br>▪ [데이터 셋 정보 저장] 버튼은 데이터 셋을 수정할 경우에만 활성화됩니다.<br>▪ 데이터 셋 정보 저장 시 인코딩이 UTF-8로 설정됩니다. |
| **앨리어스** | 해당 데이터 셋 설정 정보를 별도의 연결 정보 파일에서 가져올 것인지 여부를 설정하며 앨리어스를 사용할 경우 오즈 서버의 datasource.properties 파일에 정의한 앨리어스 이름을 설정합니다. |
| **앨리어스 정보 파일** | 데이터 셋 설정 정보 파일 이름입니다.<br>앨리어스를 사용할 경우 앨리어스 정보가 있는 데이터 셋 설정 정보 파일명을 설정합니다. |

■ **XML 예제**

```
<?xml version="1.0" encoding="KSC5601"?>
<!-- Data Sample of OZ XML Dataset-->
<XMLSET>
    <DATASET>
        <FIELDINFO>
            <FIELD NAME="ID" TYPE="INT"/>
            <FIELD NAME="NAME" TYPE="CHAR"/>
            <FIELD NAME="ADDRESS" TYPE="VARCHAR"/>
        </FIELDINFO>
        <RECORD>
            <ID>1</ID>
            <NAME>Park Chan Ho</NAME>
            <ADDRESS>USA</ADDRESS>
        </RECORD>
    </DATASET>
</XMLSET>
```

■ **CSV 데이터 셋 설정**

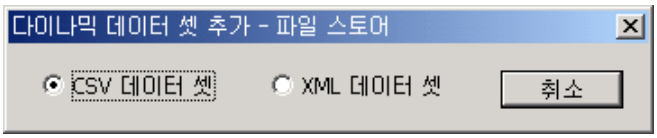



| 항목 | 설명 |
|---|---|
| **CSV 데이터 셋 이름** | 해당 데이터 셋에 대한 이름으로 사용자가 임의로 설정합니다. |
| **CSV 파일 경로** | 실제 서버에서 사용할 CSV 파일의 경로와 이름입니다. |
| **디자인시 사용할 임시 CSV 파일 경로** | CSV 파일의 경로가 오즈 서버와 디자이너에서 다른 경우 디자인 시점에만 사용하는 임시 CSV 파일의 경로와 이름입니다. |
| **필드 구분자** | 필드 구분자를 선택합니다. TAB, SPACE, SEMICOLON, COLON, COMMA는 미리 정의된 구분자로 선택 가능하며 기타 문자에 대해서는 사용자가 직접 입력해서 설정할 수 있습니다. |
| **라인 구분자** | 라인 구분자를 선택합니다. LINEFEED는 미리 정의된 구분자로 기본 값이며 기타 문자에 대해서는 사용자가 직접 입력해서 설정할 수 있습니다.<br>또한, [패러미터] 버튼을 클릭하여 사용자 지정 패러미터로 라인 구분자를 설정할 수 있습니다. |
| **필드 타입 정보 포함** | 해당 CSV 파일에 필드에 대한 타입 정보가 포함되어 있는지 유무를 선택합니다. 체크되지 않은 경우 모든 필드의 타입은 'VARCHAR'로 간주됩니다. |
| **널 마크** | NULL 데이터를 구분하기 위한 문자열을 정의합니다. |

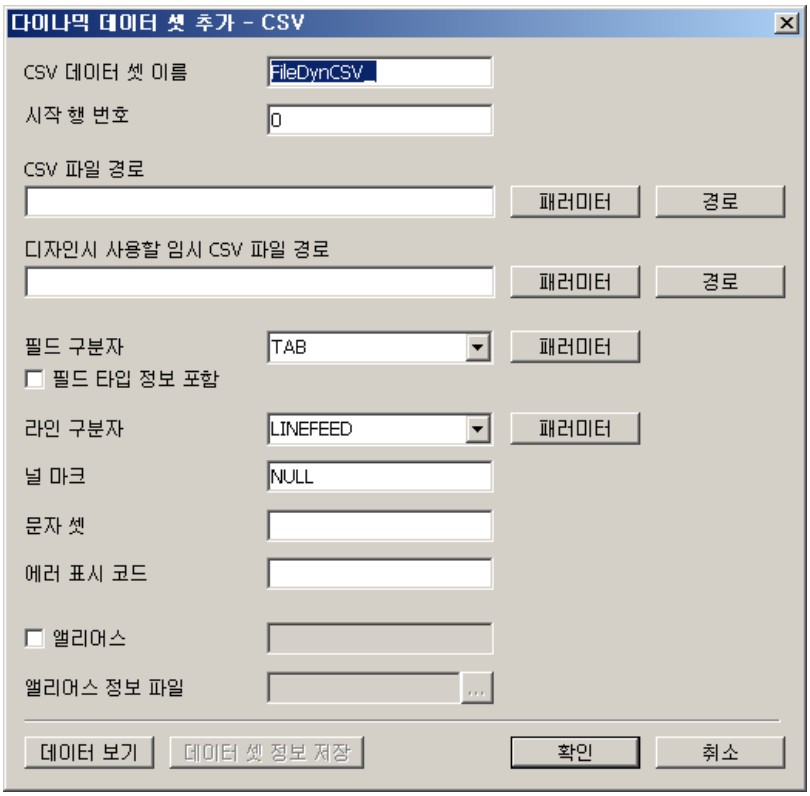
| | |
|---|---|
| **문자셋** | 파일의 문자 셋을 정의합니다. 설정하지 않을 경우 시스템의 기본 로케일 값이 이용됩니다. 사용 가능한 캐릭터셋은 본 매뉴얼 'Appendix. 문자셋과 유효한 앨리어스"을 참조하시기 바랍니다. |
| **에러 표시 코드** | 데이터 에러를 표시하기 위한 코드를 설정합니다. 설정한 코드 값이 파일의 첫 줄에 있을 경우 보고서 미리 보기를 수행하지 않고 에러 메시지 창을 실행합니다. 예를 들어 CSV 파일의 첫줄에 "ERR"라는 구문이 있고 에러 표시 코드를 "ERR"라고 입력하면 "ERR"을 데이터로 인식하지 않고 해당 파일이 적당한 CSV 데이터 파일이 아니라는 의미의 에러 메시지 창이 실행됩니다. |
| **패러미터** | 사용할 패러미터를 패러미터 마법사를 이용하여 설정합니다. 패러미터 마법사에 대한 자세한 내용은 본 매뉴얼 "Ⅴ. 쿼리 데이터 셋 디자인의 패러미터 마법사 이용하기"를 참조하시기 바랍니다. |
| **경로** | 가져올 CSV 파일을 선택합니다. |
| **데이터 보기** | 설정한 파일의 데이터를 미리보기합니다. |
| **데이터 셋 정보 저장** | 해당 데이터 셋의 정보(필드 이름, 필드 타입)를 CSV 파일로 저장합니다<br>※ 참고사항<br>▪ [데이터 셋 정보 저장] 버튼은 데이터 셋을 수정할 경우에만 활성화됩니다.<br>▪ 필드 타입은 "필드 타입 정보 포함" 옵션을 체크하였을 경우에만 저장합니다. |
| **앨리어스** | 해당 데이터 셋 설정 정보를 별도의 연결 정보 파일에서 가져올 것인지 여부를 설정하며 앨리어스를 사용할 경우 오즈 서버의 datasource.properties 파일에 정의한 앨리어스 이름을 설정합니다. |
| **앨리어스 정보 파일** | 데이터 셋 설정 정보 파일 이름입니다.<br>앨리어스를 사용할 경우 앨리어스 정보가 있는 데이터 셋 설정 정보 파일명을 설정합니다. |

■ **다이나믹 데이터 셋 설정**

파일 스토어의 다이나믹 데이터 셋은 CSV 데이터 셋과 XML 데이터 셋을 원본으로 하기 때문에 '다이나믹 데이터 셋 추가 다이얼로그' 또한 CSV 데이터 셋과 XML 데이터 셋두 가지로 구성됩니다. 다이나믹 데이터 셋에 대한 자세한 내용은 본 매뉴얼의 "Ⅹ. 다이나믹 데이터 셋"을 참조하시기 바랍니다.




- 다이나믹 데이터 셋 추가 - CSV

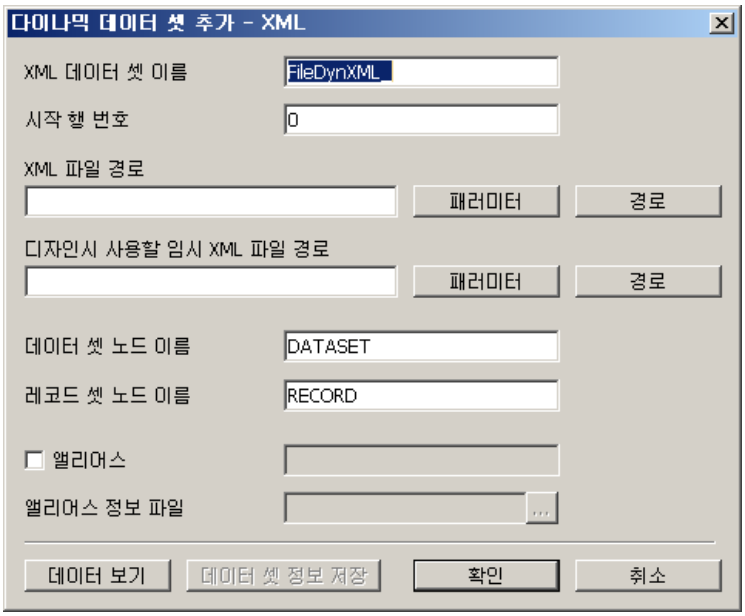



| 항목 | 설명 |
|---|---|
| **CSV 데이터 셋 이름** | 해당 데이터 셋에 대한 이름으로 사용자가 임의로 설정합니다. |
| **시작 행 번호** | 다이나믹 데이터 셋 생성시 ROW INDEX에 처음 시작되는 번호입니다. 기본 값은 '0'이며 각 ROW INDEX는 0부터 차례로 번호가 부여됩니다. 사용자가 0 이상의 값을 설정할 수 있습니다.<br>※ 주의사항 : 0 ~ 2147483647 이외의 값은 시작 행 번호로 설정할 수 없습니다. |
| **CSV 파일 경로** | 실제 서버에서 사용할 CSV 파일의 경로와 이름입니다. |
| **디자인시 사용할 임시 CSV 파일 경로** | CSV 파일의 경로가 오즈 서버와 디자이너에서 다른 경우 디자인시점에만 사용하는 임시 CSV 파일의 경로와 이름입니다. |

| | |
|---|---|
| **필드 구분자** | 필드 구분자를 선택합니다. TAB, SPACE, SEMICOLON, COLON, COMMA는 미리 정의된 구분자로 선택 가능하며 기타 문자에 대해서는 사용자가 직접 입력해서 설정할 수 있습니다. |
| **라인 구분자** | 라인 구분자를 선택합니다. LINEFEED는 미리 정의된 구분자로 기본 값이며 기타 문자에 대해서는 사용자가 직접 입력해서 설정할 수 있습니다.<br>또한, [패러미터] 버튼을 클릭하여 사용자 지정 패러미터로 라인 구분자를 설정할 수 있습니다. |
| **필드 타입 정보 포함** | 해당 CSV 파일에 필드에 대한 타입 정보가 포함되어 있는지 유무를 선택합니다. 체크되지 않은 경우 모든 필드의 타입은 'VARCHAR'로 간주됩니다. |
| **널 마크** | NULL 데이터를 구분하기 위한 문자열을 정의 합니다. |
| **문자셋** | 파일의 문자 셋을 정의합니다. 설정하지 않을 경우 시스템의 기본 로케일 값이 이용됩니다. 사용 가능한 캐릭터 셋은 본 매뉴얼 'Appendix. 문자셋과 유효한 앨리어스"을 참조하시기 바랍니다. |
| **에러 표시 코드** | 데이터 에러를 표시하기 위한 코드를 설정합니다. 설정한 코드 값이 파일의 첫 줄에 있을 경우 보고서 미리 보기를 수행하지 않고 에러 메시지 창을 실행합니다. 예를 들어 CSV 파일의 첫줄에 "ERR"라는 구문이 있고 에러 표시 코드를 "ERR"라고 입력하면 "ERR"을 데이터로 인식하지 않고 해당 파일이 적당한 CSV 데이터 파일이 아니라는 의미의 에러 메시지 창이 실행됩니다. |
| **패러미터** | 사용할 패러미터를 패러미터 마법사를 이용하여 설정합니다. 패러미터 마법사에 대한 자세한 내용은 본 매뉴얼 "V. 쿼리 데이터 셋 디자인의 패러미터 마법사 이용하기 "를 참조하시기 바랍니다. |
| **경로** | 가져올 CSV 파일을 선택합니다. |
| **데이터 보기** | 설정한 파일의 데이터를 미리보기합니다. |
| **데이터 셋 정보 저장** | 해당 데이터 셋의 정보(필드 이름, 필드 타입)를 CSV 파일로 저장합니다<br>※ 참고사항<br>▪ [데이터 셋 정보 저장] 버튼은 데이터 셋을 수정할 경우에만 활성화됩니다.<br>▪ 필드 타입은 "필드 타입 정보 포함" 옵션을 체크하였을 경우에만 저장합니다. |

| | |
|---|---|
| **앨리어스** | 해당 데이터 셋 설정 정보를 별도의 연결 정보 파일에서 가져올 것인지 여부를 설정하며 앨리어스를 사용할 경우 오즈 서버의 datasource.properties 파일에 정의한 앨리어스 이름을 설정합니다. |
| **앨리어스 정보 파일** | 데이터 셋 설정 정보 파일 이름입니다.<br>앨리어스를 사용할 경우 앨리어스 정보가 있는 데이터 셋 설정 정보 파일명을 설정합니다. |

- 다이나믹 데이터 셋 추가 - XML




| 항목 | 설명 |
|---|---|
| **XML 데이터 셋 이름** | 해당 데이터 셋에 대한 이름으로 사용자가 임의로 설정합니다. |
| **시작 행 번호** | 다이나믹 데이터 셋 생성시 ROW INDEX에 처음 시작되는 번호입니다. 기본 값은 '0'이며 각 ROW INDEX는 0부터 차례로 번호가 부여됩니다. 사용자가 0 이상의 값을 설정할 수 있습니다.<br>※ 주의사항 : 0 ~ 2147483647 이외의 값은 시작 행 번호로 설정할 수 없습니다. |
| **XML 파일 경로** | 실제 서버에서 사용할 XML 파일의 경로와 이름입니다. |
| **디자인시 사용할 임시 XML 파일명(경로)** | XML 파일의 경로가 오즈 서버와 디자이너에서 다른 경우 디자인시점에만 사용하는 임시 XML 파일의 경로와 이름입니다. |

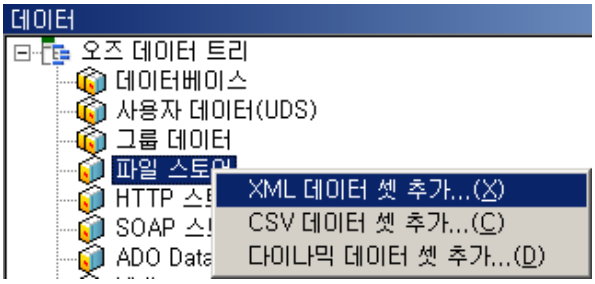
| | |
|---|---|
| **데이터 셋 노드 이름** | 해당 XML 파일에서 데이터 셋을 나타내는 노드 이름입니다. 기본 값은 'DATASET' 입니다. |
| **레코드셋 노드 이름** | 해당 XML 파일에서 레코드셋을 나타내는 노드 이름입니다. 기본 값은 'RECORD'입니다. |
| **패러미터** | 사용할 패러미터를 패러미터 마법사를 이용하여 설정합니다. 패러미터 마법사에 대한 자세한 내용은 본 매뉴얼 "Ⅴ. 쿼리 데이터 셋 디자인의 패러미터 마법사 이용하기"를 참조하시기 바랍니다. |
| **경로** | 가져올 XML 파일을 선택합니다. |
| **데이터 보기** | 설정한 파일의 데이터를 미리보기합니다. |
| **데이터 셋 정보 저장** | 해당 데이터 셋의 정보(필드 이름, 필드 타입)를 XML 파일로 저장합니다.<br>※ 참고사항<br>▪ [데이터 셋 정보 저장] 버튼은 데이터 셋을 수정할 경우에만 활성화됩니다.<br>▪ 데이터 셋 정보 저장 시 인코딩이 UTF-8로 설정됩니다. |
| **앨리어스** | 해당 데이터 셋 설정 정보를 별도의 연결 정보 파일에서 가져올 것인지 여부를 설정하며 앨리어스를 사용할 경우 오즈 서버의 datasource.properties 파일에 정의한 앨리어스 이름을 설정합니다. |
| **앨리어스 정보 파일** | 데이터 셋 설정 정보 파일 이름입니다.<br>앨리어스를 사용할 경우 앨리어스 정보가 있는 데이터 셋 설정 정보 파일명을 설정합니다. |

## 파일 스토어 연결

다음은 오즈 쿼리 디자이너와 파일 스토어 데이터베이스에 연결하여 파일 데이터 셋(XML/CSV/다이나믹 데이터 셋)을 생성하는 방법을 순차적으로 설명합니다.

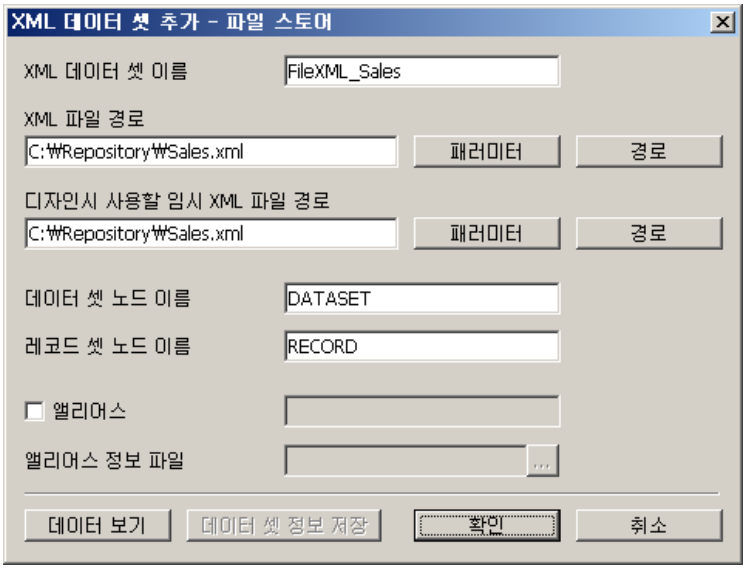
### ■ XML 데이터 셋

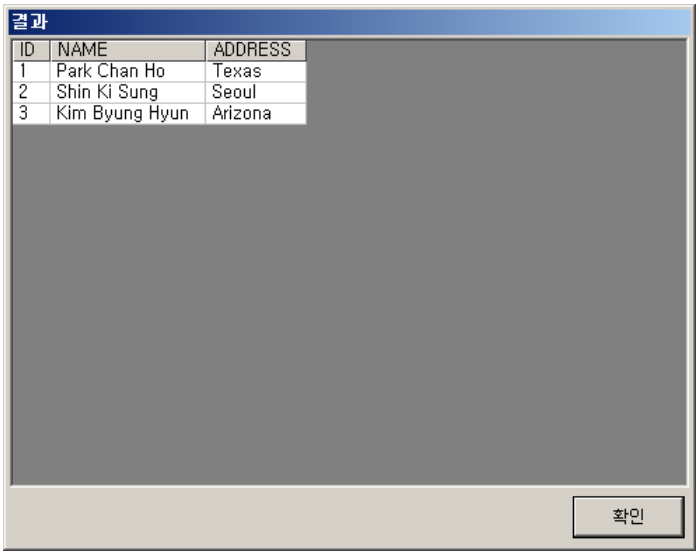
**Step 1** 오즈 쿼리 디자이너 데이터 창에서 [파일 스토어]를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [XML 데이터 셋 추가] 메뉴를 클릭합니다.



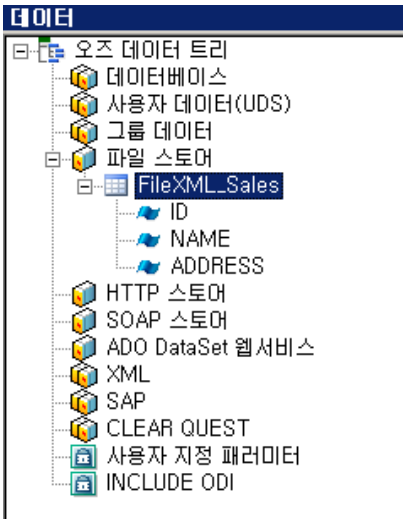

**Step 2**

'파일 스토어 설정' 관련 설명을 참조하여 'XML 데이터 셋 추가 다이얼로그'의 해당 옵션에 적절한 값을 설정한 후 [파일 보기] 버튼를 이용하여 데이터를 확인합니다.



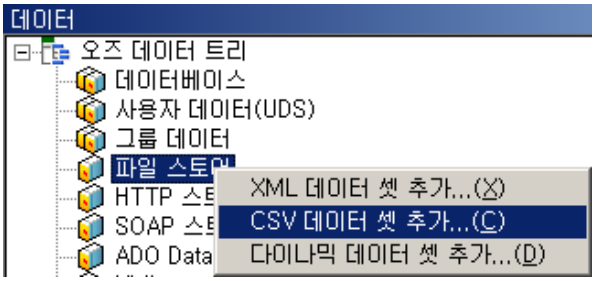



**Step 3** [확인] 버튼을 클릭하면 [파일 스토어]에 추가된 XML 데이터 셋 정보가 표시됩니다.

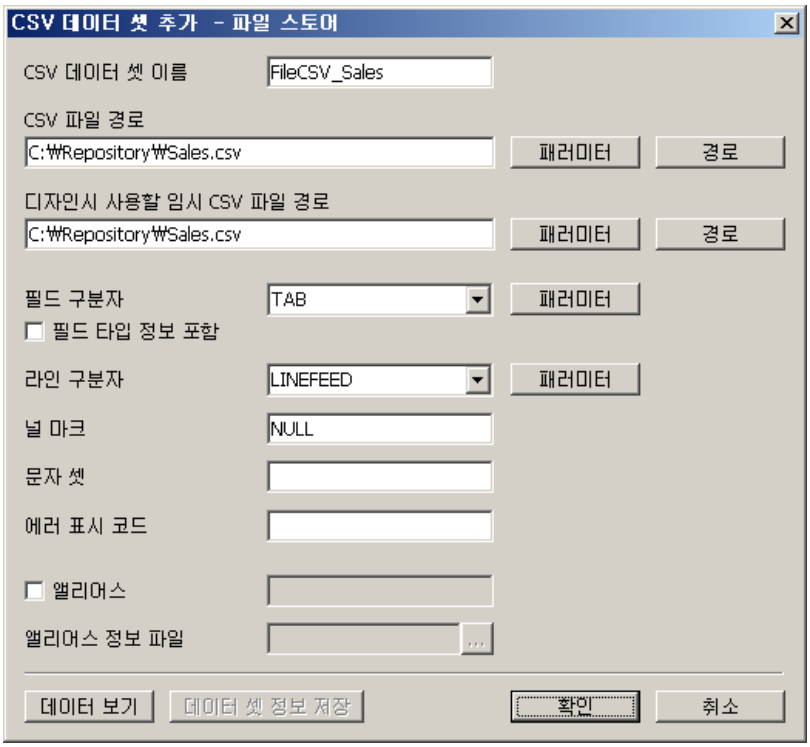


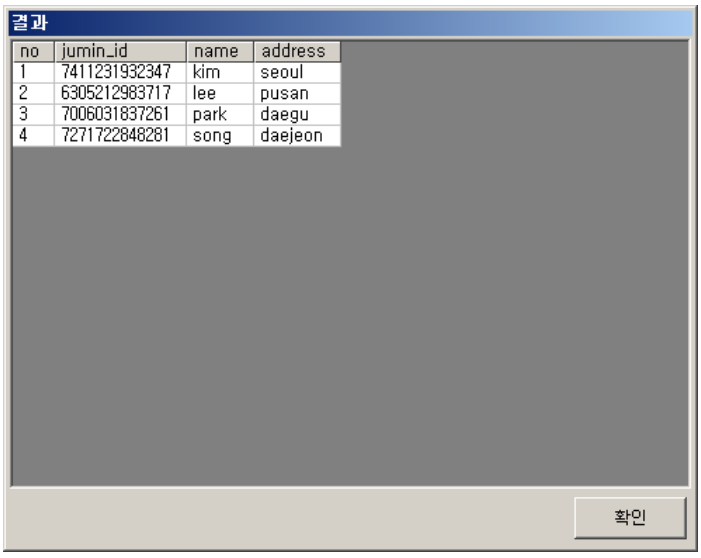

■ **CSV 데이터 셋**

**Step 1** 오즈 쿼리 디자이너 데이터 창에서 [파일 스토어]를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [CSV 데이터 셋 추가] 메뉴를 클릭합니다.

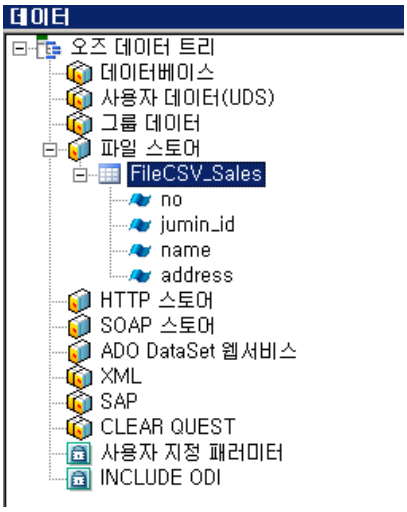



**Step 2** '파일 스토어 설정' 관련 설명을 참조하여 'CSV 데이터 셋 추가 다이얼로그'의 해당 옵션에 적절한 값을 설정한 후 [파일 보기] 버튼을 이용하여 데이터를 확인합니다.



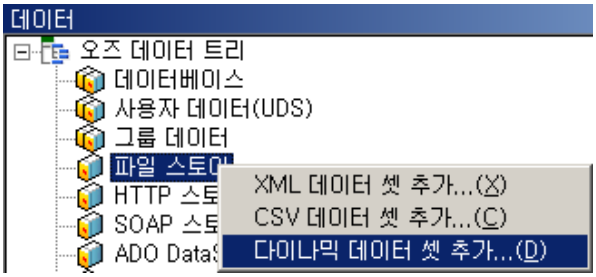




**Step 3** [확인] 버튼을 클릭하면 [파일 스토어]에 추가된 CSV 데이터 셋 정보가 표시됩니다.

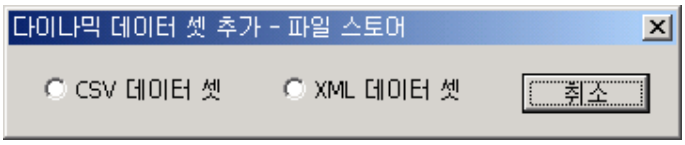



■ **다이나믹 데이터 셋**

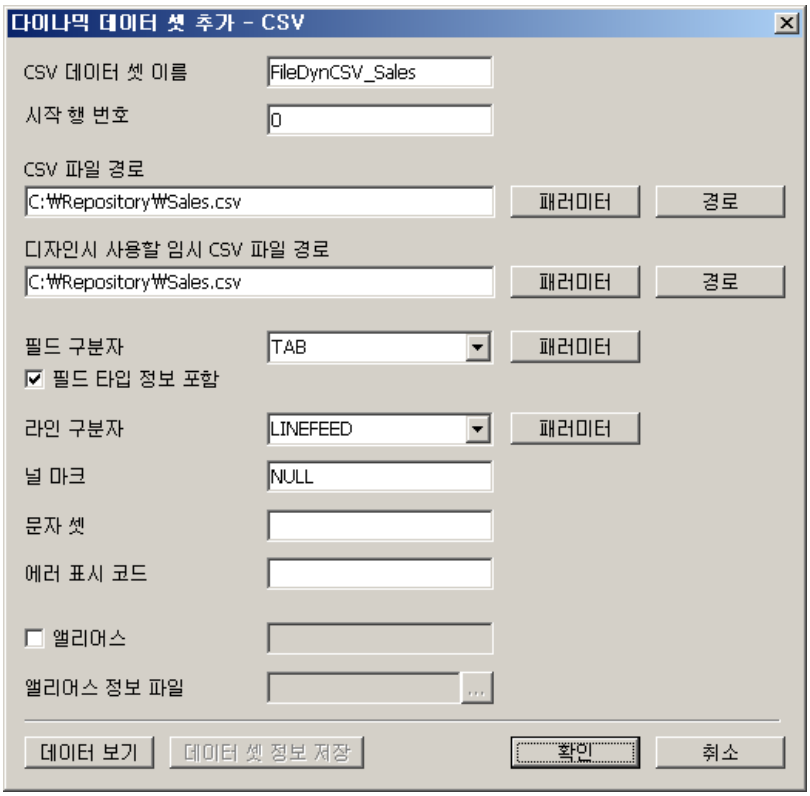
**Step 1** 오즈 쿼리 디자이너 데이터 창에서 [파일 스토어]를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [다이나믹 데이터 셋 추가] 메뉴를 클릭합니다.




**Step 2** '다이나믹 데이터 셋 추가 다이얼로그'에서 'CSV 데이터 셋'을 선택하면 '다이나믹 데이터 셋 추가-CSV 다이얼로그'가 나타납니다.

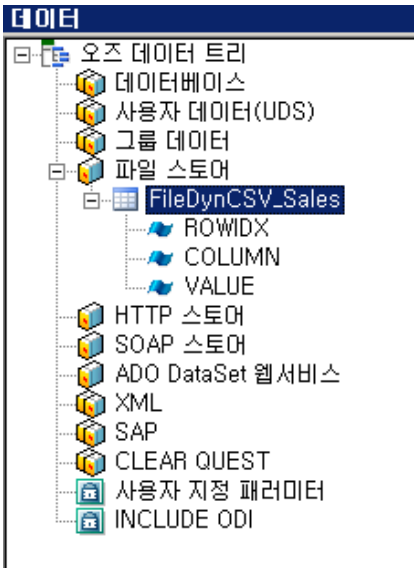



**Step 3** '파일 스토어 설정' 관련 설명을 참조하여 '다이나믹 데이터 셋 추가-CSV 다이얼로그'의 해당 옵션에 적절한 값을 설정한 후 [파일 보기] 버튼을 이용하여 데이터를 확인합니다.




**Step 4**

[확인] 버튼을 클릭하면 [파일 스토어]에 추가된 다이나믹 데이터 셋 정보가 표시됩니다.



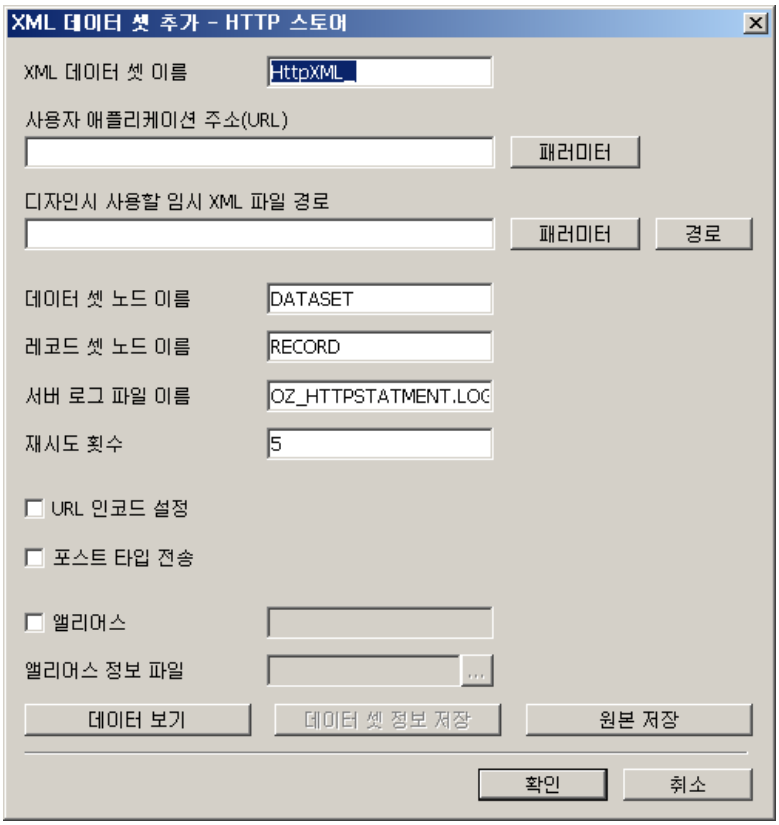
# HTTP 스토어 이용

JSP나 ASP와 같은 사용자 애플리케이션에 의해 생성되는 HTTP 스트림 데이터를 데이터 소스로 연결하고 데이터 셋을 생성하는 방법에 대해 설명합니다.

## HTTP 스토어 설정 창

XML이나 CSV 타입의 파일을 데이터 소스로 연동할 때 사용하는 창입니다.

■ **XML 데이터 셋 설정**

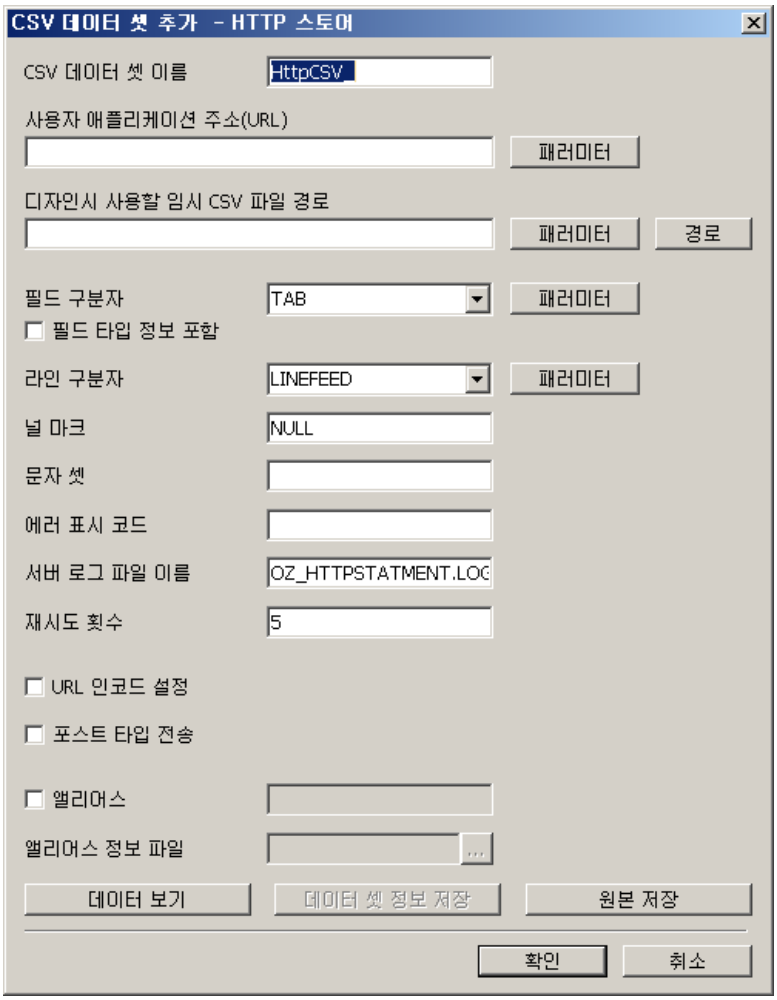



| 항목 | 설명 |
|---|---|
| **XML 데이터 셋 이름** | 해당 데이터 셋에 대한 이름으로 사용자가 임의로 설정합니다. |
| **사용자 애플리케이션 주소(URL)** | 데이터를 가져올 파일 또는 사용자 애플리케이션의 HTTP 주소입니다. |

| | |
|---|---|
| **디자인시 사용할 임시 XML 파일 경로** | 디자인 시점에만 사용할 수 있는 로컬 시스템의 파일 경로와 이름입니다. |
| **서버 로그 파일 이름** | ODI가 서버에서 바인딩될 때 전달받은 데이터에 대한 로그를 서버에 남길 경우 서버 로그 파일명을 설정합니다. 파일 이름에는 경로를 설정하지 않고 파일명만 입력하여야 합니다. |
| **재시도 횟수** | HTTP 스토어의 데이터 셋 사용 시 에러가 발생할 경우 연결을 재시도하는 횟수 (기본 값 : 5)<br>※ 참고사항 : 값을 설정하지 않거나 잘못된 값을 설정할 경우 기본 값으로 적용됩니다. |
| **URL 인코드 설정** | Server URL 인코딩 유무를 설정합니다. 데이터 요청시 URL의 패러미터 부분에 한국어나 기타 다국어 등이 포함되어 있을 경우 인코딩을 'TRUE'로 설정합니다. |
| **데이터 셋 노드 이름** | 해당 XML 파일에서 데이터 셋을 나타내는 노드 이름입니다. 기본 값은 'DATASET'입니다. |
| **레코드셋 노드 이름** | 해당 XML 파일에서 레코드 셋을 나타내는 노드 이름입니다. 기본 값은 'RECORD'입니다. |
| **포스트 타입 전송** | URL의 패러미터 전송시 GET 방식을 이용할 것인지 POST 방식을 이용할 것인지 선택합니다. |
| **페러미터** | 사용할 패러미터를 패러미터 마법사를 이용하여 설정합니다. 패러미터 마법사에 대한 자세한 내용은 본 매뉴얼 "Ⅴ. 쿼리 데이터 셋 디자인의 패러미터 마법사 이용하기"를 참조하시기 바랍니다. |
| **경로** | 가져올 XML 파일을 선택합니다. |
| **데이터 보기** | 설정한 파일의 데이터를 미리보기합니다.<br>※ 주의사항 : HTTP 스토어 XML 데이터 셋의 경우 로그 파일 이름이 설정되어 있으면 최대 행 수에 설정된 행 수와 관계 없이 모든 행을 가져온 후 데이터 보기가 되므로 데이터 보기 속도가 느려질 수 있습니다. |
| **원본 저장** | 데이터 셋의 원본 데이터를 파일로 저장합니다.<br>※ 참고사항 : [원본 저장] 버튼은 데이터 셋을 수정할 경우에만 활성화됩니다. |
| **데이터 셋 정보 저장** | 해당 데이터 셋의 정보(필드 이름, 필드 타입)를 XML 파일로 저장합니다.<br>※ 참고사항<br>▪ [데이터 셋 정보 저장] 버튼은 데이터 셋을 수정할 경우에만 활성화됩니다.<br>▪ 데이터 셋 정보 저장 시 인코딩이 UTF-8로 설정됩니다. |

| | |
|---|---|
| **앨리어스** | 해당 데이터 셋 설정 정보를 별도의 연결 정보 파일에서 가져올 것인지 여부를 설정하며 앨리어스를 사용할 경우 오즈 서버의 datasource.properties 파일에 정의한 앨리어스 이름을 설정합니다. |
| **앨리어스 정보 파일** | 데이터 셋 설정 정보 파일 이름입니다.<br>앨리어스를 사용할 경우 앨리어스 정보가 있는 데이터 셋 설정 정보 파일명을 설정합니다. |

■ **CSV 데이터 셋 설정**




| 항목 | 설명 |
|---|---|
| **CSV 데이터 셋 이름** | 해당 데이터 셋에 대한 이름으로 사용자가 임의로 설정합니다. |
| **사용자 애플리케이션 주소(URL)** | 데이터를 가져올 파일 또는 사용자 애플리케이션의 HTTP 주소입니다. |

| | |
|---|---|
| **디자인시 사용할 임시 CSV 파일 경로** | 디자인 시에만 사용할 수 있는 로컬 시스템의 파일 경로와 이름입니다. |
| **서버 로그 파일 이름** | ODI가 서버에서 바인딩될 때 전달받은 데이터에 대한 로그를 서버에 남길 경우 서버 로그 파일명을 설정합니다. 파일 이름에는 경로를 설정하지 않고 파일명만 입력하여야 합니다. |
| **재시도 횟수** | HTTP 스토어의 데이터 셋 사용 시 에러가 발생할 경우 연결을 재시도하는 횟수 (기본 값 : 5)<br>※ 참고사항 : 값을 설정하지 않거나 잘못된 값을 설정할 경우 기본 값으로 적용됩니다. |
| **URL 인코드 설정** | Server URL 인코딩 유무를 설정합니다. 데이터 요청시 URL의 패러미터 부분에 한국어나 기타 다국어 등이 포함되어 있을 경우 인코딩을 'TRUE'로 설정합니다. |
| **필드 구분자** | 필드 구분자를 선택합니다. TAB, SPACE, SEMICOLON, COLON, COMMA는 미리 정의된 구분자로 선택 가능하며 기타 문자에 대해서는 사용자가 직접 입력해서 설정할 수 있습니다. |
| **라인 구분자** | 라인 구분자를 선택합니다. LINEFEED는 미리 정의된 구분자로 기본 값이며 기타 문자에 대해서는 사용자가 직접 입력해서 설정할 수 있습니다.<br>또한, [패러미터] 버튼을 클릭하여 사용자 지정 패러미터로 라인 구분자를 설정할 수 있습니다. |
| **필드 타입 정보 포함** | 해당 CSV 파일에 필드에 대한 타입 정보가 포함되어 있는지 유무를 선택합니다. 'FALSE'인 경우 모든 필드의 타입은 'VARCHAR'로 간주됩니다. (TRUE/FALSE) |
| **널 마크** | NULL 데이터를 구분하기 위한 문자열을 정의 합니다. |
| **문자셋** | 해당 파일의 문자 셋을 정의합니다. 로컬 시스템의 문자 셋과 동일할 경우 설정하지 않아도 됩니다. 사용 가능한 캐릭터셋은 본 매뉴얼 'Appendix. 문자셋과 유효한 앨리어스"을 참조하시기 바랍니다. |
| **에러 표시 코드** | 데이터 에러를 표시하기 위한 코드를 설정합니다. 설정한 코드 값이 파일의 첫 줄에 있을 경우 보고서 미리보기를 수행하지 않고 에러 메시지를 창에 띄웁니다. 예를 들어 CSV파일의 첫줄에 "ERR"라는 구문이 있고 에러 표시 코드를 "ERR"라고 입력하면 "ERR"을 데이터로 인식하지 않고 해당 파일이 적당한 CSV 데이터 파일이 아니라는 의미의 에러 메시지 창이 실행됩니다. |
| **포스트 타입 전송** | URL의 패러미터 전송시 GET 방식을 이용할 것인지 POST 방식을 이용할 것인지 선택합니다. |

| | |
|---|---|
| **패러미터** | 사용할 패러미터를 패러미터 마법사를 이용하여 설정합니다. 패러미터 마법사에 대한 자세한 내용은 본 매뉴얼 "Ⅴ. 쿼리 데이터 셋 디자인의 패러미터 마법사 이용하기"를 참조하시기 바랍니다. |
| **경로** | 가져올 CSV 파일을 선택합니다. |
| **데이터 보기** | 설정한 파일의 데이터를 미리보기합니다. |
| **원본 저장** | 데이터 셋의 원본 데이터를 파일로 저장합니다.<br>※ 참고사항 : [원본 저장] 버튼은 데이터 셋을 수정할 경우에만 활성화됩니다. |
| **데이터 셋 정보 저장** | 해당 데이터 셋의 정보(필드 이름, 필드 타입)를 CSV 파일로 저장합니다.<br>※ 참고사항<br>▪ [데이터 셋 정보 저장] 버튼은 데이터 셋을 수정할 경우에만 활성화됩니다.<br>▪ 필드 타입은 "필드 타입 정보 포함" 옵션을 체크하여 데이터 셋을 추가 또는 수정하였을 경우에만 저장합니다. |
| **앨리어스** | 해당 데이터 셋 설정 정보를 별도의 연결 정보 파일에서 가져올 것인지 여부를 설정하며 앨리어스를 사용할 경우 오즈 서버의 datasource.properties 파일에 정의한 앨리어스 이름을 설정합니다. |
| **앨리어스 정보 파일** | 데이터 셋 설정 정보 파일 이름입니다.<br>앨리어스를 사용할 경우 앨리어스 정보가 있는 데이터 셋 설정 정보 파일명을 설정합니다. |

■ **다이나믹 데이터 셋 설정**

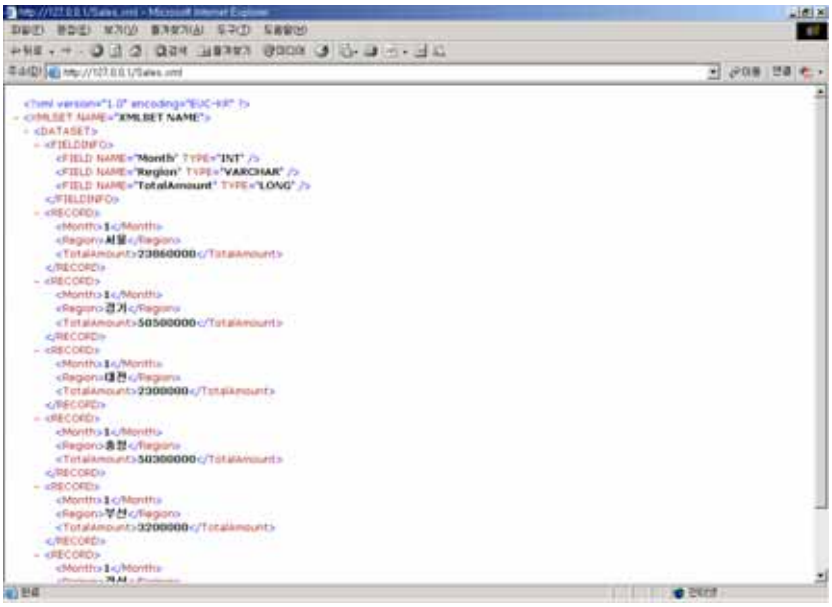
앞의 파일 스토어의 "다이나믹 데이터 셋 추가"을 참조하시기 바랍니다.

## HTTP 스토어 연결

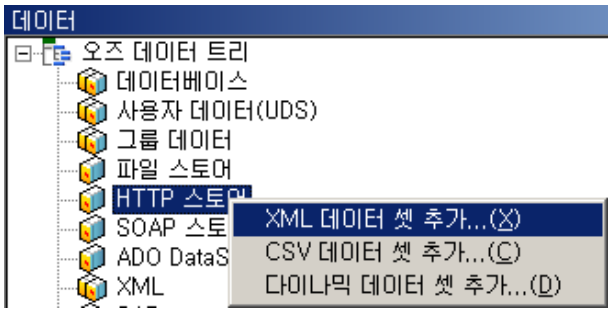
다음은 오즈 쿼리 디자이너와 HTTP 스토어를 연결하여 HTTP 데이터 셋(XML/CSV/다이나믹 데이터 셋)을 생성하는 방법을 순차적으로 설명합니다.

■ **XML 데이터 셋**

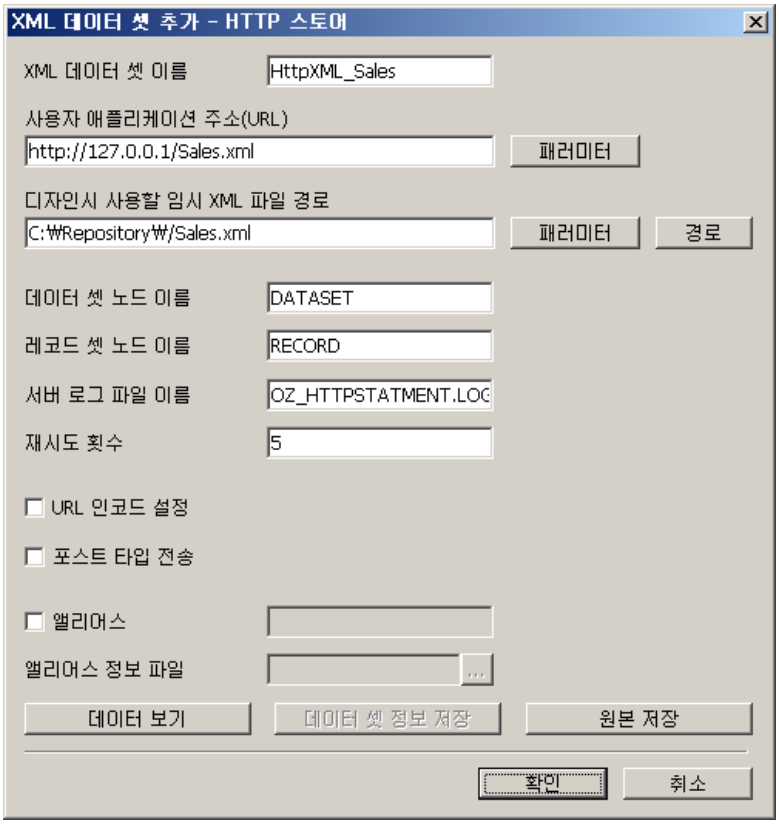
**Step 1** 아래의 예와 같이 웹 브라우저를 통해 XML 파일(Sales.xml)을 받아올 수 있는 환경을 가정합니다.

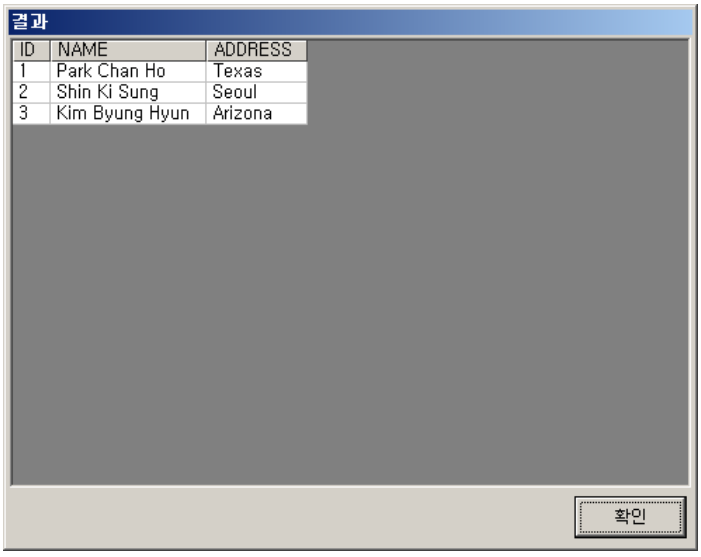


**Step 2** 오즈 쿼리 디자이너 데이터 창에서 [HTTP 스토어]를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [XML 데이터 셋 추가] 메뉴를 클릭합니다.

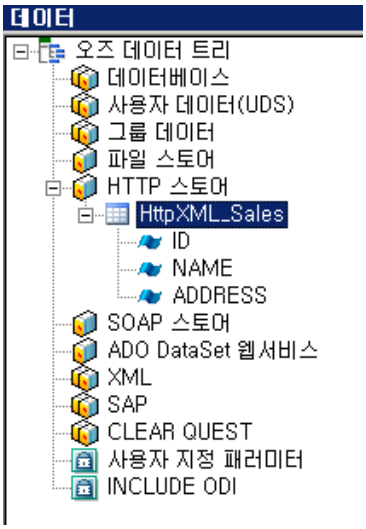



**Step 3** 'HTTP 스토어 설정' 관련 설명을 참조하여 'XML 데이터 셋 추가 다이얼로그'의 해당 옵션에 적절한 값을 설정한 후 [파일 보기] 버튼을 이용하여 데이터를 확인합니다.



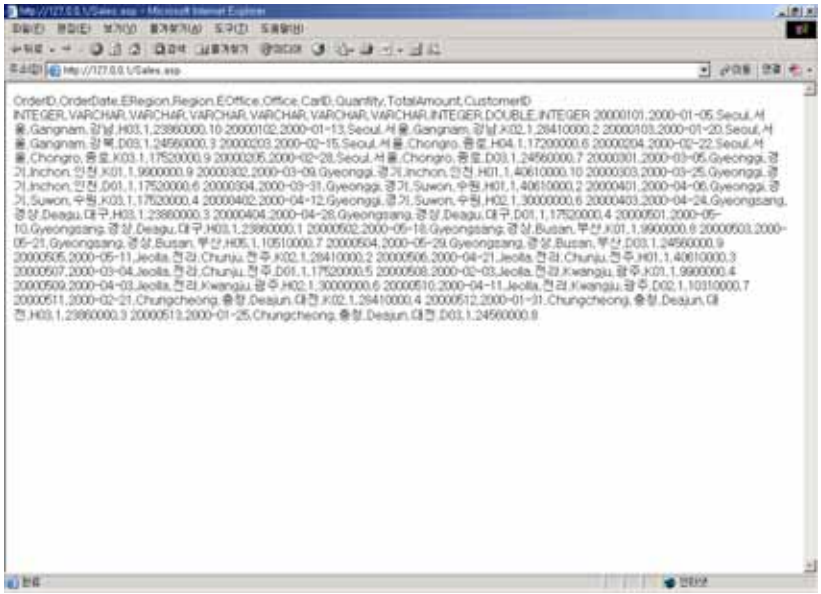

**Step 4** [확인] 버튼을 클릭하면 [HTTP 스토어]에 추가된 XML 데이터 셋 정보가 표시됩니다.



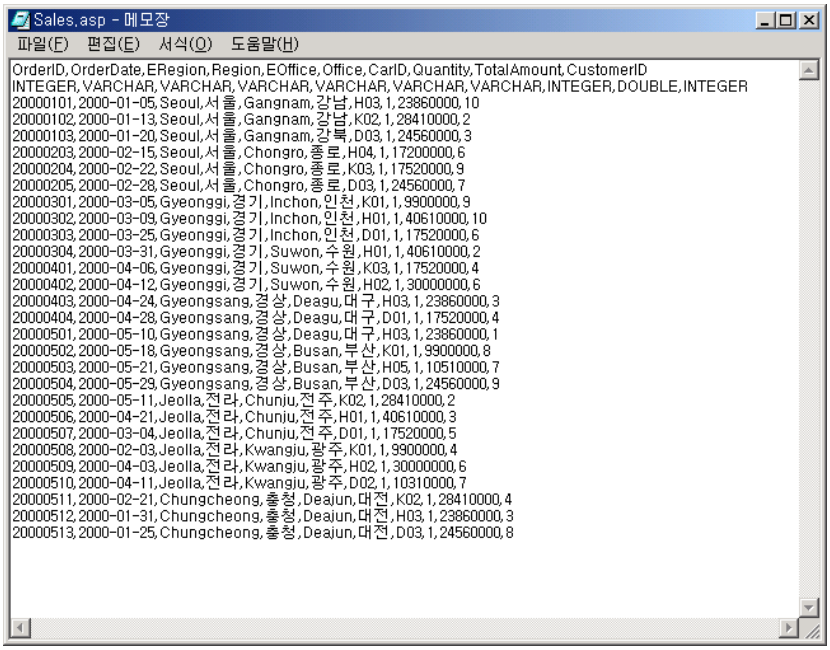

■ **CSV 데이터 셋**

**Step 1** 아래의 보기와 같이 HTTP를 통해 CSV 파일 형태의 데이터를 전송하는 사용자 애플리케이션을 가정합니다.

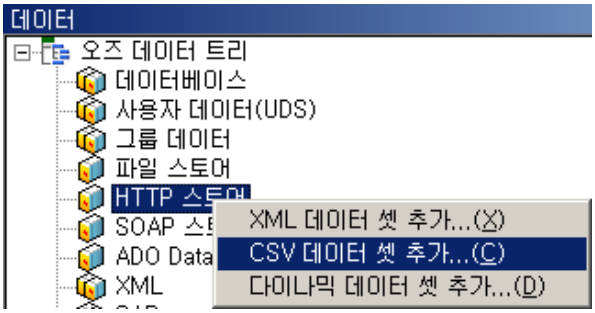
http를 이용하여 CSV 정보를 전달 받을 경우 브라우저 환경은 다음과 같습니다.



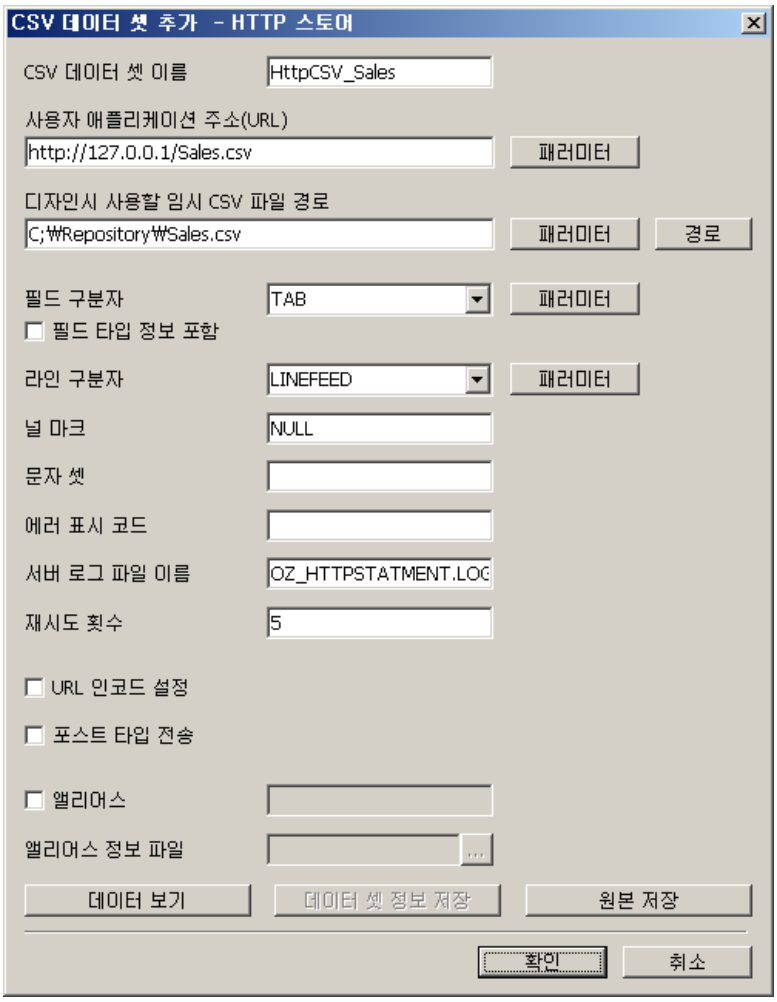
위의 브라우저에 표시된 CSV 파일은 다음과 같습니다.

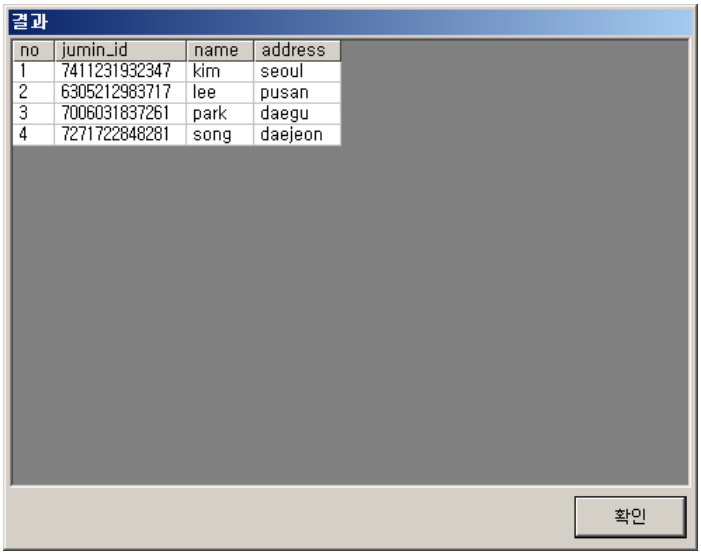



**Step 2** 오즈 쿼리 디자이너 데이터 창에서 [HTTP 스토어]를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [CSV 데이터 셋 추가] 메뉴를 클릭합니다.




**Step 3** 'HTTP 스토어 설정' 관련 설명을 참조하여 'CSV 데이터 셋 추가 다이얼로그'의 해당 옵션에 적절한 값을 설정한 후 [파일 보기]를 이용하여 데이터를 확인합니다.


CSV 데이터 셋 추가 - HTTP 스토어
CSV 데이터 셋 이름
HttpCSV_Sales
사용자 애플리케이션 주소(URL)
http://127.0.0.1/Sales.csv
패러미터
디자인시 사용할 임시 CSV 파일 경로
C;₩Repository₩Sales.csv
패러미터
경로
필드 구분자
TAB
패러미터
필드 타입 정보 포함
라인 구분자
LINEFEED
패러미터
널 마크
NULL
문자 셋
에러 표시 코드
서버 로그 파일 이름
OZ_HTTPSTATMENT.LOG
재시도 횟수
5
URL 인코드 설정
포스트 타입 전송
앨리어스
앨리어스 정보 파일
데이터 보기
데이터 셋 정보 저장
원본 저장
확인
취소


결과
no
jumin_id
name
address
1
7411231932347
kim
seoul
2
6305212983717
lee
pusan
3
7006031837261
park
daegu
4
7271722848281
song
daejeon
확인

**Step 4** [확인] 버튼을 클릭하면 [HTTP 스토어]에 추가된 CSV 데이터 셋 정보가 표시됩니다.




■ **다이나믹 데이터 셋 설정**

파일 스토어의 "다이나믹 데이터 셋 추가"를 참조하시기 바랍니다.

# X. SOAP 스토어

- SOAP 개요
- SOAP 스토어 이용

# SOAP 개요

## SOAP 개요

SOAP(Simple Object Access Protocol)은 분산 환경에서 구조화된 정보를 교환하기 위한 경량 프로토콜 통신 규약입니다.

SOAP은 확장성 있는 메시지를 정의하기 위해 XML을 사용하며, 이 XML로 통해 객체(Object) 데이터가 손쉽게 교환될 수 있는 메시지 구조를 제공합니다.

## SOAP 스토어 개요

OZ에서는 비지니스 서비스를 기술하여 웹에서 비즈니스들끼리 접근하는 방법을 제공하기 위해 만든 WSDL(Web Service Description Language)을 통하여 웹서비스들을 사용할 수 있습니다.

# SOAP 스토어 이용

다음은 SOAP 스토어를 이용하여 웹 서비스를 통해 데이터 셋을 생성하는 방법에 대해 설명합니다.

※ 주의사항 : 쿼리 디자이너에서 SOAP 스토어 기능을 사용하기 위해서는 SOAP Toolkit 3.0 라이브러리가 필요합니다. 또한 MSSOAP30.dll, WHSC30.dll 파일을 쿼리 디자이너가 설치된 경로의 bin/soap/register_soap.bat 파일을 실행하여 레지스트리에 등록하여야 합니다.

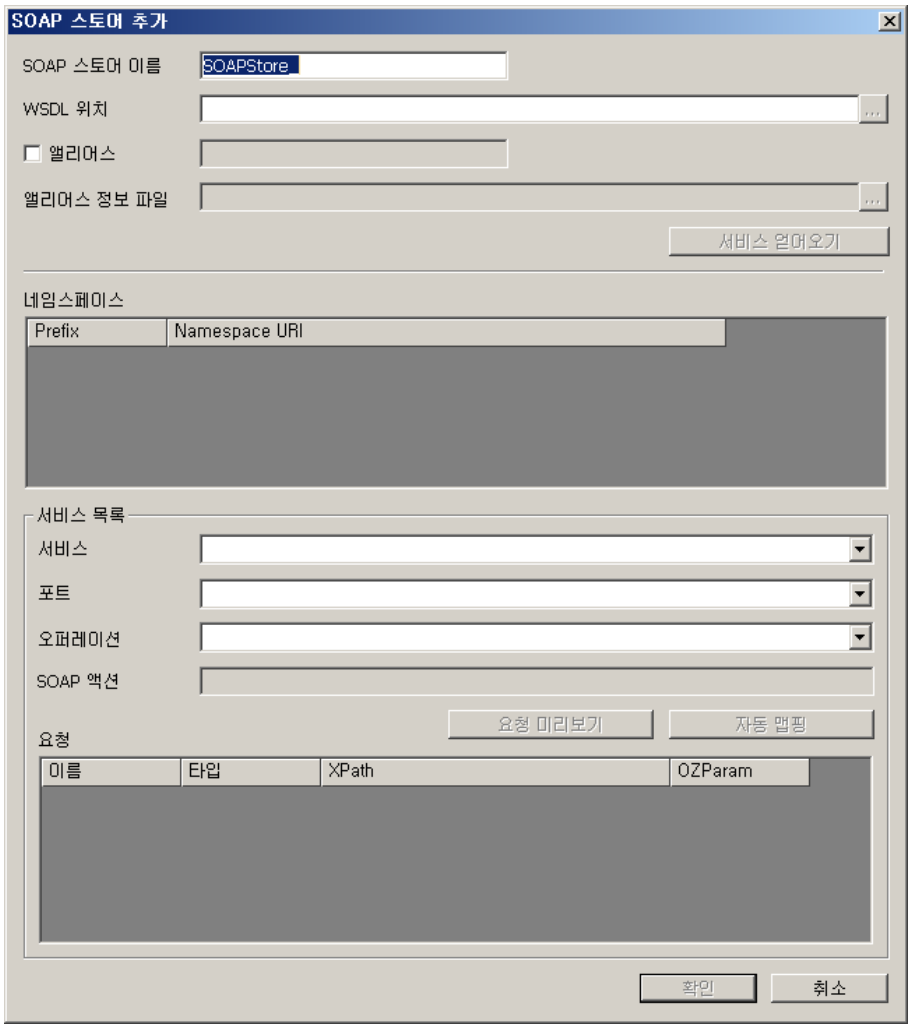
※ 제약사항

- WSDL 정의의 메시지에 논리적 타입을 지정해 주기 위해서는 part와 type이 1:1 매핑이 되어야하나 n:1인 WSDL인 경우 첫번째 part와 type이 1:1 매핑됩니다.
- 자동으로 사용자 패러미터 필드 지정 시 XML 스키마의 "DateTime" 타입은 "VARCHAR" 타입으로 생성됩니다.
- 디자인 모드를 매뉴얼 모드에서 디자인 뷰 모드로는 변경할 수 없습니다.

## SOAP 스토어 설정 창

웹 서비스와 연동하여 데이터 소스 가져올 때 사용하는 창입니다.

■ **SOAP 스토어 설정**




| 항목 | 설명 |
|---|---|
| **SOAP 스토어 이름** | 추가할 SOAP 스토어 이름을 입력합니다. |
| **WSDL 위치** | WSDL 문서 위치를 입력합니다. |
| **앨리어스** | 해당 데이터 셋 설정 정보를 별도의 연결 정보 파일에서 가져올 것인지 여부를 설정하며 앨리어스를 사용할 경우 오즈 서버의 datasource.properties 파일에 정의한 앨리어스 이름을 설정합니다. |
| **앨리어스 정보 파일** | 데이터 셋 설정 정보 파일 이름입니다.<br>앨리어스를 사용할 경우 앨리어스 정보가 있는 데이터 셋 설정 정보 파일명을 설정합니다. |

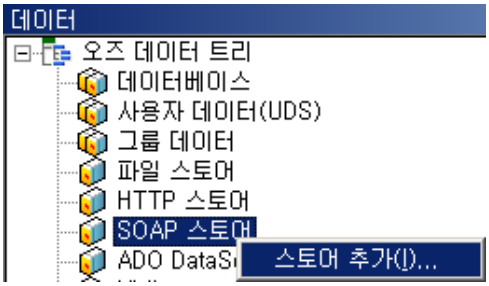
| | | |
|---|---|---|
| **서비스 얻어오기** | 서비스에 사용된 네임스페이스 목록과 서비스 목록을 표시합니다.<br>※ 참고사항 : "WSDL 위치" 옵션 값을 입력하여야 해당 버튼이 활성화되며 입력한 WSDL 주소에 접속이 되어야만 네임스페이스 목록과 서비스 목록이 표시됩니다. | |
| **네임스페이스** | WDSL의 타입에 정의된 Prefix와 NameSpace URI를 리스트 형태로 표시하며 사용하고자 하는 Prefix를 설정합니다. | |
| **서비스 목록** | **서비스** | WSDL에서 제공하는 서비스 명이 콤보박스 리스트에 표시되며 사용하고자 하는 서비스를 설정합니다. |
| | **포트** | 서비스에 속한 바인딩 목록이 콤보박스 리스트에 표시되며 사용하고자 하는 포트를 설정합니다. |
| | **오퍼레이션** | 포트에 속한 바인딩 목록이 콤보박스 리스트에 표시되며 사용하고자 하는 Operation을 설정합니다. |
| | **SOAP 액션** | 선택된 Operation의 SOAPaction 정보가 표시됩니다. |
| | **요청** | Operation의 입력 정보인 이름, 타입, XPath, OZParam을 리스트 형태로 표시하며 사용하고자 하는 OZParam을 "#OZParam.패러미터 명#" 형태로 입력합니다. |
| | **요청 미리보기** | 요청한 SOAP 메시지를 XML 형태로 표시하는 다이얼로그를 띄웁니다. |
| | **자동 매핑** | 요청 옵션의 "OZParam" 값을 자동으로 설정하며 설정한 값은 "#OZParam.<operation name>_<name>#" 형태로 표시됩니다. |

## SOAP 스토어 연결

다음은 오즈 쿼리 디자이너와 SOAP 스토어를 연결하고 데이터 셋을 생성하는 방법을 순차적으로 설명합니다.

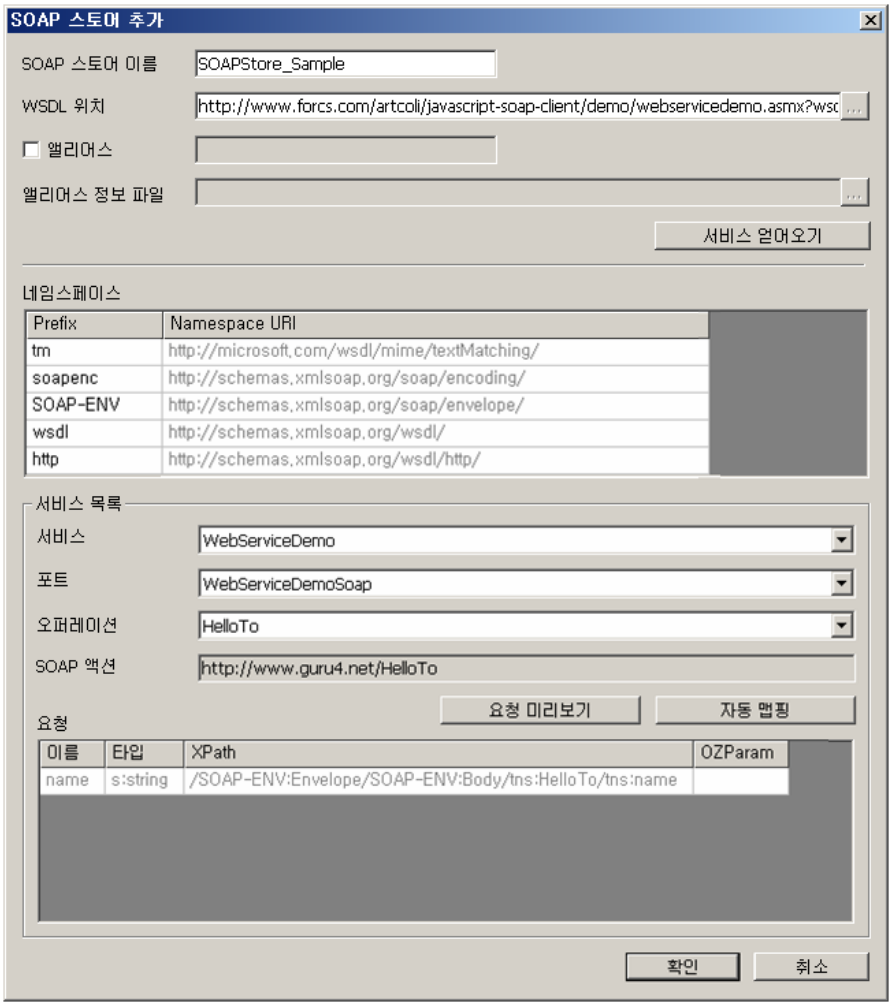
### ■ SOAP 스토어와 데이터 셋

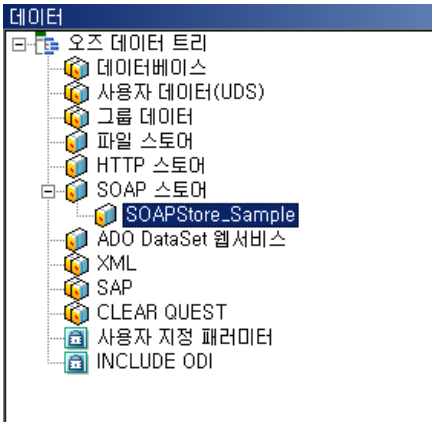
**Step 1** 오즈 쿼리 디자이너 데이터 창에서 [SOAP 스토어]을 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [SOAP 데이터 스토어 추가] 메뉴를 클릭합니다.




**Step 2**

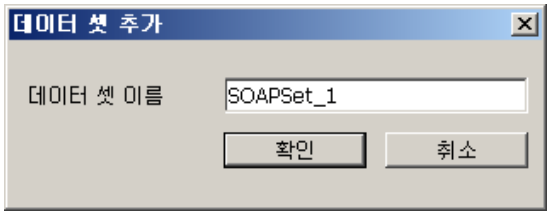
'SOAP 스토어 추가 다이얼로그'에서 SOAP 스토어 이름, WSDL 위치를 설정한 후 [서비스 얻어오기] 버튼을 클릭하여 네임스페이스 목록, 서비스 목록을 표시한 후 [확인] 버튼을 클릭하면 'SOAP 스토어'가 나타납니다.



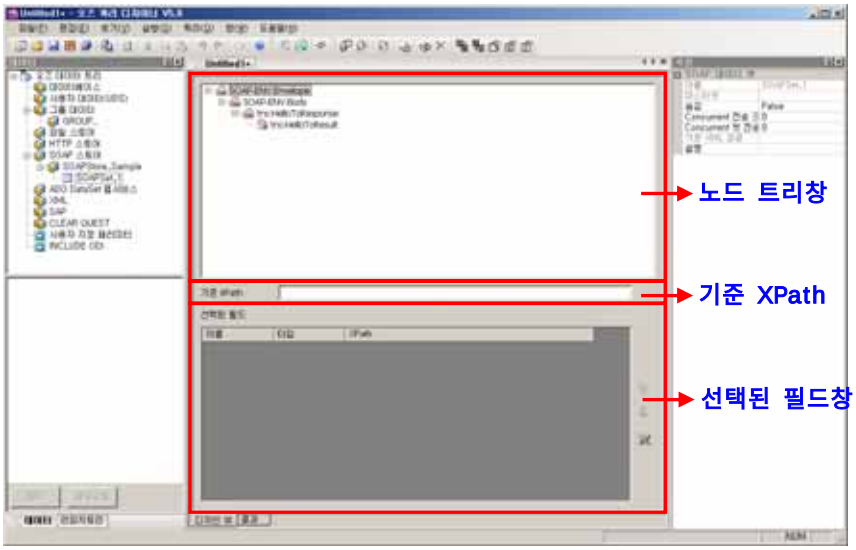



**Step 3**

데이터 창에 생성된 SOAP 스토어를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [데이터 셋 추가] 메뉴를 클릭하면 데이터 셋 이름을 입력하는 다이얼로그가 나타납니다.




데이터 셋 이름을 입력한 후 [확인] 버튼을 클릭하면 쿼리 디자이너의 쿼리 창에 아래 그림처럼 SOAP 스토어 정보를 설정할 수 있도록 변경됩니다.



- 노드 트리 창

  추가된 SOAP 스토어의 노드 구조가 트리 모양으로 표시됩니다.

  노드명 앞에는 노드 각각의 특성을 나타내는 아이콘이 있어 노드에 대한 식별을 용이하게 합니다. 노드의 아이콘에 대한 설명은 다음과 같습니다.

| 아이콘 | 설명 |
|---|---|
| | 애트리뷰트 |
| | 최소 발생 1, 최대 발생 1인 엘리먼트(기본 값) |
| | 최소 발생 0, 최대 발생 제한 없는 엘리먼트 |
| | 최소 발생 0, 최대 발생 1인 엘리먼트 |
| | 최소 발생 1, 최대 발생 제한 없는 엘리먼트 |
| | 트리에서 확장된 엘리먼트 |
| | 필드로 선택 불가능한 엘리먼트 |
| | 기준 XPath로 지정된 엘리먼트 |

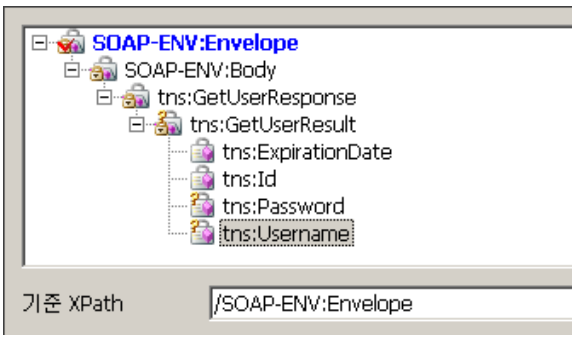
  노트 트리 창에서 기준 XPath를 지정한 후 노드를 더블 클릭하여 데이터 셋에 추가합니다.

- 기준 XPath

  **[디자인 모드]**

  노드 트리 창에서 기준 XPath로 지정할 노드를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [기준 XPath 지정] 메뉴를 클릭합니다. 기준 XPath로 지정된 노드는 아이콘이 체크된 상태( )로 표시되고 노드 이름이 하이라이트되어 표시되며, 기준 XPath에 지정한 기준 XPath가 표시됩니다



이미 설정된 기준 XPath를 수정하려면 기준 XPath를 나타내는 텍스트 박스에서 다른 값으로 입력하거나 트리 창에서 기준 XPath로 변경할 노드 아이템을 마우스 오른쪽 버튼으로 선택하여 나타나는 [기준 XPath 지정] 메뉴를 클릭하여 수정할 수 있습니다.

**[매뉴얼 모드]**

기준 XPath에 기준 XPath를 직접 입력합니다.

기준 XPath 경로를 입력할 때 기준 XPath로 지정할 노드를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [노드 경로 복사] 메뉴를 이용하여 XPath 경로를 손쉽게 입력할 수 있습니다.

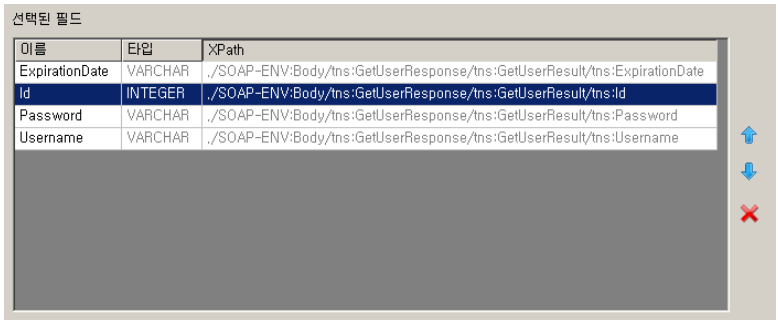



- 선택된 필드 창

**[디자인 모드]**

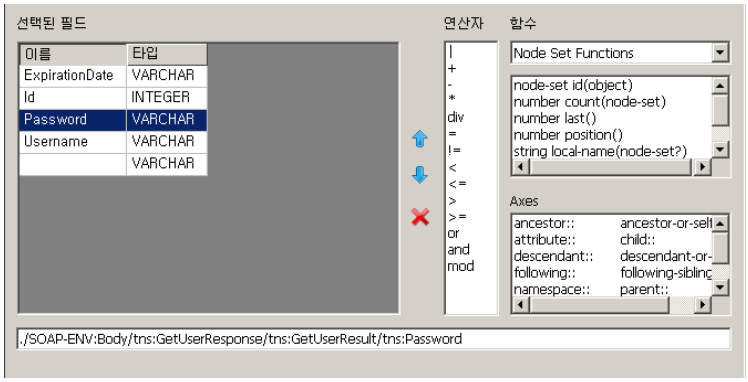
선택한 필드에 이름, 타입, XPath를 리스트 형태로 보여주며 선택한 필드의 순서를 변경( )하거나 삭제( )할 수 있습니다.

선택된 필드 창에서는 필드의 이름만 편집할 수 있으며, 필드의 타입과 XPath는 편집할 수 없습니다.




**[매뉴얼 모드]**

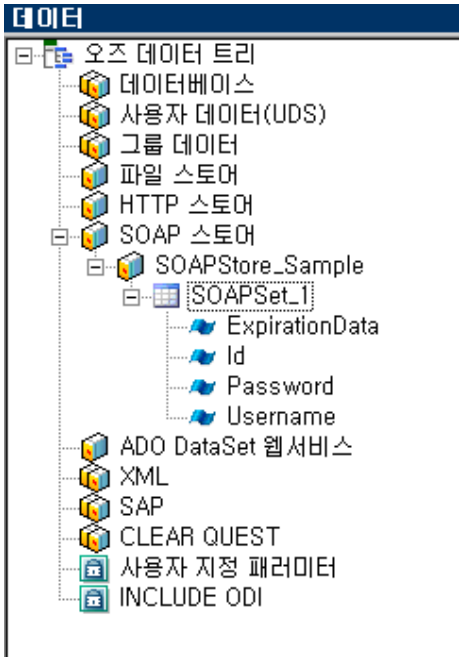
선택한 필드에 이름, 타입, XPath를 리스트 형태로 보여주며 선택한 필드의 순서를 변경( )하거나 삭제( )할 수 있습니다.




매뉴얼 모드에서는 노드 트리 창에서 노드를 더블 클릭하면 선택된 필드에 해당 노드 이름이 추가되지만 노드의 경로는 입력되지 않으므로, 사용자가 직접 입력하여야 합니다. 노드 트리 창에서 노드를 클릭하지 않고 추가하고자 할 경우에는 선택된 필드 맨 아래에 추가된 행에 노드 이름을 직접 입력하여야 합니다.

**Step 4**

데이터 셋에 필드를 모두 추가한 후 실행 아이콘( )을 클릭하면 아래 그림과 같이 추가된 필드를 확인할 수 있습니다.



# XI. ADO DataSet 웹서비스 스토어

- ADO DataSet 웹서비스 개요
- ADO DataSet 웹서비스 스토어 이용

# ADO DataSet 웹서비스 개요

## ADO DataSet 웹서비스 스토어 개요

OZ에서는 닷넷 웹 서비스에서 ADO DataSet 객체를 연동할 수 있도록 ADO DataSet 웹서비스 스토어를 제공합니다.

※ 제약사항

- ADO DataSet 웹서비스 스토어는 닷넷 서버에서만 지원되므로 ADO DataSet 웹서비스 스토어를 사용할 경우 서버 타입을 닷넷으로 설정하시기 바랍니다.
- 애플리케이션 또는 리포트 디자이너에서 폼 및 보고서를 디자인할 때 ADO DataSet 웹서비스 스토어가 포함된 ODI를 사용할 경우 로컬 서버 타입을 닷넷으로 설정하시기 바랍니다.
- ADO DataSet 웹서비스 스토어는 자바 스크립트 함수를 지원하지 않습니다.

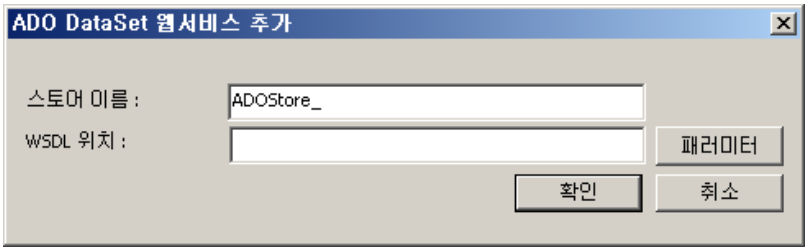
# ADO DataSet 웹서비스 스토어 이용

다음은 ADO DataSet 웹서비스 스토어를 이용하여 ADO DataSet 객체를 연동하여 데이터 셋을 생성하는 방법에 대해 설명합니다.

## ADO DataSet 웹서비스 스토어 설정 창

ADO DataSet 객체와 연동하여 데이터 소스 가져올 때 사용하는 창입니다.

■ **ADO DataSet 웹서비스 스토어 설정**




| 항목 | 설명 |
|---|---|
| **스토어 이름** | 추가할 ADO DataSet 웹서비스 스토어 이름을 입력합니다. |
| **WSDL 위치** | WSDL 위치를 입력합니다.<br>WSDL 위치는 직접 입력하거나 패러미터를 이용하여 설정할 수 있습니다. [패러미터] 버튼을 클릭하면 패러미터 마법사가 실행되며, 설정 가능한 패러미터를 선택할 수 있습니다. |

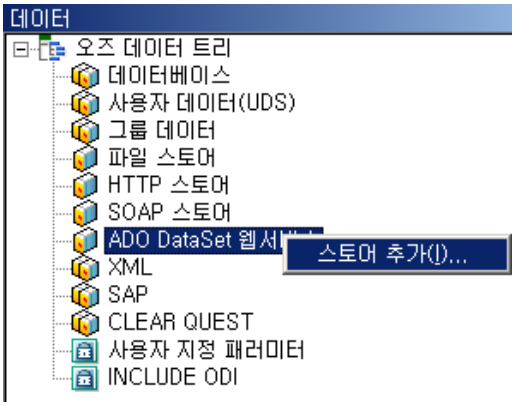
## ADO DataSet 웹서비스 스토어 연결

다음은 오즈 쿼리 디자이너와 ADO DataSet 웹서비스 스토어를 연결하고 데이터 셋을 생성하는 방법을 순차적으로 설명합니다.

■ **ADO DataSet 웹서비스 스토어와 데이터 셋**

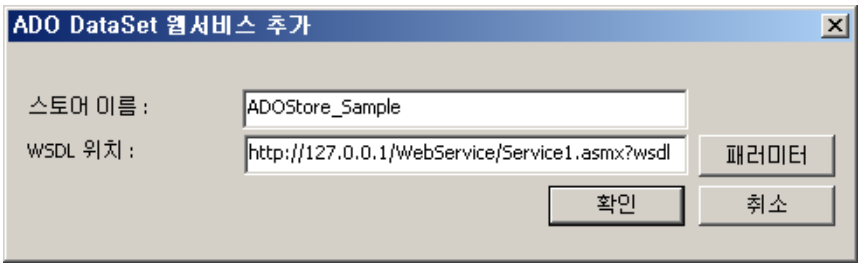
**Step 1** 오즈 쿼리 디자이너와 연동되는 서버 타입을 닷넷으로 설정한 후 데이터 창에서 [ADO DataSet 웹서비스]를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메

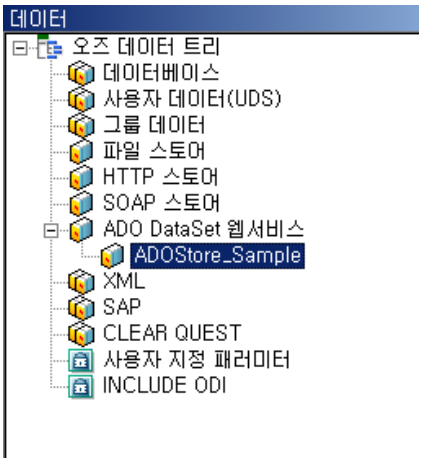
뉴에서 [스토어 추가] 메뉴를 클릭합니다.




**Step 2**

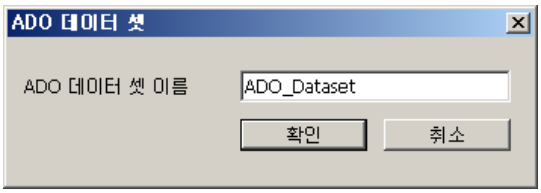
'ADO DataSet 웹서비스 추가 다이얼로그'에서 ADO DataSet 웹서비스 스토어 이름과 WSDL 위치를 입력한 후 [확인] 버튼을 클릭하여 스토어를 추가합니다.



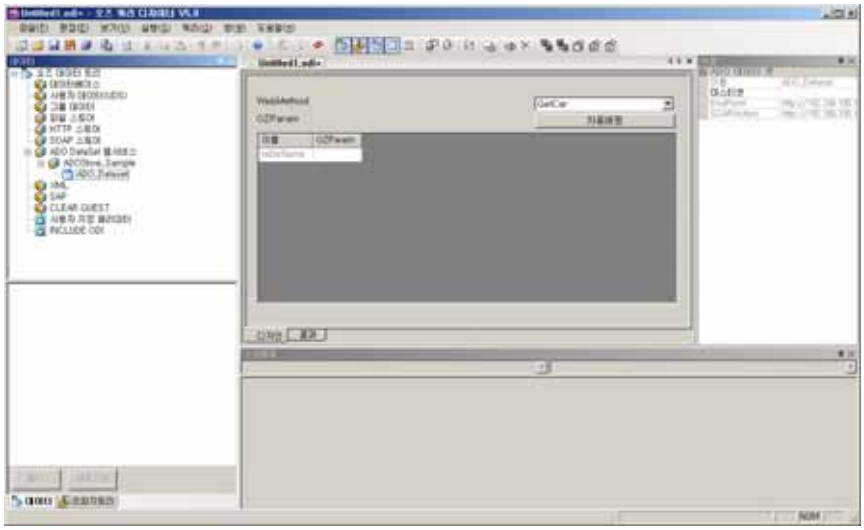




**Step 3**

데이터 창에 생성된 SOAP 스토어를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [데이터 셋 추가] 메뉴를 클릭하면 데이터 셋 이름을 입력하는 다이얼로그가 나타납니다.



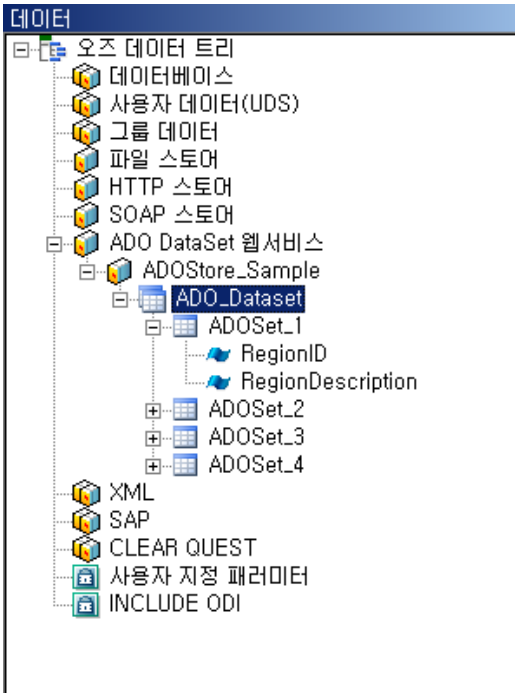

데이터 셋 이름을 입력한 후 [확인] 버튼을 클릭하면 쿼리 디자이너의 쿼리 창이 아래 그림처럼 ADO DataSet 웹서비스 데이터 셋 정보를 설정할 수 있도록 변경됩니다.



| 항목 | 설명 |
|---|---|
| **WebMethod** | WSDL에 정의된 Web Method가 콤보박스에 리스트 형태로 표시됩니다. |
| **OZParam** | WSDL에 정의된 패러미터 이름을 표시하고 해당 패러미터 값을 설정합니다. [자동 매핑] 버튼을 클릭하면 WSDL에 정의된 패러미터를 사용자 지정 패러미터로 추가하고 WSDL에 정의된 패러미터의 값으로 추가한 사용자 지정 패러미터를 지정합니다. 이때 추가된 사용자 지정 패러미터 이름은 "#OZParam.WebMethod이름_WSDL패러미터 이름#" 형태로 설정됩니다. |

**Step 4**

데이터 셋 정보를 설정한 후 실행 아이콘( )을 클릭하면 'ADO 데이터 셋 정보 다이얼로그'가 표시되고 SET_NAME을 설정한 후 [확인] 버튼을 클릭하면 아래 그림과 같이 추가된 데이터 셋과 필드를 확인할 수 있습니다.


데이터
오즈 데이터 트리
데이터베이스
사용자 데이터(UDS)
그룹 데이터
파일 스토어
HTTP 스토어
SOAP 스토어
ADO DataSet 웹서비스
ADOStore_Sample
ADO_Dataset
ADOSet_1
RegionID
RegionDescription
ADOSet_2
ADOSet_3
ADOSet_4
XML
SAP
CLEAR QUEST
사용자 지정 패러미터
INCLUDE ODI

# XII. XML 스토어

- XML 개요
- XML 스토어 이용

# XML 개요

## XML 개요

XML은 웹 상에서 서로 다른 업체의 응용프로그램 간의 데이터 교환을 위한 데이터 표현 언어로써, The World Wide Web Consortium(W3C)이 정의한 데이터를 응용 프로그램으로부터 분리하는 웹 언어입니다. 예를 들어, HTML이 웹 상에서 표준 UI를 구현하는 웹 언어라면 XML은 웹 상에서의 데이터 교환을 위한 웹 언어라고 할 수 있습니다.

## XML DTD/XSD 개요

DTD/XSD는 XML 문서에 존재할 수 있는 엘리먼트의 종류와 엘리먼트 간의 구조적 연관 관계 등을 나타내는 문법 규칙(Schema)을 정의하는 표준 중의 하나입니다. 따라서 DTD/XSD는 XML 문서에 존재할 수 있는 태그와 각 태그간의 포함 관계를 정의하고, 태그의 순서와 포함할 수 있는 태그의 개수를 정의합니다. 또한 태그가 가질 수 있는 속성과 그 속성이 갖는 값들을 정의합니다.

XML 표준은 XML 분석 단계에서 DTD/XSD와 같은 XML 구성 정보를 이용해 문서의 유효성을 검증함으로써 XML 데이터를 이용하는 응용 프로그램이 데이터를 검증해야 하는 번거로움을 제거합니다. 따라서 XML 데이터를 이용하는 응용프로그램은 XML 표준을 이용함으로써 다른 업체의 응용 프로그램과의 데이터 교환이 손쉬워지는 이점을 갖게 됩니다.

## XML 스토어 개요

XML 스토어는 오즈가 제공하는 XML 기반 데이터 소스로써, DTD나 XSD가 없는 XML만을 지원하는 파일/HTTP 스토어와는 달리 DTD나 XSD를 이용하는 XML 파일을 데이터 소스로 이용할 수 있습니다.

따라서 XML 스토어는 지원하는 문법 규칙에 따라 XML DTD 스토어와 XML XSD 스토어로 구성됩니다.

XML에 대한 자세한 내용은 XML 기본 서적을 참고하시기 바랍니다

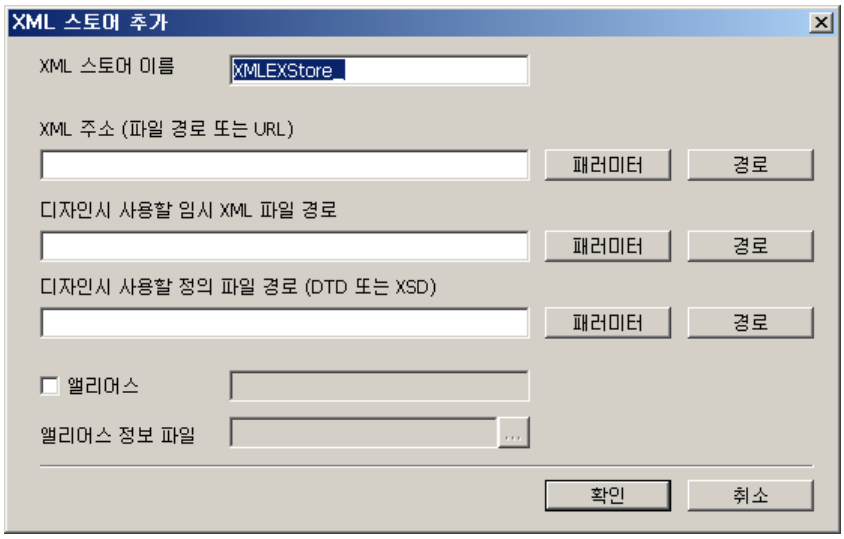
# XML 스토어 이용

다음은 XML 스토어를 이용해 XML DTD/XSD 데이터 소스를 연결하고 데이터 셋을 생성하는 방법에 대해 설명합니다.

## XML 스토어 설정 창

XML 파일을 데이터 소스로 연동할 때 사용하는 창입니다.

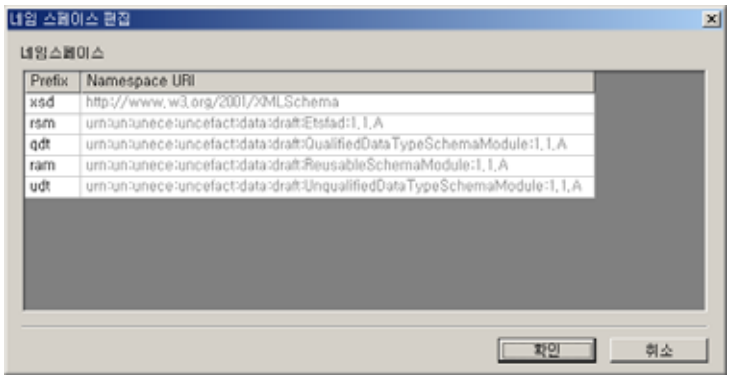
■ **XML 스토어 설정**




| 항목 | 설명 |
|---|---|
| **XML 스토어 이름** | 데이터 스토어에 대한 이름으로 사용자가 임의로 설정합니다. |
| **XML 주소(파일 경로 또는 URL)** | 서버에서 운용 시 사용할 XML 파일 경로 또는 사용자 애플리케이션 경로 이름입니다. |
| **디자인시 사용할 임시 XML 파일 경로** | 디자인 시 사용할 XML 파일의 로컬 시스템 파일 경로와 이름입니다. |
| **디자인시 사용할 정의 파일 경로(DTD 또는 XSD)** | 디자인 시 사용할 DTD/XSD 파일의 로컬 시스템 파일 경로와 이름입니다. |

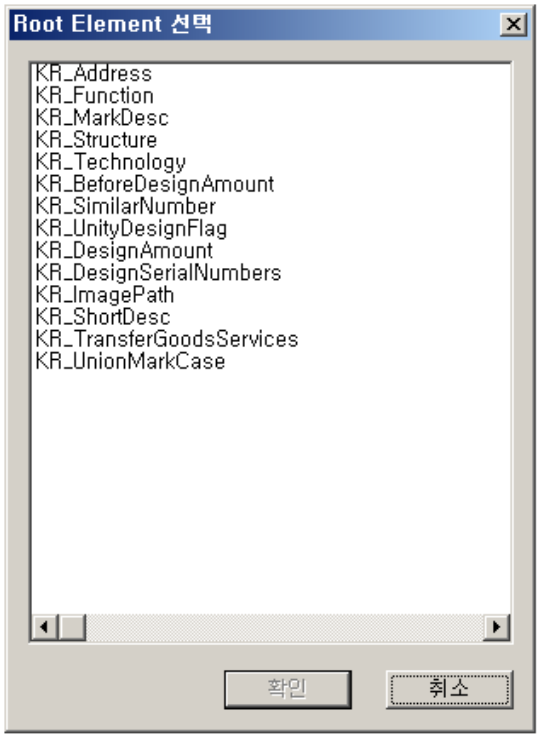
| | |
|---|---|
| **패러미터** | 사용할 패러미터를 패러미터 마법사를 이용하여 설정합니다. 패러미터 마법사에 대한 자세한 내용은 본 매뉴얼 "V. 쿼리 데이터 셋 디자인의 패러미터 마법사 이용하기"를 참조하시기 바랍니다. |
| **경로** | 가져올 XML 혹은 DTD 파일을 선택합니다. |
| **앨리어스** | 해당 데이터 셋 설정 정보를 별도의 연결 정보 파일에서 가져올 것인지 여부를 설정하며 앨리어스를 사용할 경우 오즈 서버의 datasource.properties 파일에 정의한 앨리어스 이름을 설정합니다. |
| **앨리어스 정보 파일** | 데이터 셋 설정 정보 파일 이름입니다.<br>앨리어스를 사용할 경우 앨리어스 정보가 있는 데이터 셋 설정 정보 파일명을 설정합니다. |

※ 참고사항 : "디자인시 사용할 정의 파일 경로"를 XSD로 설정하고 [확인] 버튼을 누르면 '네임 스페이스 편집 다이얼로그'가 나타납니다. '네임 스페이스 편집 다이얼로그'에서는 정의된 Prefix와 Namespace URI를 리스트 형태로 보여줍니다.




| 항목 | 설명 |
|---|---|
| **Prefix** | XSD 파일에 설정되어 있는 Prefix를 표시하며 사용하고자 하는 Prefix로 수정할 수 있습니다. XSD 파일에서 Prefix가 설정되어 있지 않다면 공백으로 표시됩니다. 만일 공백으로 표시되어 있거나 설정할 경우에는 '데이터 셋 추가 다이얼로그'의 노드 트리 창에서 "ns1, ns2, ns3…" 형태로 나타납니다. |
| **Namespace URI** | 네임 스페이스 URI를 표시하며, 사용자가 수정할 수 없습니다. |

※ 참고사항 : XML 주소에 설정한 XML 파일에 디자인시 사용할 정의 파일 경로에 설정한 DTD 파일에 정의된 Root Element가 없고, Root Element가 여러 개일 경우 [확인] 버튼을 클릭하면 다음 그림과 같이 'Root Element 선택 다이얼로그'가 표시됩니다.




사용자가 디자인 시 사용할 Root Element를 다이얼로그에서 선택한 후 [확인] 버튼을 클릭하면 XML 파일과 관계 없이 XML스토어를 추가할 수 있습니다.

## XML 스토어 연결

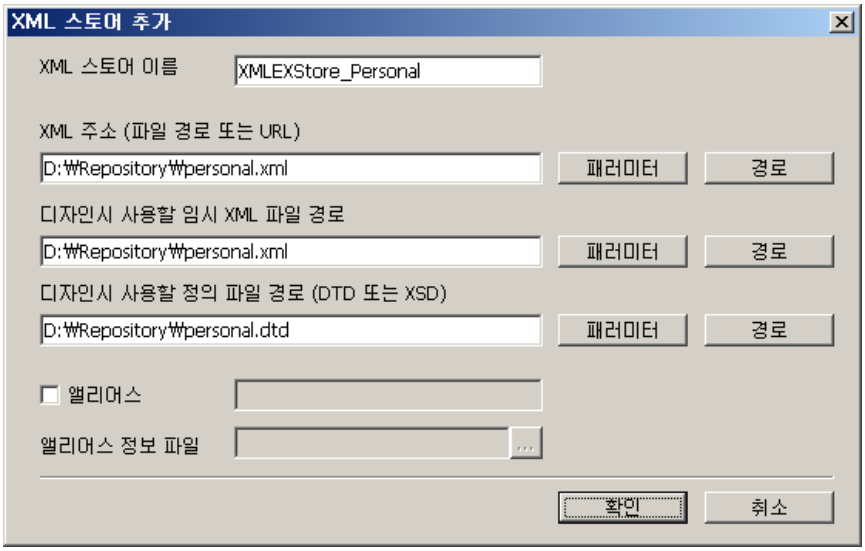
다음은 오즈 쿼리 디자이너와 XML 스토어(DTD/XSD 스토어) 데이터베이스를 연결하고 데이터 셋을 생성하는 방법을 순차적으로 설명합니다. XSD와 DTD 설정은 거의 차이가 없기 때문에 본 매뉴얼에서는 DTD 스토어에 대해서만 알아봅니다.

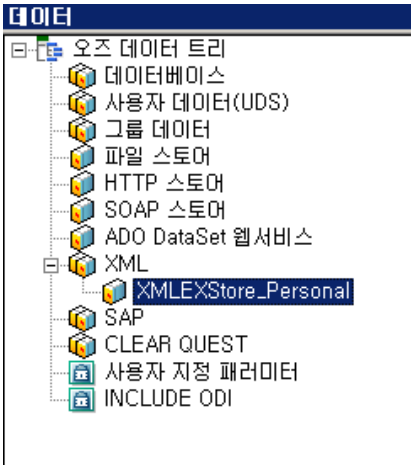
### ■ XML DTD 스토어와 데이터 셋

**Step 1** 오즈 쿼리 디자이너 데이터 창에서 [XML]을 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [XML 데이터 스토어 추가] 메뉴를 클릭합니다.



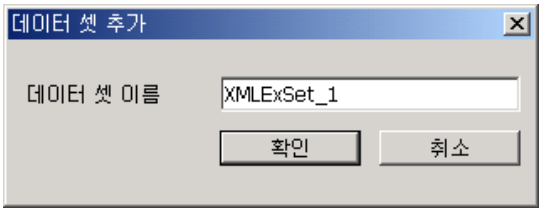

**Step 2** 'XML 스토어 추가 다이얼로그'에서 해당 옵션을 설정한 후 [확인] 버튼을 클릭하면 데이터 창에 추가된 'XML 스토어'가 나타납니다.



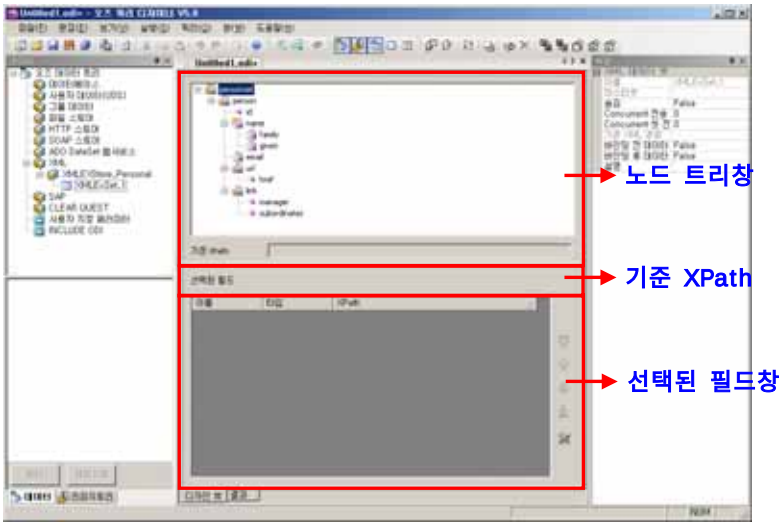



**Step 3**

데이터 창에 생성된 XML 스토어를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [데이터 셋 추가] 메뉴를 클릭하면 데이터 셋 이름을 입력하는 다이얼로그가 나타납니다.




데이터 셋 이름을 입력한 후 [확인] 버튼을 클릭하면 쿼리 디자이너의 쿼리 창에 아래 그림처럼 XML 스토어 정보를 설정할 수 있도록 변경됩니다.




- 노드 트리 창

  추가된 XML 스토어의 노드 구조가 트리 모양으로 표시됩니다.

  노드명 앞에는 노드 각각의 특성을 나타내는 아이콘이 있어 노드에 대한 식별을 용이하게 합니다. 노드의 아이콘에 대한 설명은 다음과 같습니다.

| 아이콘 | 설명 |
|---|---|
|  | 애트리뷰트 |
|  | 최소 발생 1, 최대 발생 1인 엘리먼트(기본 값) |
|  | 최소 발생 0, 최대 발생 제한 없는 엘리먼트 |
|  | 최소 발생 0, 최대 발생 1인 엘리먼트 |
|  | 최소 발생 1, 최대 발생 제한 없는 엘리먼트 |
|  | 트리에서 확장된 엘리먼트 |

| | |
|---|---|
| | 필드로 선택 불가능한 엘리먼트 |
| | 기준 XPath로 지정된 엘리먼트 |

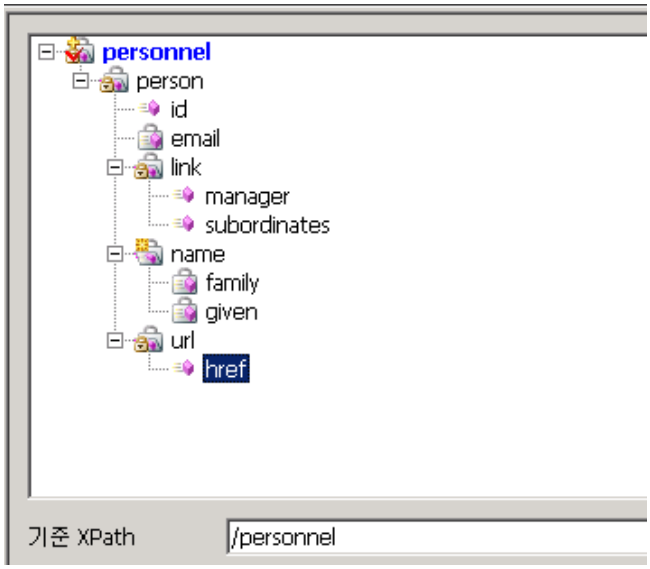
노트 트리 창에서 기준 XPath를 지정한 후 노드를 더블 클릭하여 데이터 셋에 추가합니다.

- 기준 XPath

**[디자인 모드]**

노드 트리 창에서 기준 XPath로 지정할 노드를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [기준 XPath 지정] 메뉴를 클릭합니다. 기준 XPath로 지정된 노드는 아이콘이 체크된 상태( )로 표시되고 노드 이름이 하이라이트되어 표시되며, 기준 XPath에 지정한 기준 XPath가 표시됩니다



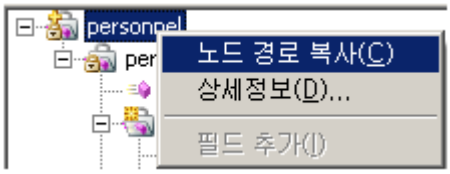




이미 설정된 기준 XPath를 수정하려면 기준 XPath를 나타내는 텍스트 박스에서 다른 값으로 입력하거나 트리 창에서 기준 XPath로 변경할 노드 아이템을 마우스 오른쪽 버튼으로 선택하여 나타나는 [기준 XPath 지정] 메뉴를 클릭하여 수정할 수 있습니다.

**[매뉴얼 모드]**

기준 XPath에 기준 XPath를 직접 입력합니다.

기준 XPath 경로를 입력할 때 기준 XPath로 지정할 노드를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [노드 경로 복사] 메뉴를 이용하여 XPath 경로를 손쉽게 입력할 수 있습니다.

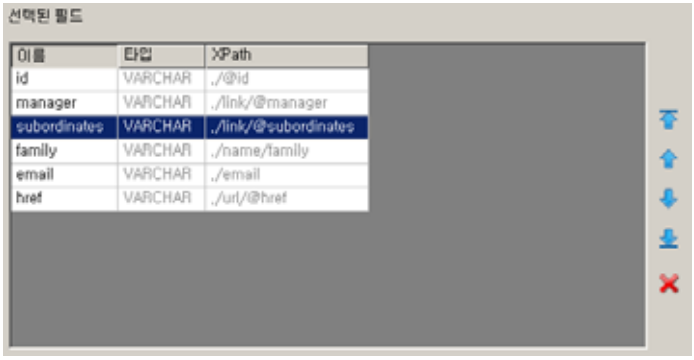



- 선택된 필드 창

**[디자인 모드]**

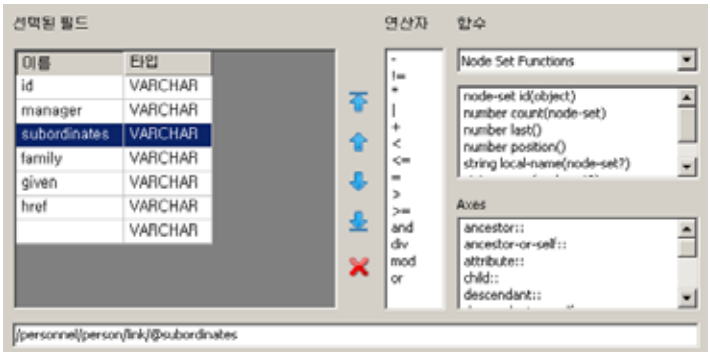
선택한 필드에 이름, 타입, XPath를 리스트 형태로 보여주며 선택한 필드의 순서를 변경( )하거나 삭제( )할 수 있습니다.

선택된 필드 창에서는 필드의 이름만 편집할 수 있으며, 필드의 타입과 XPath는 편집할 수 없습니다.




**[매뉴얼 모드]**

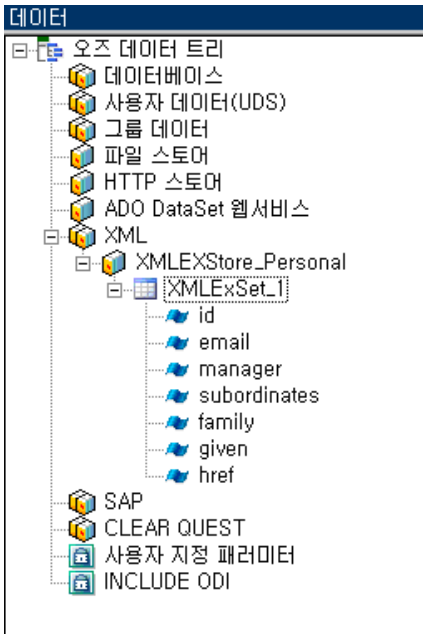
선택한 필드에 이름, 타입, XPath를 리스트 형태로 보여주며 선택한 필드의 순서를 변경( ))하거나 삭제( )할 수 있습니다.




매뉴얼 모드에서는 노드 트리 창에서 노드를 더블 클릭하면 선택된 필드에 해당 노드 이름이 추가되지만 노드의 경로는 입력되지 않으므로, 사용자가 직접 입력하여야 합니다. 노드 트리 창에서 노드를 클릭하지 않고 추가하고자 할 경우에는 선택된 필드 맨 아래에 추가된 행에 노드 이름을 직접 입력하여야 합니다.

**Step 4**

데이터 셋에 필드를 모두 추가한 후 실행 아이콘( )을 클릭하면 아래 그림과 같이 추가된 필드를 확인할 수 있습니다.



# XIII. SAP 스토어

- SAP 개요
- SAP 스토어 이용

# SAP 개요

## SAP 개요

SAP은 데이터베이스를 이용하여 통합된 업무용 애플리케이션을 제공하며, 최근의 SAP R/3 시스템에서는 문서들을 자동으로 사용자들에게 전달하는 워크 플로우 기능까지 제공하고 있습니다. SAP은 기본적으로 클라이언트 서버 환경을 기본으로 하며, 대부분의 운영 체제 플랫폼을 지원합니다.

## SAP 스토어 개요

SAP의 데이터를 이용하여 보고서를 작성하기 위해서는 별도의 데이터 추출 프로그램 작성 및 데이터 추출 프로그램 호출을 위한 SAP 클라이언트 애플리케이션을 별도로 설치해야 하는 번거로움이 있습니다. SAP 스토어는 별도의 데이터 추출 프로그램 작성없이 간단히 데이터 셋을 작성할 수 있도록 도와줍니다.

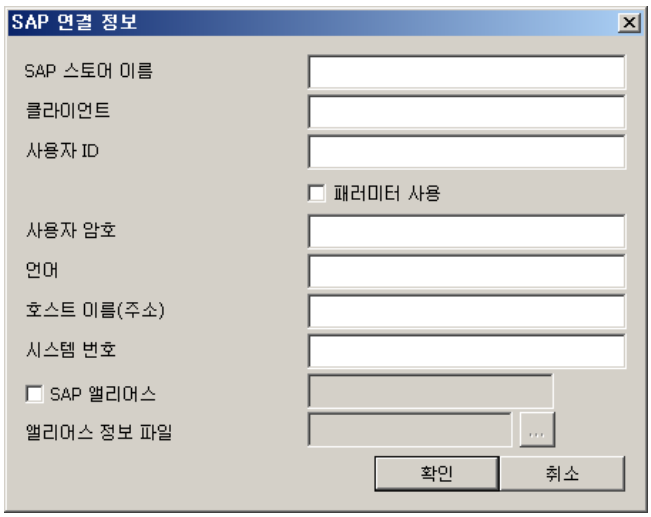
SAP 스토어에서 데이터를 전달받기 위해서는 RFC(Remote Function Call:시스템 밖에서 R/3에 접근하여 데이터를 얻을 수 있는 방법) 모듈을 참조하여 전달받는 결과값을 테이블 형태로 구성합니다.

# SAP 스토어 이용

다음에서는 SAP 스토어 연동 방법에 대해 설명합니다.

## SAP 스토어 설정 창

■ **SAP 연결 정보 설정**



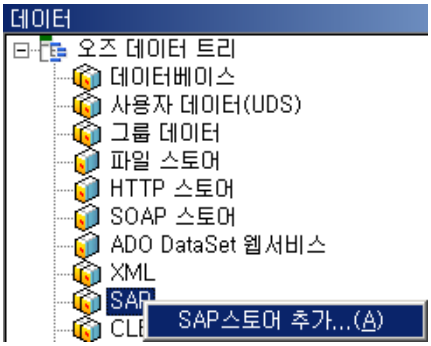

| 항목 | 설명 |
|---|---|
| **SAP 스토어 이름** | 오즈 쿼리 디자이너에서 사용할 스토어 이름을 설정합니다. |
| **클라이언트 이름** | 로그인할 클라이언트 번호를 설정합니다. |
| **사용자 ID** | 로그인시 사용할 아이디를 설정합니다. |
| **사용자 암호** | 로그인시 사용할 암호를 설정합니다. |
| **패러미터 사용** | 사용자 암호를 사용자 지정 패러미터 값으로 설정합니다. |
| **언어** | 로그인시 사용할 언어를 설정합니다. |
| **호스트 이름 (주소)** | SAP 서버의 IP Address를 설정합니다. |
| **시스템 번호** | SAP 서버의 시스템 번호를 설정합니다. |
| **SAP 앨리어스** | SAP 스토어 정보를 앨리어스로 설정할지 여부를 체크하고, 사용할 앨리어스 이름을 입력합니다. |

| | |
|---|---|
| **앨리어스 정보 파일** | 앨리어스 정보 파일(sap.properties)을 설정합니다.<br>※ 참고사항 : sap.properties에 대한 설명은 "오즈 엔터프라이즈 서버 관리자 매뉴얼"을 참조하시기 바랍니다. |

## SAP 스토어 연결

다음은 오즈 쿼리 디자이너와 SAP 스토어를 연결하는 방법을 순차적으로 설명합니다.

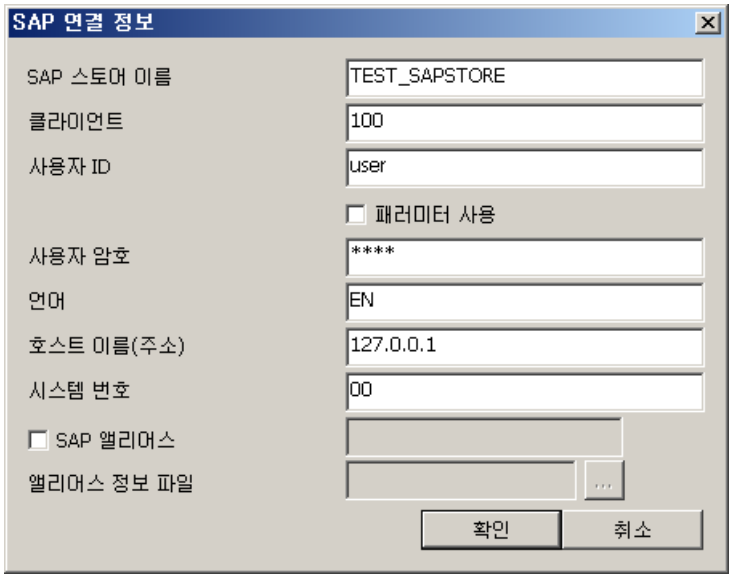
**Step 1** 오즈 쿼리 디자이너 데이터 창에서 [SAP]을 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [SAP 스토어 추가] 메뉴를 클릭합니다.



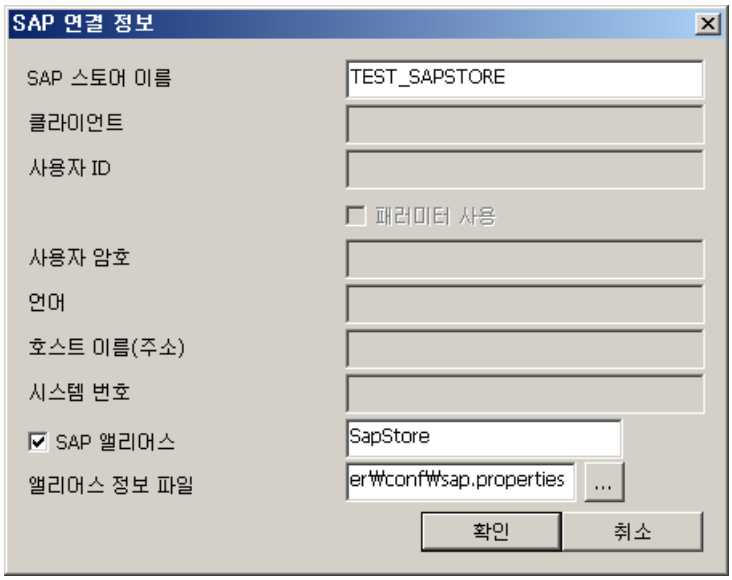

**Step 2** [SAP 스토어 추가] 메뉴를 클릭하여 나타나는 'SAP 연결 정보 다이얼로그'에 연결할 SAP 정보를 입력합니다.

SAP 연결 정보를 직접 설정하여 입력하거나 앨리어스 정보 파일을 이용하여 설정할 수 있습니다.

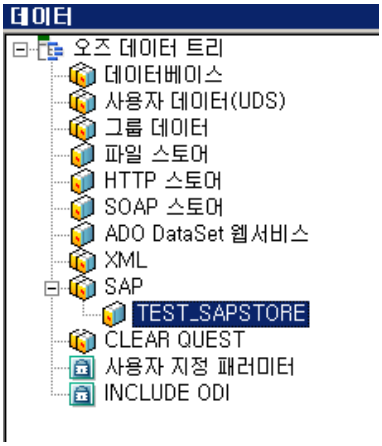
- 연결 정보를 직접 설정한 경우




- 앨리어스 정보 파일을 이용한 경우




'SAP 연결 정보 다이얼로그'에서 해당 옵션을 설정한 후 [확인] 버튼을 클릭하면 데이터 창에 추가된 'SAP 스토어'가 나타납니다.



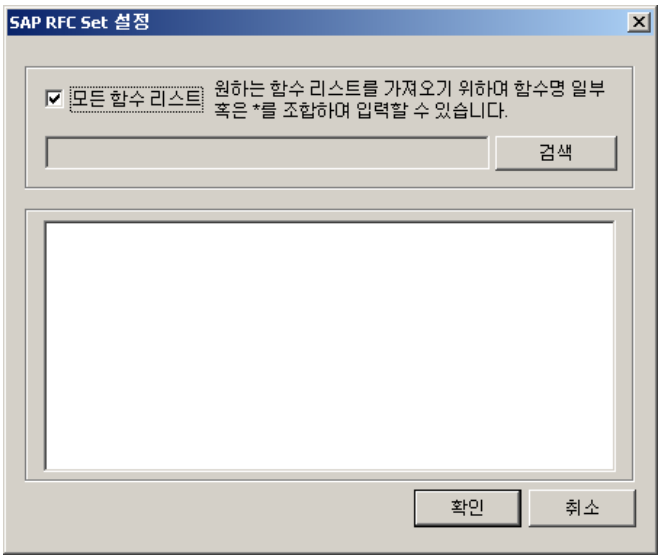

**Step 3**

데이터 창에 추가된 SAP 스토어를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [함수 추가] 메뉴를 클릭하면 SAP 데이터 셋을 생성하기 위한 'SAP RFC Set 설정 다이얼로그'가 나타납니다.

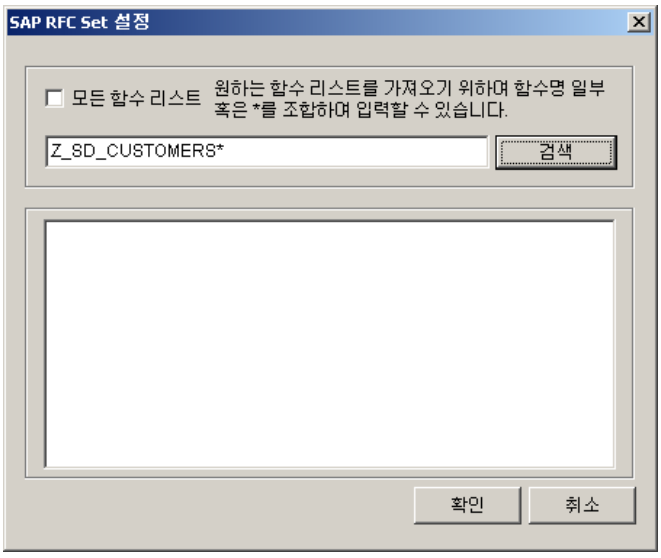



RFC를 선택하는 방법은 크게 두 가지로 형태로 구분됩니다.

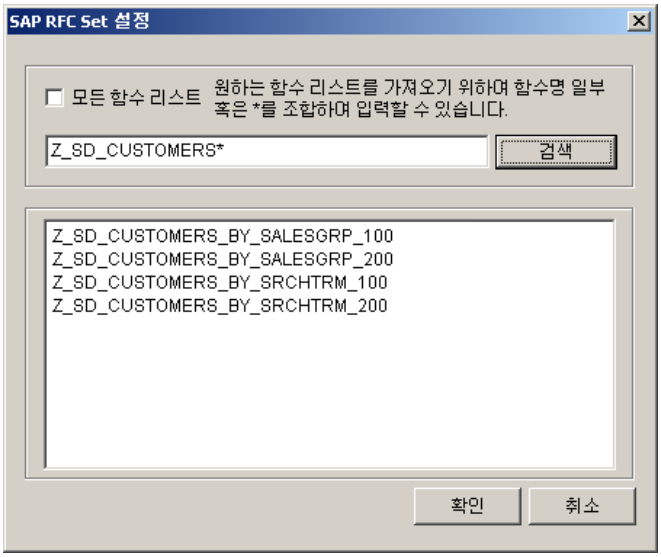
첫 번째로, 모든 함수 리스트를 체크한 후 [검색] 버튼을 클릭하여 나타나는 함수 선택 창에서 원하는 함수 이름을 선택합니다.




두 번째로, 모든 함수 리스트를 체크하지 않고 아래 에디터 박스에 '*', '?'과 문자를 조합하여 입력한 후 [검색] 버튼을 클릭하면 조건에 해당하는 RFC 리스트를 보여줍니다.



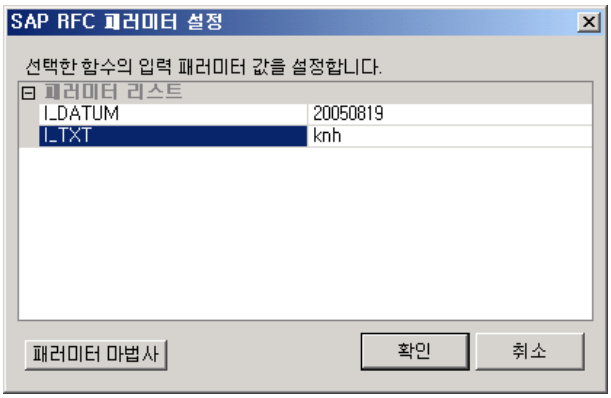

위와 같은 검색조건을 입력한 후 [검색] 버튼을 클릭하면 다음과 같이 검색한 결과(RFC) 리스트를 보여줍니다.



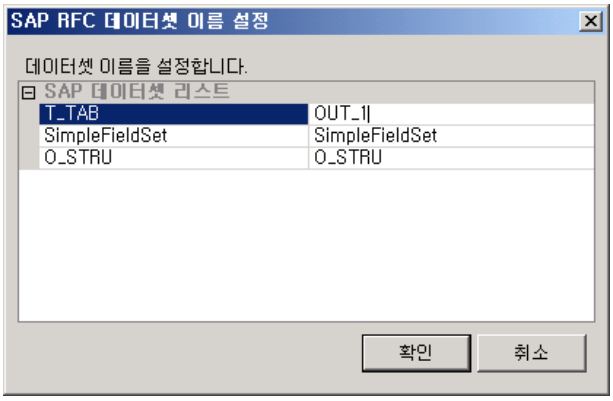

검색된 리스트에서 원하는 RFC 항목을 더블 클릭한 후 [확인] 버튼을 클릭합니다.

**Step 4**

다음은 선택한 함수에 대한 입력 패러미터 정보를 가져옵니다. 검색된 리스트에서 원하는 RFC 항목을 더블 클릭한 후 [확인] 버튼을 클릭하면 다음과 같이 선택된 RFC에 대한 패러미터 값을 입력하는 창이 나타납니다.




입력 패러미터 값을 지정한 후 [확인] 버튼을 선택하면 RFC로 통해서 생성될 데이터 셋 리스트 다이얼로그가 나타납니다. 기본적으로 RFC에서 생성된 정보가 나타나며 사용자가 설정값을 변경할 수도 있습니다.




**Step 5**

패러미터를 입력한 후 [확인] 버튼을 클릭하면 해당 RFC에 대한 결과 데이터 셋이 생성됩니다.

하나의 RFC 데이터 소스는 다중 데이터 셋을 생성할 수 있으며, 데이터 셋은 해당 RFC의 필드 정보를 포함합니다

# XIV. CLEAR QUEST 스토어

- CLEAR QUEST 스토어 개요
- CLEAR QUEST 스토어 이용

# CLEAR QUEST 스토어 개요

## CLEAR QUEST 스토어 개요

CLEAR QUEST 스토어는 형상 관리 툴인 IBM Rational Clear Quest 프로그램에서 생성한 DB를 연동할 때 이용하는 데이터 소스입니다.

※ 참고사항

- CLEAR QUEST 스토어를 사용하려면 사용자 PC에 IBM Rational Clear Quest 프로그램이 설치되어 있어야 합니다.
- CLEAR QUEST 스토어를 사용할 경우 데이터 창에 사용 가능한 테이블 리스트가 표시되지 않습니다.
- CLEAR QUEST에서 지원하는 쿼리 방식 데이터 셋 추가는 현재 지원하지 않습니다.
- CLEAR QUEST에서 지원하는 다이나믹 방식 데이터 셋의 필터 정의문은 표시되지 않습니다.
- 디자인 모드를 매뉴얼 모드에서 디자인 뷰 모드로는 변경할 수 없습니다.
- CLEAR QUEST에 접속한 사용자에 SQL 보기 권한이 없을 경우 디자인 모드를 매뉴얼 모드로 변경하여 사용할 수 없습니다.

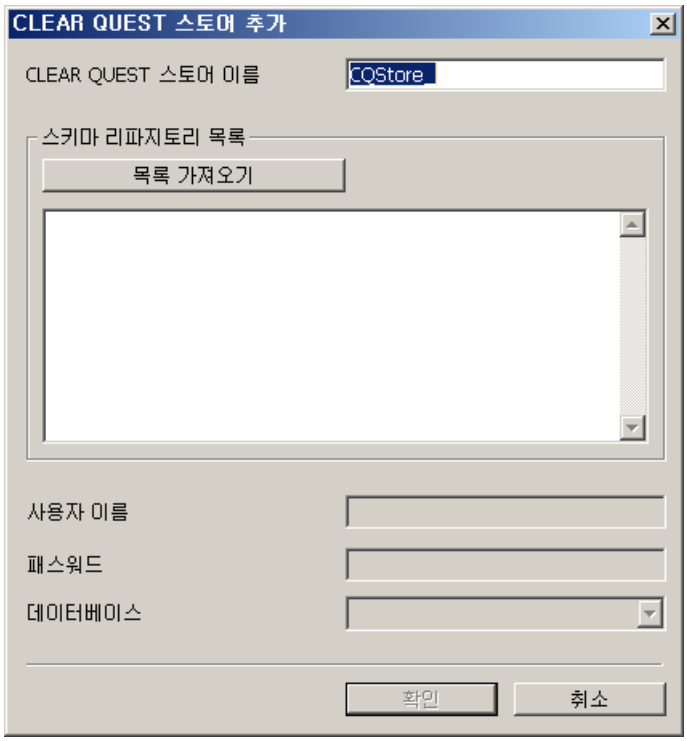
# CLEAT QUEST 스토어 이용

다음은 CLEAT QUEST 스토어를 이용하여 IBM CLEAT QUEST 프로그램에서 생성한 DB를 연동하여 데이터 셋을 생성하는 방법에 대해 설명합니다.

## CLEAT QUEST 스토어 설정 다이얼로그

IBM CLEAT QUEST 프로그램에서 생성한 DB에 연동하여 데이터 소스 가져올 때 사용하는 창입니다.

■ **CLEAT QUEST 스토어 설정**




| 항목 | 설명 |
|---|---|
| **CLEAR QUEST 스토어 이름** | 추가할 CLEAR QUEST 스토어 이름을 입력합니다. |
| **스키마 리파지토리 목록** | CLEAR QUEST에서 생성한 스키마 목록을 가져옵니다. |
| **사용자 이름** | 스키마 사용자 이름을 입력합니다. |
| **패스워드** | 스키마 사용자 패스워드를 입력합니다. |

| 데이터베이스 | 연결할 데이터베이스를 선택합니다. |
|---|---|

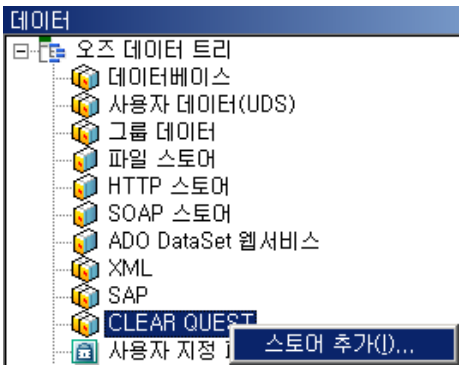
## CLEAR QUEST 스토어 연결

다음은 오즈 쿼리 디자이너에서 CLEAR QUEST 스토어를 연결하고 데이터 셋을 생성하는 방법을 디자인 모드가 매뉴얼 모드일 경우와 디자인 뷰 모드일 경우로 나누어 순차적으로 설명합니다.

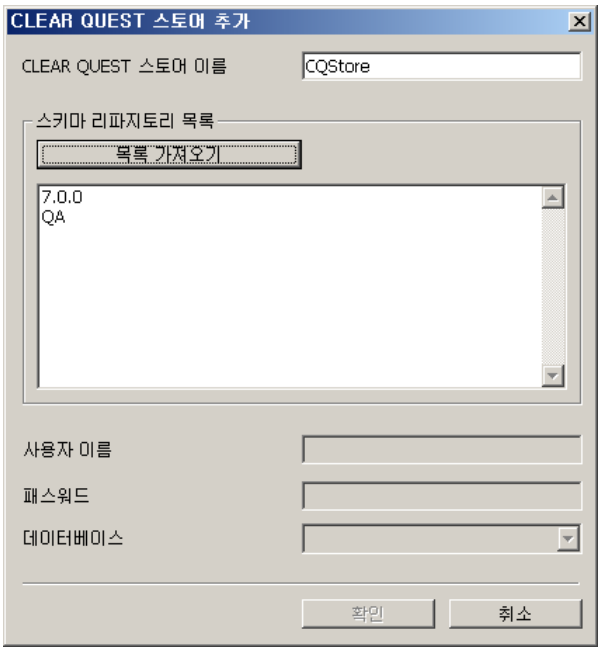
■ **매뉴얼 모드**

디자인 모드가 매뉴얼 모드일 경우 CLEAR QUEST 데이터 셋 추가 시 쿼리 창에 쿼리문을 직접 입력하여 쿼리를 디자인합입니다.

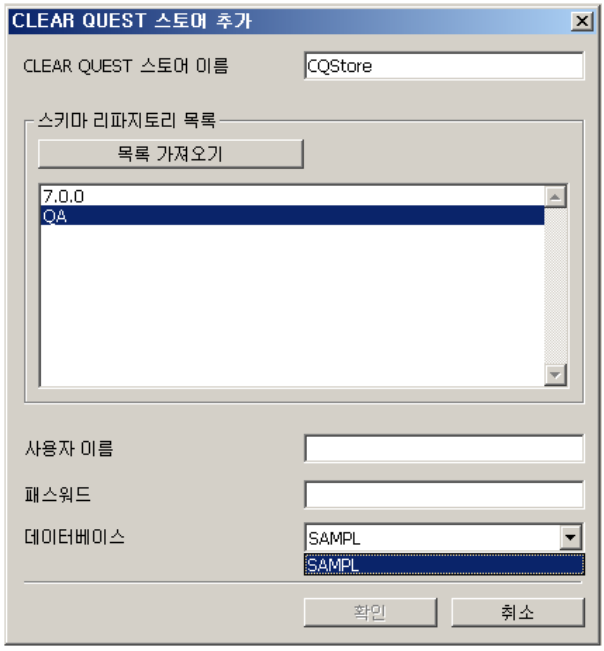
**Step 1** 오즈 쿼리 디자이너 데이터 창에서 [CLEAR QUEST]을 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [스토어 추가] 메뉴를 클릭합니다.



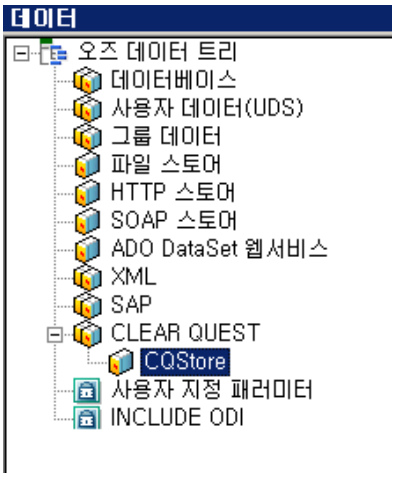

**Step 2** 'CLEAR QUEST 스토어 추가 다이얼로그'에서 CLEAR QUEST 스토어 이름을 입력한 후 스키마 리파지토리 목록의 [목록 가져오기] 버튼을 클릭하여 CLEAR QUEST에 추가된 스키마 리파지토리 목록을 가져와서 리스트 형태로 표시합니다.



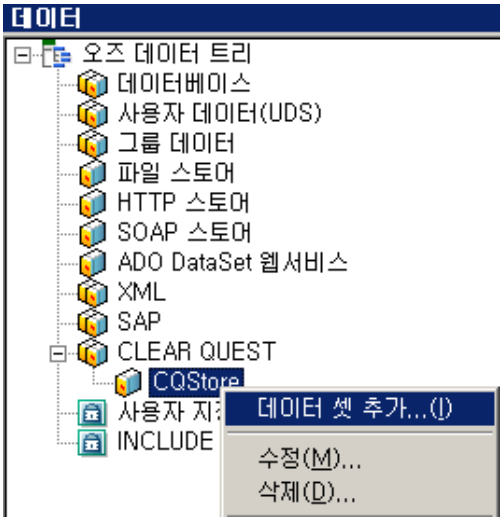

**Step 3** 표시된 스키마 리파지토리 목록 중 연결할 스키마를 더블 클릭하여 해당 스키마의 데이터베이스 목록을 가져옵니다.



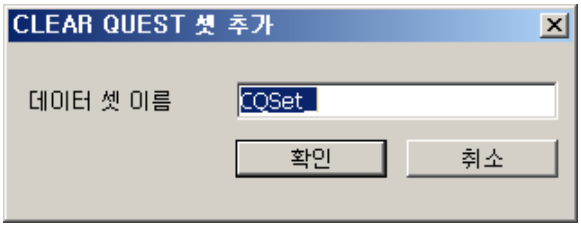

**Step 4** 연결할 데이터베이스를 선택한 후 사용자 이름과 패스워드를 입력한 후 [확인] 버튼을 클릭하면 데이터 창에 CLEAR QUEST 스토어가 표시됩니다.



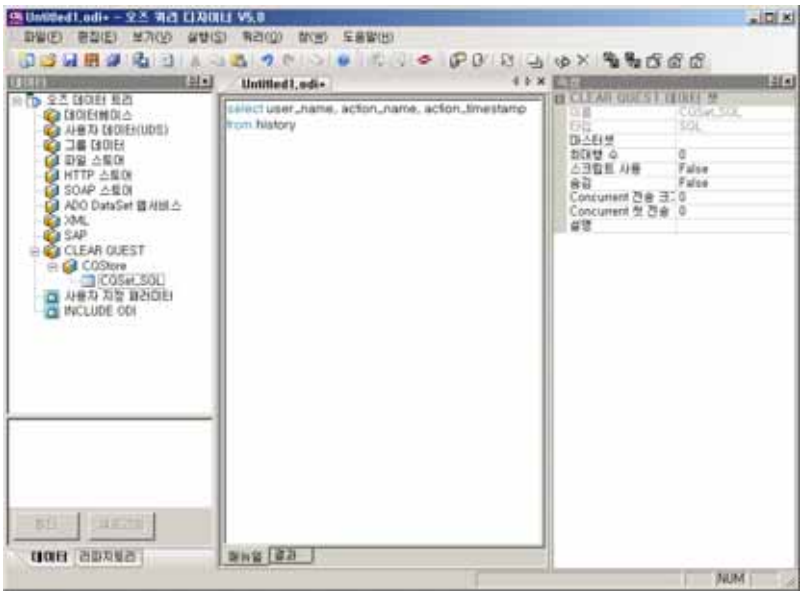

**Step 5** 데이터 창에 생성된 "CLEAR QUEST"를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [데이터 셋 추가] 메뉴를 클릭합니다.



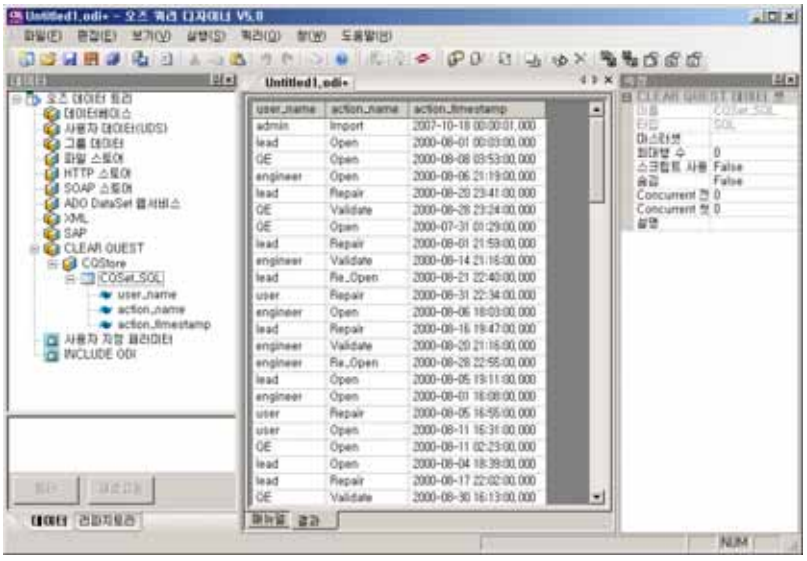

**Step 6** 'CLEAR QUEST 셋 추가 다이얼로그'에서 데이터 셋 이름을 입력한 후 [확인] 버튼을 클릭하여 데이터 셋을 추가합니다.




**Step 7** 추가한 데이터 셋의 쿼리 창에 실행할 쿼리문을 직접 입력합니다.



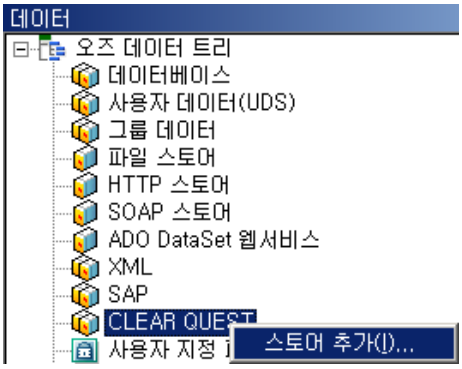
**Step 8** 쿼리문을 입력한 후 툴바에서 쿼리문 실행( )아이콘을 클릭하여 쿼리문을 실행합니다.



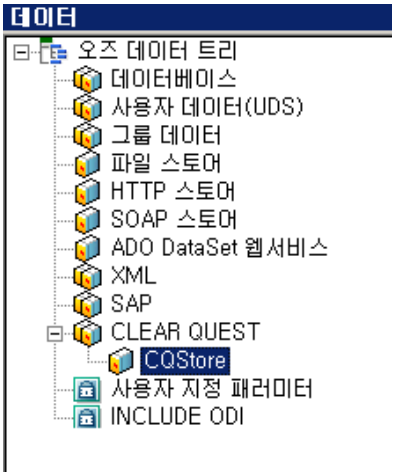
■ **디자인 뷰 모드**

디자인 모드가 디자인 뷰 모드일 경우 CLEAR QUEST 데이터 셋 추가 시 쿼리문을 사용자가 직접 입력하지 않고 마법사 형식의 UI에서 옵션을 선택하여 실행할 쿼리문을 설정합니다.

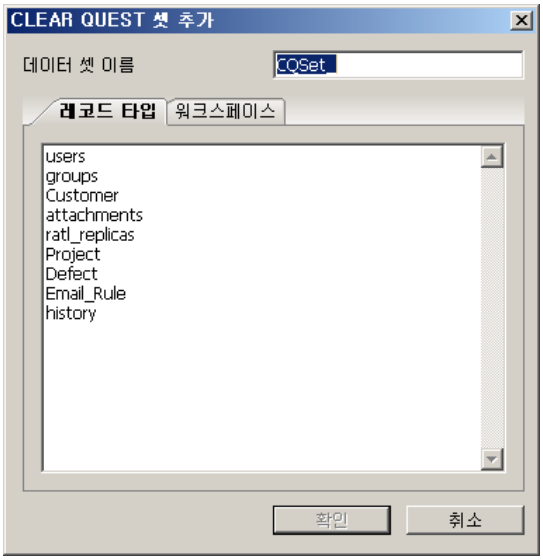
**Step 1** 오즈 쿼리 디자이너 데이터 창에서 [CLEAR QUEST]을 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [스토어 추가] 메뉴를 클릭합니다.




**Step 2** 'CLEAR QUEST 스토어 추가 다이얼로그'에서 CLEAR QUEST 스토어 이름을 입력한 후 연결할 스키마를 선택하고 사용자 이름, 패스워드 등을 설정한 후 [확인] 버튼을 클릭하여 CLEAR QUEST 스토어를 추가합니다.



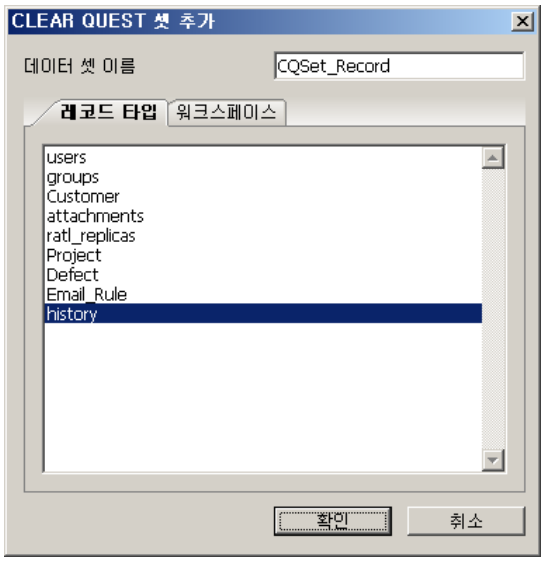

**Step 3** 데이터 창에 생성된 "CLEAR QUEST"를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [데이터 셋 추가] 메뉴를 클릭하여 'CLEAR QUEST 셋 추가 다이얼로그'를 표시합니다.



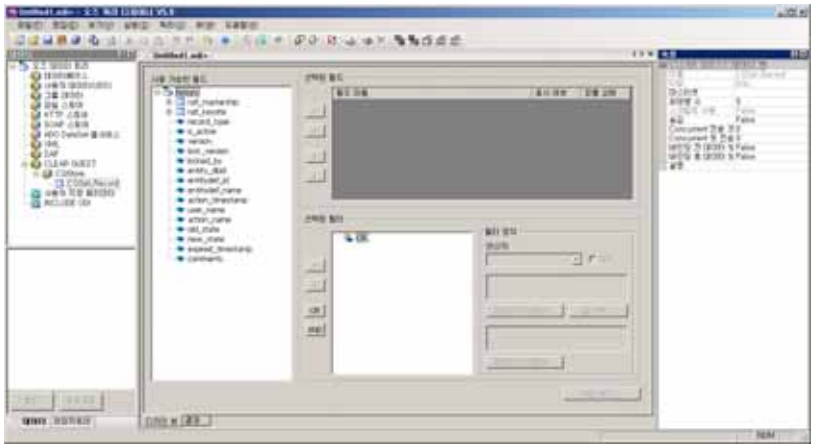

- 레코드 타입

  레코드 타입은 CLEAR QUEST의 레코드 타입을 선택한 후 해당 레코드 타입과 연동하여 데이터 셋을 추가하는 방식입니다.

  - 'CLEAR QUEST 셋 추가 다이얼로그'에서 [레코드 타입] 탭을 선택합니다.




  - 다이얼로그에서 데이터 셋 이름을 입력하고 연결할 레코드 타입을 선택한 후 [확인] 버튼을 클릭하면 데이터 셋이 추가되고 [디자인 뷰] 탭에 아래 그림처럼 데이터 셋 정보를 설정할 수 있는 화면이 표시됩니다.

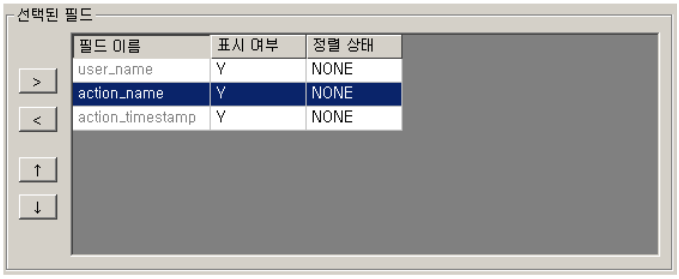


**[사용 가능한 필드 창]**

선택한 레코드의 사용 가능한 필드가 표시됩니다.

**[선택된 필드 창]**

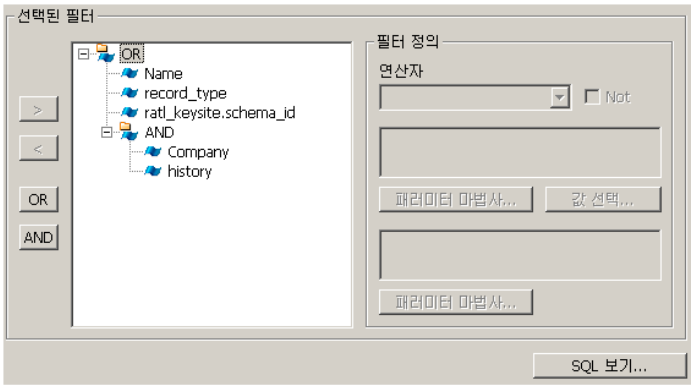
사용 가능한 필드 창에서 필드를 선택한 후 필드 추가(>), 삭제(<) 버튼을 클릭하여 사용할 필드를 선택하고 필드 순서 이동 버튼(↑, ↓)을 클릭하여 표시할 필드 순서를 설정합니다.




| 옵션 명 | 설명 |
|---|---|
| 필드 이름 | 필드 이름을 표시합니다. |
| 표시 여부 | 결과 셋에 해당 필드를 표시할지 여부를 설정합니다. |
| 정렬 상태 | 필드의 정렬 방식을 선택합니다. |

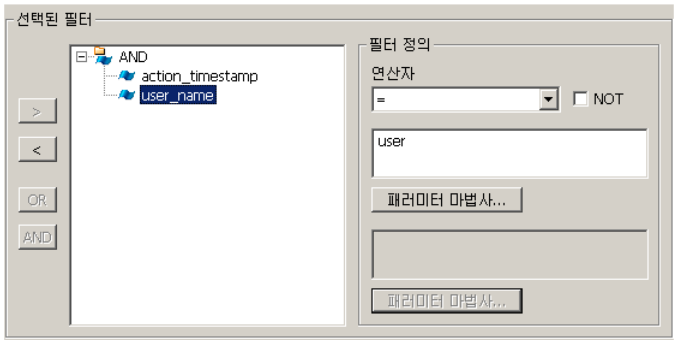
**[선택 필터 창]**

사용 가능한 필드 창에서 필드를 선택한 후 필드 추가(>), 삭제(<) 버튼을 클릭하여 사용할 필드를 선택하고 연산자 버튼(OR, AND)을 클릭하여 선택한 필드의 연산자를 변경합니다.

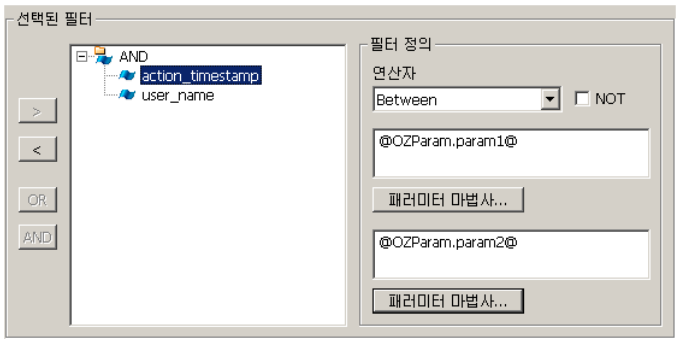



| 옵션 명 | 설명 |
|---|---|
| 연산자 | 조건문을 구성하는데 사용할 수 있는 연산자 리스트를 표시합니다. |
| NOT | 연산자 "NOT"을 설정합니다. |
| 값 | 조건문에 설정할 값을 입력합니다.<br>※ 참고사항 : 두 번째 값을 설정하는 텍스트 박스는 연산자가 "Between"일 경우에만 활성화됩니다. |
| 값 선택 | 필터의 연산자가 "IN"일 경우 여러 개의 값을 설정합니다.<br>버튼을 클릭하면 '값 선택 다이얼로그'가 표시되며 해당 다이얼로그에서 값을 여러 개 선택하여 추가할 수 있습니다. |
| 패러미터 마법사 | 조건문에 입력할 값을 패러미터로 설정합니다.<br>버튼을 클릭하면 "패러미터 마법사"가 표시되며 해당 다이얼로그에서 패러미터를 추가할 수 있습니다. |
| SQL 보기 | 현재 편집 중인 쿼리의 SQL문을 표시합니다.<br>버튼을 클릭하면 SQL문을 표시하는 창이 표시됩니다. |

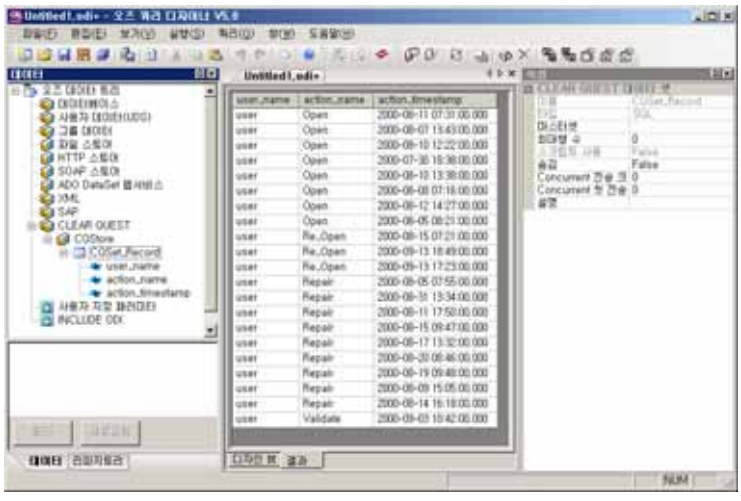
① 값을 직접 입력하여 필터를 설정한 예




② 패러미터를 설정하여 필터를 설정한 예



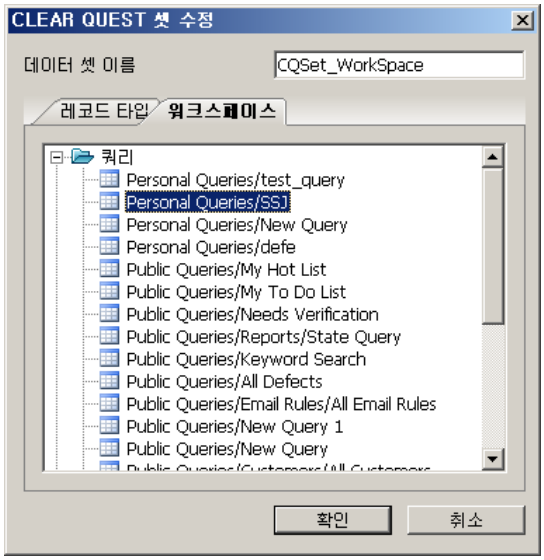

▪ 옵션을 설정한 후 툴바에서 쿼리문 실행( )아이콘을 클릭하여 쿼리문을 실행합니다.



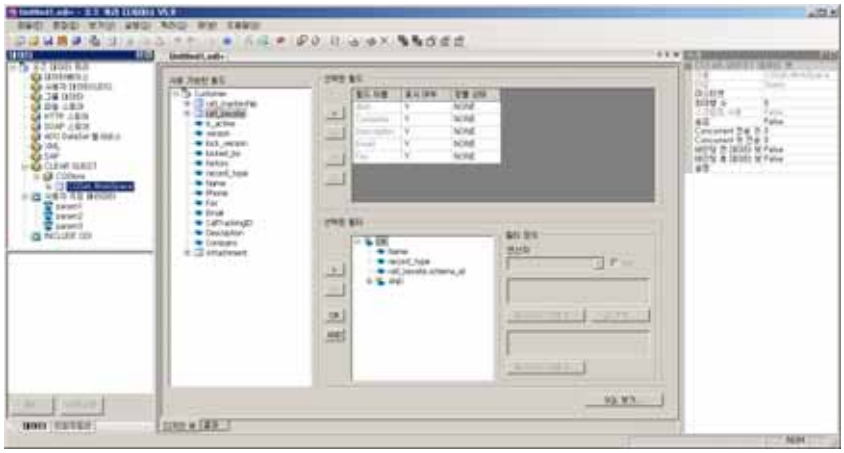
- 워크스페이스

워크스페이스는 CLEAR QUEST에 추가된 워크스페이스를 선택한 후 해당 워크스페이스와 연동하여 데이터 셋을 추가하는 방식입니다.

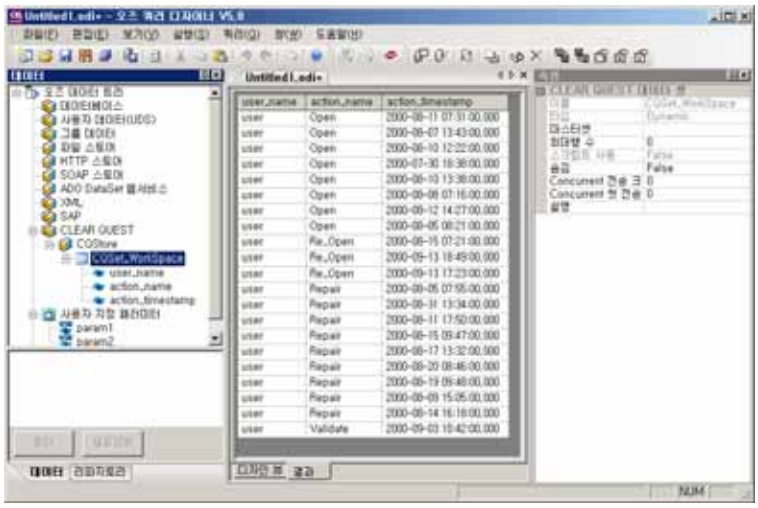
- 'CLEAR QUEST 셋 추가 다이얼로그'에서 [워크스페이스] 탭을 선택합니다.




- 다이얼로그에서 데이터 셋 이름을 입력하고 연결할 워크스페이스를 선택한 후 [확인] 버튼을 클릭하면 데이터 셋이 추가되고 [디자인 뷰] 탭에 아래 그림처럼 선택한 워크스페이스에 설정된 필드 및 필터 정보가 화면에 표시됩니다.
  변경할 사항이 있을 경우 표시된 필드와 필터 정보를 사용자가 수정할 수 있습니다.



- 옵션을 설정한 후 툴바에서 쿼리문 실행( )아이콘을 클릭하여 쿼리문을 실행합니다.



# XV. 서버 리파지토리 사용

- 서버 리파지토리 개요
- 서버 리파지토리 연결
- 서버로부터 파일 열기
- 카테고리 및 아이템 관리
- 체크인, 체크아웃
- 히스토리 보기
- 아이템 찾기

**본 장에서는 오즈 서버 리파지토리 서비스의 개념을 설명하고 서버 리파지토리에 연결하고 파일을 업로드하고 다운로드하는 방법을 설명합니다.**

## 서버 리파지토리 개요

서버 리파지토리는 ODI, OZR, OZA, 데이터 파일 등 각종 파일을 저장하는 원격 저장소로, 다수의 개발자가 원격으로 파일을 공유할 수 있으며, 각 사용자마다 접근 권한을 달리할 수 있습니다. 또한 파일 버전 관리 기능을 통해 개발의 효율을 높여줍니다.

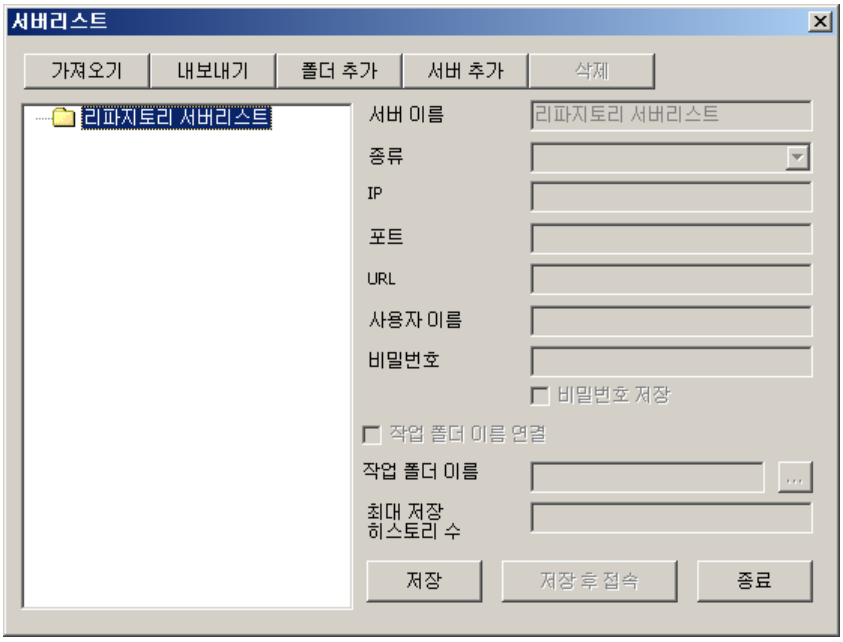
# 서버 리파지토리 연결

## 서버 리파지토리 연결

**Step 1** 오즈 쿼리 디자이너의 리파지토리 창에서 '리파지토리 서버'를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [리파지토리 열기] 메뉴를 클릭합니다.




**Step 2** 아래 그림과 같이 저장된 리파지토리 서버 리스트와 접속 정보를 나타내는 '서버 리스트 다이얼로그'가 표시됩니다




'서버 리스트 다이얼로그'의 설정 버튼 및 옵션의 기능은 아래와 같습니다.

| Button | Description |
|---|---|
| 가져오기 | 서버 리스트 정보를 저장하고 있는 XML 파일을 가져와서 서버 리스트를 표시합니다. |

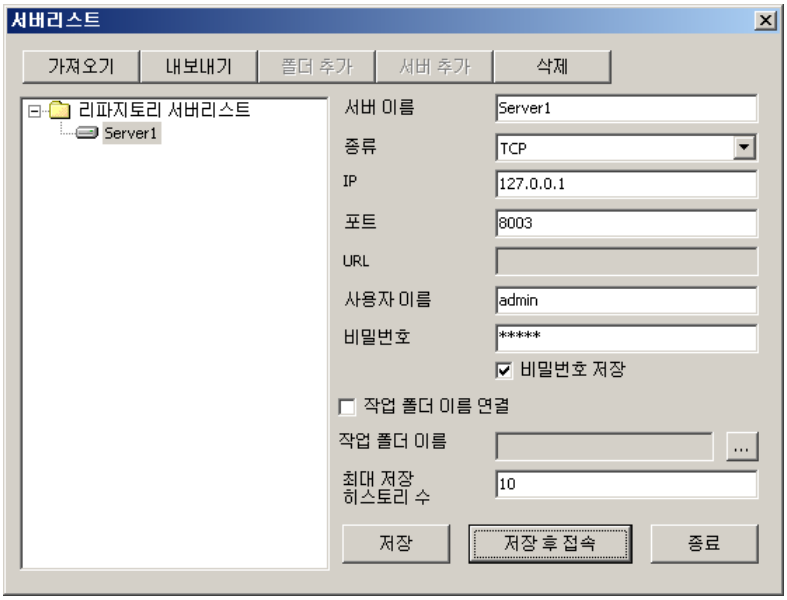
| | |
|---|---|
| **내보내기** | 현재 다이얼로그에 추가되어 있는 서버 리스트 정보를 XML 파일로 저장합니니다. |
| **폴더 추가** | 서버 리파지토리에 카테고리를 추가합니다. |
| **서버 추가** | 새로운 서버 리파지토리를 추가합니다. [서버 추가] 버튼을 클릭하면 새로운 서버 리파지토리를 추가할 수 있도록 옵션 설정이 활성화됩니다. |
| **삭제** | 선택된 서버 또는 카테고리를 삭제합니다. |
|  | 서버 리파지토리에 연결할 작업 폴더 이름을 선택할 수 있도록 작업 폴더 매니저를 실행합니다. |
| **저장** | 현재 서버 리스트 정보를 저장합니다. |
| **저장 후 접속** | 현재 서버 리스트 정보를 저장한 후 선택된 리파지토리 서버로 접속합니다. |
| **종료** | '서버 리스트 다이얼로그'를 닫습니다. |

| Option | Description |
|---|---|
| **서버 이름** | 접속할 서버 리파지토리의 이름을 입력합니다. |
| **종류** | 연결할 OZ Server의 타입을 설정합니다. Daemon 타입일 경우 TCP로, Servlet 타입일 경우 HTTP로 설정합니다. |
| **IP** | Daemon 타입의 서버로 설정할 경우 OZ Server가 실행 중인 IP 주소를 입력합니다. |
| **포트** | Daemon 타입의 서버로 설정할 경우 OZ Server의 Port를 입력합니다. |
| **URL** | Servlet 타입의 서버로 설정할 경우 OZ Server의 URL을 입력합니다. |
| **이름** | 서버 리파지토리에 등록된 사용자 계정을 입력합니다. |
| **비밀번호** | 해당 사용자 계정의 패스워드를 입력합니다. |
| **비밀번호 저장** | 비밀번호 저장 여부를 설정합니다. |
| **작업 폴더 이름 연결** | 서버 리파지토리 접속 시 작업 폴더를 작업 폴더 이름에 설정된 경로로 자동으로 변경할지 여부를 설정합니다.<br>※ 참고사항 : 작업 폴더 이름 연결에 따라 작업 폴더가 변경되는 규칙은 아래 "리파지토리 서버 접속 시 작업 폴더 설정 규칙" 참고 사항을 참조하시기 바랍니다. |
| **작업 폴더 이름** | 서버 접속 시 연결할 작업 폴더 이름이 표시됩니다. |

| | |
|---|---|
| **최대 저장 히스토리 수** | '리파지토리 파일 다이얼로그'에서 표시되는 폴더 히스토리의 개수 중 해당 서버의 폴더 히스토리 개수를 설정합니다. |

**Step 3**

서버 리파지토리 연결 정보를 입력한 다음 [확인] 버튼을 클릭합니다. 본 매뉴얼에서는 데몬 타입의 오즈 서버를 실행한 후 아래와 같이 입력하여 서버 리파지토리를 연결합니다.




비밀번호는 'admin'으로 입력하였습니다.

※ 참고사항 : 리파지토리 서버 접속 시 작업 폴더 설정 규칙

리파지토리 서버 접속 시 작업 폴더 매니저에 설정된 작업 폴더 이름과 연결하여 작업 폴더를 자동으로 변경할 수 있으며, 작업 폴더가 변경되는 규칙은 다음과 같습니다.

1) "작업 폴더 이름 연결" 옵션을 체크한 경우

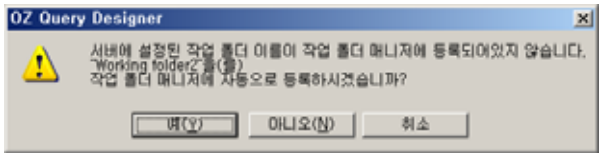
① 작업 폴더 이름이 작업 폴더 매니저에 등록된 경우

리파지토리 서버 연결 시 작업 폴더 매니저에 등록된 경로로 작업 폴더가 자동으로 변경됩니다. 단, 작업 폴더 매니저에 등록된 경로가 로컬에 없을 경우 작업 폴더를 생성한 후 해당 경로로 작업 폴더가 변경됩니다.

② 작업 폴더 이름이 작업 폴더 매니저에 등록되지 않은 경우

리파지토리 서버 연결 시 아래와 같이 작업 폴더 이름을 작업 폴더 매니저에 등록할지 여부를 묻는 메시지 창이 표

시됩니다.

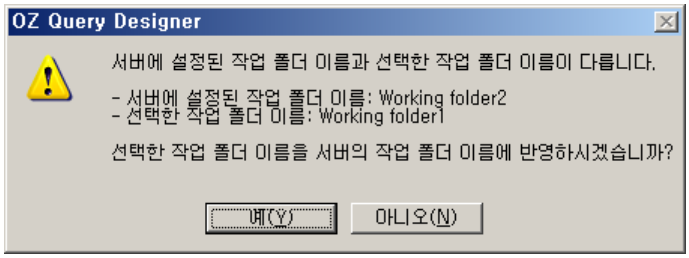



[예] 버튼을 클릭할 경우 작업 폴더 이름과 현재 작업 폴더 경로가 작업 폴더 매니저에 자동으로 등록되며 리파지토리 서버에 접속하고, (이 때, 작업 폴더 트리 제목에 등록된 작업 폴더 이름이 표시됩니다.) [아니오] 버튼을 클릭할 경우 작업 폴더 매니저에 등록하지 않고 리파지토리 서버에 접속하며, [취소] 버튼을 클릭할 경우 작업 폴더 매니저에 등록하지 않고 리파지토리 서버에 접속하지 않습니다.

2) "작업 폴더 이름 연결" 옵션을 체크하지 않은 경우
작업 폴더를 변경하지 않고 리파지토리 서버에 접속합니다.

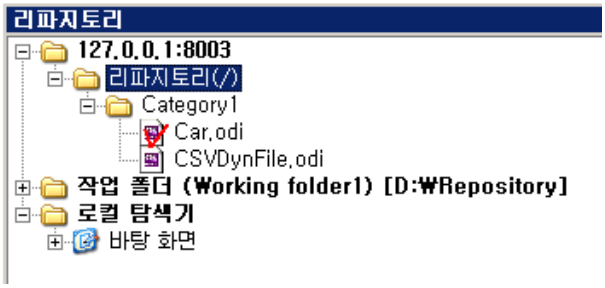
※ 참고사항 : 작업 폴더 이름 변경 규칙

리파지토리 서버에 접속된 상태에서 작업 폴더 매니저 또는 리파지토리 창의 [작업 폴더 설정] 메뉴를 클릭하여 작업 폴더를 변경할 때 접속된 서버에 "작업 폴더 이름 연결" 옵션이 체크되어 있고, 작업 폴더 이름이 작업 폴더 매니저에 등록되어 있으며, 서버에 설정된 작업 폴더 이름과 선택한 작업 폴더 이름이 다를 경우 아래와 같이 서버에 설정된 작업 폴더 이름을 변경할지 여부를 묻는 메시지 창이 표시됩니다



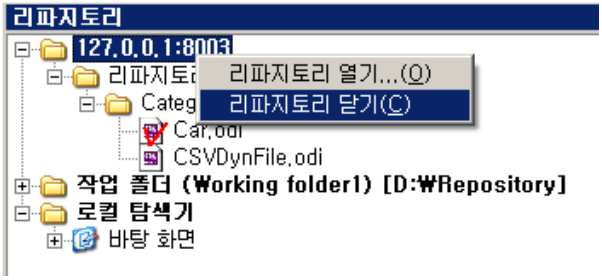

[예] 버튼을 클릭할 경우 선택한 작업 폴더 이름이 서버에 반영되고, [아니오] 버튼을 클릭할 경우 서버에 반영되지 않습니다.

**Step 4** 서버 리파지토리 연결이 완료되면 리파지토리 창에 서버 리파지토리의 폴더 및 파일이 표시됩니다.




## 서버 리파지토리 연결 해제

서버 리파지토리 연결을 해제할 때에는 접속된 리파지토리 서버를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 [리파지토리 닫기] 메뉴를 클릭하면 접속이 해제됩니다.



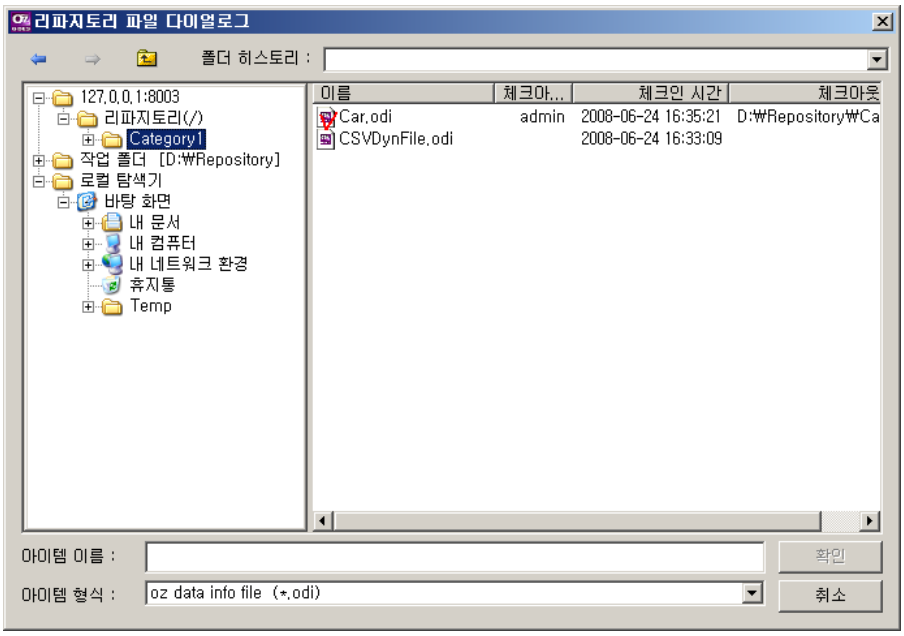
# 서버로부터 파일 열기

서버 리파지토리에 연결된 상태에서는 서버 리파지토리에 저장된 ODI 파일을 직접 열어 편집할 수 있습니다.

## 서버로부터 파일 열기

[파일] 메뉴의 [열기] 메뉴나 툴바의 열기 아이콘()을 클릭하여 서버 리파지토리의 파일을 선택하거나 리파지토리 창에서 해당 파일을 더블 클릭하여 저장된 파일을 읽어 옵니다.

[파일] 메뉴의 [열기] 메뉴나 툴바의 열기 아이콘()을 클릭하면 아래 그림과 같은 '리파지토리 파일 다이얼로그'가 화면에 나타납니다.




| Button | Description |
|---|---|
|  | 현재 선택한 폴더의 이전 폴더의 위치로 이동합니다.<br>※ 참고사항 : 최초로 다이얼로그를 구동하였을 경우 아이콘이 Disable 상태로 표시되며, '리파지토리 파일 다이얼로그'를 새로 띄우면 이전의 폴더를 선택한 기록은 남지 않고 초기화 됩니다. |

| | |
|---|---|
|  | 이전 폴더로 이동하는 아이콘을 클릭하여 현재 선택된 폴더의 이전 폴더로 이동한 후 해당 아이콘을 클릭하면 이동되기 이전의 폴더 위치로 다시 이동합니다.<br>※ 참고사항 : 최초로 다이얼로그를 구동하였을 경우 아이콘이 Disable 상태로 표시되며, '리파지토리 파일 다이얼로그'를 새로 띄우면 이전의 폴더를 선택한 기록은 남지 않고 초기화 됩니다. |
|  | 현재 선택한 폴더의 상위 폴더로 이동합니다.<br>※ 참고사항 : 현재 선택된 폴더가 최상위 폴더인 경우 아이콘이 Disable 상태로 표시됩니다. |

| Option | Description |
|---|---|
| **폴더 히스토리** | 최근에 파일을 열었던 폴더의 순서로 목록이 표시됩니다.<br>로컬에서 파일을 열었던 폴더의 목록이 먼저 표시되고 현재 접속중인 서버 리파지토리에서 파일을 열었던 폴더의 목록이 그 아래 표시됩니다.<br>로컬 리파지토리 폴더의 경우에는 "ozp://" 형식으로, 서버 리파지토리 폴더의 경우에는 "ozpserver://" 형식으로 표시됩니다.<br>목록에 표시되는 히스토리 수의 기본 값은 로컬/서버 리파지토리 각 10개씩 즉, 20개입니다.<br>※ 주의사항 : 히스토리에서 선택한 파일 목록의 경로가 실제 리파지토리에 존재하지 않는 경로일 경우 현재 선택되어진 폴더가 표시됩니다.<br>※ 참고사항<br>▪ 폴더 히스토리에 표시되는 히스토리의 최대 개수는 '서버 리스트 다이얼로그'에서 설정한 "Max history count"와 로컬 리파지토리의 히스토리 수가 합쳐진 수만큼 목록에 표시됩니다. 예를 들어 서버 리파지토리의 히스토리 개수가 10이고 로컬 리파지토리의 히스토리 개수가 10인 경우 '리파지토리 파일 다이얼로그'의 폴더 히스토리 목록에 표시되는 최대 개수는 20개입니다.<br>▪ 폴더 히스토리의 개수는 conf/OZServerList.xml 파일에서 확인할 수 있습니다.<br>▪ 서버 리파지토리의 히스토리 수는 해당 서버의 "MaxHistoryCount" 옵션 즉, "<SERVER Name=..."의 "MaxHistoryCount"에 설정되어있는 값이고, 로컬 리파지토리의 히스토리 수는 "<LOCAL MaxHistoryCount="에 설정되어있는 값입니다. |

## 서버로부터 파일 열기의 주의사항

- 서버로부터 열기한 파일의 편집한 내용을 서버 리파지토리에 반영하기 위해서는 반드시 체크아웃을 먼저하고 내용을 편집한 다음 체크인을 하는 과정을 거쳐야 합니다.

- 서버로부터 열기한 파일이 체크아웃 상태에서 편집을 마친 후에는 반드시 체크인하거나 체크아웃을 취소하여야 합니다.

- 파일을 서버로부터 열기할 때 다운로드하여 로컬의 작업 폴더에 저장한 경우에는 [파일] 메뉴의 [저장] 메뉴나 툴바의 저장 아이콘()을 클릭하여 저장하는 등의 작업은 로컬의 파일에 저장시키는 것입니다. 즉 다시 파일을 서버로부터 열기를 하면 서버 리파지토리에 있는 파일이 열리기 때문에 로컬의 작업 폴더에 저장한 내용과 다를 수 있습니다.

# 카테고리 및 아이템 관리

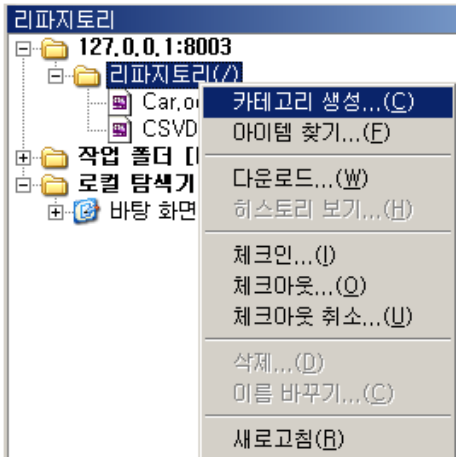
## 카테고리 및 아이템의 개요

오즈 리파지토리에 저장되는 데이터 정보 및 폼 파일들을 아이템이라 하며, 해당 아이템을 그룹핑한 아이템 그룹을 카테고리로 정의합니다.

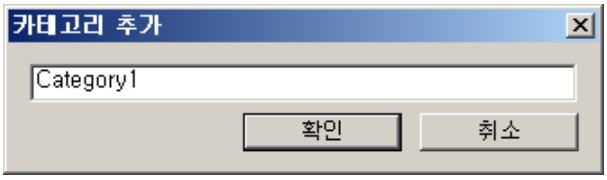
리파지토리 창을 이용하여 리파지토리 서버에 카테고리를 추가하고, 아이템 즉 파일 업로드, 다운로드, 삭제, 이름 변경 등을 할 수 있습니다.

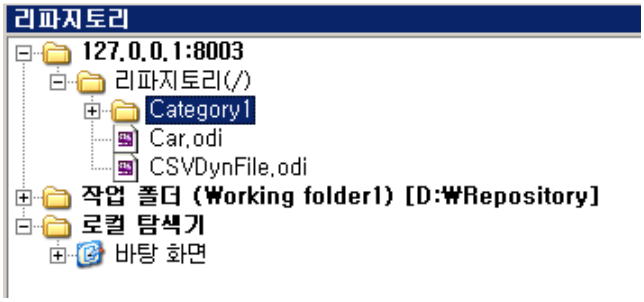
## 카테고리 추가

카테고리를 추가하고자 하는 카테고리에서 마우스 오른쪽 버튼을 클릭하여 나타나는 팝업 메뉴에서 [카테고리 생성] 메뉴를 클릭합니다.



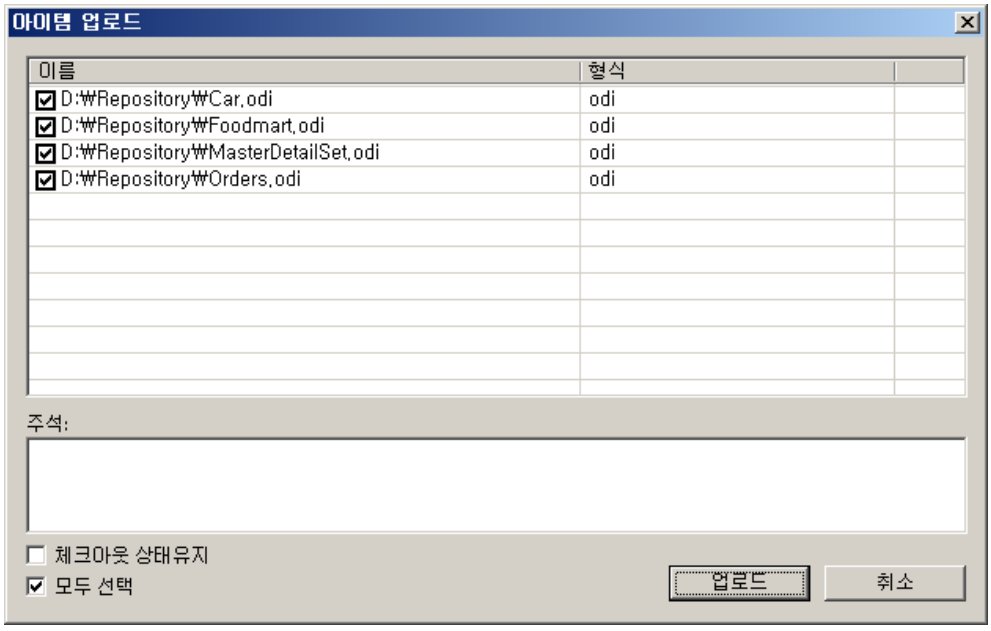

'카테고리 추가 다이러로그'가 나타나면 추가하고자 하는 카테고리명을 입력한 후 [확인] 버튼을 클릭하면 새로운 카테고리가 만들어집니다.






## 파일 업로드

작업 폴더에서 업로드할 파일을 선택한 후 리파지토리 서버의 원하는 카테고리로 드래그&드롭하면 '아이템 업로드 다이러로그'가 표시되고 업로드할 아이템 리스트를 보여줍니다.




[업로드] 버튼을 클릭하게 되면 선택된 아이템이 리파지토리 서버의 지정한 카테고리에 추가되게 됩니다. 한 개 이상의 파일을 선택할 때에는 <Ctrl> 키나 <Shift> 키를 누른 상태에서 원하는 파일을 마우스로 클릭합니다. 원하는 파일들을 선택한 후 드래그하여 추가하고자 하는 카테고리에 가져가면 선택된 아이템 리스트가 기록된 '아이템 업로드 다이얼로그'가 나타납니다. 추가하고자 하는 아이템을 선택한 후 [업로드] 버튼을 클릭하면 선택된 모든 아이템이 추가됩니다.

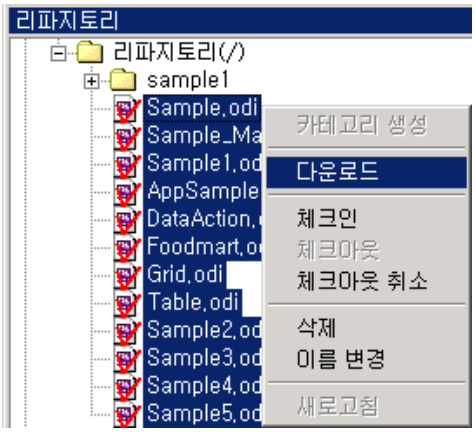
이때 선택된 아이템 중 특정 아이템을 제외하고자 하는 경우 '아이템 업로드 다이얼로그'에 나타난 아이템 리스트 중에서 이름에서의 체크 박스에 체크된 부분을 지우면 특정 아이템을 제외하고

추가할 수 있습니다

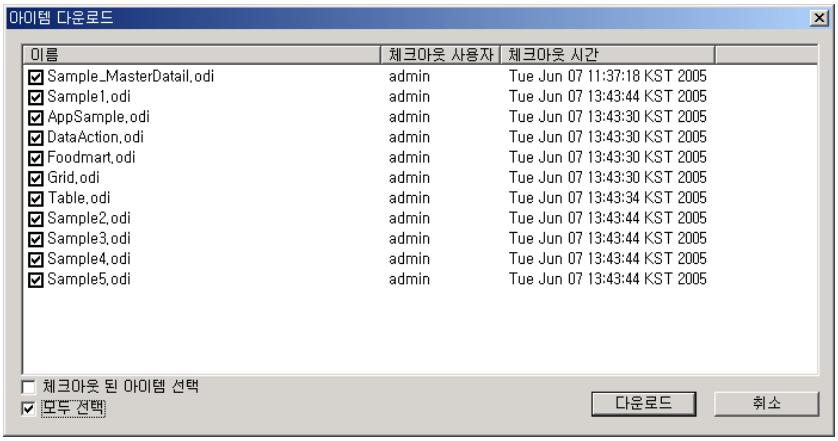
※ 주의사항 : 쿼리 디자이너에 포함된 리파지토리 클라이언트에서는 odi 파일만 아이템으로 추가할 수 있습니다.

## 파일 다운로드

리파지토리 서버에서 다운로드할 파일을 선택한 후 마우스 오른쪽 버튼을 눌러 나타나는 팝업 메뉴에서 [다운로드] 메뉴를 클릭합니다.



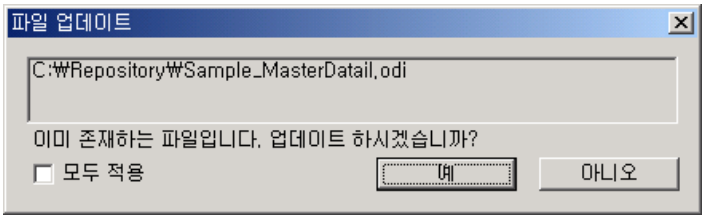

'아이템 다운로드 다이얼로그'에서 다운로드할 아이템을 선택한 후 [다운로드] 버튼을 클릭하여 로컬 시스템의 작업 폴더로 다운로드 합니다.




'아이템 다운로드 다이얼로그'에서 '체크아웃된 아이템 선택' 옵션을 체크하면 현재 체크 아웃된 아이템(다른 사용자가 체크 아웃한 아이템 포함)에 대해서도 다운로드를 할 수 있습니다.

로컬 시스템의 작업 폴더 내에 동일한 이름의 파일이 있을 경우에는 업데이트 할 것인지 묻

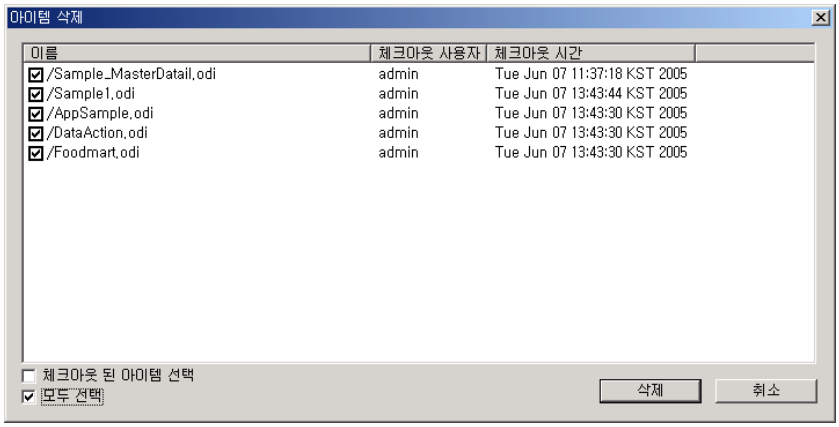
는 확인 창이 나타나고 그렇지 않을 경우에는 바로 다운로드 됩니다.




## 카테고리, 파일 삭제

삭제하고자 하는 카테고리 또는 파일을 마우스 오른쪽 버튼을 클릭하여 나타나는 팝업 메뉴에서 [삭제] 메뉴를 클릭하면 '아이템 삭제 다이얼로그'가 표시되고 삭제되는 아이템 리스트를 나타납니다.

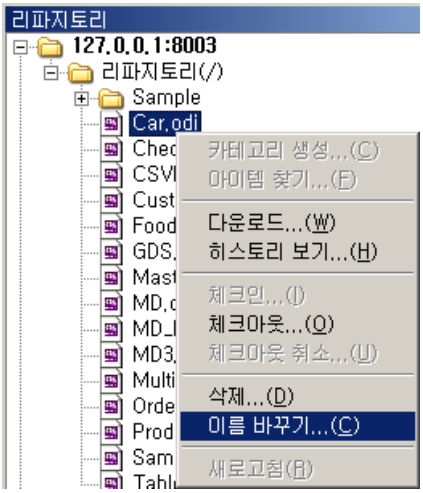


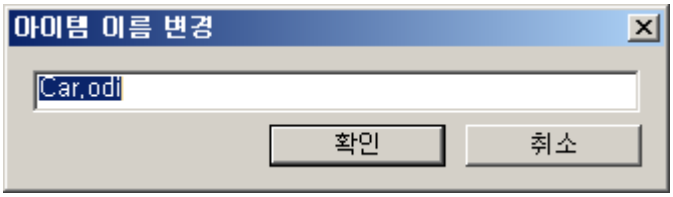



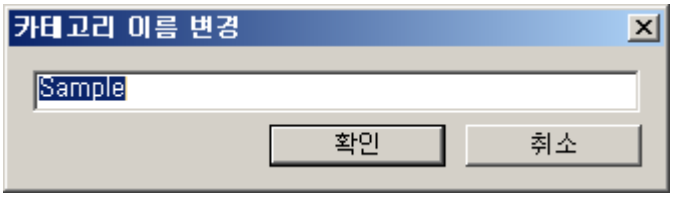

삭제할 아이템을 선택한 후 [삭제] 버튼을 클릭하면 리파지토리 서버에서 해당 아이템이나 카테고리가 삭제됩니다.

## 이름 변경

이름을 변경하고자 하는 카테고리 또는 아이템에서 마우스 오른쪽 버튼을 클릭하여 나타나는 팝업 메뉴에서 [이름 변경] 메뉴를 클릭하면 해당 카테고리 또는 아이템의 이름을 변경할 수 있는 '카테고리 이름 변경 다이얼로그'나 '아이템 이름 변경 다이얼로그'가 나타납니다. 해당 창에 변경하고자 하는 이름을 입력하고 [확인] 버튼을 클릭하면 아이템이나 카테고리 이름이 변경됩니다.



# 체크인, 체크아웃

오즈 리파지토리 서비스는 다중 사용자 개발 환경을 제공하며, 다중 개발자 환경에서 파일의 안전한 공유와 갱신을 위해 체크인, 체크아웃 기능을 제공합니다.
Admin 계정으로 'None' 타입의 서버에 접속한 경우에는 모든 파일에 대해 항상 체크아웃한 상태로 서버에 접속되며, 체크인, 체크아웃 기능을 정상적으로 사용하기 위해서는 오즈 서버의 리파지토리 타입이 BuiltIn 또는 RDB 방식이어야 합니다. 자세한 설명은 '오즈 엔터프라이즈 서버 관리자 매뉴얼'을 참조하십시오.

## 체크인, 체크아웃

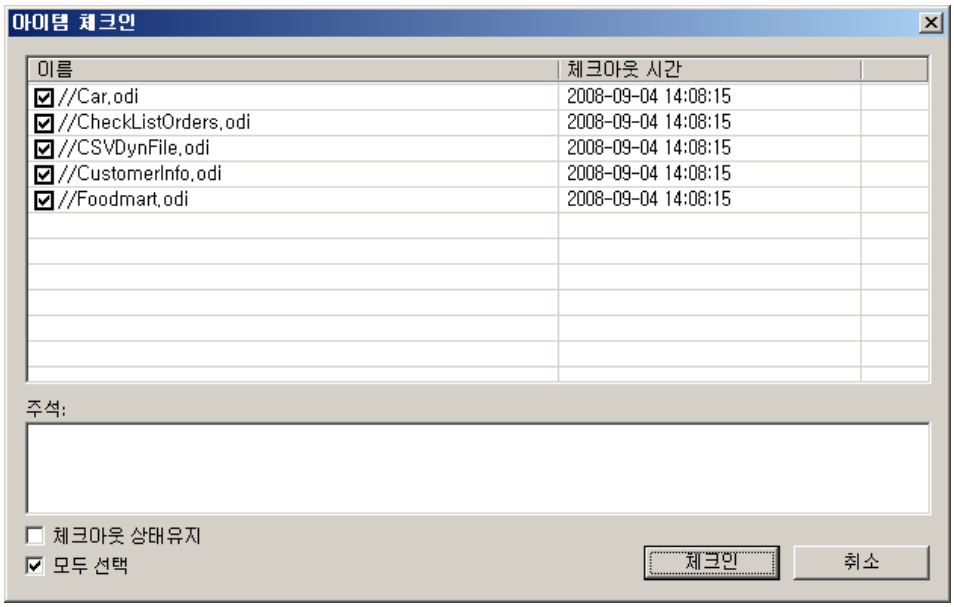
오즈 리파지토리 서비스는 파일의 안전한 공유와 갱신을 위해 하나의 파일이 한 명 이상의 개발자에 의해 동시에 수정되거나 갱신되는 것을 허용하지 않습니다. 이를 위해 리파지토리 서버는 어떤 순간에도 각각의 파일에 대한 수정 권한을 오직 한 명의 개발자에게만 부여하고, 리파지토리 서버가 관리하는 파일을 갱신하기 위해서는 반드시 리파지토리 서버로부터 갱신 권한을 받도록 제한합니다.
이렇게 리파지토리 서비스가 정의하는 원칙에 따라 리파지토리 서버의 파일을 수정하고 갱신할 수 있는 권한을 받는 과정을 체크아웃이라하고, 반대로 수정된 파일을 리파지토리에 등록하고 갱신 권한을 반납하는 과정을 체크인이라고 합니다.

■ **체크인**

체크인은 서버 리파지토리에서 체크아웃 후 수정된 내용을 서버 리파지토리에 저장하고 갱신 권한을 반납하는 기능입니다. 리파지토리 창에서 체크인할 파일을 선택한 후 마우스 오른쪽 버튼을 클릭하여 나타나는 팝업 메뉴에서 [체크인] 메뉴를 클릭합니다.



'체크인 다이얼로그'에서 체크인할 파일을 체크한 후 [Check In] 버튼을 클릭하여 체크인 합니다.




만일 체크아웃 상태인 파일을 수정하였을 경우 수정된 내용을 서버 리파지토리에 저장하고 갱신 권한을 반납하지 않고자 할 경우에는 '체크아웃 상태 유지' 옵션을 체크한 후 [체크인] 버튼을 클릭하십시오. 이 옵션을 이용하면 체크아웃 상태인 파일을 서버 리파지토리에 저장하고 갱신 권한을 계속 갖고자 할 경우 체크인을 하고 다시 체크아웃을 해야 하는 번거로움을 줄일 수 있습니다.

※ 주의사항 : 서버 리파지토리의 파일을 체크인으로 변경한 상태에서 서버 리파지토리 연결을 해제한 후 다시 Admin 계정으로 'None' 타입의 서버 리파지토리를 연결할 경우 Admin 계정은 모든 파일에 대해 항상 체크아웃한 상태로 서버에 접속되기 때문에, 접속 전에 체크인으로 변경한 파일도 모두 체크아웃 상태로 자동 변경됩니다.

■ **체크아웃**

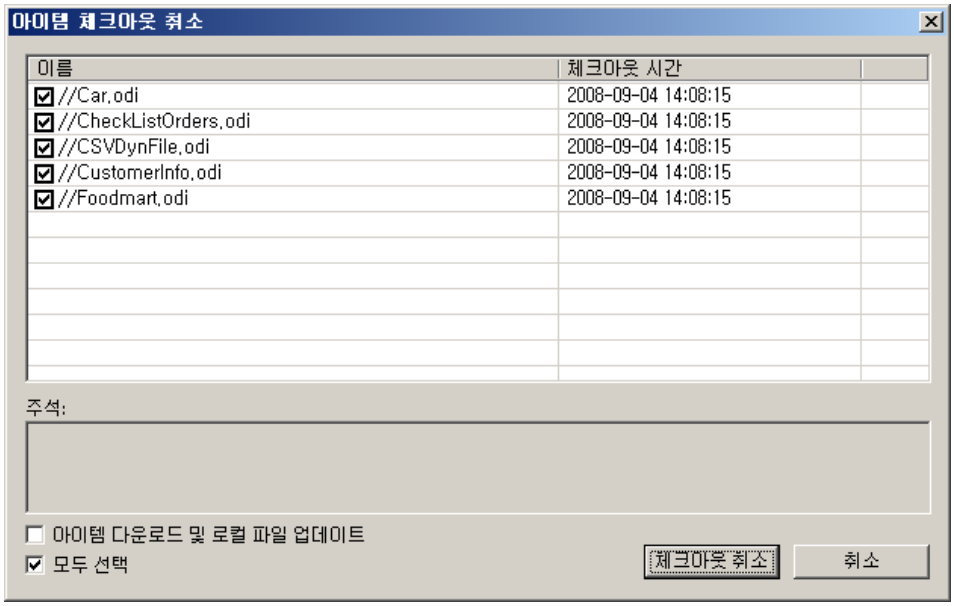
체크아웃은 서버 리파지토리에 저장된 파일의 수정을 위해 리파지토리 서버로부터 특정 파일에 대한 갱신 권한을 얻어오는 기능입니다.

리파지토리 창에서 체크아웃할 파일을 선택한 후 마우스 오른쪽 버튼을 클릭하여 나타나는 팝업 메뉴에서 [체크아웃] 메뉴를 클릭하고, '체크아웃 다이얼로그'에서 체크아웃할 파일을 체크한 후 [체크아웃] 버튼을 클릭하여 체크아웃 합니다.

■ **체크아웃 취소**

체크아웃 취소는 체크아웃된 파일의 수정 내용을 서버 리파지토리에 반영하지 않고 단순히 갱신 권한을 반납하는 기능입니다.

리파지토리 창에서 체크아웃 취소할 파일을 선택한 후 마우스 오른쪽 버튼을 클릭하여 나타나는 팝업 메뉴에서 [체크아웃 취소] 메뉴를 클릭하고, '체크아웃 취소 다이얼로그'에서 체크아웃 취소할 파일을 체크한 후 [체크아웃 취소] 버튼을 클릭하여 체크아웃을 취소합니다.



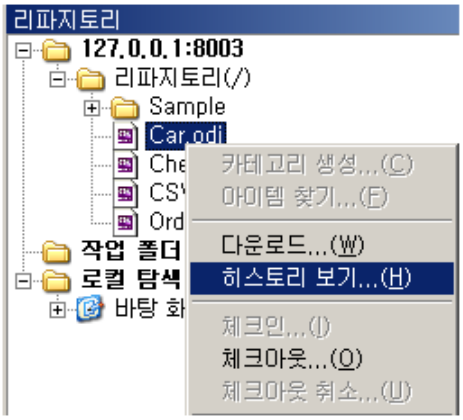

체크아웃을 취소할 때 만일 로컬 시스템의 작업 폴더에 선택한 파일을 다운로드하여 로컬 시스템의 작업 폴더의 파일을 최신 버전으로 갱신할 경우에는 '아이템 다운로드 및 로컬 파일 업데이트' 옵션을 체크하십시오.

# 히스토리 보기

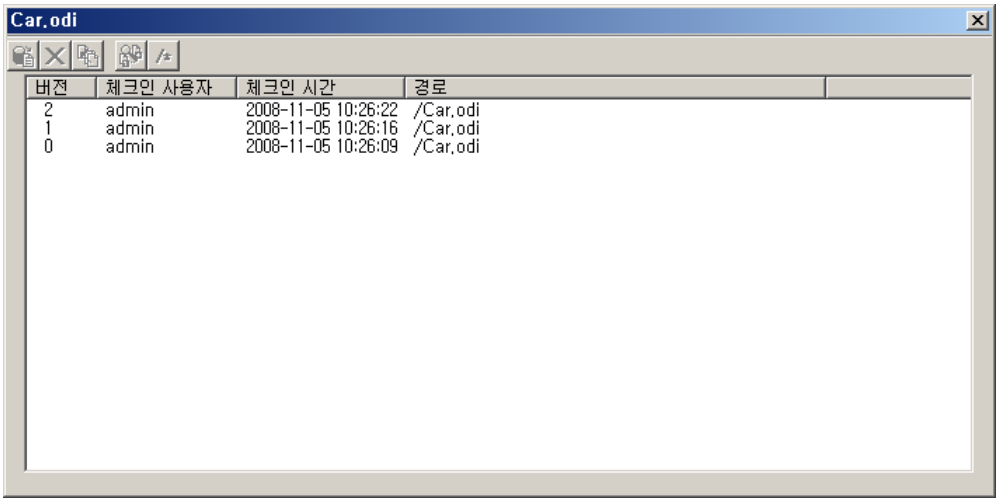
아이템을 체크인 또는 체크아웃하여 각 아이템별로 히스토리를 남길 경우 히스토리 보기 기능을 사용하여 아이템의 내역을 볼 수 있습니다.

## 아이템 히스토리 보기

서버 리파지토리에 있는 아이템을 선택한 후 마우스 오른쪽 버튼을 클릭하여 나타나는 팝업 메뉴에서 [히스토리 보기] 메뉴를 클릭합니다.




[히스토리 보기] 메뉴를 클릭하면 선택한 아이템의 히스토리를 나타낸 다이얼로그가 실행됩니다.




버전은 아이템의 히스토리 버전으로 아이템 체크인 시에 자동으로 부여되며, 낮은 숫자일수록 오

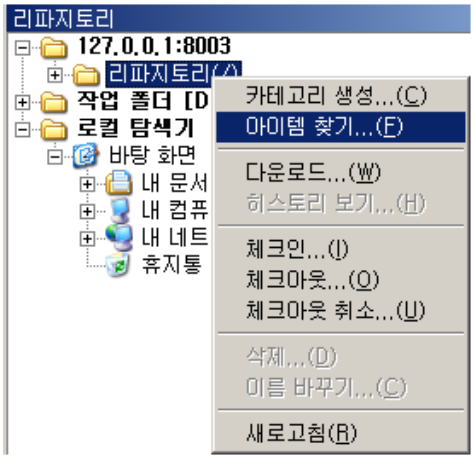
래된 버전입니다.
특정 버전을 선택하여 다운로드, 주석 조회, 롤백, 히스토리 삭제 등의 기능을 실행할 수 있 으며, 두 개의 버전을 선택하여 버전 간 스크립트를 비교할 수 있습니다.

버전 간 스크립트를 비교하는 방법은 본 매뉴얼의 "IV. 오즈 쿼리 디자이너 사용 – 스크립트 비교" 부분을 참조하시기 바랍니다.

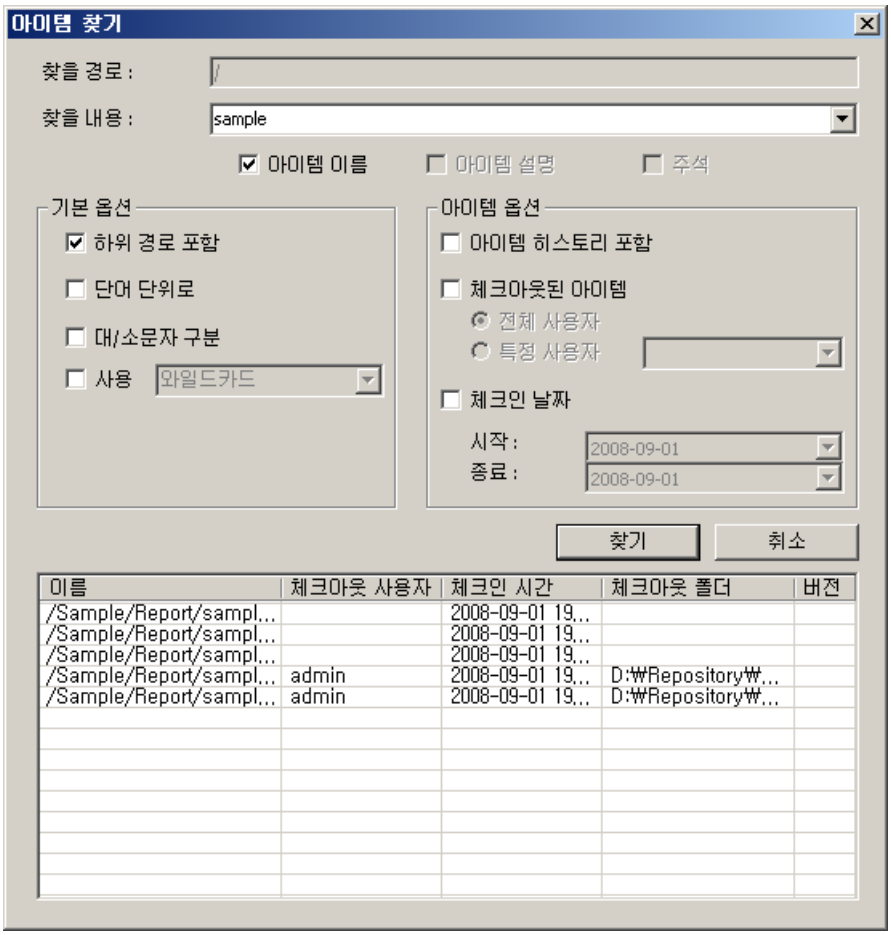
# 아이템 찾기

## 아이템 찾기

서버 리파지토리에서 검색할 카테고리를 선택한 후 마우스 오른쪽 버튼을 클릭하여 나타나는 팝업 메뉴에서 [아이템 찾기] 메뉴를 클릭합니다.




[아이템 찾기] 메뉴를 클릭하면 "아이템 찾기 다이얼로그"가 실행되고, 찾을 경로에는 선택한 카테고리가 자동으로 표시됩니다. 찾을 내용, 기본 옵션, 아이템 옵션 등을 설정한 후 [찾기] 버튼을 클릭하면 검색된 아이템이 리스트 형태로 표시됩니다.




검색된 아이템을 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴를 이용하여 체크인, 체크아웃, 체크아웃 취소 등의 기능을 실행할 수 있습니다.



## 아이템 찾기 관련 주의사항

- 아이템 찾기 기능은 서버 리파지토리에서만 사용 가능합니다. (로컬 작업 폴더에서는 사용할 수 없습니다.)

- 주석으로 찾을 경우 검색 옵션 중 "아이템 히스토리 포함" 옵션을 반드시 체크하시기 바랍니다.

- 이력 정보 기능을 사용하지 않는 일반적인 None 타입 서버의 경우 체크인, 체크아웃 등 히스토리 기능을 지원하지 않는 서버이므로, "아이템 찾기 다이얼로그"에서 "아이템 옵션"이 모두 비활성화되어 표시됩니다.

- 버전 정보는 "주석", "아이템 히스토리 포함" 옵션이 체크된 경우에만 버전 정보를 가져와서 표시하며, 다른 경우에는 버전 정보를 표시하지 않습니다. 버전의 숫자가 낮을수록 오래된 버전이며, 버전 정보가 있는 히스토리 아이템의 경우 해당 아이템의 팝업 메뉴를 띄우면 [다운로드] 메뉴와 [주석] 메뉴만 활성화됩니다.

# Appendix. 문자셋과 유효한 앨리어스

다음은 데이터베이스 연결정보의 인코딩, 디코딩 셋에 들에 들어가는 문자셋 이름입니다. 각 문자셋 이름과 해당 문자셋과 같은 의미를 가지는 앨리어스 이름을 나열하면 다음과 같습니다.

※ 주의사항 : "유효한 앨리어스 이름"이 공란으로 되어 있는 문자셋 이름은 사용할 수 없습니다.

| 문자셋 이름 | 유효한 앨리어스 이름 |
|---|---|
| **ASCII** | us-ascii, ascii, 646, iso646-us, default, ascii7 |
| **Big5** | big5 |
| **Cp037** | ibm037, ibm-037, cp037, 037 |
| **Cp1006** | ibm1006, ibm-1006, cp1006, 1006 |
| **Cp1025** | ibm1025, ibm-1025, cp1025, 1025 |
| **Cp1026** | ibm1026, ibm-1026, cp1026, 1026 |
| **Cp1046** | |
| **Cp1097** | ibm1097, ibm-1097, cp1097, 1097 |
| **Cp1098** | ibm1098, ibm-1098, cp1098, 1098 |
| **Cp1112** | ibm1112, ibm-1112, cp1112, 1112 |
| **Cp1122** | ibm1122, ibm-1122, cp1122, 1122 |
| **Cp1123** | ibm1123, ibm-1123, cp1123, 1123 |
| **Cp1124** | ibm1124, ibm-1124, cp1124, 1124 |
| **Cp1140** | |
| **Cp1141** | |
| **Cp1142** | |
| **Cp1143** | |
| **Cp1144** | |
| **Cp1145** | |
| **Cp1146** | |
| **Cp1147** | |
| **Cp1148** | |
| **Cp1149** | |
| **Cp1250** | windows-1250 |
| **Cp1251** | windows-1251 |

| 문자셋 이름 | 유효한 앨리어스 이름 |
|---|---|
| **Cp1252** | windows-1252 |
| **Cp1253** | windows-1253 |
| **Cp1254** | windows-1254 |
| **Cp1255** | windows-1255 |
| **Cp1256** | windows-1256 |
| **Cp1257** | windows-1257 |
| **Cp1258** | windows-1258 |
| **Cp1381** | ibm1381, ibm-1381, cp1381, 1381 |
| **Cp1383** | ibm1383, ibm-1383, cp1383, 1383 |
| **Cp273** | ibm273, ibm-273, cp273, 273 |
| **Cp277** | ibm277, ibm-277, cp277, 277 |
| **Cp278** | ibm278, ibm-278, cp278, 278 |
| **Cp280** | ibm280, ibm-280, cp280, 280 |
| **Cp284** | ibm284, ibm-284, cp284, 284 |
| **Cp285** | ibm285, ibm-285, cp285, 285 |
| **Cp297** | ibm297, ibm-297, cp297, 297 |
| **Cp33722** | Ibm33722, ibm-33722, cp33722, 33722 |
| **Cp420** | ibm420, ibm-420, cp420, 420 |
| **Cp424** | ibm424, ibm-424, cp424, 424 |
| **Cp437** | ibm437, ibm-437, cp437, 437 |
| **Cp500** | ibm500, ibm-500, cp500, 500 |
| **Cp737** | ibm737, ibm-737, cp737, 737 |
| **Cp775** | ibm775, ibm-775, cp775, 775 |
| **Cp838** | ibm838, ibm-838, cp838, 838 |
| **Cp850** | ibm850, ibm-850, cp850, 850, cspc850multilingual |
| **Cp852** | ibm852, ibm-852, cp852, 852, cspcp852 |
| **Cp855** | ibm855, ibm-855, cp855, 855, cspcp855 |
| **Cp856** | ibm856 ibm-856, cp856 856 |
| **Cp857** | ibm857, ibm-857, cp857, 857, csibm857 |
| **Cp858** | |

| 문자셋 이름 | 유효한 앨리어스 이름 |
|---|---|
| **Cp860** | ibm860, ibm-860, cp860, 860, csibm860 |
| **Cp861** | ibm861, ibm-861, cp861, cp-is, 861, csibm861 |
| **Cp862** | ibm862, ibm-862, cp862, cp862, 862, cspc862latinhebrew |
| **Cp863** | ibm863, ibm-863, cp863, 863, csibm863 |
| **Cp864** | ibm864, ibm-864, cp864, csibm864 |
| **Cp865** | ibm865, ibm-865, cp865, 865, csibm865 |
| **Cp866** | ibm866, ibm-866, cp866, 866, csibm866 |
| **Cp868** | ibm868, ibm-868, cp868, 868 |
| **Cp869** | ibm869, ibm-869, cp869, 869, cp-gr, csibm869 |
| **Cp870** | ibm870, ibm-870, cp870, 870 |
| **Cp871** | ibm871, ibm-871, cp871, 871 |
| **Cp874** | ibm874, ibm-874, cp874, 874 |
| **Cp875** | ibm875, ibm-875, cp875, 875 |
| **Cp918** | ibm918, ibm-918, cp918, 918 |
| **Cp921** | ibm921, ibm-921, cp921, 921 |
| **Cp922** | ibm922, ibm-922, cp922, 922 |
| **Cp923** | |
| **Cp930** | ibm930, ibm-930, cp930, 930 |
| **Cp933** | ibm933, ibm-933, cp933, 933 |
| **Cp935** | ibm935, ibm-935, cp935, 935 |
| **Cp937** | ibm937, ibm-937, cp937, 937 |
| **Cp939** | ibm939, ibm-939, cp939, 939 |
| **Cp942** | ibm942, ibm-942, cp942, 942 |
| **Cp942C** | cp942c |
| **Cp943** | ibm943, ibm-943, cp943, 943 |
| **Cp943C** | cp943c |
| **Cp948** | ibm948, ibm-948, cp948, 948 |
| **Cp949** | ibm949, ibm-949, cp949, 949 |
| **Cp949C** | cp949c |
| **Cp950** | ibm950, ibm-950, cp950, 950 |

| 문자셋 이름 | 유효한 앨리어스 이름 |
|---|---|
| **Cp964** | ibm964, ibm-964, cp964, 964 |
| **Cp970** | ibm970, ibm-970, cp970, 970 |
| **DBCS_ASCII** | |
| **DBCS_EBCDIC** | |
| **Default** | |
| **DoubleByte** | |
| **EUC** | |
| **EUC_CN** | gb2312, gb2312-80, gb2312-1980, euc-cn, euccn |
| **EUC_JP** | eucjis, euc-jp, eucjp, extended_unix_code_packed_format_for_japanese, cseucpkdfmtjapanese, x-euc-jp, x-eucjp |
| **EUC_KR** | ksc5601, euc-kr, euckr, ks_c_5601-1987, ksc5601-1987, ksc5601_1987, ksc_5601, 5601 |
| **EUC_TW** | cns11643, euc-tw, euctw |
| **GBK** | |
| **ISO2022** | |
| **ISO2022CN** | |
| **ISO2022JP** | jis, iso-2022-jp, csiso2022jp, jis_encoding, csjisencoding |
| **ISO2022KR** | |
| **ISO8859_1** | 8859_1, iso_8859-1:1978, iso-ir-100, iso_8859-1, iso-8859-1, iso8859-1, latin1, l1, ibm819, ibm-819, cp819, 819, csisolatin1 |
| **ISO8859_15_FDIS** | 8859_15, iso-8859-15, iso_8859-15, iso8859-15, ibm923, ibm-923, cp923, 923, latin0, latin9, csisolatin0, csisolatin9 |
| **ISO8859_2** | 8859_2, iso_8859-2:1987, iso-ir-101, iso_8859-2, iso-8859-2, iso8859-2, latin2, l2, ibm912, ibm-912, cp912, 912, csisolatin2, |
| **ISO8859_3** | 8859_3, iso_8859-3:1988, iso-ir-109, iso_8859-3, iso-8859-3, iso8859-3, latin3, l3, ibm913, ibm-913, cp913, 913, csisolatin3 |
| **ISO8859_4** | 8859_4, iso_8859-4:1988, iso-ir-110, iso_8859-4, iso-8859-4, iso8859-4, latin4, l4, ibm914, ibm-914, cp914, 914, csisolatin4 |

| 문자셋 이름 | 유효한 앨리어스 이름 |
|---|---|
| **ISO8859_5** | 8859_5, iso_8859-5:1988, iso-ir-144, iso_8859-5, iso-8859-5,iso8859-5, cyrillic, csisolatincyrillic, ibm915, ibm-915, cp915, 915, |
| **ISO8859_6** | 8859_6, iso_8859-6:1987, iso-ir-127, iso_8859-6, iso-8859-6, iso8859-6, ecma-114, asmo-708, arabic, csisolatinarabic, ibm1089, ibm-1089, cp1089, 1089 |
| **ISO8859_7** | 8859_7, iso_8859-7:1987, iso-ir-126, iso_8859-7, iso-8859-7, iso8859-7, elot_928, ecma-118, greek, greek8, csisolatingreek, ibm813, ibm-813, cp813, 813 |
| **ISO8859_8** | 8859_8, iso_8859-8:1988, iso-ir-138, iso_8859-8, iso-8859-8, iso8859-8, hebrew, csisolatinhebrew, ibm916, ibm-916, cp916, 916 |
| **ISO8859_9** | 8859_9, iso-ir-148, iso_8859-9, iso-8859-9, iso8859-9, latin5, l5, ibm920, ibm-920, cp920, 920, csisolatin5 |
| **ISO8859_13** | 8859_13, iso_8859-13, iso-8859-13, iso8859-13, |
| **JIS0201** | |
| **JIS0208** | |
| **JIS0212** | |
| **JISAutoDetect** | jis auto detect |
| **Johab** | ksc5601-1992, ksc5601_1992, ms1361 |
| **KOI8_R** | koi8-r, koi8, cskoi8r |
| **MS874** | windows-874 |
| **MS932** | shift_jis, ms_kanji, csshiftjis, windows-31j, cswindows31j, x-sjis |
| **MS932DB** | |
| **MS936** | |
| **MS949** | windows-949 |
| **MS950** | |
| **MacArabic** | |
| **MacCentralEurope** | |
| **MacCroatian** | |

| 문자셋 이름 | 유효한 앨리어스 이름 |
|---|---|
| **MacCyrillic** | |
| **MacDingbat** | |
| **MacGreek** | |
| **MacHebrew** | |
| **MacIceland** | |
| **MacRoman** | |
| **MacRomania** | |
| **MacSymbol** | |
| **MacThai** | |
| **MacTurkish** | |
| **MacUkraine** | |
| **SJIS** | シフト符?現表化, pck |
| **SingleByte** | |
| **TIS620** | tis620.2533 |
| **UTF8** | utf-8, unicode-1-1-utf-8 |
| **UTF16** | utf-16 |
| **Unicode** | unicode |
| **UnicodeBig** | unicode-1-1, iso-10646-ucs-2, utf-16be, x-utf-16be |
| **UnicodeLittle** | utf-16le, x-utf-16le |