

# **CUBRID 2008 R4.0 매니저 매뉴얼**

---



# Table of Contents

<b>CUBRID 매니저</b>	<b>1</b>
<b>CUBRID 매니저 소개</b>	<b>2</b>
CUBRID 매니저 구조	2
CUBRID 매니저 클라이언트	2
<b>CUBRID 매니저 실행</b>	<b>9</b>
CUBRID 매니저 서버 환경 설정	9
CUBRID 매니저 서버 실행	10
CUBRID 매니저 클라이언트 실행	11
<b>호스트 관리</b>	<b>13</b>
기본 호스트 정보	13
호스트 연결 및 연결 해제	14
호스트 그룹 관리	15
호스트 추가 및 삭제	18
<b>서비스 관리</b>	<b>20</b>
<b>사용자 관리</b>	<b>22</b>
사용자 권한 설정	22
사용자 추가, 편집 및 삭제	23
<b>속성 관리</b>	<b>25</b>
<b>질의 편집기</b>	<b>31</b>
질의 편집기 구조	31
툴바 기능	32
질의 편집 창	35
질의 결과 창	37
질의 편집기 옵션	41
질의 실행 계획 보기	43
<b>데이터베이스</b>	<b>47</b>
데이터베이스 기능 구조	47
데이터베이스 연결	48
데이터베이스 시작	49
데이터베이스 생성	50
데이터베이스 설정	54
사용자	56
테이블	58
뷰	70
트리거	72

시리얼	74
저장 프로시저	75
작업 자동화	77
저장 공간	81
데이터베이스 로드	83
데이터베이스 언로드	85
데이터베이스 백업	87
데이터베이스 복구	89
데이터베이스 이름 변경	91
데이터베이스 복사	92
데이터베이스 최적화	93
데이터베이스 공간 정리	94
데이터베이스 검사	95
데이터베이스 잠금 정보	95
데이터베이스 트랜잭션 정보	98
데이터베이스 삭제	99
OID 탐색기	99
다중 호스트 환경에서의 백그라운드 운영	100
사용 중인 매개 변수 정보	101
질의 수행 계획 캐쉬 정보	102
<b>브로커</b>	<b>105</b>
브로커 구조	105
개별 브로커 기능	105
SQL 로그	105
<b>상태 모니터</b>	<b>109</b>
<b>대시보드</b>	<b>114</b>
대시보드 추가	114
대시보드 관리	118
대시보드 창	119
<b>로그</b>	<b>128</b>
<b>CUBRID 매니저 관리자 안내서</b>	<b>131</b>
CUBRID 매니저 관리자 유틸리티	132
CUBRID 매니저 관리자 유틸리티	133
DBMT 사용자	134
DBMT 사용자 추가	135
DBMT 사용자 삭제	137
DBMT 사용자 정보 출력	138

DBMT 사용자 권한 변경	139
DBMT 사용자 비밀번호 변경	140
DBMT 사용자의 데이터베이스 정보 추가	141
DBMT 사용자의 데이터베이스 정보 삭제	143
DBMT 사용자의 데이터베이스 정보 변경	144



# CUBRID 매니저

CUBRID 매니저는 CUBRID 데이터베이스 관리 및 질의 기능을 GUI 환경에서 제공하는 CUBRID 데이터베이스 전용 관리 도구이다. CUBRID 매니저를 사용하면 다양한 관리 작업을 쉽게 할 수 있다. CUBRID 매니저는 데이터베이스 관리 및 응용 개발에 필수적인 질의 편집기 기능을 제공한다.

CUBRID 매니저는 데이터베이스 서버, 브로커가 설치된 호스트에서 동작하는 매니저 서버와 GUI 기반의 클라이언트로 구성되어 있다. CUBRID 매니저 클라이언트 프로그램은 Java 기반으로 작성되어 Java가 실행될 수 있는 환경이면 사용 가능하다.

이 장에서는 설명하는 주요 내용은 다음과 같다.

- CUBRID 매니저 소개
- CUBRID 매니저 실행
- CUBRID 매니저 클라이언트 기능

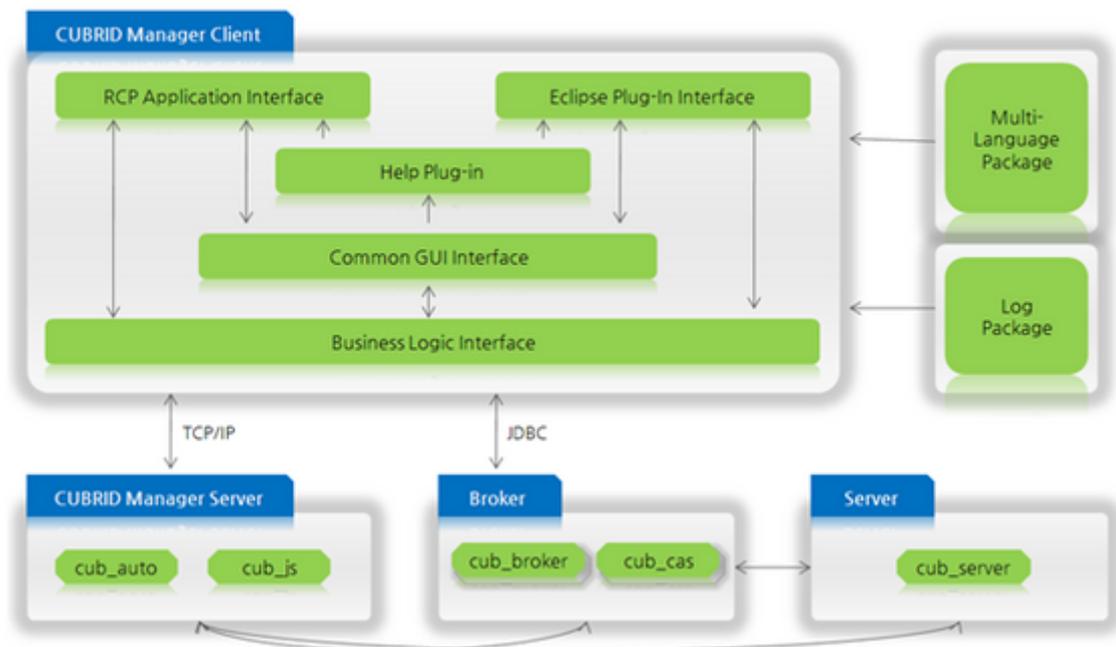
# CUBRID 매니저 소개

## CUBRID 매니저 구조

CUBRID 매니저의 구조는 아래 그림과 같다. 데이터베이스 서버가 설치된 호스트에 CUBRID 매니저 서버가 동작하고, 이 CUBRID 매니저 서버에 CUBRID 매니저 클라이언트가 접속한다.

CUBRID 매니저 서버는 **cub\_auto** 프로세스와 **cub\_js** 프로세스로 구성된다. CUBRID 매니저 클라이언트에서 CUBRID 매니저 서버에 접속하기 위해서는 반드시 CUBRID 매니저 서버가 구동되어 있어야 하며, **cub\_auto** 프로세스와 **cub\_js** 프로세스와 상응하는 네트워크(TCP/IP) 포트가 설정되어 있어야 한다. 질의를 수행하기 위해서는 JDBC 연결을 위한 브로커도 구동되어 있어야 한다.

- **cub\_auto** 프로세스는 CUBRID 매니저 클라이언트 사용자의 인증처리 및 주기적인 자동화 작업과 진단 정보를 수집하는 기능을 수행한다.
- **cub\_js** 프로세스는 CUBRID 매니저 클라이언트로부터 전송된 사용자의 요구를 수행한다.

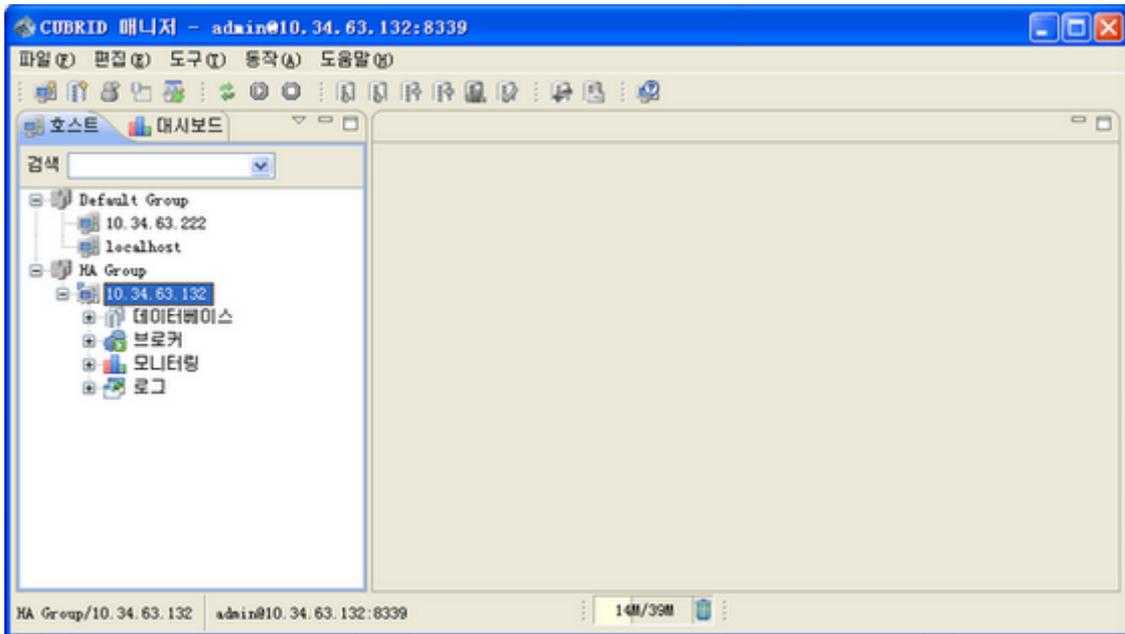


서버 구동에 대한 상세한 정보는 [CUBRID 매니저 서버 환경 설정](#)을 참고한다.

## CUBRID 매니저 클라이언트

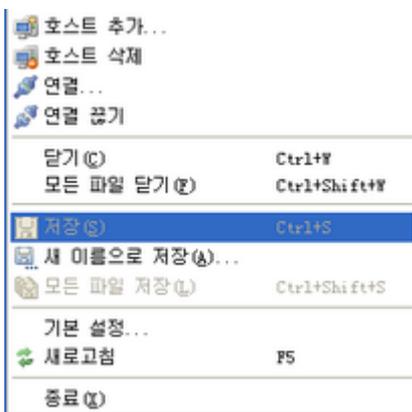
### 화면 구성

클라이언트는 메뉴, 탐색기, 보기 창 그리고 상태 표시줄 4개 영역으로 이루어지며, 각 영역별로 고유의 정보를 제공한다.



### 파일 메뉴

[파일] 메뉴는 호스트 관리 및 연결 메뉴, 파일 저장 메뉴, CUBRID 매니저 기본 설정 메뉴 등으로 구성되어 있다.



### 편집 메뉴

[편집] 메뉴는 CUBRID 매니저 내부에서 사용하는 편집 관련 메뉴로 구성되어 있다.



### 도구 메뉴

[도구] 메뉴는 CUBRID 매니저의 주 메뉴로 서비스 시작/정지를 통제하는 메뉴와 사용자 관리 메뉴로 구성되어 있다.



### 동작 메뉴

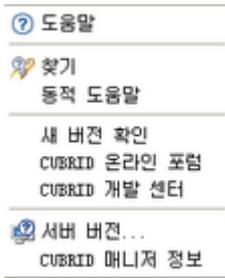
[동작] 메뉴는 동적으로 구성되는 메뉴로 탐색 창에서 마우스 오른쪽 버튼을 클릭해서 수행할 수 있는 기능을 동적으로 보여 주는 메뉴이다. 아래 그림은 테이블을 선택했을 때의 [동작] 메뉴이다.



### 도움말 메뉴

[도움말] 메뉴는 CUBRID 데이터베이스 매뉴얼을 볼 수 있는 [도움말]과 이를 동적으로 구성하고 찾을 수 있는 메뉴로 구성되어 있다.

[CUBRID 온라인 포럼] 및 [CUBRID 개발 센터]를 선택하면 CUBRID 사용 관련 팁 및 개발 관련 소식을 접할 수 있다.



### 툴바

툴바는 전체 기능 중 자주 사용되고 중요한 기능으로 구성되며, 해당 기능이 수행 가능할 때 활성화되고 수행 불가능할 때에는 비활성화된다.

툴바의 아이콘에 마우스 포인터를 대면 아래 그림과 같이 툴팁 도움말이 나타난다.



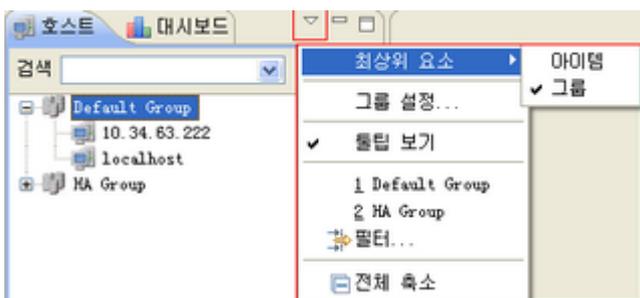
툴바에서 제공하는 기능은 다음과 같다.



호스트 추가	데이터베이스 언로드
데이터베이스 생성	데이터베이스 로드
사용자 추가	데이터베이스 백업
테이블 추가	데이터베이스 복구
새 질의 편집기	데이터베이스 최적화
새로 고침	데이터베이스 검사
시작	데이터베이스 트랜잭션 정보
정지	데이터베이스 잠금 정보
	서버 버전

### 탐색기 메뉴

탐색기 메뉴는 호스트 탐색기와 대시보드 탐색기를 위한 기본적인 기능을 제공한다.



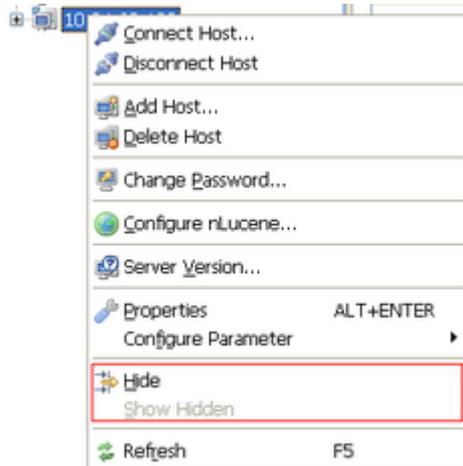
- **최상위 요소:** 호스트 탐색기에서 최상위 요소로 표시할 요소를 선택한다. 호스트 그룹 관리에 사용한다.
- **그룹 설정:** 그룹 관리를 설정한다. 자세한 내용은 [호스트 그룹 관리](#)를 참고한다.
- **툴팁 보기:** 탐색기에서 호스트, 데이터베이스, 브로커에 마우스 포인터를 올렸을 때 툴팁 정보를 표시할지 여부를 설정한다.



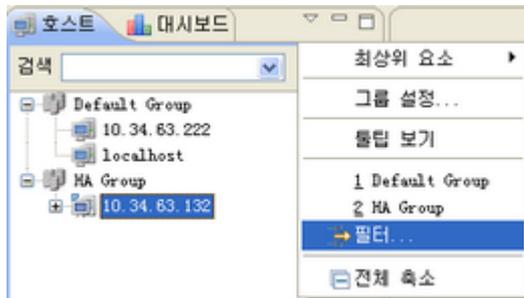
- **필터:** 보고 싶지 않은 노드를 탐색기에서 숨기도록 설정할 수 있다.
- **전체 축소:** 탐색기의 하위 요소들을 숨긴다.

탐색기 필터는 다음과 같은 방법으로 설정할 수 있다.

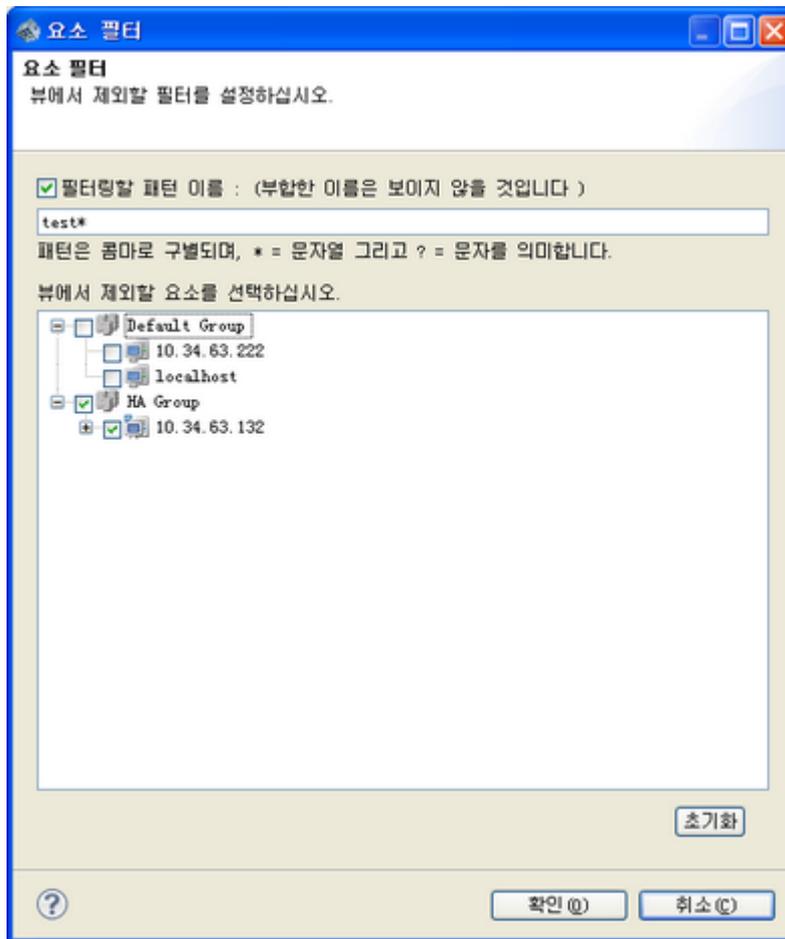
- 탐색기 메뉴에서 [Default Group] 등의 최상위 요소를 선택하여 숨길 수 있다. 선택하면 숨기고 선택하지 않으면 표시한다.
- 탐색기의 노드를 마우스 오른쪽 버튼으로 클릭하여 [Hide]를 선택하면 해당 노드와 그 하위 노드를 숨기고, [Show Hidden]를 선택하면 해당 노드의 모든 하위 노드가 표시된다. 예를 들어 호스트의 바로가기 메뉴는 다음과 같다.



- 탐색기 메뉴에서 [필터]를 선택하면 다음과 같은 대화 상자가 나타나 탐색기에서 숨길 요소를 선택할 수 있다.



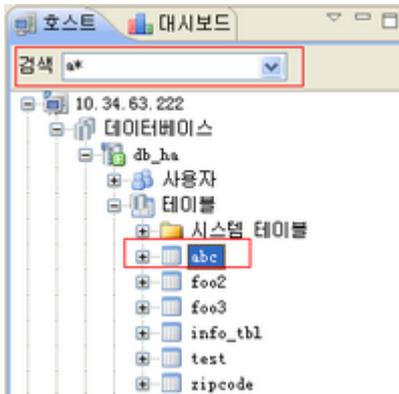
[필터링할 패턴 이름]을 선택하면 해당 패턴과 일치하는 노드를 숨긴다. 트리 메뉴에서 숨길 노드를 선택할 수도 있다.



### 탐색기 툴바

탐색기 툴바는 아이템을 찾아서 표시하는 기능을 제공하는 검색 콤보박스를 제공한다. <Enter> 키를 누르면 검색을 시작하고 <F3> 키를 누르면 다음 아이템을 찾는다.

검색어에는 임의의 문자열을 나타내는 \*와 임의의 문자 한 개를 나타내는 ?를 사용할 수 있다.



# CUBRID 매니저 실행

---

## CUBRID 매니저 서버 환경 설정

CUBRID 매니저 서버의 환경 설정 파일은 **\$CUBRID/conf**에 위치하며, 파일 이름은 **cm.conf**이다.

CUBRID 매니저의 환경 설정 파일에서 주석은 "#"으로 처리되며, 매개 변수 이름과 값이 저장된다. 매개 변수 이름과 값 사이에는 공백 또는 등호 부호(=)로 구분한다. **cm.conf**에서 설정할 수 있는 매개 변수는 다음과 같다.

### cm\_port

CUBRID 매니저 서버와 클라이언트 사이의 통신 포트를 설정하는 매개 변수로, 기본값은 **8001**로 설정된다. **cm\_port**는 **cub\_auto**가 사용하는 포트이며, **cm\_js**는 자동으로 **cm\_port**로 설정한 값보다 1만큼 큰 값을 사용한다. 예를 들어, **cm\_port**가 8001로 설정된 경우 **cub\_auto**는 8001 포트를 사용하고, **cm\_js**는 8002 포트를 사용한다. 따라서 방화벽이 설정된 환경에서 CUBRID 매니저를 구동하려면 반드시 실제로 사용되는 두 개의 포트를 열어야 한다.

### monitor\_interval

**cub\_auto**의 모니터링 주기를 초 단위로 설정하는 매개 변수로, 기본값은 **5**이다.

### allow\_user\_multi\_connection

CUBRID 매니저 서버에 클라이언트가 중복 접속하는 것을 허용하기 위한 매개 변수로, 기본값은 **YES**이다. 따라서 CUBRID 매니저 서버에는 두 개 이상의 CUBRID 매니저 클라이언트가 접속할 수 있으며, 같은 사용자 이름으로 접속할 수도 있다.

### server\_long\_query\_time

서버의 진단 항목 중 **slow\_query** 항목을 설정할 경우 몇 초 이상을 늦은 질의로 판별할지 결정하는 매개 변수로, 기본 값은 **10**이다. 서버에서 수행된 질의 수행 시간이 매개 변수 설정 값보다 큰 경우, **slow\_query**의 개수가 증가한다.

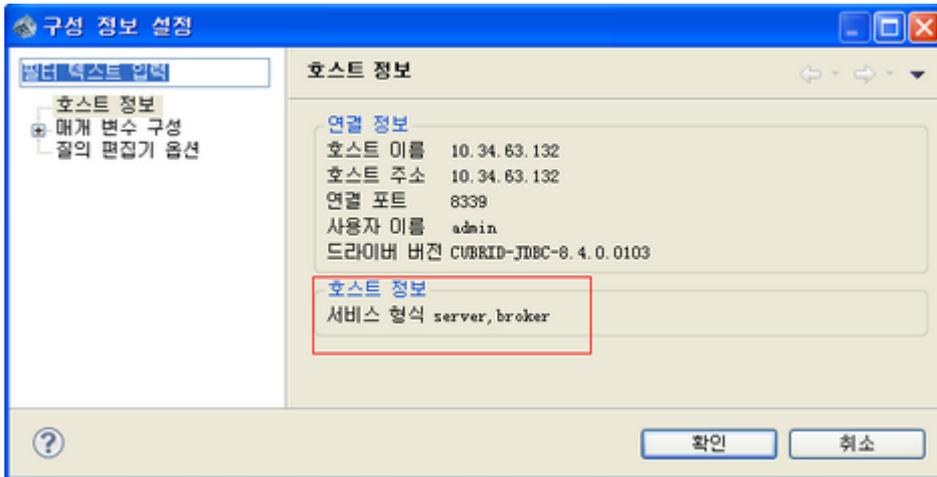
### cm\_target

브로커와 데이터베이스 서버가 분리된 구조에서 매니저의 메뉴를 해당 서비스에 맞게 출력할 목적으로 지원되는 매개 변수이다. 기본값은 브로커와 데이터베이스 서버가 같이 설치되어 있는 환경을 의미하며, 다음과 같이 설정할 수 있다.

- **cm\_target broker, server** : 브로커와 데이터베이스 서버가 같이 있을 경우
- **cm\_target broker** : 브로커만 있을 경우
- **cm\_target server** : 데이터베이스 서버만 있을 경우

브로커만 설정하면 매니저에서 브로커 관련 메뉴만 출력되고, 데이터베이스 서버만 설정하면 서버 관련 메뉴만 출력된다.

탐색 트리에서 호스트를 마우스 오른쪽 버튼 클릭하고 [속성]을 선택하면, 설정된 정보를 [호스트 정보]에서 확인할 수 있다.

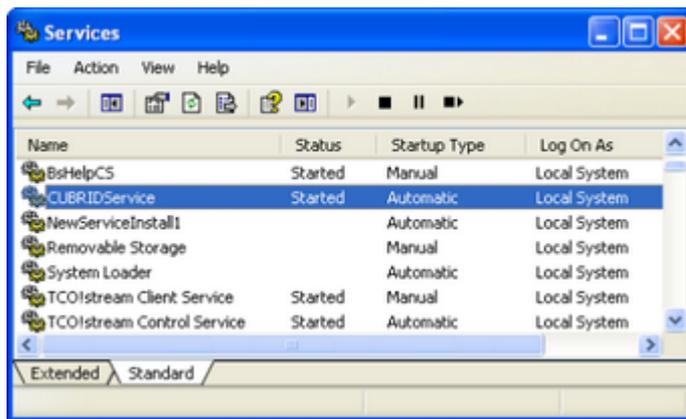


## CUBRID 매니저 서버 실행

CUBRID 매니저는 서버와 클라이언트로 구성되며, CUBRID 매니저 서버가 실행 중일 때에만 CUBRID 매니저 클라이언트를 실행할 수 있다.

### Windows 환경에서 실행

- [제어판] > [성능 및 유지 관리] > [관리도구] > [서비스]에서 CUBRIDService를 통해 실행할 수 있다.



- CUBRID 서비스 트레이(CUBRID Service Tray)를 클릭하고 [CUBRID Server] > [Start]를 선택한다.



**참고** CUBRID가 실행 중일 때, CUBRID 서비스 트레이에서 [Exit]를 선택하면 해당 서버에서 실행 중인 모든 서비스와 프로세스가 중지되므로 주의한다.

### Linux 환경에서 실행

셸에서 다음의 명령 중 하나를 입력하면 CUBRID 매니저 서버가 실행된다.

- %cubrid service start
- %cubrid manager start

### CUBRID 매니저 서버 로그

CUBRID 매니저 서버를 통해 이루어지는 작업이나 오류에 관한 로그는 **\$CUBRID/log/manager**에 기록되며, 실행 중인 서버의 PID는 **\$CUBRID/var/manager**에 기록된다. 기록되는 파일은 다음과 같다.

- **cub\_js.access.log** : cub\_js가 처리하는 작업을 기록한다.
- **cub\_js.error.log** : cub\_js의 작업 처리 중 발생한 에러를 기록한다.
- **cub\_auto.access.log** : 서버에 로그인한 클라이언트에 대한 정보를 기록한다.
- **cub\_auto.error.log** : 서버 로그인에 실패한 클라이언트에 대한 정보를 기록한다.
- **connlist** : 현재 CUBRID 매니저 서버에 접속 중인 클라이언트들의 리스트를 기록한다.

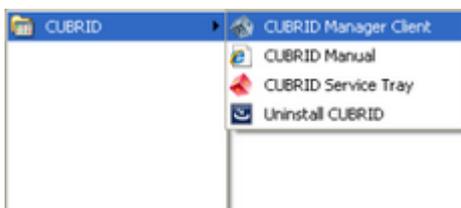
## CUBRID 매니저 클라이언트 실행

CUBRID 매니저는 서버와 클라이언트로 구성되며, CUBRID 매니저 서버가 실행 중일 때에만 CUBRID 매니저 클라이언트를 실행할 수 있다.

CUBRID 매니저 클라이언트는 Java 응용 프로그램으로 JRE 혹은 JDK 1.6 이상 버전에서만 실행되며, 낮은 버전에서 실행할 경우 "지원하지 않는 JRE 버전입니다. CUBRID 매니저를 사용하기 위해서는 JRE 1.6 이상 버전을 사용해야 합니다."라는 오류 메시지가 출력된다.

### Windows 환경에서 실행

- [시작] > [모든 프로그램] > [CUBRID] > [CUBRID Manager Client]를 선택한다.



- CUBRID 서비스 트레이를 클릭하고 [Tools] > [CUBRID Manager]를 선택한다.



### Linux 환경에서 실행

셸에서 다음 중 하나의 명령을 입력하여 CUBRID 매니저 클라이언트를 실행할 수 있다.

- `%cd $CUBRID/cubridmanager`
- `%./cubridmanager`

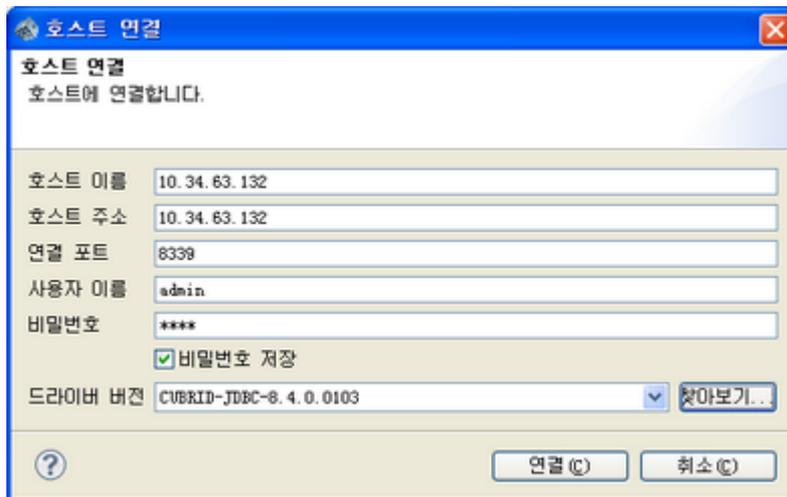
### CUBRID 매니저 클라이언트 로그

CUBRID 매니저 클라이언트를 수행 중에 발생한 오류 로그는 **\$CUBRID/cubridmanager/logs**에 **cubridmanager.log**라는 파일로 생성된다. 사용자는 CUBRID 매니저 개선 및 오류 분석 시 해당 오류 로그와 현상을 CUBRID 매니저 개발 사이트 이슈로 등록하여 CUBRID 매니저 개선 활동에 참여할 수 있다.

## 호스트 관리

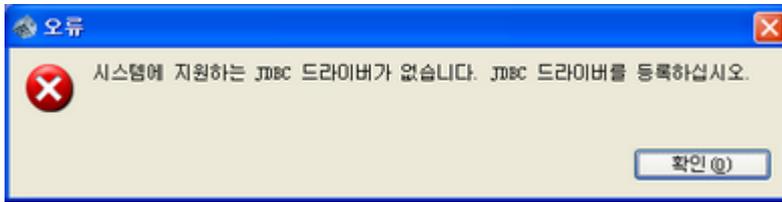
### 기본 호스트 정보

CUBRID 매니저를 설치 후 처음 실행하게 되면 호스트 탐색 트리에 "localhost"라는 호스트가 기본으로 설정되어 있다. 현재 CUBRID 매니저 클라이언트가 실행되는 PC에 구동 중인 CUBRID 매니저 서버가 있다는 가정 아래 설정된 것이며, CUBRID 매니저 접속에 대한 기본 정보를 알려주기 위해 제공된다.

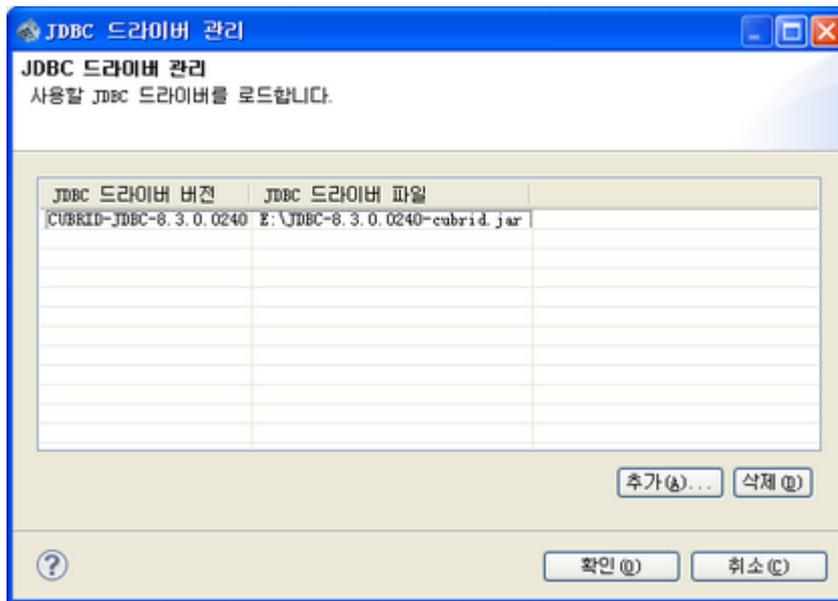


- **호스트 이름:** 호스트 이름은 CUBRID 매니저 내에서 관리하고자 하는 호스트를 구분하는 구분자로, 유일한 이름을 부여한다. 내부적으로 관리하는 호스트 구분자는 [호스트 주소 + 연결 포트]이다. 호스트 이름으로는 숫자와 알파벳만 입력할 수 있으며, 공백을 포함할 수 없다. 호스트 이름의 길이는 최소 4자리 최대 32자리이다.
- **호스트 주소:** 호스트 주소는 공백이거나 공백을 포함할 수 없다.
- **연결 포트:** 연결 포트는 1024와 65535 사이의 정수만 입력할 수 있다. 기본 포트 값은 **8001**이며, 접속이 안 될 경우 **\$CUBRID/conf/cm.conf** 파일의 **cm\_port** 값을 확인한다.
- **사용자 이름:** CUBRID 매니저의 관리자 권한을 가진 사용자는 **admin**이며, 설치 후 처음 사용할 때에는 반드시 **admin** 계정으로 접속해야 한다. 사용자 이름으로는 숫자와 알파벳만 입력할 수 있으며, 공백을 포함할 수 없다. 사용자 이름의 길이는 최소 4자리 최대 32자리이다.
- **비밀번호:** CUBRID 매니저의 관리자 권한을 가진 **admin** 사용자의 최초 비밀번호는 "admin"이다. **admin** 사용자가 최초 연결할 때 비밀번호를 재설정하지 않으면 접속할 수 없다. 이때 재설정된 **admin** 사용자의 비밀번호는 분실하지 않도록 주의한다. 비밀번호는 "admin"으로 재설정할 수 없다. 비밀번호는 공백을 포함할 수 없으며, 비밀번호는 길이는 최소 4자리 최대 32자리이다.
- **비밀번호 저장:** 비밀번호를 로컬에 저장한다. 선택하면 더블클릭해서 자동으로 이 호스트에 로그인할 수 있다. 선택하지 않으면 비밀번호를 입력하는 대화상자가 나타난다.
- **드라이버 버전:** CUBRID 매니저에서 사용할 CUBRID JDBC 드라이버를 선택할 수 있다. 기본값은 "Auto Detect"로, 먼저 서버의 버전을 확인한 후 CUBRID 매니저에 등록된 JDBC 드라이버 중 동일한 버전이

있는가를 확인하여 자동으로 연결한다. 연결할 수 있는 JDBC 드라이버가 없으면 다음과 같이 JDBC를 선택하여 등록하라는 확인 메시지가 출력된다.



- **찾아보기:** 클릭하면 다음과 같이 [JDBC 드라이버 관리] 대화상자가 나타난다.

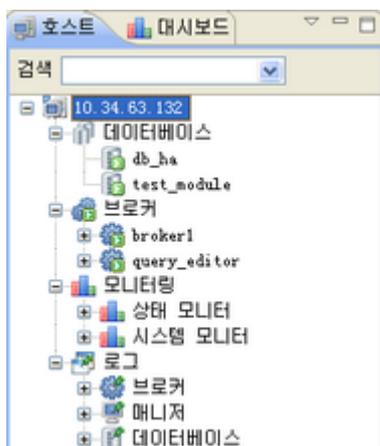


## 호스트 연결 및 연결 해제

### 호스트 연결

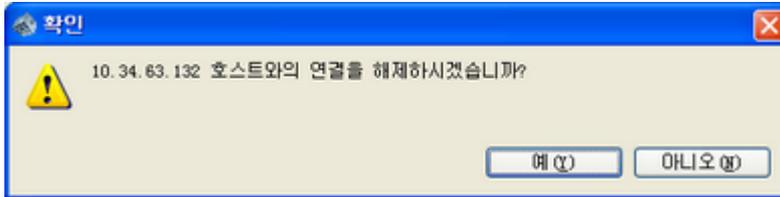
호스트 탐색 트리에 등록된 호스트를 더블 클릭하면 해당 호스트로 연결된다. 또는 해당 호스트를 마우스 오른쪽 버튼 클릭하여 [연결]을 선택하고, 사용자 이름을 확인한 후 [연결]을 클릭하면 호스트에 연결된다.

연결이 성공하면 호스트 아이콘이 에서 로 변경된다. 연결 후 해당 호스트의 초기 화면은 다음과 같다.



## 호스트 연결 해제

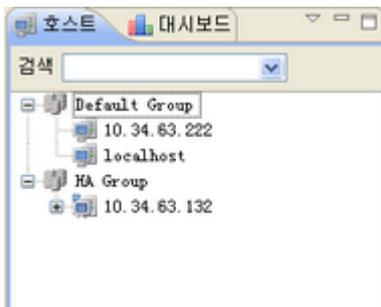
호스트 탐색 트리에서 호스트를 마우스 오른쪽 버튼 클릭하여 [연결 끊기]를 선택한다.



## 호스트 그룹 관리

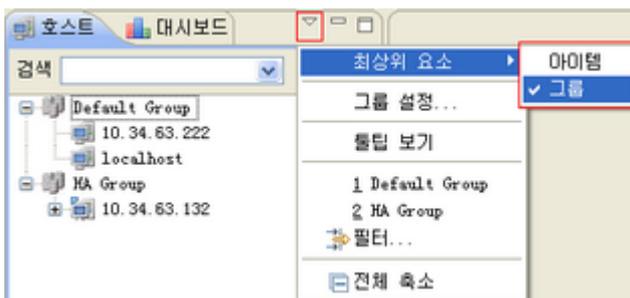
### 개요

관리할 호스트가 많을 때에는 호스트 그룹으로 묶어서 아래 그림과 같이 보기 쉽게 관리할 수 있다.



### 호스트 그룹 보기/숨기기

사용자는 다음 메뉴를 사용하여 호스트 그룹을 보거나 숨길 수 있다.

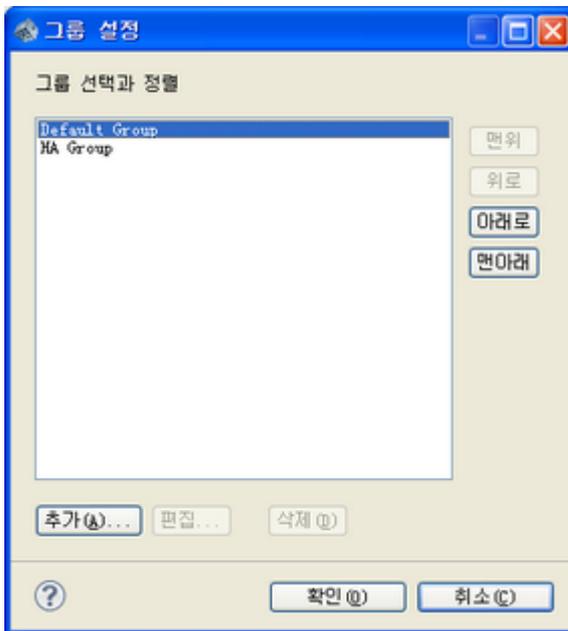


- **아이템:** 호스트만 표시하고 그룹 노드는 숨긴다.
- **그룹:** 그룹 노드를 표시한다.

### 호스트 그룹 설정



메뉴에서 [그룹 설정]을 선택하면 다음과 같은 [그룹 설정] 대화상자가 나타난다.



[맨위], [위로], [아래로], [맨아래] 버튼을 클릭하여 그룹의 순서를 변경할 수 있다. [추가], [편집] 버튼을 클릭하면 그룹을 추가하거나 편집할 수 있는 대화상자가 나타난다.

### 호스트 그룹 추가/편집

[그룹 설정] 대화상자에서 [추가]를 클릭하면 새로운 그룹의 이름과 그 그룹에 속할 호스트를 입력할 수 있다.



[그룹 설정] 대화상자에서 [편집]을 클릭하면 호스트 그룹을 편집할 수 있다.



대화 상자의 왼쪽에 해당 호스트 그룹에 속하지 않은 호스트 목록이 보이고, 오른쪽에는 해당 호스트 그룹에 속한 호스트 목록이 보인다. [>] 버튼을 클릭하면 왼쪽 목록에서 선택한 호스트가 오른쪽으로 이동하고, [>>] 버튼을 클릭하면 왼쪽 목록의 모든 호스트가 오른쪽으로 이동한다. [<] 버튼을 클릭하면 오른쪽 목록에서 선택한 호스트가 왼쪽으로 이동하고, [<<] 버튼을 클릭하면 오른쪽 목록의 모든 호스트가 왼쪽으로 이동한다.

### 호스트 그룹 삭제

[그룹 설정] 대화상자에서 삭제할 호스트 그룹을 선택하고 [삭제] 버튼을 클릭한다. [Default Group]은 삭제하거나 편집할 수 없으며, 아무 그룹에도 속하지 않는 호스트를 포함한다.

### 끌어다 놓기

여러 개의 호스트를 끌어다 다른 호스트 그룹으로 옮기거나 현재 그룹 내에서 호스트 노드의 표시 순서를 변경할 수 있다. 그룹 노드를 숨겼을 때에도 호스트 노드의 표시 순서를 변경할 수 있다.

## 호스트 추가 및 삭제

### 호스트 추가

호스트를 추가하여, 여러 호스트에 존재하는 데이터베이스를 하나의 매니저에서 관리할 수 있다. 호스트를 추가하려면 다음 중 하나를 실행한다.

- 툴바에서 [호스트 추가] 버튼을 클릭한다.
- 메뉴에서 [파일] > [호스트 추가]를 선택한다.
- 호스트 탐색 트리에서 마우스 오른쪽 버튼을 클릭하고 [호스트 추가]를 선택한다.

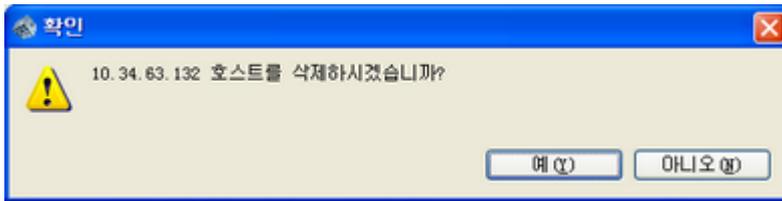


호스트 이름, 호스트 주소, 연결 포트, 사용자 이름 그리고 비밀번호에 대한 자세한 내용은 [기본 호스트 정보](#)를 참조한다.

정보를 입력하고 [추가] 버튼을 클릭하면 호스트 탐색 트리에 추가만 하고 연결은 하지 않는다. 반면, [연결]을 클릭하면 탐색 트리에 추가하고 해당 호스트로 연결한다.

### 호스트 삭제

호스트 탐색 트리에 등록된 호스트 정보를 삭제한다. 호스트 삭제 시 하부 노드의 모든 정보가 같이 삭제된다. 단, CUBRID 매니저의 클라이언트 구성 정보만 삭제되는 것으로, CUBRID 매니저 서버에 구성된 정보까지 삭제되는 것은 아니다.



## 서비스 관리

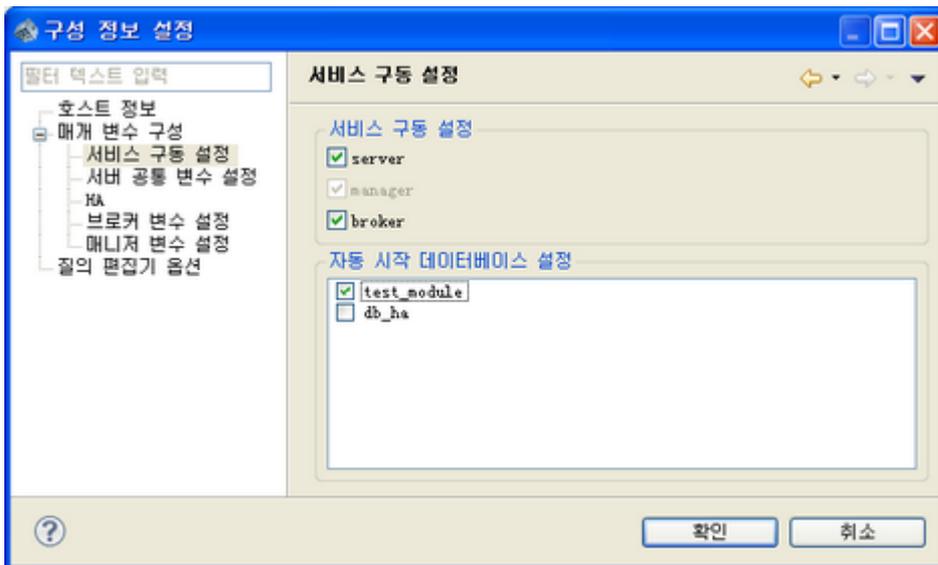
CUBRID 매니저로 CUBRID 관련 서비스를 시작/정지할 수 있다. 이것은 CUBRID 데이터베이스 명령어인 cubrid 명령과 동일한 기능이다. 단, CUBRID 매니저 서버는 CUBRID 매니저 클라이언트를 통해서 구동/정지할 수 없다.



### 서비스 시작

메뉴에서 [도구] > [서비스 시작]을 선택하면, [서비스 구동 설정]에서 선택한 서비스와 데이터베이스를 시작한다. 시작하는 서비스와 데이터베이스는, 호스트를 마우스 오른쪽 버튼 클릭하고 [속성]을 선택하여 [서비스 구동 설정]에서 선택한다.

[자동 시작 데이터베이스 설정]은 [서비스 구동 설정]에서 [server]를 선택했을 경우에만 활성화된다.



## 서비스 정지

메뉴에서 [도구] > [서비스 정지]를 선택하면, [서비스 구동 설정]에서 선택한 서비스와 데이터베이스를 정지한다.

## 데이터베이스 시작/정지

선택한 데이터베이스를 시작 및 정지하려면, 다음 중 한 가지를 실행한다.

- 툴바에서 [시작 ] 또는 [정지 ]를 클릭한다.
- 데이터베이스를 마우스 오른쪽 버튼 클릭하고 [데이터베이스 시작] 또는 [데이터베이스 정지]를 선택한다.
- 메뉴에서 [동작] > [데이터베이스 시작] 또는 [동작] > [데이터베이스 정지]를 선택한다.

## 브로커 시작/정지

선택한 브로커를 시작 및 정지하려면, 다음 중 한 가지를 실행한다.

- 툴바에서 [시작 ] 또는 [정지 ]를 클릭한다.
- 브로커를 마우스 오른쪽 버튼 클릭하고 [브로커 시작] 또는 [브로커 정지]를 선택한다.
- 메뉴에서 [동작] > [브로커 시작] 또는 [동작] > [브로커 정지]를 선택한다.

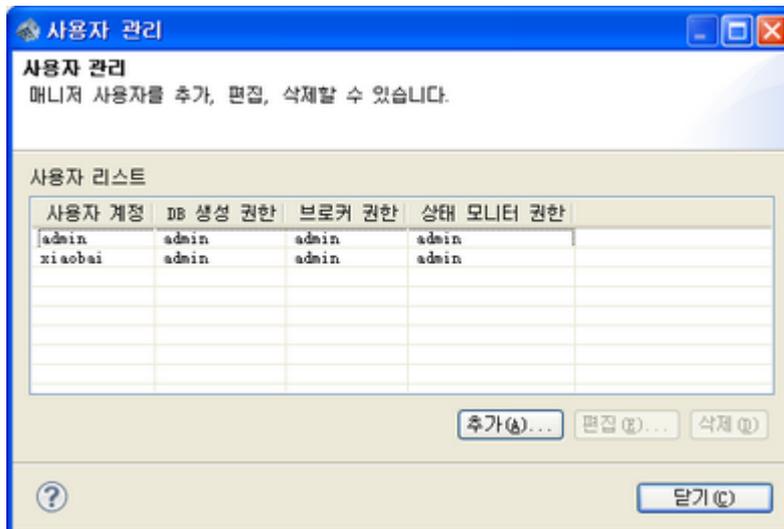
## 사용자 관리

### 사용자 권한 설정

하나의 데이터베이스에 여러 사용자를 등록하고 사용자 권한을 관리하듯이, 매니저 서버에도 여러 사용자를 등록하고 사용자 권한을 관리할 수 있다. 매니저 사용자 정보는 CUBRID 매니저 서버에서 저장되며, 해당 호스트에 접속하기 위해서는 반드시 사전에 CUBRID 매니저 사용 권한을 취득해야 한다.

관리자 계정인 **admin**과 **admin**이 등록된 일반 사용자만 CUBRID 매니저에 접속할 수 있다. CUBRID 매니저 사용자 관리 기능은 **admin**에게만 허용되며, [도구] > [사용자 관리]를 선택해서 실행할 수 있다.

사용자 관리에서는 각 사용자에게 다음의 권한을 설정할 수 있다.



#### DB 생성 권한

- **admin** : 새 데이터베이스를 생성할 수 있는 권한을 의미하며, 오직 관리자 계정인 **admin** 사용자한테만 부여된다.
- **none** : 권한이 없음을 의미한다.

#### 브로커 권한

- **admin** : 브로커를 시작/정지할 수 있고 브로커를 추가, 편집, 삭제할 수 있는 관리 권한을 의미한다.
- **monitor** : 브로커의 상태 보기 기능을 통해 브로커의 진행 상황을 모니터링할 수 있는 권한을 의미한다.
- **none** : 권한이 없음을 의미한다.

#### 상태 모니터 권한

- **admin** : 상태 모니터를 수행할 수 있고 추가, 편집, 삭제할 수 있는 관리 권한을 의미한다.
- **monitor** : 상태 모니터를 수행하여 모니터링할 수 있는 권한을 의미한다.

- **none** : 권한이 없음을 의미한다.

## 사용자 추가, 편집 및 삭제

### 사용자 추가

사용자 추가는 사용자 계정 정보 및 권한 설정과 데이터베이스 접속 권한 설정으로 이루어진다.

#### 사용자 계정 및 권한 설정

- **사용자 이름**: 사용자 이름의 길이는 최소 4자리 최대 32자리이다. 사용자 이름으로는 숫자와 알파벳만 입력할 수 있으며, 공백을 포함할 수 없다. 사용자 이름으로 "admin"을 사용할 수 없으며, 사용자 이름은 해당 호스트 내에서 유일해야 한다.
- **비밀번호**: 비밀번호의 길이는 최소 4바이트 최대 32바이트이며, 공백을 포함할 수 없다. 비밀번호로 "admin"을 사용할 수 없다.

#### 데이터베이스 권한 설정

데이터베이스 이름	연결 권한	DB 계정	브로커 주소	브로커 포트
db_ha	Yes	public	10.34.63.132	query_editor [45000/0K]
test_module	No			broker1 [49000/0K]

- **연결 권한:** 추가 중인 CUBRID 매니저 사용자가 연결할 수 있는 데이터베이스를 선택한다. [Yes]를 선택한 데이터베이스만 호스트 탐색 트리에 출력된다.
- **DB 계정:** 추가 중인 CUBRID 매니저 사용자가 해당 데이터베이스에 접속할 때 사용하는 데이터베이스 계정 정보를 입력한다. "dba" 또는 "public"과 같은 값을 입력할 수 있다.
- **브로커 주소:** 해당 데이터베이스에 접속하기 위해 사용할 브로커 주소를 입력한다. 기본값은 데이터베이스 서버의 주소와 동일하다. 브로커 서버가 분리되어 있다면 이 값을 변경하여 접속 정보를 부여할 수 있다.
- **브로커 포트:** 추가 중인 CUBRID 매니저 사용자가 해당 데이터베이스에 접속할 때 사용할 브로커 포트를 정의한다. 브로커 포트는 현재 정의되어 있는 브로커 정보를 통해서 선택할 수 있다. 브로커 정보는 "브로커 이름[포트/구동상태]"로 구성되어 있다.

## 사용자 편집

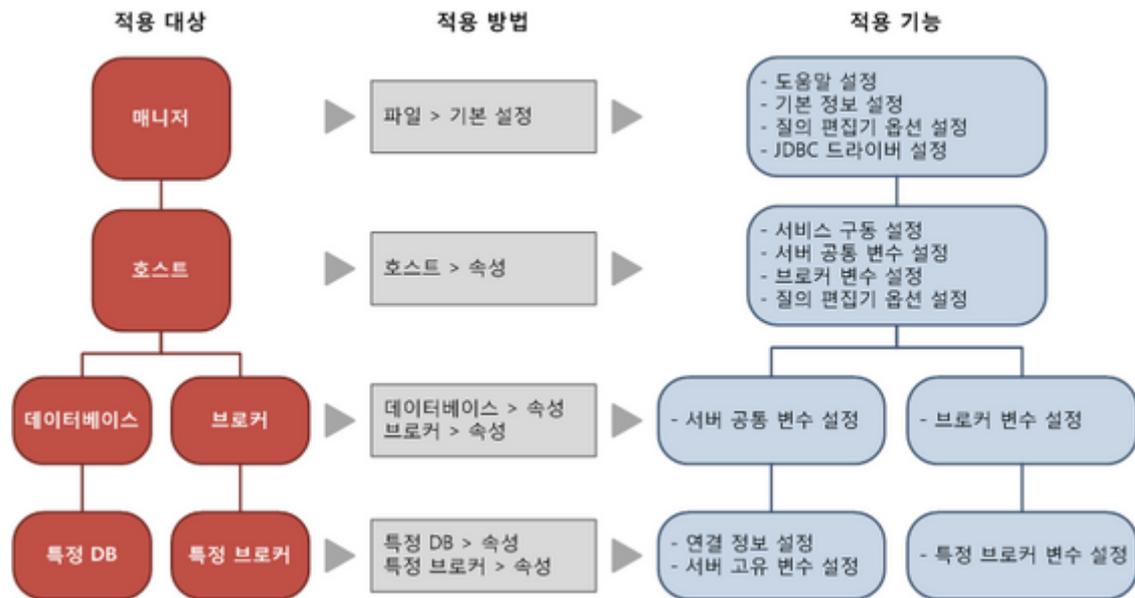
편집할 사용자를 리스트에서 선택한 후, 사용자 추가와 동일한 방법으로 사용자를 편집할 수 있다. 단, **admin** 계정에 대한 편집은 비밀번호만 변경할 수 있으며, 나머지 권한은 **admin** 권한에서 다른 권한으로 변경할 수 없다.

## 사용자 삭제

삭제할 사용자를 리스트에서 선택한 후 [삭제] 버튼을 클릭하면 사용자를 삭제할 수 있다. 단, **admin** 계정은 삭제할 수 없다.

## 속성 관리

CUBRID 매니저를 통해서 서비스, 데이터베이스, 브로커, 매니저 서버 그리고 질의 편집기의 구동 환경을 설정할 수 있다. 환경 설정은 각 단계별로 설정할 수 있으며 다음과 같은 구성을 가진다.



### 도움말 설정

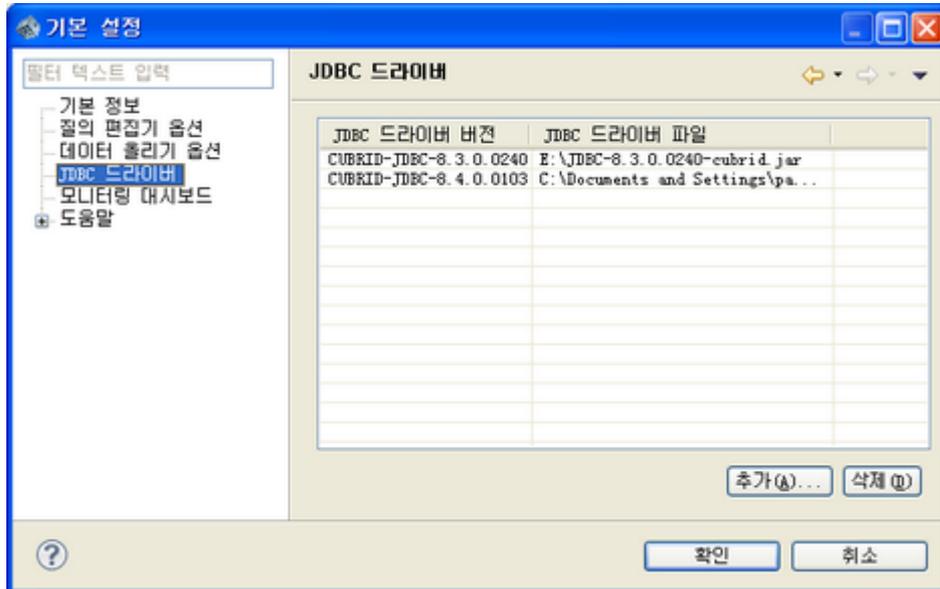
내부 도움말 창을 통해 도움말을 볼지 또는 외부 브라우저로 도움말을 볼지 선택할 수 있으며, Eclipse 플러그인으로 실행했을 때 도움말 열기 모드를 선택할 수 있다.

### 기본 정보 설정

기본 정보 설정에서는 다음과 같은 내용을 설정할 수 있다.

- CUBRID 매니저 창을 최대 크기로 열거나 또는 이전 크기로 열도록 설정
- CUBRID 매니저를 종료할 때 확인 여부 설정
- 시작할 때 CUBRID 소식 보기 여부 설정
- 탐색 트리에서 정보 창을 열 때 마우스 한 번 클릭으로 열지 또는 더블클릭으로 열지 설정(기본값은 더블클릭)

### JDBC 드라이버 설정

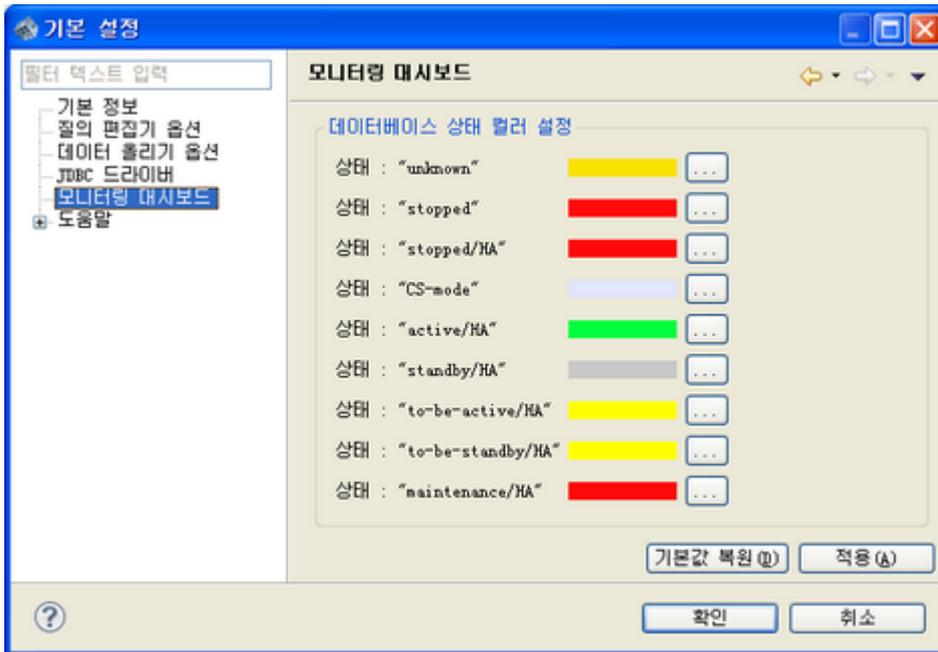


CUBRID 2008 R2.1 버전부터는 다중 JDBC 드라이버를 지원하며, 하위 버전 데이터베이스 서버에 접근하기 위하여 해당 데이터베이스 서버의 JDBC 드라이버를 추가로 설정할 수 있다. CUBRID JDBC 드라이버는 **cubridmanager/plugins**에 포함되어 배포된다. CUBRID 2008 R2.1 버전의 매니저는 일반적으로 CUBRID-JDBC-8.2.1.x 드라이버를 사용하여 해당 버전의 데이터베이스 서버에 접근할 수 있으나, 다중 JDBC 드라이버를 이용하면 CUBRID 매니저 버전보다 낮은 버전의 데이터베이스에도 접근할 수 있다. 이를 위해서는 [기본 설정] > [JDBC 드라이버]을 선택하여 사용할 JDBC 드라이버를 추가로 선택해야 한다.

단, 접속하고자 하는 데이터베이스 서버의 버전에 따라 일부 기능이 제한될 수 있다. 예를 들어, 왼쪽 트리 구조에서 관리할 수 있는 기능은 CUBRID 2008 R2.0 이상의 데이터베이스에 대해서는 정상적으로 수행되지만 이전 버전에 대해서는 제한되며, 질의 편집 및 수행 기능은 CUBRID 2008 R1.4 이상의 데이터베이스에 대해서는 정상 수행되지만 이전 버전에 대해서는 제한된다.

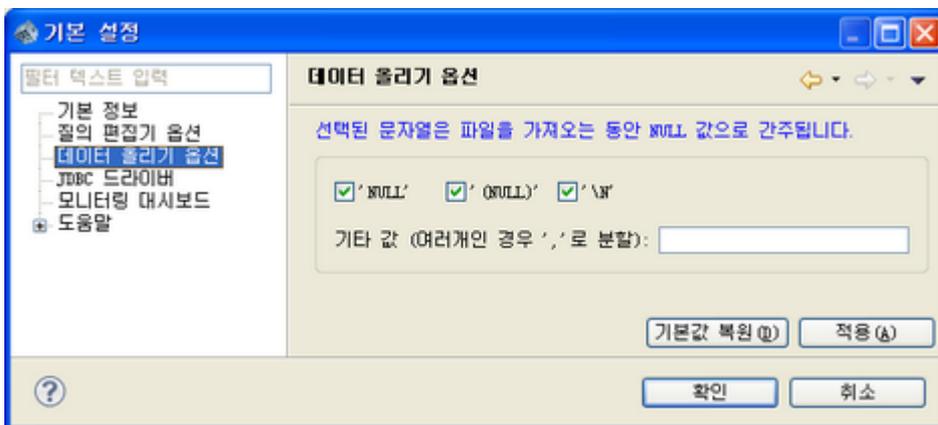
### 모니터링 대시보드

호스트, 데이터베이스 서버 및 브로커의 상태를 모니터링하는 대시보드의 상태별 색상을 사용자가 설정할 수 있다.



### 데이터 올리기 옵션

데이터를 올리거나 업데이트할 때 **NULL**로 처리할 문자열을 설정한다.



### 질의 편집기 옵션

질의 편집기를 통해서 데이터베이스와 접속할 때 적용되는 매개 변수를 설정한다. 자세한 것은 [질의 편집기 옵션](#)을 참조한다.

### 서비스 구동 설정

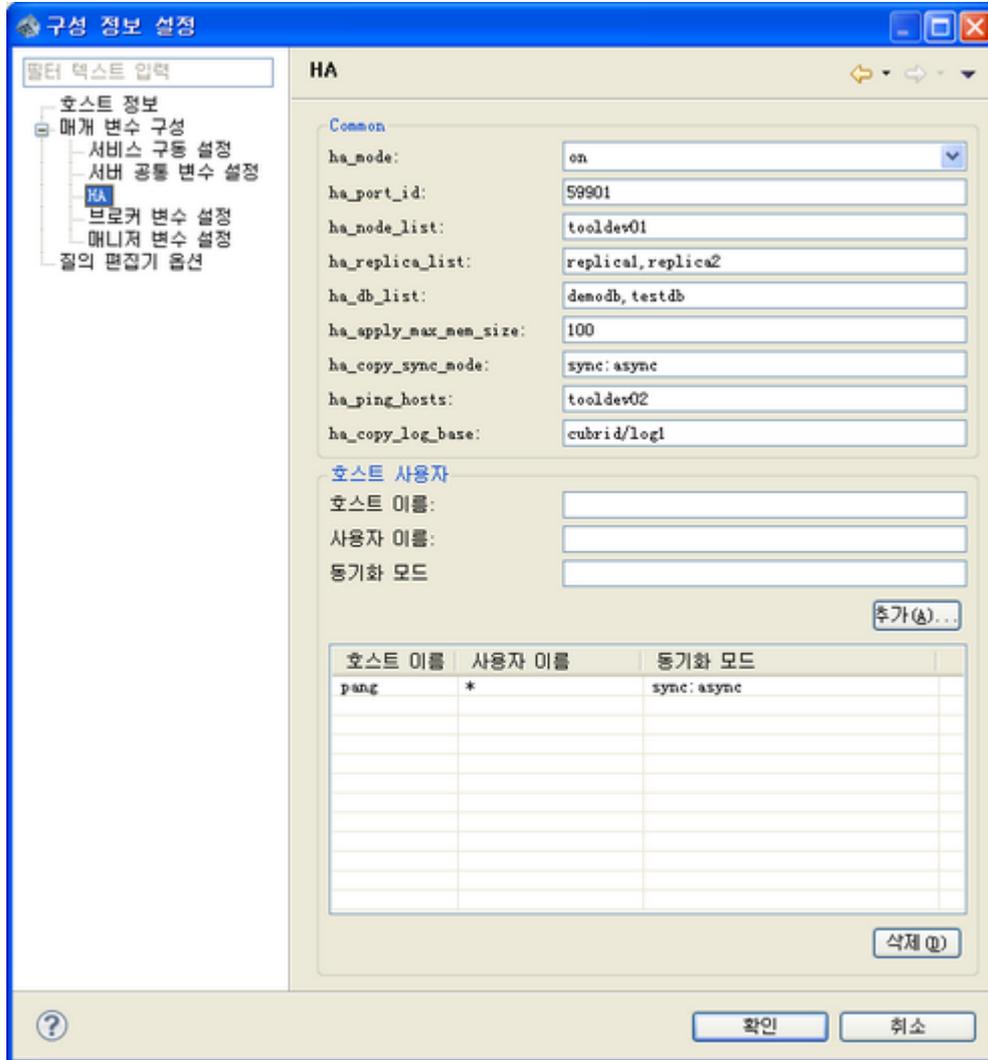
서비스 구동과 관련된 옵션을 설정한다. 데이터베이스 자동 시작을 설정할 수 있다.

### 서버 공통/고유 변수 설정

데이터베이스 공통 변수를 설정한다. 자세한 것은 [데이터베이스 설정](#)을 참조한다.

### HA 설정

데이터베이스 HA 파라미터를 설정한다. HA 설정은 CUBRID R4.0 이상 버전에서 지원한다.



### 특정 브로커 변수 설정

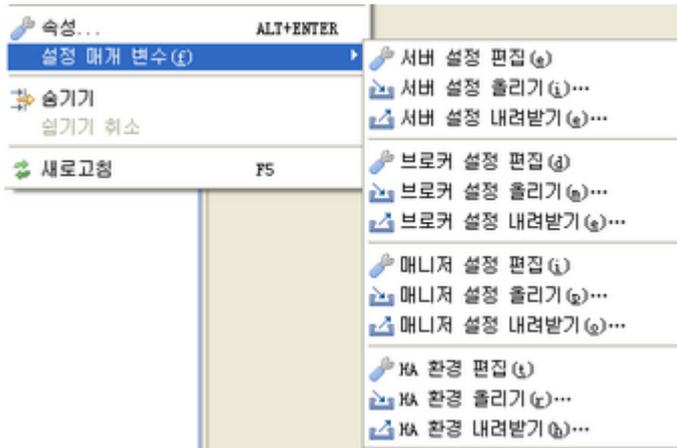
브로커 정보를 설정한다.

### 연결 정보 설정

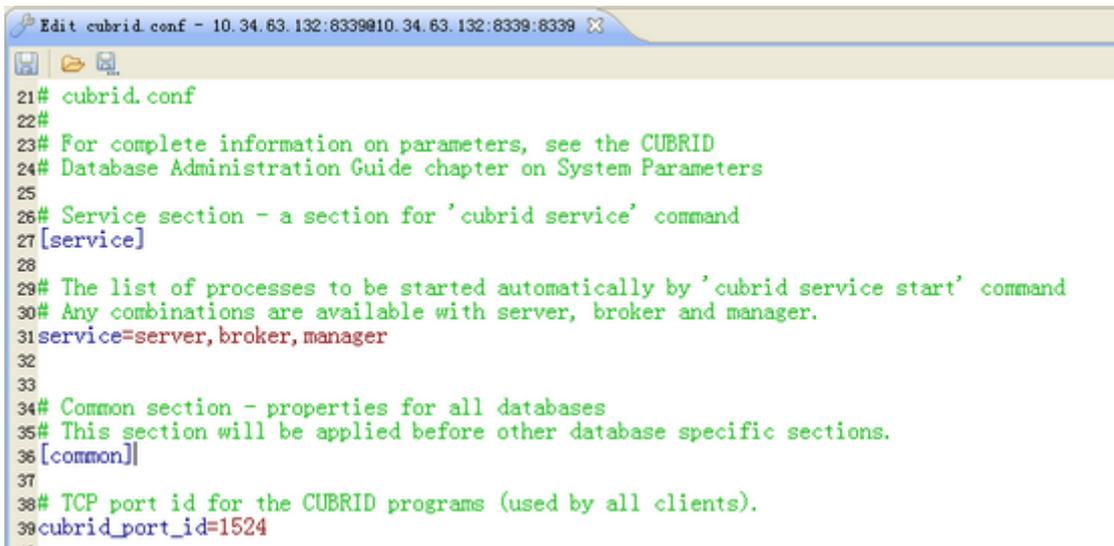
연결 정보 설정을 통해 데이터베이스에 접속하는 포트와 문제 세트를 설정할 수 있다.

### 설정 파일 편집

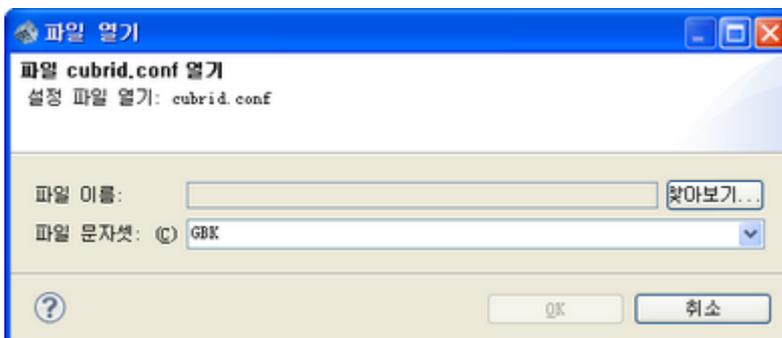
관리자는 호스트 노드를 마우스 오른쪽 버튼으로 클릭하고 [설정 매개 변수]를 선택하여 직접 설정 파일을 편집할 수 있다. 직접 편집할 수 있는 설정 파일은 **cubrid.conf**, **cubrid\_broker.conf**, **cm.conf**, **ha.conf**(서버가 HA 서버이고 CUBRID 버전이 2008 R4.0 이상인 경우)이다.



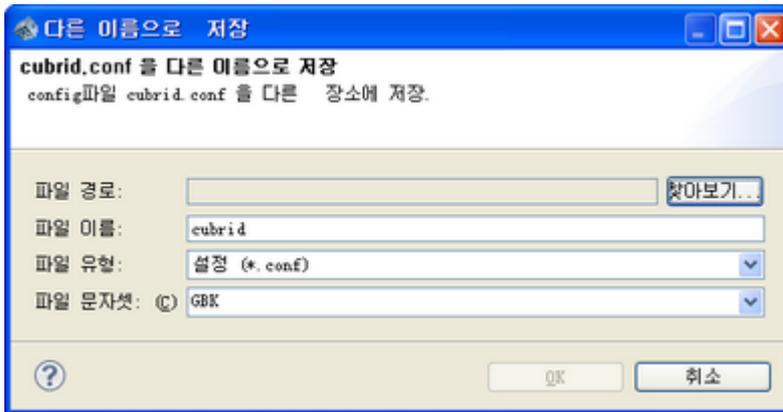
만약 **cubrid.conf** 파일을 직접 편집하려면 [서버 설정 편집]을 선택하여 편집기에서 **cubrid.conf** 파일을 연다.



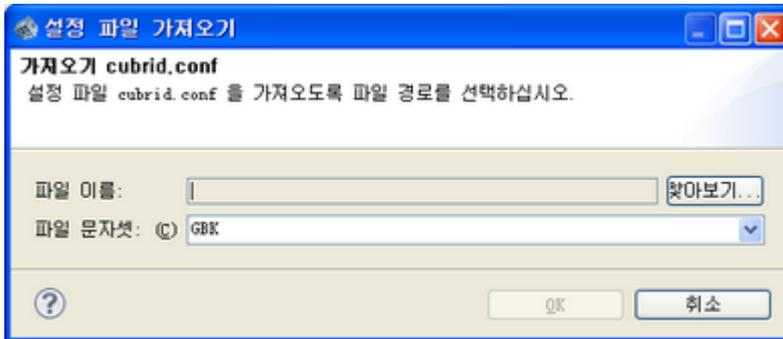
📁 버튼을 클릭하면 다음과 같은 대화 상자가 나타나 로컬 파일을 열 수 있다.



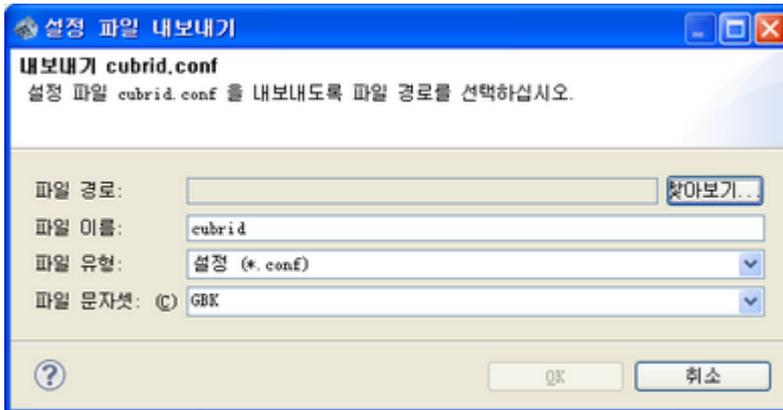
파일을 저장하려면 왼쪽 위의 버튼을 클릭하고, 다른 이름으로 저장하려면 버튼을 클릭한다.



로컬에 저장된 설정 파일을 가져와 올릴 수도 있다. 만약 **cubrid.conf** 파일을 올리려면 [서버 설정 올리기]를 선택한다.



서버의 설정 파일을 내려받으려면 [서버 설정 내려받기]를 선택한다.



## 질의 편집기

### 질의 편집기 구조

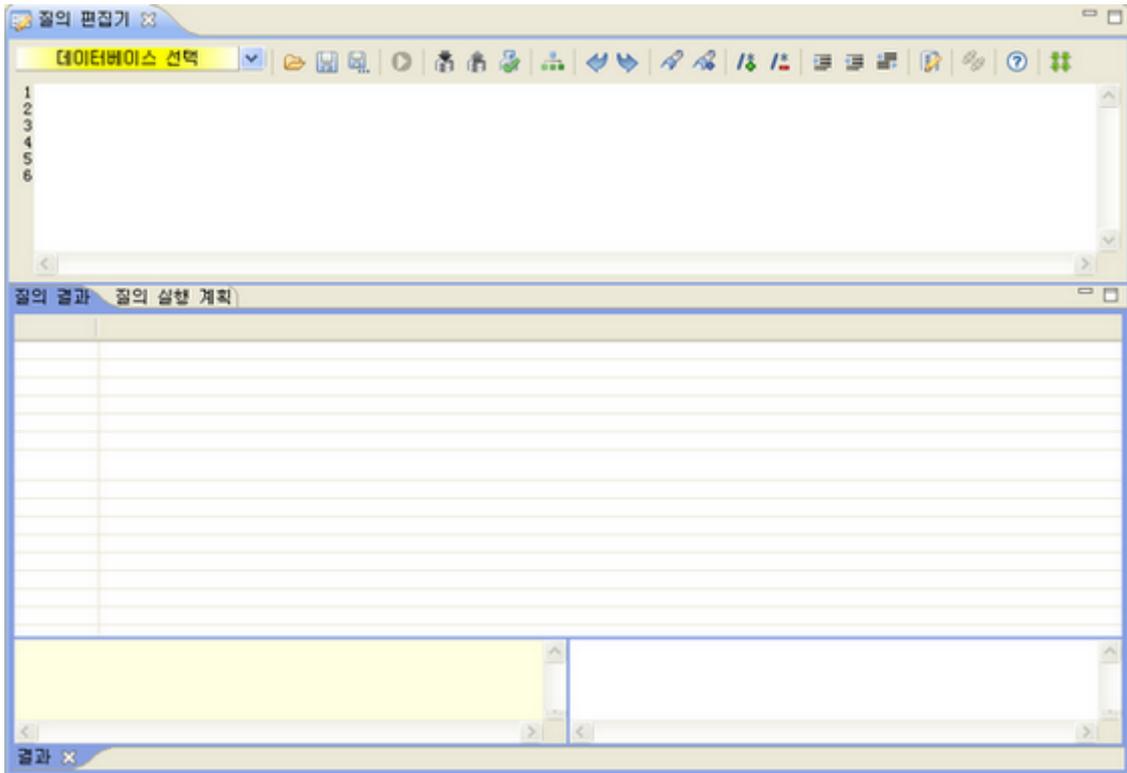
CUBRID 매니저 질의 편집기는 모든 **DML**, **DDL**, **DCL** 문의 수행을 지원하는 질의 도구로서, 보다 쉽고 편리하게 질의를 편집하고 실행할 수 있다. 질의 편집기를 실행하려면 메뉴에서 [도구] > [새 질의 편집기]를 선택하거나, 툴바에서 [새 질의 편집기 ]를 클릭한다. 또는 해당 데이터베이스를 마우스 오른쪽 버튼 클릭하여 [새 질의 편집기]를 선택한다.

데이터베이스를 마우스 오른쪽 버튼 클릭하여 [새 질의 편집기]를 선택하면, 해당 데이터베이스에 로그인한 기본 정보로 질의 편집기를 실행한다.

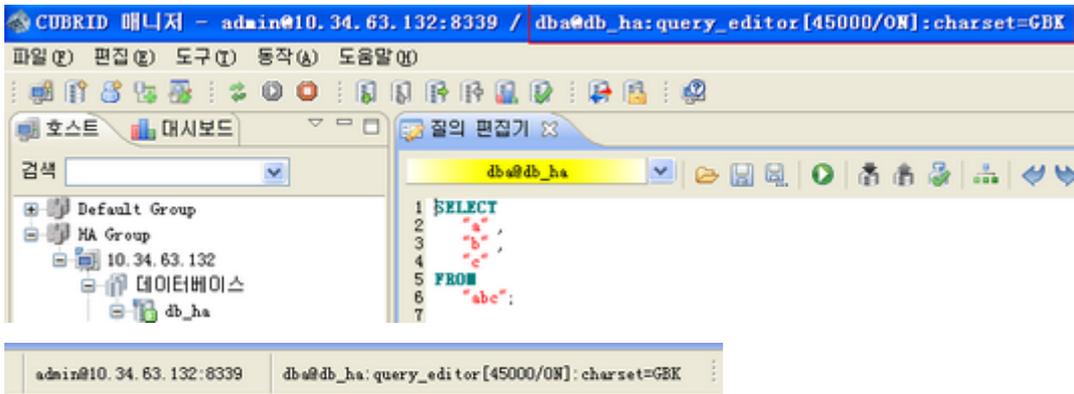
메뉴 또는 툴바에서 [새 질의 편집기]를 선택하면 아래 그림과 같이 사용자가 직접 접속하고자 하는 데이터베이스 로그인 정보, 문자 집합(character set) 및 JDBC 드라이버 버전 정보를 설정하여 로그인할 수 있다. 브로커가 다른 서버에서 구동 중이면 브로커에 접속하기 위한 정보를 별도로 설정할 수 있다.



질의 편집기 화면은 위쪽의 질의 편집 창과 아래쪽의 질의 결과 창으로 구분할 수 있다. 질의 편집 창에서는 수행할 질의를 입력하고 편집하며, 질의 편집기에서 주로 사용하는 기능을 아이콘으로 보여주는 툴바를 포함한다. 질의 결과 창에서는 수행된 질의별 결과를 탭 형식으로 보여주며, 수행한 질의 및 해당 질의의 수행 시간을 확인할 수 있다.



질의 편집기를 여러 개 열고 사용하는 경우, CUBRID 매니저의 제목 표시줄 및 상태 표시줄에서 각 질의 편집기의 접속 정보를 확인할 수 있다. 해당 질의 편집기에서 작성된 질의가 수행되는 대상 데이터베이스, 로그인한 사용자, 브로커 포트, 문자 집합 정보가 표시된다.



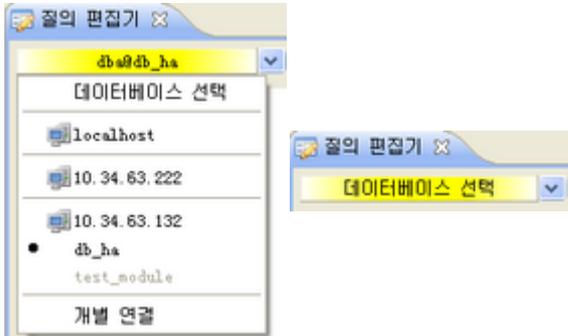
## 툴바 기능

### 데이터베이스 선택

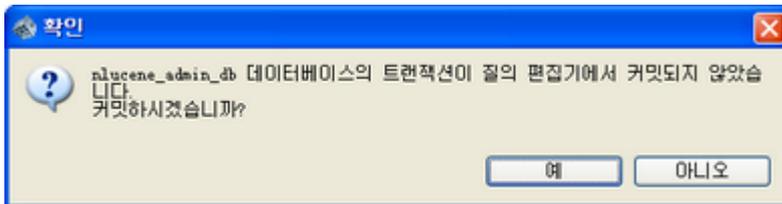
하나의 질의 편집기에서 다중 호스트의 다중 데이터베이스를 접속할 수 있도록 지원한다.

현재 접속 중인 데이터베이스를 표시하며, 현재 CUBRID 매니저에서 접속한 다른 호스트의 데이터베이스도 선택하여 접속할 수 있다. 아래 그림과 같이 DBA로 로그인한 경우와 데이터베이스를 선택하지 않은 경우는

노란색으로 강조하여, 현재 어떤 계정으로 로그인하여 작업 중인지 확인하고, 데이터베이스 연결 정보가 없을 때 선택할 수 있도록 지원한다.



만약 현재 접속 중인 데이터베이스의 트랜잭션을 종료하지 않고 다른 데이터베이스를 선택하면 아래와 같이 수행 중인 트랜잭션을 커밋할 것인지 아니면 롤백할 것인지 묻는다.



### 툴바 기능

질의 편집기 툴바의 기능은 다음과 같으며, 편집 기능은 메뉴의 [편집] 메뉴와 동기화되어 있다. 중요 기능은 단축키를 제공한다.



기능	아이콘	단축키	설명
열기			텍스트 형태의 SQL 문을 열어서 질의 편집창에 보여 준다.
저장			질의 편집창의 내용을 저장한다.
새 이름으로 저장			질의 편집창의 내용을 다른 이름으로 저장한다.
실행		F5	질의 편집창의 모든 질의를 수행한다. 또는 블록으로 선택한 질의만 수행한다. 수행할 수 없는 상태이면 아이콘이 비활성화(🚫)된다.
커밋			자동 커밋 모드에서는 비활성화(🚫) 상태를 유지하고, 자동 커밋 모드가 아닌 경우에 트랜잭션이 발생하면 활성화(👉)되어 사용자가 명시적으로 커밋을 수행할 수 있다.
롤백			자동 커밋 모드에서는 비활성화(🚫) 상태를 유지하고, 자동 커밋 모드가 아닌 경우에 트랜잭션이

			발생하면 활성화(  )되어 사용자가 명시적으로 롤백을 수행할 수 있다.
자동 커밋			질의 편집창에서 수행하는 질의를 자동 커밋한다. 아이콘은 토글 형태로 표시되며, 자동 커밋 설정 상태는  이고 해제 상태는  이다. 단, 해당 설정은 해당 질의 편집기에만 적용되며, 해당 데이터베이스 질의 편집기 옵션의 기본값은 변경되지 않는다.
질의 실행 계획		F6	선택한 질의의 질의 실행 계획을 본다. 자세한 내용은 <a href="#">질의 실행 계획 보기</a> 를 참조한다.
실행 취소		Ctrl+Z	편집 기능으로 편집한 내용을 취소한다.
다시 실행		Ctrl+Y	편집 기능으로 실행 취소한 내용을 다시 실행한다.
찾기/바꾸기		Ctrl+F	질의 편집창 내에서 찾고 바꿀 수 있는 기능을 제공한다.
다음 찾기		F3	한번 찾은 내용에 대해 다음 찾기 기능을 제공한다.
주석 추가		Ctrl+/	질의 편집 창에서 사용자가 선택한 영역이나 커서가 있는 라인의 주석을 삽입한다. 주석은 '--'로 삽입된다.
주석 삭제		Ctrl+/	질의 편집 창에서 사용자가 선택한 영역이나 커서가 있는 라인의 주석을 제거한다.
들여 쓰기		Tab	질의 편집 창에서 사용자가 선택한 영역에 들여 쓰기를 한다.
내어 쓰기		Shift+Tab	질의 편집 창에서 사용자가 선택한 영역에 내어 쓰기를 한다.
SQL 서식화		Ctrl+Shift+F	질의 편집 창에서 선택한 SQL 문에 대해 서식화를 수행한다.
OID 정보 보기			[질의 편집기 옵션]의 [OID 정보 보기]를 툴바에서 설정할 수 있다. [OID 정보 보기]를 선택하면, 이후 수행하는 질의 부터 질의 결과 창에서 데이터를 직접 수정/삭제할 수 있다. 단, 툴바에서 이 기능을 설정하면 해당 질의 편집기에만 적용되며, 해당 데이터베이스 질의 편집기 옵션의 값은 변경되지 않는다.
정의된 SQL 매개			질의 편집기에서 Prepared Statement 를 작성하고 해당 문장을 블록 선택한 다음 이 아이콘을 클릭하

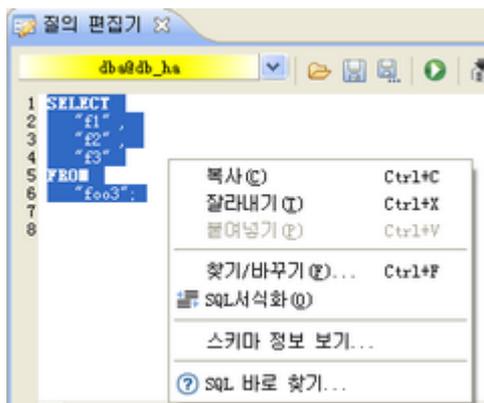
변수 설정		면, 해당 문장의 매개 변수 값을 설정할 수 있다.
도움말		편집기에서 선택한 텍스트에 대한 도움말을 본다.
질의 결과 보기/ 숨기기		질의 결과 창을 보거나 숨긴다.

## 질의 편집 창

질의 편집 창에서는 데이터베이스 조작을 위한 질의를 입력하고 편집할 수 있으며, 툴바에서 지원하는 모든 기능을 사용할 수 있다. 질의 편집 창은 구문 자동 완성 기능 및 팝업 메뉴를 통한 편집 및 스키마 정보 보기 기능을 제공한다.

### 바로가기 메뉴

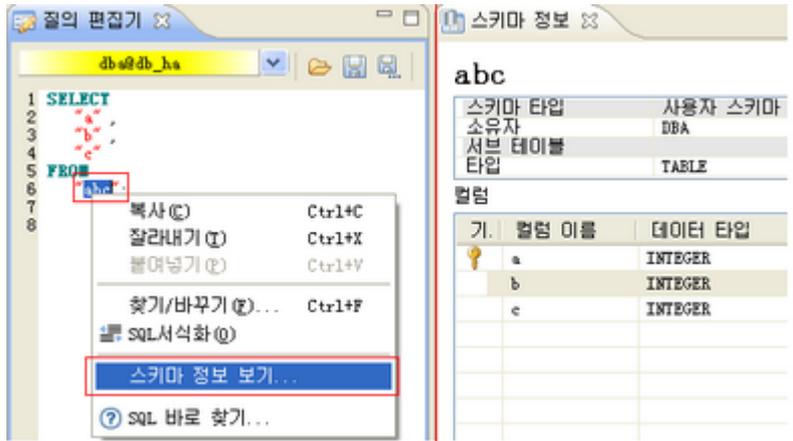
질의 편집 창에서 마우스 오른쪽 버튼을 클릭하여 복사, 잘라내기, 붙여넣기, 찾기/바꾸기, SQL서식화, 스키마 정보 보기를 실행할 수 있다. 편집 기능은 메뉴의 [편집] 기능과 동기화되어 있다.



### 스키마 정보 보기

테이블 이름을 선택한 후 [스키마 정보 보기]를 실행하면 질의 편집기 뷰와 스키마 정보 뷰를 한 화면에서 보면서 질의를 작성할 수 있다.

탐색기에서 질의 편집기를 실행했을 때에만 해당 테이블의 스키마 정보를 확인할 수 있으며, 툴바로 질의 편집기를 실행했을 때에는 지원하지 않는다.



### 구문 자동 완성

큐브리드 데이터베이스 키워드를 입력을 하면 자동으로 구문을 찾아 완성시켜 주므로 사용하기 편리하고 사용자 접근성이 높다.



### 다중 질의 수행 기능

질의문 마지막에 세미콜론(;)을 입력하여, 하나의 질의가 끝나고 다음 질의가 시작됨을 명시한다. 여러 개의 질의가 존재할 경우, 각 질의는 순차적으로 수행되며, 질의별로 질의 결과 창에 결과 탭이 형성된다.

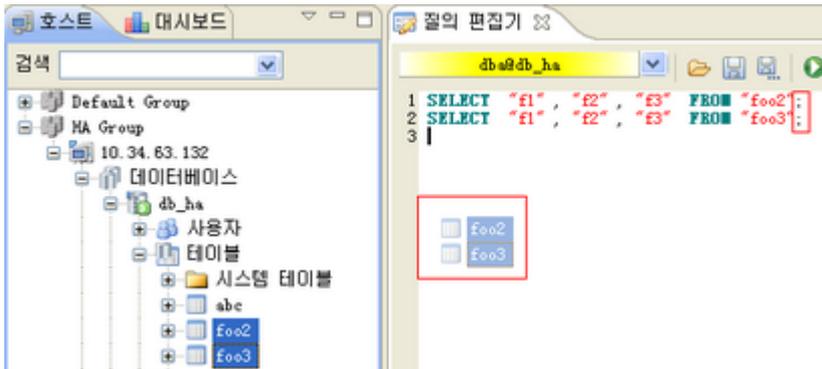
세미콜론으로 구분하지 않고 다중 질의를 수행하면, 첫 번째 질의만 수행되고 이후의 질의는 무시된다.

### 빠른 수행

<Ctrl + Enter> 키를 누르면 현재 커서가 위치한 질의문을 수행한다.

### 끌어다 놓기

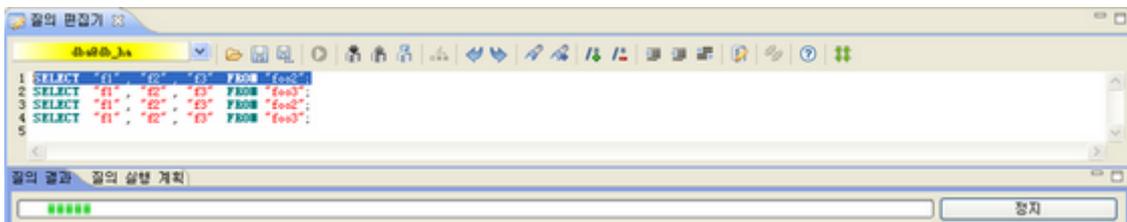
조회하려는 테이블을 호스트 탐색 트리에서 질의 편집기의 편집 창으로 끌어다 놓으면 해당 테이블에 대한 **SELECT** 질의문이 자동 생성된다. 여러 개의 테이블을 한꺼번에 끌어다 놓을 수 있으며, 테이블의 특정 컬럼들만 질의 편집기에 끌어다 놓을 수도 있다.



### 질의 취소 기능

수행 중인 질의를 중지시킬 수 있는 기능이다. 질의 수행 중 [정지]를 클릭하면 상황에 따라 다음과 같이 동작한다.

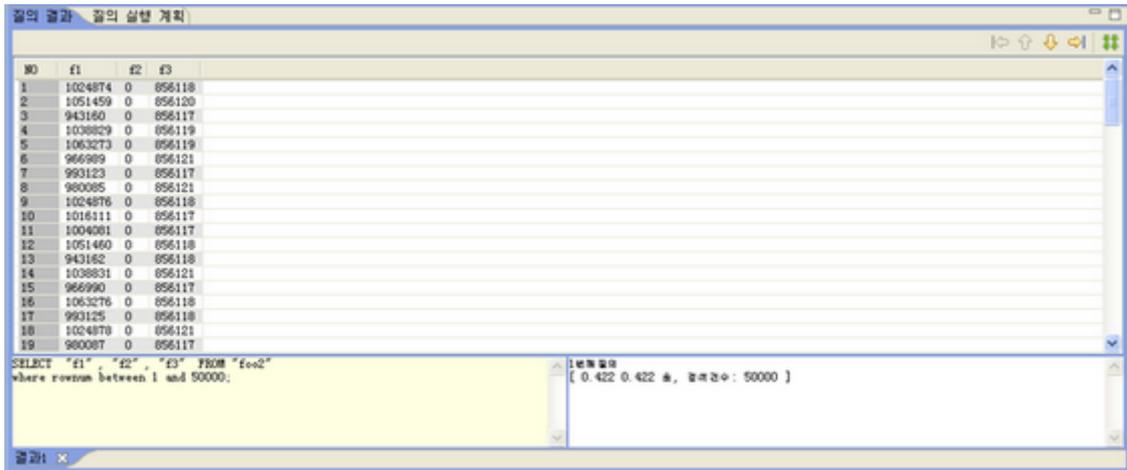
- **다중 질의 수행 시** : 자동 커밋이 설정되어 있을 때 다중 질의 수행 시 정지 버튼을 클릭하면, 현재 취소되는 질의 이전에 처리한 질의들은 정상적으로 반영되고, 현재 취소된 질의와 이후의 질의는 반영되지 않는다. 자동 커밋이 설정되어 있지 않으면, 수행된 모든 질의를 반영하지 않는다.
- **긴 트랜잭션 수행 시** : 정지 호출 시 CUBRID 매니저 > JDBC > 브로커 > 서버의 각 단계로 질의 취소 명령이 전달되어 최종적으로 데이터베이스 서버에서 해당 질의가 정지되어야 실제 질의가 취소된다. 단, Windows 버전의 경우, CUBRID R2.2 이상 버전에서만 질의 취소 기능을 지원한다.



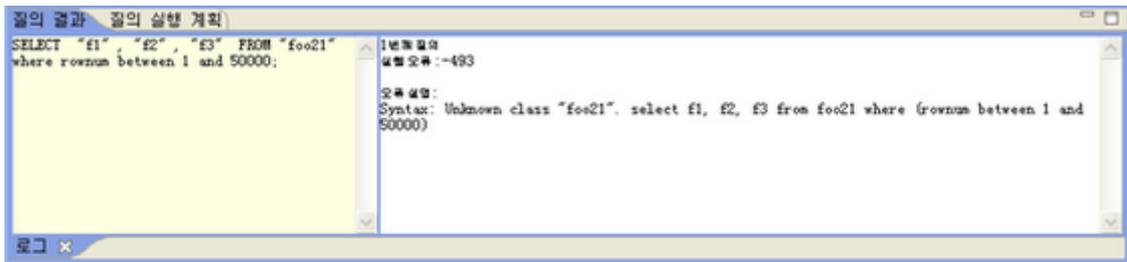
### 질의 결과 창

질의 결과 창은 수행된 질의의 결과를 보여주는 창으로, 여러 개의 질의가 수행된 경우 질의 결과 창 아래에 질의별 탭으로 구분된다. 각 탭을 선택하여 해당 질의의 수행 결과를 확인할 수 있다.

질의 결과 창은 결과 탐색, 수행 질의 정보, 수행 시간 및 검색 건수를 제공하는 영역으로 구분된다.



**SELECT** 문을 제외한 다른 SQL문을 수행하거나, 수행한 **SELECT** 문에 오류가 있으면 로그 탭에 수행 질의 정보와 오류 메시지 또는 수행 시간 정보가 출력된다.

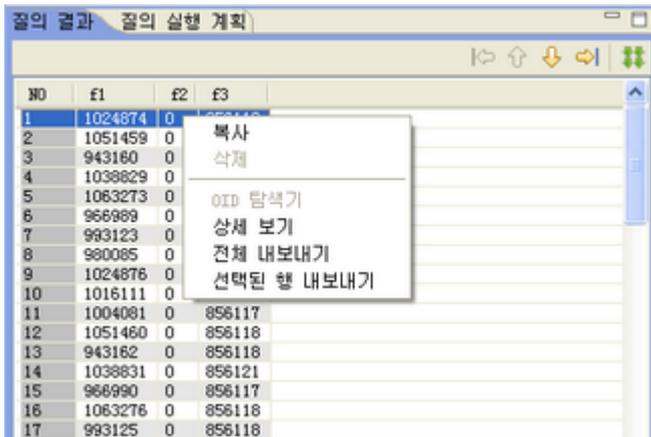


### 질의 결과 창 구조

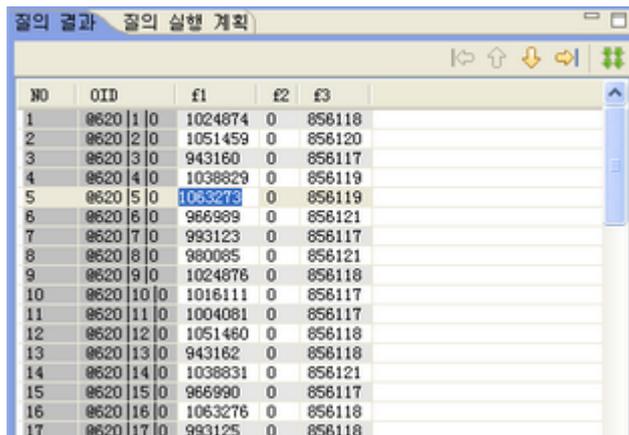
- **결과 탐색:** 전체 검색 결과에서 [질의 편집기 옵션]의 [결과 창의 페이징 단위 설정]에서 설정한 값만큼 이동하며 탐색한다.
- **수행 질의 정보:** 현재 질의 결과가 어떤 질의를 수행한 결과인지 알려 준다.
- **수행 시간 및 검색 건수:** 현재 질의 결과를 얻는데 소요된 서버의 수행 시간과 전체 검색 건수 정보를 제공한다.

### 질의 결과 창 기능

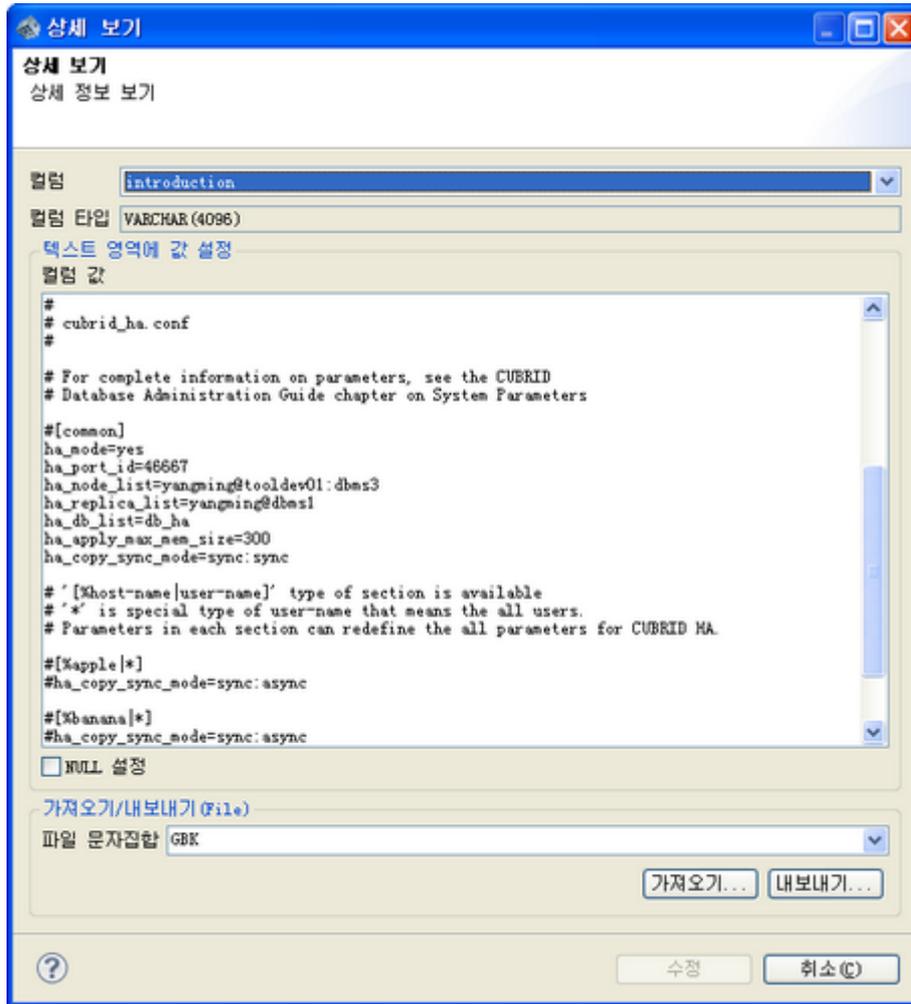
질의 결과 창에는 다음 그림의 바로가기 메뉴와 같이 복사, 수정, 삭제, OID 탐색, 상세 보기, 전체 내보내기 그리고 선택된 행 내보내기와 같은 기능이 있다.



- **복사**: 한 행 전체를 복사한다. 한 행의 특정 컬럼 값만 복사하려면 [상세 보기]를 선택한다.
- **수정**: 질의 결과 창에서 데이터를 직접 수정할 수 있다. 수정하고자 하는 데이터를 더블 클릭하여 편집할 수 있는 상태로 바꾼 후 데이터를 수정한다. 단, 이 수정 기능은 해당 호스트의 속성(🔑)의 [질의 편집기 옵션] 대화 상자에서 [OID 정보 보기]가 선택되어 있거나, 질의 편집기 툴바의 OID 정보 보기(🔑)가 선택되어 있어야만 가능하다.

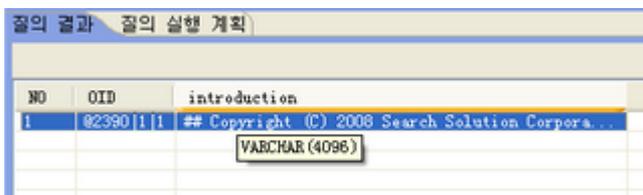


- **삭제**: 질의 결과 창에서 데이터를 직접 삭제할 수 있으며, 삭제할 행을 마우스 오른쪽 버튼 클릭하고 [삭제]를 선택하여 수행한다. 단, 삭제 기능은 수정과 마찬가지로 [OID 정보 보기]가 선택되어 있어야만 가능하다.
- **OID 탐색기**: OID 데이터를 마우스 오른쪽 버튼 클릭하면 활성화된다. 해당 OID에 대해 직접 탐색할 수 있는 기능을 제공한다.
- **상세 보기**: 행으로 출력되는 데이터가 많거나 클 경우 한 행의 데이터를 상세히 보기 어렵다. 이 때 해당 행을 마우스 오른쪽 버튼 클릭하고 [상세 보기]를 선택하여 한 행의 모든 데이터를 상세히 볼 수 있다. [OID 정보 보기]가 선택되어 있다면, 상세 보기 창에서 직접 수정할 수도 있다. **BLOB**, **CLOB** 타입의 데이터는 별도의 파일로 내보내고 가져올 수 있다.



- **전체 내보내기:** 결과 창의 모든 데이터를 엑셀 또는 CSV 파일로 내보낸다.
- **선택된 행 내보내기:** 결과 창에서 선택한 행의 데이터만 엑셀 또는 CSV 파일로 내보낸다. 내보낸 데이터가 저장되는 파일의 문자 집합을 설정할 수 있다.
- **BLOB/CLOB 데이터 내보내기:** **BLOB, CLOB, BIT VARYING**(>100)인 데이터에 대해서는 내보내기 및 가져오기 기능을 실행할 수 없지만, 질의 편집기 결과 창의 상세 정보에서는 내보낼 수 있다. 단, 질의 결과 창에서 해당 타입의 데이터를 직접 표시하지 않고 (BLOB), (CLOB), (BIT)와 같은 타입 태그만 표시한다.
- **BLOB/CLOB 데이터 가져오기:** 질의 편집기 결과 창의 상세 정보에서 **STRING, CHAR, VARCHAR, NCHAR, NCHAR VARYING, BIT VARYING, BIT, BLOB, CLOB** 타입 데이터가 저장된 파일을 직접 열어서 데이터를 입력하거나 편집할 수 있도록 지원한다.

결과 창에서 컬럼명에 마우스 포인터를 올리면 해당 컬럼의 데이터 타입과 크기를 알 수 있다.



## 질의 편집기 옵션

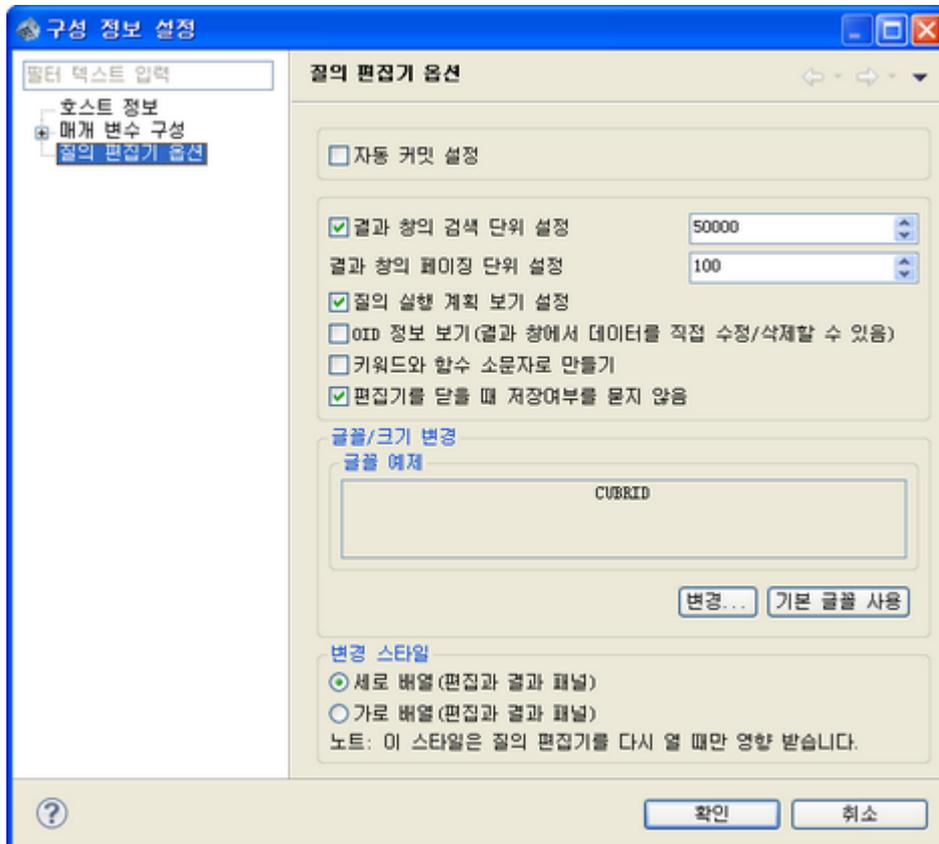
질의 편집기에 대한 옵션은 다음과 같이 설정할 수 있으며, 다음과 같은 적용의 우선 순위를 가진다.

구분	적용	내용
[파일] > [기본 설정] > [질의 옵션]	3 순위	현재 CUBRID 매니저의 기본 정보이다. 각 호스트와 데이터베이스에 적용한 값이 없을 경우 최후에 적용되는 기본 값이다.
[호스트] > [속성] > [질의 편집기 옵션]	2 순위	[파일] > [기본 설정]보다 우선 적용되며, 해당 호스트 내에 공통으로 적용된다.
[데이터베이스] > [속성] > [연결 정보]	1 순위	[호스트] > [설정]보다 우선 적용되며, 설정할 수 있는 정보는 연결 포트와 문자 세트다. 이 정보는 수정 후 여는 질의 편집기부터 적용된다.

### [파일] > [기본 설정] > [질의 옵션]

현재 사용하는 CUBRID 매니저에 대한 기본 정보를 설정한다. 설정 항목은 호스트의 설정과 동일하다.

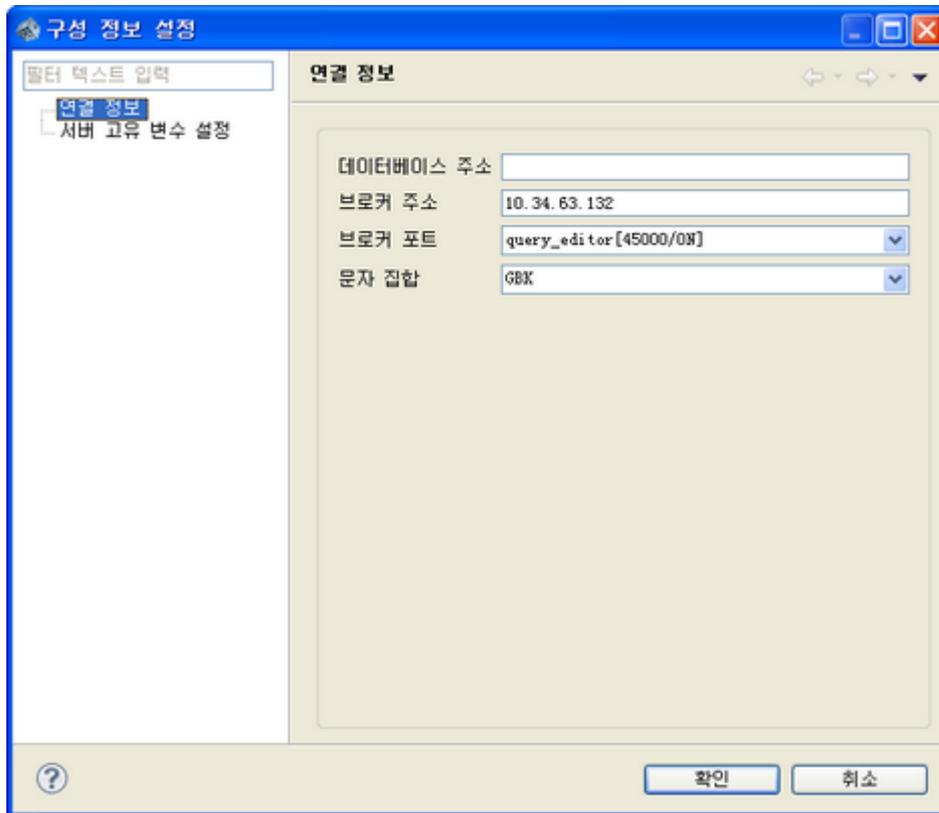
### [호스트] > [속성] > [질의 편집기 옵션]



[질의 편집기 옵션] 대화 상자에서는 다음과 같은 옵션을 설정할 수 있다.

- **자동 커밋 설정:** 질의 편집 창에서 질의를 실행한 후 자동으로 커밋을 수행하도록 기본값으로 설정할 수 있다. [자동 커밋 설정]이 선택되어 있어도 질의 편집기의 툴바에서 자동 커밋 기능을 설정() /해제()할 수 있다.
- **결과 창의 검색 단위 설정:** 질의를 실행한 후, 데이터베이스로부터 한 번에 가져올 결과 행의 개수를 지정한다. 즉, 5,000개로 설정한 상태에서 검색 결과 행 개수가 7,000개라면, 5,000개의 데이터를 가져온 후에 계속 검색 결과를 가져올지 선택할 수 있다.  
또한, 다음의 조건을 충족하는 질의를 수행할 경우에는 자동으로 이 설정 값을 기준으로 **ROWNUM**을 추가하여 불필요하게 서버 자원을 많이 사용하는 것을 방지한다.
- **WHERE** 조건 절이 없는 경우
- **GROUP BY**를 사용하지 않는 경우
- **ORDER BY**를 사용하지 않는 경우
- 집합 함수(**SUM, COUNT, MIN, MAX, AVG, STDDEV, VARIANCE**)를 사용하지 않는 경우
- 계층적 질의(Hierarchical Query)를 사용하지 않은 경우
- **결과 창의 페이징 단위 설정:** 질의 결과 창에서 지정한 데이터 건수 만큼 페이징(   )하여 탐색할 수 있다.
- **질의 실행 계획 보기 설정:** 질의를 실행하기 전후에 해당 질의에 대한 실행 계획을 확인할 수 있다. 이 옵션을 선택하면, 이후 수행하는 질의에 대해 미리 플랜 정보를 생성하게 되므로 질의 수행 시간이 약간의 영향을 받을 수 있다.
- **OID 정보 보기:** 질의 수행 시 OID 정보를 가져온다. 이를 통해 질의 결과 창에서 데이터를 직접 수정/삭제 할 수 있다. 단, 조인 질의와 같이 OID를 가져올 수 없는 경우는 "NONE"이라고 출력한다. 이 옵션을 선택하면, 선택을 안 했을 때보다 질의 수행 시간이 길어질 수 있다.
- **키워드와 함수 소문자로 만들기:** 자동 완성 기능에서 키워드를 소문자로 표시한다.
- **편집기를 닫을 때 저장여부를 묻지 않음:** 파일이 열려있지 않으면 편집기에 작성 중인 내용이 있어도 편집기를 닫을 때 확인하지 않는다.
- **글꼴/크기 변경:** 질의 편집기에서 사용하는 글꼴과 글꼴 크기를 변경한다.
- **변경 스타일:** 질의 편집기 편집 창과 결과 창의 스타일을 변경한다.

## [데이터베이스] &gt; [속성] &gt; [연결 정보]



- **데이터베이스 주소:** 해당 데이터베이스의 IP 주소
- **브로커 주소:** 데이터베이스에 접속하기 위해 사용하는 브로커 주소는 기본적으로 DB 서버의 IP로 설정된다. 만약, 브로커 서버가 분리되어 있을 경우, 이 값을 변경하여 해당 브로커를 통해 접속할 수 있다. 단, 브로커 주소는 CUBRID 매니저 **admin** 사용자만 변경할 수 있으며, 그 밖의 사용자는 **admin** 사용자가 지정한 브로커 주소를 통해서만 접속할 수 있다.
- **브로커 포트:** 데이터베이스에 접속하기 위해 사용하는 브로커 포트는 CUBRID 매니저 **admin** 사용자에 의해 부여된 포트가 출력된다. 단, **admin** 사용자만 지정한 포트를 변경하여 사용할 수 있으며, 그 밖의 사용자는 **admin** 사용자가 지정한 브로커 포트를 통해서만 접속할 수 있다.
- **문자 집합:** 해당 데이터베이스에 적용된 문자 세트로 설정할 수 있다. 한 데이터베이스에 하나의 문자 세트만 사용하는 것을 권장한다. 기본값은 CUBRID 매니저가 수행 중인 시스템(PC)에 설정된 문자 세트이다.

## 질의 실행 계획 보기

질의 편집기 옵션에서 [질의 실행 계획 보기 설정]을 선택하면, 질의 편집기의 툴바에서 [질의 실행 계획 보기] 가 활성화된다.

사용자는 툴바에서 [질의 실행 계획 보기] 를 클릭하거나 <Ctrl+L>을 눌러서, 질의를 실제로 수행하지 않고도 선택된 질의가 어떻게 수행될지에 대한 실행 계획을 확인할 수 있다. 이미 수행된 질의에 대한 실행 계획도 확인할 수 있다.

질의 편집기 하단에는 질의 실행 계획이 항상 보이므로 데이터베이스에 접속하지 않은 상태에서도 실행 계획 이력 파일을 열어서 기존의 실행 계획을 점검해 볼 수 있다.

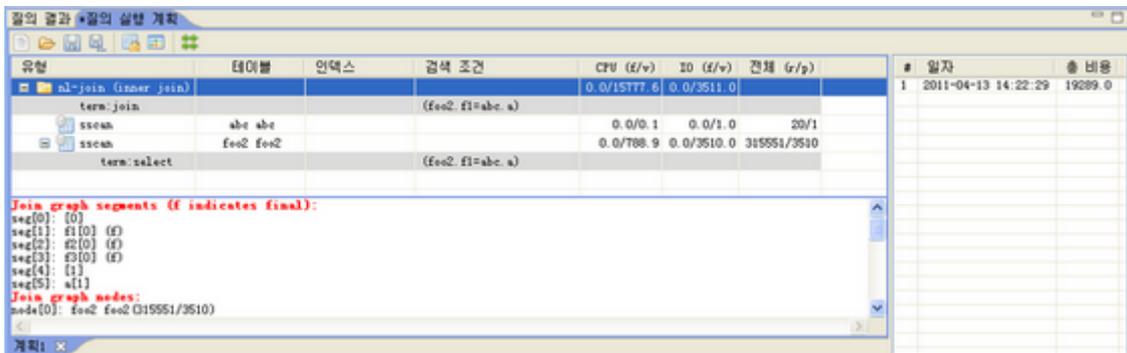
질의 실행 계획 기능은 SQL의 실행 계획을 조회하는 것으로, 일회성이 아닌 실행 계획을 지속적으로 관리하고 조회하기 위한 수집 목적으로 사용한다. 실행 계획을 조회할 때마다 질의 실행 계획 이력에 누적되고, 누적된 자료를 .xml 파일로 저장할 수 있다. 저장된 .xml 파일을 열면 당시의 실행 계획과 실행된 SQL을 조회해 볼 수 있다. 일회성 조회가 목적이라면, 툴바에서 [수집 이력 안 남기기] 를 선택하여 이력을 남기지 않고 현재 실행 계획을 임시로 조회할 수 있다.

질의 실행 계획은 툴바, 실행 계획 출력 창, 선택된 실행 계획에 대한 원문 출력 창 그리고 실행 계획 이력 창으로 구성되어 있다.

### 질의 실행 계획 탭

질을 선택하고 툴바에서 [질의 실행 계획 보기] 를 클릭하면 아래 그림과 같이 [질의 실행 계획] 탭에 질의 수행 계획이 출력된다. [질의 실행 계획] 탭은 해당 질의에 대한 실행 계획을 트리 구조로 요약하여 보여준다.

[질의 실행 계획] 탭은 [질의 결과] 탭 오른쪽에 있으며, 질의 결과를 보다가 [질의 실행 계획] 탭으로 전환할 수 있다.



### 질의 실행 계획 툴바

질의 실행 계획 툴바에는 다음과 같은 기능이 있다.

- **새 파일** : 현재까지 수집하고 조회하던 질의 실행 계획 이력을 모두 초기화하고 다시 수집한다. 기존에 하던 작업을 초기화할 때 사용한다.
- **파일 열기** : 기존에 저장된 실행 계획 이력 파일을 불러온다. .xml 파일을 열면 당시의 실행 계획을 조회해 볼 수 있다.
- **질의 실행 계획 저장** , **질의 실행 계획 새 이름으로 저장** : 수집한 실행 계획을 외부 파일로 저장할 수 있다. 파일의 확장자는 xml이다.
- **수집 이력 안 남기기** : 질의 편집기에서 질의 실행 계획 보기를 이용하여 실행 계획을 조회하면 항상 실행 계획 이력 창에 이력이 추가된다. 이력에 남기지 않고 임시로 조회하려면 [수집 이력 안 남기기] 를 클릭한다. 실행 계획 출력 창 아래의 탭 이름은 "계획"으로만 출력된다.

- **실행 계획 형식**  : 클릭하면 텍스트형  또는 계층형 으로 전환된다. 데이터베이스에서 생성하는 가공되지 않은 실행 계획 소스를 텍스트 형식으로 볼 수 있다.
- **수집 이력창 출력/숨김**  : 오른쪽의 수집 이력 창을 보거나 숨길 수 있다.
- **실행 계획 이력 파일**: 실행 계획 이력을 파일로 저장하였거나 기존에 저장된 파일을 열었을 경우에는 현재 이용중인 실행 이력 파일 이름이 출력된다.



### 실행 계획 출력 창

실행 계획 출력 창에는 단계별로 실행된 실행계획이 계층형(tree)으로 출력된다.

세로축의 각 항목을 노드(node)라고 부르며 각 노드마다 데이터의 내용은 다르다. 위에서 아래로 내려가면서 본다.

가로축은 항목(item)이라고 부르며 유형, 테이블, 인덱스, 검색조건, CPU I/O 비용, Disk I/O 비용, 전체(ROW/PAGE)으로 구분된다.

유형	테이블	인덱스	검색 조건	CPU (f/v)	IO (f/v)	전체 (r/p)
nl-join (inner join)				0.0/15777.6	0.0/3511.0	
term:join			(foo2.f1=abc.a)			
sscan	abc abc			0.0/0.1	0.0/1.0	20/1
sscan	foo2 foo2			0.0/788.9	0.0/3510.0	315551/3510

- **유형**: **sscan**, **iscan**, **idx-join** 등과 같이 실행 계획상의 각 노드별 질의 최적화기의 실행 방법을 의미한다.
- **테이블**: 해당 노드가 실행될 때 참조하는 테이블(class), 뷰(virtual class) 이름과 별명(alias)이 함께 출력이 된다.
- **인덱스**: **iscan** 유형의 경우 사용되는 인덱스명이 출력된다.
- **검색 조건**: 결합(join) 조건, 필터 조건 등이 출력된다. 가독성을 위해 내용은 숨겨져 있으며 +를 클릭하여 하위 노드를 펼치면 상세히 출력된다. 또한 검색 조건별로 별도의 색깔로 볼 수 있다.
- **비용**: 실행 계획에서 출력하는 비용은 CPU, Disk I/O 비용이다. 각각 고정 비용과 가변 비용으로 구분하여 출력된다.
- **전체(r/p)**: 가져올(fetch) 데이터를 만들기 위해 사용되는 테이블의 전체 행의 개수(rows)와 페이지 수(pages)가 출력된다.

질의 실행 계획 출력 창의 아래쪽에는 실행 계획 창에서 선택한 실행 계획의 원문이 출력된다.

```

Join graph segments (f indicates final):
seg[0]: [0]
seg[1]: f1[0] (f)
seg[2]: f2[0] (f)
seg[3]: f3[0] (f)
seg[4]: [1]
seg[5]: a[i]
Join graph nodes:
node[0]: foo2 foo2 (315551/3510)
node[1]: abc abc (20/1)
    
```

### 실행 계획 이력 창

실행 계획 이력 창에는 실행 계획을 수행할 때마다 이력이 누적되어 출력된다. #은 누적 순번이며 실행 계획 출력 창 아래의 탭 번호와 일치한다. 일자란 실행계획을 실행한 일시이며, 총 비용은 CPU와 Disk I/O의 합이다. 이 실행 계획 이력 창의 항목을 더블 클릭을 하면 해당 실행 계획을 실행 계획 출력 창에서 다시 볼 수 있다.

#	일자	총 비용
1	2011-04-13 14:22:29	19269.0
2	2011-04-13 14:31:25	1.0
3	2011-04-13 14:31:25	1.0
4	2011-04-13 14:31:26	1.0
5	2011-04-13 14:31:26	1.0
6	2011-04-13 14:31:26	1.0
7	2011-04-13 14:31:26	1.0
8	2011-04-13 14:31:27	1.0
9	2011-04-13 14:31:27	1.0

### 질의 실행 계획 활용

질의 실행 계획 기능을 활용하면 실행 계획과 해당 테이블의 스키마를 동시에 보면서 분석할 수 있다.

질의 실행 계획 출력 창에서 테이블이 있는 행을 마우스 오른쪽 버튼 클릭하여 [스키마 정보 보기]를 선택하면, 해당 테이블의 스키마 정보를 별도로 열어서 같이 볼 수 있다.



스키마 정보 창을 CUBRID 매니저 밖으로 끌어서 놓으면 별도의 창으로 볼 수도 있다. 이는 다중 모니터를 사용하는 환경에서 편리하게 활용할 수 있다.

# 데이터베이스

## 데이터베이스 기능 구조

데이터베이스 관리를 위한 CUBRID 매니저 기능은 크게 전체 관리 기능과 개별 데이터베이스 관리 및 개발 기능으로 나누어 볼 수 있다.

### 전체 관리 기능

호스트 탐색 트리의 데이터베이스를 마우스 오른쪽 버튼 클릭하여 실행할 수 있다.

데이터베이스를 생성할 수 있으며, [속성]을 선택하면 해당 호스트 전체에 영향을 주는 데이터베이스 공통 매개 변수를 설정할 수 있다.



### 개별 데이터베이스 관리 및 개발 기능

개별 데이터베이스 관리 기능은 탐색 트리와 바로가기 메뉴로 제공된다.

데이터베이스 내의 객체와 작업 자동화 관련 기능이 탐색 트리로 제공된다. 탐색 트리로 제공되는 기능은 데이터베이스가 구동 중일 때에만 사용할 수 있다.



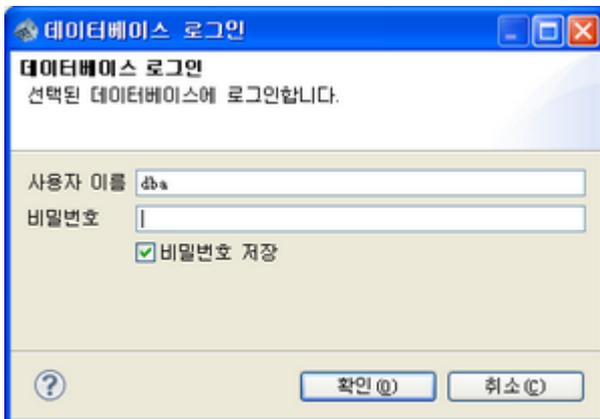
바로가기 메뉴는 운영에 관련된 모든 기능을 제공하며, OID 탐색 기능을 제외한 모든 기능은 데이터베이스가 정지된 상태에서도 사용할 수 있다.



## 데이터베이스 연결

CUBRID 매니저에서 데이터베이스를 구동하려면 데이터베이스에 로그인해야 한다. 호스트 탐색 트리에서 데이터베이스 항목을 마우스 오른쪽 버튼 클릭하고 [로그인]을 선택하거나, 더블 클릭한다.

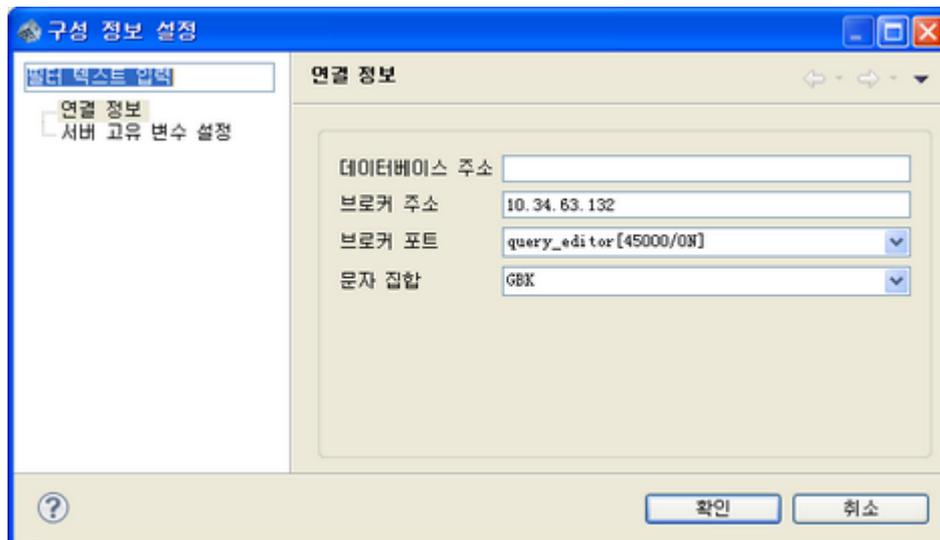
[데이터베이스 로그인] 창에서 사용자 이름과 비밀번호를 입력한다. 기본 설정된 사용자 이름은 **dba**이며, 비밀번호는 없다. [비밀번호 저장]을 선택하여 로컬에 비밀번호를 저장하면 더블클릭만으로 해당 데이터베이스에 로그인할 수 있다.



데이터베이스 로그인 과정에서는 사용자 권한을 검사하고, 로그인에 성공하면 해당 사용자 권한에 맞는 트리 메뉴가 나타난다. 해당 데이터베이스를 구동하기 위해서는 왼쪽 탐색 트리에서 데이터베이스 항목을 마우스 오른쪽 버튼 클릭하고 [데이터베이스 시작]을 선택한다. 데이터베이스가 구동된 상태에서 [사용자]를 클릭하면 사용자를 추가하거나 기존 사용자의 비밀번호를 변경할 수 있다.

CUBRID 매니저가 정상적으로 구동되지 않는 경우, "서버에 접속할 수 없습니다. CUBRID 매니저 서버의 구동 및 기타 접속 환경을 점검하시기 바랍니다."라는 오류 메시지가 나타난다. 이 때, 사용자는 다음 사항을 확인하여 문제를 해결할 수 있다.

- CUBRID 매니저 서버가 구동 중인지 확인한다.
- CUBRID 매니저 서버의 환경 설정 파일을 열고 **cm\_port** 파라미터의 값이 등록된 통신 포트와 같은지 확인한다. [CUBRID 매니저 서버 환경 설정](#)을 참고한다.
- 매니저 클라이언트가 구동되는 현재 시스템에 방화벽이 설정되어 있다면, 매니저 클라이언트 접속을 위한 통신 포트(**cm\_port**, **cm\_port+1**)를 모두 연다. 예를 들어, **cm\_port**가 8001이면, 8001 포트와 8002 포트를 열어야 한다.
- "Cannot execute the current operation because the previous operation is already running."이라는 메시지가 나타나면, 서버가 이미 동일한 작업을 수행하고 있는 것이다. 이 메시지가 나타나면 해당 작업을 재시도한다.
- CUBRID 매니저 클라이언트에서 브로커에 접속할 때 사용하는 IP와 데이터베이스 서버에서 사용하는 IP의 망(Network)이 분리된 환경인 경우, 데이터베이스 속성의 연결 정보에서 데이터베이스 주소와 브로커 주소를 각각 다르게 설정해야 한다. 호스트 탐색 트리에서 데이터베이스 항목을 마우스 오른쪽 버튼 클릭하고 [속성]을 선택하여 연결 정보를 변경한다.



## 데이터베이스 시작

CUBRID 매니저에서 질의를 실행하려면 데이터베이스를 시작해야 한다. 데이터베이스를 선택한 후, 툴바에서 [시작] 을 클릭한다.

## 데이터베이스 생성

데이터베이스 생성은 바로가기 메뉴 또는 상단 툴바에서 [데이터베이스 생성]을 클릭하여 수행 할 수 있다. 데이터베이스 생성 마법사는 총 5단계로 구성되어 있다.

CUBRID 매니저 **admin** 계정만 데이터베이스를 생성할 수 있으며, 이에 대한 자세한 사항은 [사용자 권한 설정](#)을 참조한다.

### 1 단계: 기본 설정

데이터베이스 이름, 일반 볼륨 정보 로그 볼륨 정보와 같은 기본 정보를 설정한다.

The screenshot shows a 'Database Creation Wizard' window with the following fields and values:

- 기본 정보 (Basic Information):**
  - 데이터베이스 이름 (Database Name): [Empty text box]
  - 페이지 크기 (Byte) (Page Size): 4096
- 일반 볼륨 정보 (General Volume Information):**
  - 볼륨 크기 (Mbyte) (Volume Size): 40
  - 페이지 수 (Page) (Page Count): 10240
  - 일반 볼륨 경로 (General Volume Path): /home/yangming/cubrid/databases
- 로그 볼륨 정보 (Log Volume Information):**
  - 로그 페이지 크기 (Byte) (Log Page Size): 4096
  - 볼륨 크기 (Mbyte) (Volume Size): 40
  - 페이지 수 (Page) (Page Count): 10240
  - 로그 볼륨 경로 (Log Volume Path): /home/yangming/cubrid/databases

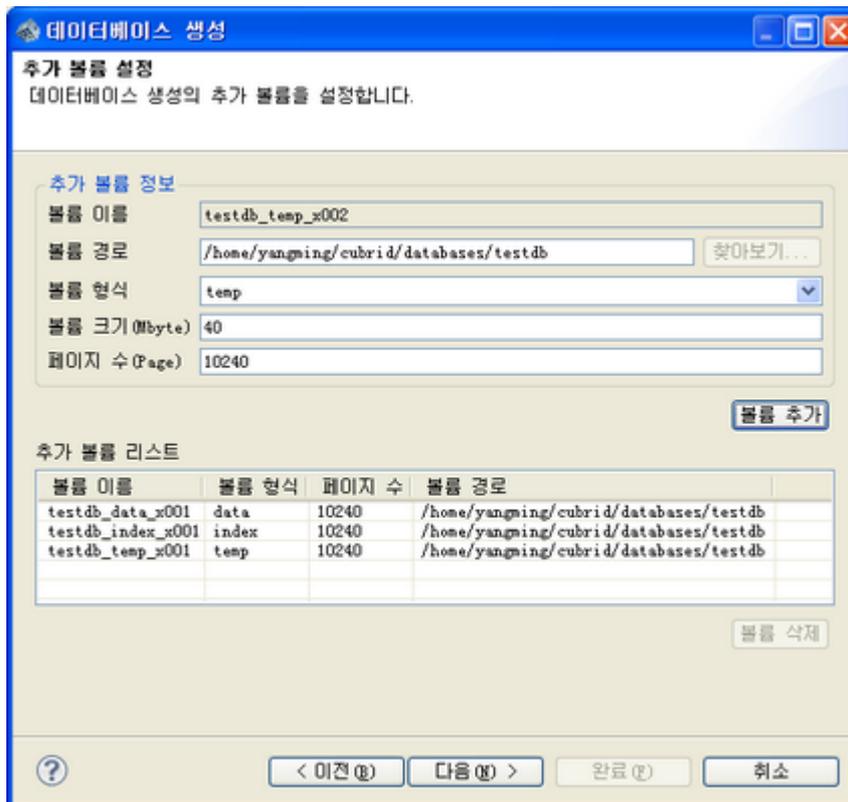
- **데이터베이스 이름:** 데이터베이스 이름은 해당 호스트 내에서 유일해야 하며, 같은 이름이 있으면 경고 메시지를 출력한다. 데이터베이스 이름은 다음과 같은 규칙을 지켜야 한다.
  - 데이터베이스 이름으로는 숫자와 알파벳만 입력할 수 있으며, 최대 길이는 16자이다.
  - Linux (Unix) 파일명으로 사용할 수 없는 글자(공백, \*, &, %, \$, |, ^, /, ~, ₩)와 특수 문자는 사용할 수 없다.
  - "."과 ".."은 데이터베이스 이름으로 사용할 수 없다.
- **페이지 크기:** 1024, 2048, 4096, 8192, 16384 중에서 선택한다. 기본값은 **4096**이다. 데이터베이스를 생성한 후에는 변경할 수 없으므로 생성하는 데이터베이스의 사용 목적에 따라 조정한다. 특별한 경우가 아니면 기본값을 사용하는 것을 권장한다.
- **일반 볼륨 정보:** 일반 볼륨(generic volume)의 볼륨 크기를 Mbyte 단위로 입력하면, 생성하고자 하는 볼륨의 페이지 수를 자동으로 연산하여 보여 준다. 일반 볼륨 경로의 기본값은 데이터베이스 위치 정보

파일(\$CUBRID\_DATABASES/databases.txt)에 설정된 값이다. CUBRID 매니저와 동일한 서버(PC)에 데이터베이스를 생성한다면, [찾아보기]를 클릭해서 디렉토리를 선택할 수 있다.

- **로그 볼륨 정보:** 로그 볼륨(log volume)의 볼륨 크기 및 로그 볼륨 경로를 입력한다. 로그 볼륨 경로의 기본값은 일반 볼륨 경로와 동일한 경로이다.

## 2 단계: 추가 볼륨 설정

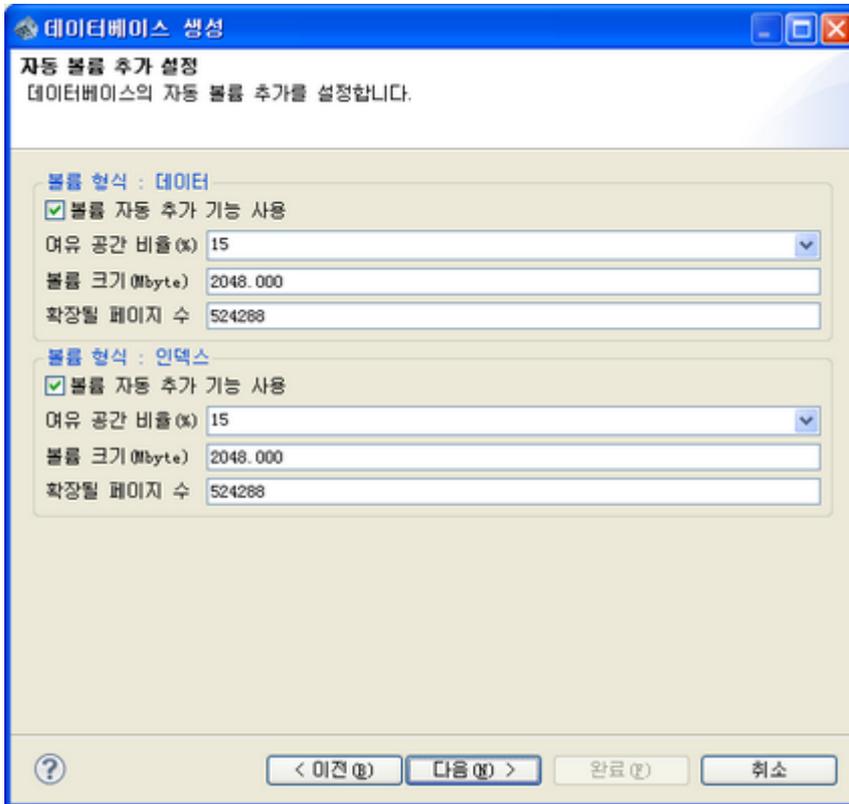
볼륨 형식(generic, data, index, temp)별로 추가 볼륨에 관한 정보를 입력한다. 단, 추가 볼륨 설정에서는 볼륨 형식별 자동 볼륨 추가 설정을 위해 반드시 데이터 볼륨과 인덱스 볼륨을 설정해야 한다.



- **추가 볼륨 정보:** 데이터베이스 생성 시에 추가 용량이 예측되는 경우, 이 단계에서 추가될 볼륨에 관한 정보를 설정한다. 새로 추가할 볼륨의 이름과 경로, 볼륨 타입 및 볼륨 크기를 입력하고, [볼륨 추가]를 클릭하면 입력된 설정 정보에 따라 볼륨이 새로 추가된다. 볼륨 형식은 목적에 따라 data, index, temp, generic 중 하나를 선택할 수 있다. 추가 볼륨에 데이터 볼륨과 인덱스 볼륨은 반드시 추가로 입력한다.
- **볼륨 추가/볼륨 삭제:** [볼륨 추가]를 클릭하면 입력한 설정 정보에 따라 추가될 볼륨이 리스트에 표시된다. 또한, 볼륨 리스트에서 볼륨을 선택한 후, [볼륨 삭제] 버튼을 클릭하면 추가 볼륨 리스트에서 삭제되어 생성되지 않는다.
- **추가 볼륨 리스트:** 데이터베이스 생성 단계에서 추가될 볼륨의 정보를 나열한다.

### 3 단계: 자동 볼륨 추가 설정

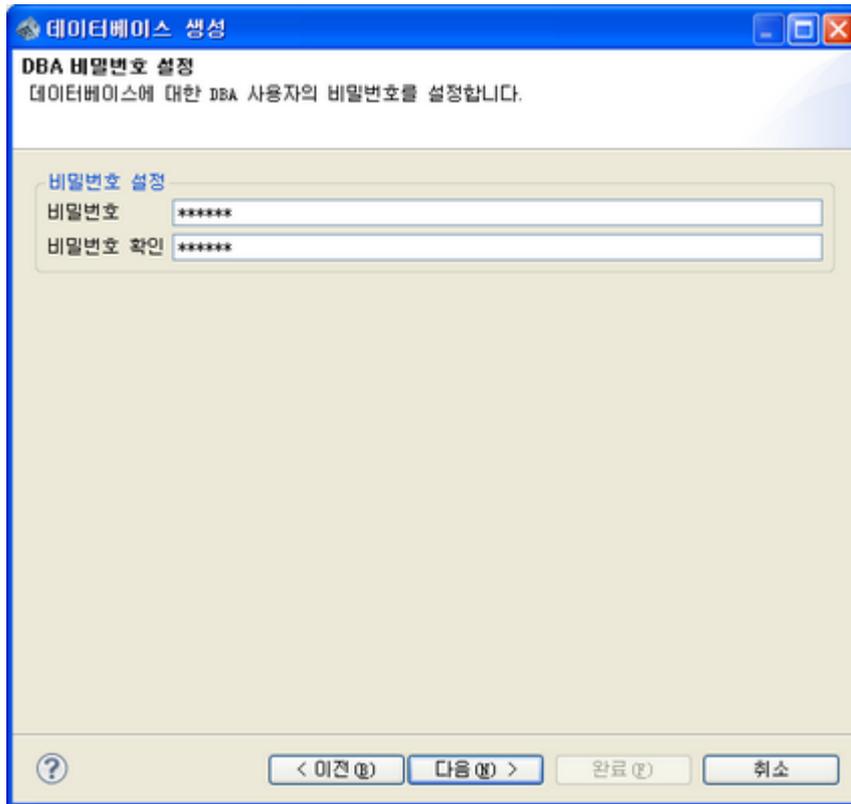
볼륨 형식(data, index)별로 남은 공간이 부족할 경우 자동으로 해당 볼륨을 추가할 수 있는 정보를 설정한다. 데이터 볼륨 또는 인덱스 볼륨에 대해서만 자동 추가 기능을 사용할 수 있다.



- **볼륨 자동 추가 기능 사용:** 선택하면 해당 형식의 볼륨에 대해 자동 추가 기능을 사용한다.
- **여유 공간 비율:** [여유 공간 비율]에 지정한 값과 남은 볼륨이 같을 때, 자동으로 볼륨을 추가한다. 예를 들어, 여유 공간 비율이 15%이면, 기존 볼륨의 여유 공간이 15% 남았을 때, 자동으로 데이터 볼륨을 추가한다. 최소 값은 15이고 최대 값은 30이다.
- **볼륨 크기:** 자동으로 추가할 볼륨의 크기를 설정한다.

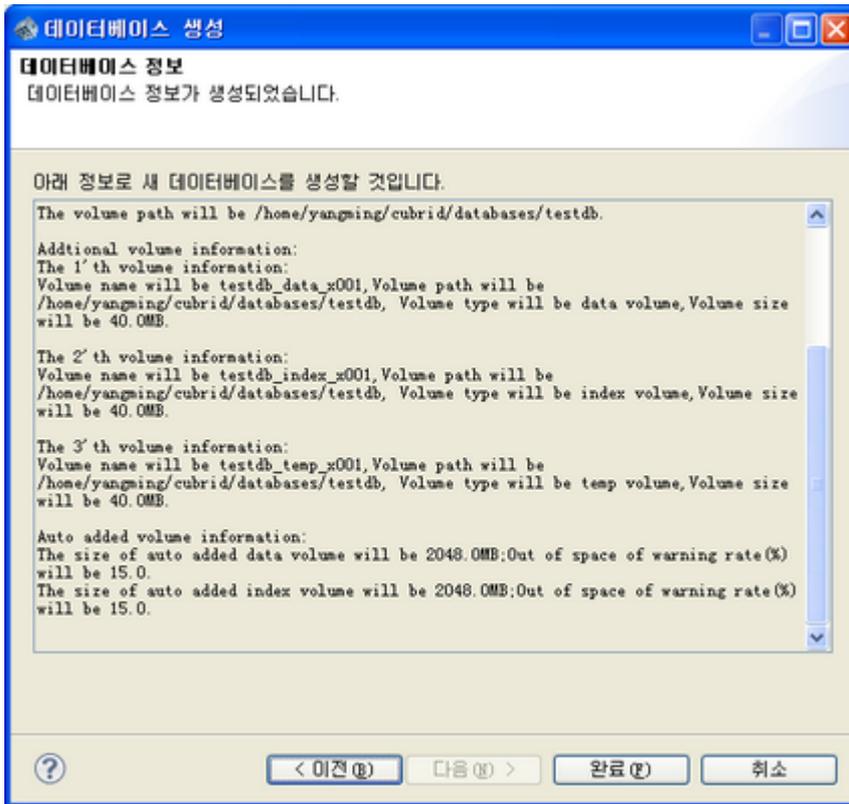
### 4 단계: DBA 비밀번호 설정

새로 생성하는 데이터베이스의 DBA 계정의 비밀번호를 설정한다. 비밀번호는 공백을 포함할 수 없고 4자리 이상이어야 한다.



#### 5 단계: 생성 정보 확인

4단계까지 설정한 정보를 확인 한 후 데이터베이스를 생성한다. 수정 사항이 있으면 [이전]을 클릭하면 이전 단계로 이동하여 수정할 수 있다.



## 데이터베이스 설정

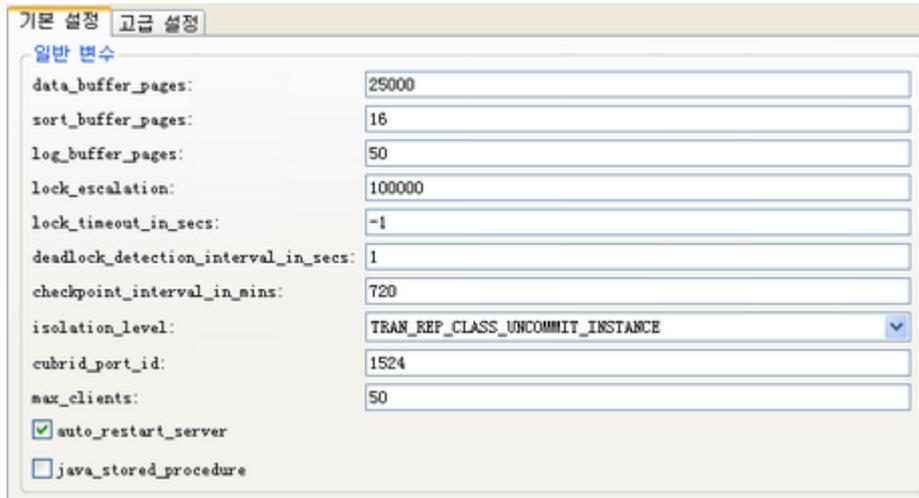
데이터베이스 서버 매개 변수를 설정하는 방법은 다음과 같다.

- 호스트 내의 모든 데이터베이스에 공통으로 적용되는 데이터베이스 서버 매개 변수를 설정하려면, 해당 호스트 또는 [데이터베이스]를 마우스 오른쪽 버튼 클릭하고 [속성]을 선택한 후 [서버 공통 변수 설정]에서 설정한다.
- 특정 데이터베이스의 매개 변수를 설정하려면, 해당 데이터베이스를 마우스 오른쪽 버튼 클릭하고 [속성]을 선택한 후, [서버 고유 변수 설정]에서 설정한다.

### 서버 공통 변수 설정

\$CUBRID/conf/cubrid.conf 파일의 [common] 섹션에 대한 공통 매개 변수를 설정한다. 설치 시 기본으로 포함된 매개 변수를 설정하는 [기본 설정] 탭과 나머지 모든 매개 변수를 설정하는 [고급 설정] 탭으로 구성되어 있다. 각 매개 변수에 대한 자세한 설명은 CUBRID 온라인 매뉴얼의 "데이터베이스 서버 설정"을 참조한다.

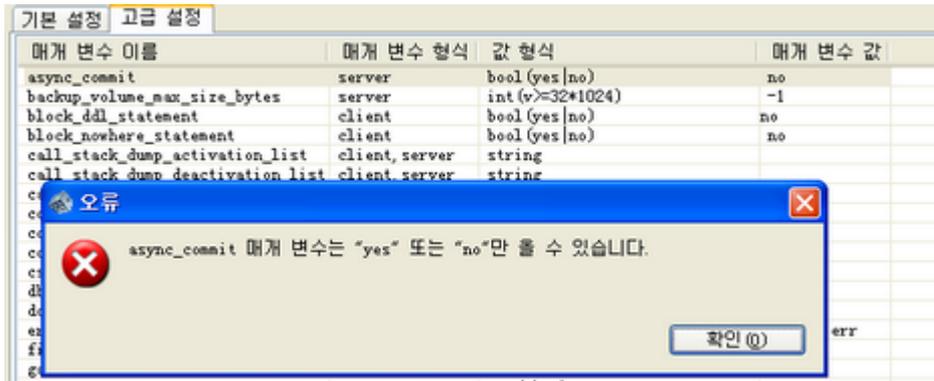
- 기본 설정



- 고급 설정

매개 변수 이름	매개 변수 형식	값 형식	매개 변수 값
async_commit	server	bool (yes no)	no
backup_volume_max_size_bytes	server	int (v)>=32*1024)	-1
block_ddl_statement	client	bool (yes no)	no
block_nowhere_statement	client	bool (yes no)	no
call_stack_dump_activation_list	client, server	string	
call_stack_dump_deactivation_list	client, server	string	
call_stack_dump_on_error	client, server	bool (yes no)	yes
compactdb_page_reclaim_only	utility only	int	0
compat_numeric_division_scale	client, server	bool (yes no)	no
compat_primary_key	client	bool (yes no)	no
csql_history_num	client	int (v)=1&&v<=200)	50
db_hosts	client	string	
dont_reuse_heap_file	server	bool (yes no)	no
error_log	client, server	string	cubrid_err
file_lock	server	bool (yes no)	yes
garbage_collection	client	bool (yes no)	no
group_commit_interval_in_msecs	server	int (v)>=0)	0
ha_mode	server	string (on off yes no replica)	on
hostvar_late_binding	client	bool (yes no)	no
index_scan_in_oid_order	client	bool (yes no)	no
index_scan_oid_buffer_pages	server	float (v)=0.05&&v<=16)	4
insert_execution_mode	client	int (v)=1&&v<=7)	1
intl_mbs_support	client	bool (yes no)	no
log_max_archives	server	int	0
lock_timeout_message_type	server	int (v)=0&&v<=2)	0
max_plan_cache_entries	client, server	int	1000
max_query_cache_entries	server	int	-1
media_failure_support	server	bool (yes no)	yes
oracle_style_empty_string	client	bool (yes no)	no
oracle_style_outerjoin	client	bool (yes no)	no
pthread_scope_process	server	bool (yes no)	yes
query_cache_mode	server	int (v)=0&&v<=2)	0
query_cache_size_in_pages	server	int	-1

서버 공통 변수 설정에서 설정하는 모든 값은 편집 시 바로 유효성 검사를 수행하여 사용자의 실수를 최소화한다.

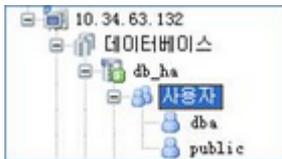


### 서버 고유 변수 설정

서버 고유 변수 설정은 서버 공통 변수 설정과 동일한 기능을 제공한다. 단, 설정한 값은 해당 데이터베이스에만 적용 되는 것으로 \$CUBRID/conf/cubrid.conf 파일의 [@DBNAME] 섹션에만 추가된다.

## 사용자

데이터베이스에 로그인한 후 탐색 트리에서 [사용자]를 마우스 오른쪽 클릭하면, 사용자를 추가, 편집, 삭제할 수 있다. [사용자]의 하부 노드에는 현재 데이터베이스의 모든 사용자가 출력된다.



### 사용자 추가

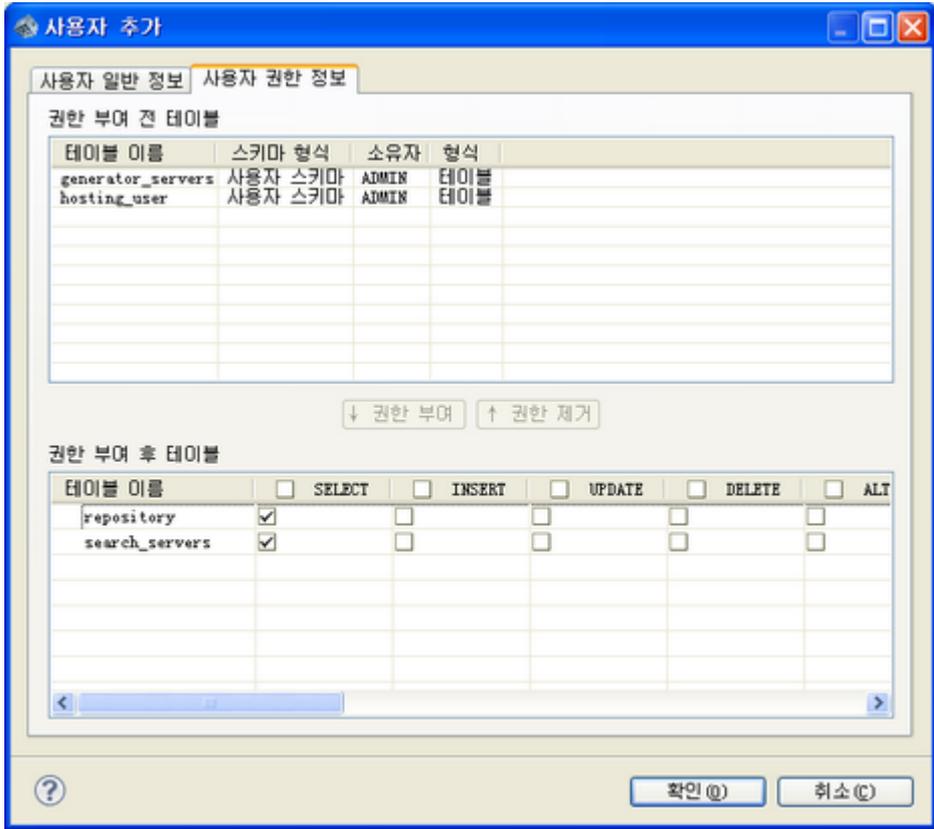
탐색 트리에서 [사용자]를 마우스 오른쪽 클릭하여 [사용자 추가]를 선택하거나, 툴바 메뉴에서 [사용자 추가]를 클릭하여 사용자를 추가할 수 있다.

## 사용자 일반 정보

- **사용자 이름/비밀번호:** 추가할 사용자의 이름, 비밀번호를 입력한다. 사용자 이름의 길이는 최대 32자리이다.
- **모든 사용자:** 추가할 사용자의 그룹으로 선택할 수 있는 사용자 목록을 보여준다.
- **이 사용자가 가진 권한:** 추가할 사용자가 가입한 그룹 목록을 보여준다. 화살표 버튼을 통해 [모든 사용자] 리스트로부터 추가할 사용자가 가입할 그룹을 지정할 수 있다. 단, **dba**와 **public** 계정은 사용자 권한을 조정할 수 없다.
- **이 사용자의 권한을 가진 사용자:** 사용자의 권한을 가진 사용자 목록을 보여준다.

## 사용자 권한 정보

[사용자 권한 정보] 탭에서는 테이블 별로 권한을 부여하고 철회할 수 있다. 권한 부여 대상 리스트는 각 컬럼별로 정렬하여 볼 수 있으며, 부여 대상 테이블을 다중 선택하여 권한을 부여할 수 있다.



### 사용자 편집

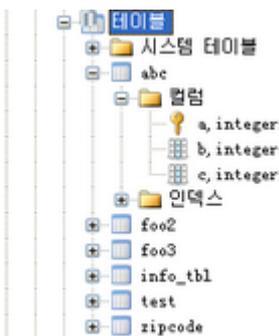
사용자 추가에서 입력한 내용을 편집할 수 있다. 단, 사용자 이름은 변경할 수 없다.

### 사용자 삭제

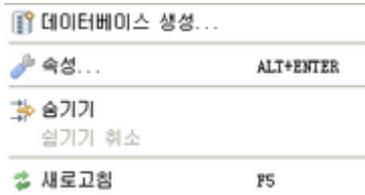
선택한 데이터베이스 사용자 계정을 삭제한다.

## 테이블

데이터베이스에 로그인하면, 접근할 수 있는 테이블, 시스템 테이블, 컬럼, 인덱스를 탐색 트리에서 볼 수 있다.

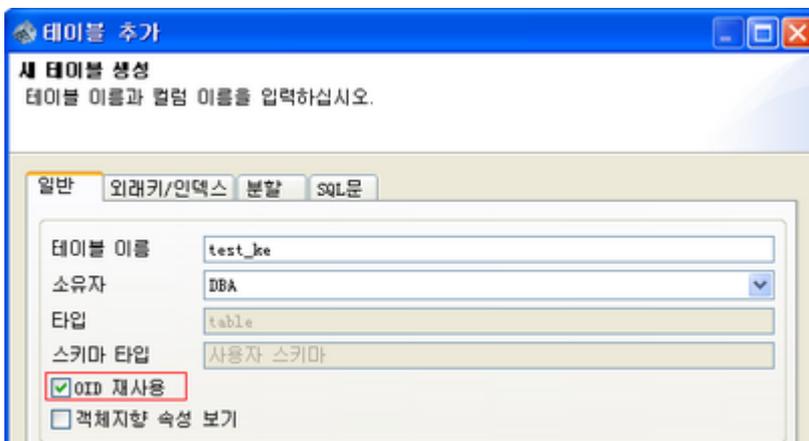


탐색 트리에서 [테이블]을 마우스 오른쪽 버튼 클릭하면, [테이블 추가], [동일 구조 테이블 추가], [], [새로고침] 메뉴가 나타난다.



## 테이블 추가

탐색 트리의 [테이블]을 마우스 오른쪽 버튼 클릭해서 [테이블 추가]를 선택하거나, 툴바에서 [테이블 추가]를 클릭하면, 새 테이블을 생성할 수 있는 마법사가 나타난다.

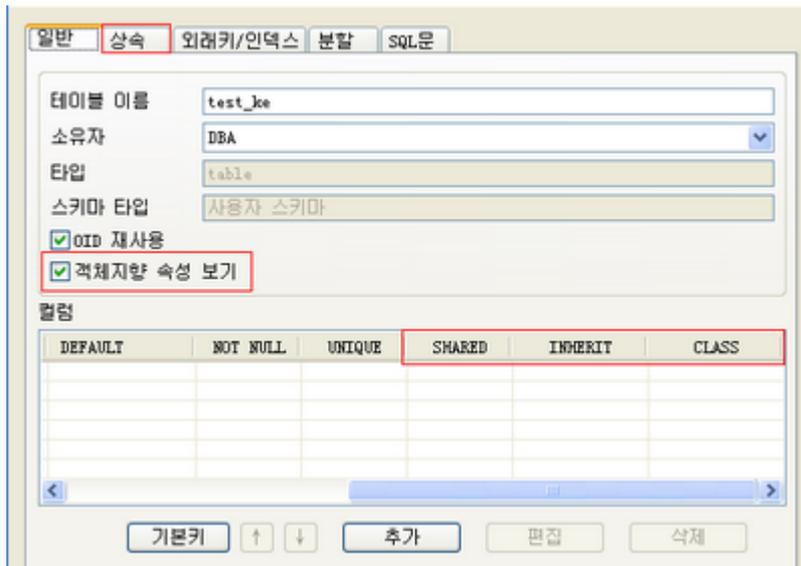


- **[일반] 탭** : 추가할 테이블 이름을 정의하고 컬럼을 추가, 편집, 삭제할 수 있다. 정의된 컬럼에 대해 기본키(PK)를 설정할 수 있으며, 테이블을 생성하기 전에는 [↑], [↓]를 클릭해서 선택한 컬럼의 위치를 조정할 수 있다. 추가할 컬럼 이름, 컬럼 타입, 기본 값 및 컬럼 제약 조건을 컬럼 추가 마법사로 설정할 수 있다. 문법에 맞지 않는 내용이 입력되면 경고 메시지를 출력하거나 문법에 맞지 않는 내용은 선택할 수 없도록 비활성화한다.
  - **OID 재사용** : REUSE\_OID 테이블 옵션이 적용되어 테이블을 생성하며, 이러한 테이블에 대해서는 OID를 이용한 기능은 제한된다. OID 재사용 테이블에 대한 자세한 내용은 CUBRID 온라인 매뉴얼의 "테이블 옵션(REUSE\_OID)"을 참고한다.
- **[외래키/인덱스] 탭** : 외래키와 인덱스를 설정할 수 있다.
  - **외래키 추가** : 외래키 이름, 참조 테이블 이름, 참조 테이블의 기본키를 외래키 추가 마법사로 설정할 수 있으며, 참조 관계 무결성을 유지하기 위한 트리거 동작을 설정할 수 있다. **ON UPDATE, ON DELETE, ON CACHE OBJECT** 옵션을 제공한다.
  - **인덱스 추가** : 인덱스 이름, 인덱스 타입, 인덱스를 설정하고자 하는 컬럼을 인덱스 제약 조건 추가 마법사로 설정할 수 있다. 오름차순(asc) 정렬과 내림차순(desc) 정렬은 지원되는 범위 내에서 선택할 수 있다. **REVERSE** 인덱스는 내림차순 정렬만 지원된다.
- **[분할] 탭** : 해당 테이블에 대한 분할 설정 및 변경을 지원한다.
  - **분할 추가** : 분할 추가 마법사로 분할 형식과 분할 표현식을 설정할 수 있다. **RANGE, LIST, HASH** 분할을 지원한다.

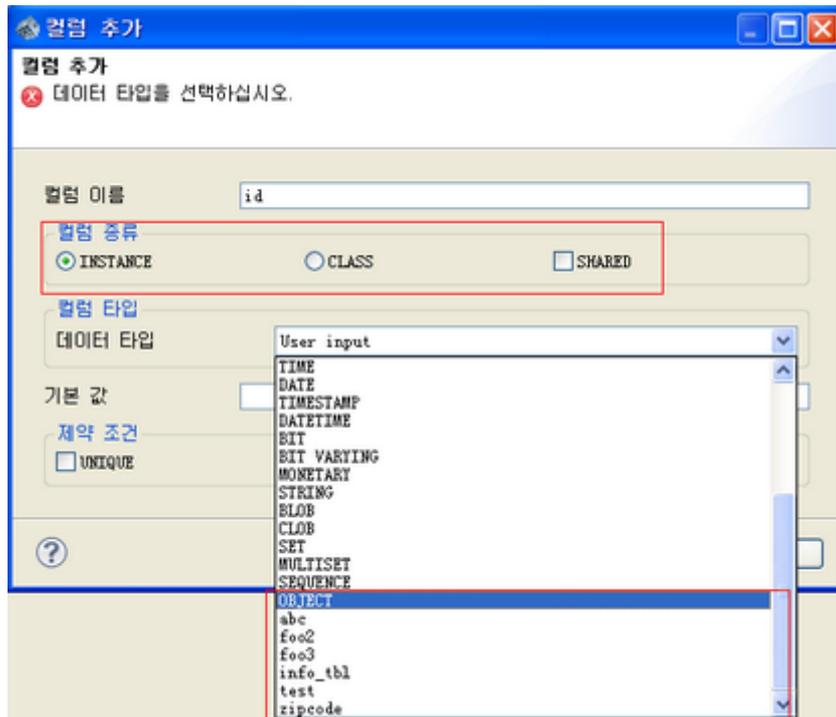
- 분할 편집/삭제 : 분할 관리 기능 중 분할 편집 및 삭제를 수행할 수 있다.
- [SQL문] 탭 : [일반] 탭과 [외래키/인덱스] 탭에서 설정한 대로 생성된 SQL 문을 확인하고 복사할 수 있다.

**객체 지향 테이블 추가**

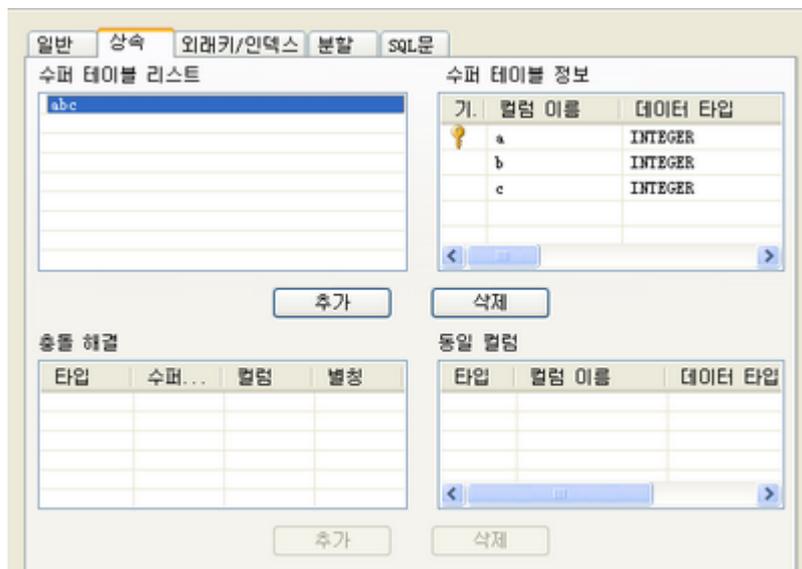
객체 지향 속성을 가지는 테이블을 추가하려면 [객체 지향 속성 보기]를 선택한다. [객체 지향 속성 보기]를 선택하면, 상속 탭이 추가된다.



- [일반] 탭: [객체 지향 속성 보기]를 선택하면, 컬럼 항목으로 SHARED, INHERIT, CLASS 정보가 추가된다. 컬럼 추가 및 편집 시에는 아래 그림과 같이 컬럼 형식을 선택할 수 있고, 데이터 타입으로 OBJECT와 데이터베이스 내에 존재하는 테이블을 선택할 수 있다.



- [상속] 탭: 상속하고자 하는 슈퍼 테이블을 정의할 수 있으며, 컬럼 이름이 충돌하면 조정할 수 있는 기능을 제공한다.



### 전체 조회

탐색 트리에서 각 테이블을 마우스 오른쪽 버튼 클릭하여 [전체 조회]를 선택하면 수행할 수 있다. 또는 [질의 편집기]가 열려 있을 때 질의 편집기의 편집 창 또는 결과 창에 해당 테이블을 끌어서 놓아서 수행할 수 있다. 수행하면 새로운 질의 편집기가 열리면서 전체 데이터를 조회한다.

### 개수 조회

해당 테이블의 전체 데이터 개수를 조회한다. 다음 구문과 같은 기능을 수행한다.

```
SELECT COUNT(*) FROM table_name
```

**DELETE ALL**

해당 테이블의 전체 데이터를 삭제한다. 다음 구문과 같은 기능을 수행한다.

```
DELETE FROM table_name
```

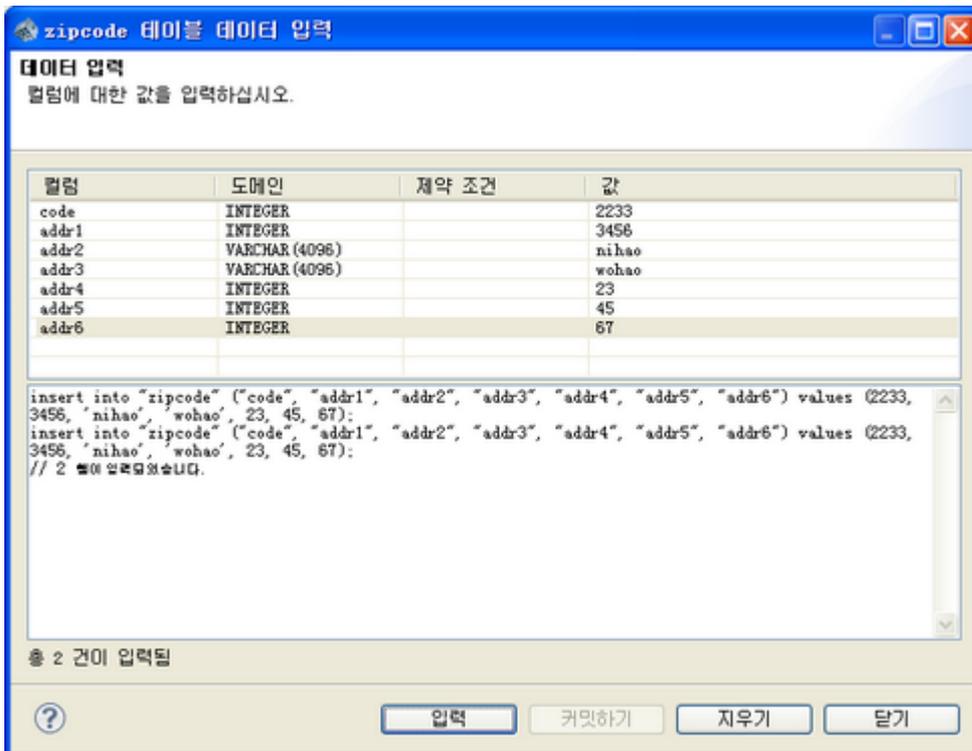
**TRUNCATE TABLE**

해당 테이블의 전체 데이터를 삭제한다. 테이블에 정의된 모든 인덱스와 제약 조건을 포함하여 모든 레코드를 한 번에 삭제하기 때문에, [DELETE ALL]보다 빠르다. [TRUNCATE TABLE]을 사용해서 삭제하면 **ON DELETE** 트리거가 활성화되지 않는다. 다음 구문과 같은 기능을 수행한다.

```
TRUNCATE TABLE table_name
```

**데이터 입력**

해당 테이블의 타입과 제약 조건 등을 보면서 각 컬럼별로 값을 입력할 수 있다.

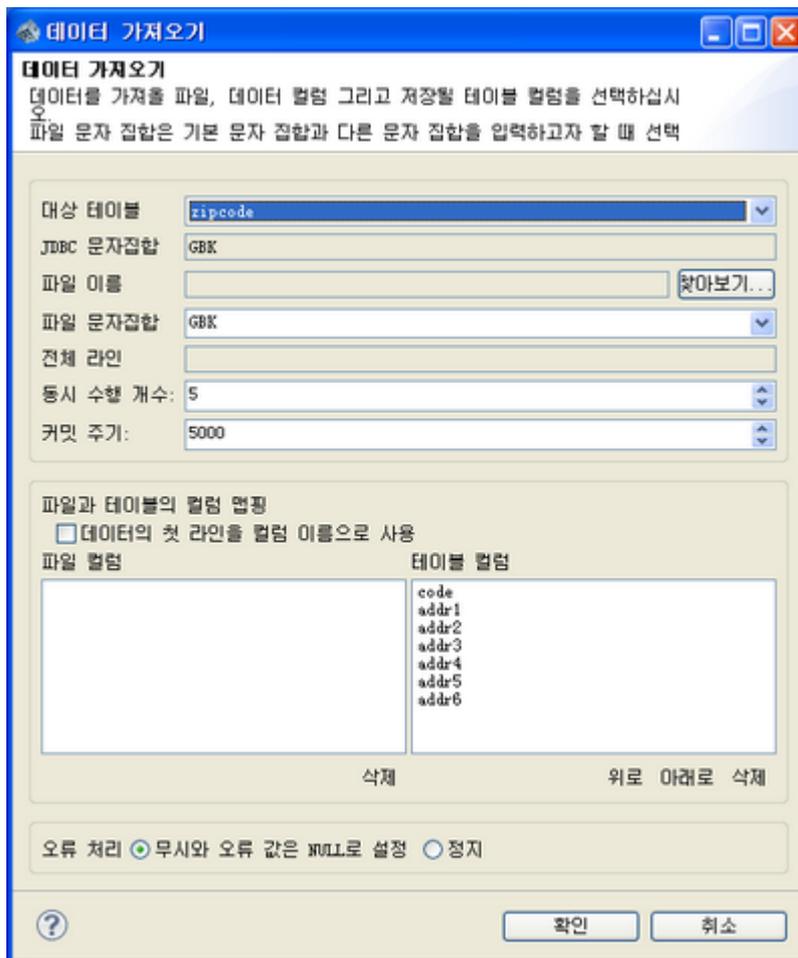


- 여러 인스턴스를 추가할 때, 각 질의와 결과 사이에 줄바꿈 문자를 추가하여 구분할 수 있다.
- 각 필드의 값을 입력할 때 <Enter> 키를 누르면 다음 필드로 커서가 이동한다.
- [지우기] 버튼을 클릭하면 입력 상자의 값, 수행 이력까지 초기화한다.
- 수행 이력창은 편집할 수 없다.
- **DATE, TIME, TIMESTAMP, DATETIME** 데이터 타입에 대해서는 타입 별로 시스템 데이터를 입력할 수 있다. 예를 들어, **DATE** 타입 같은 경우 **SYSDATE, SYS\_DATE, CURRENT\_DATE, DATE'2009-07-05'**와 같은 데이터를 입력할 수 있다.

## 데이터 가져오기

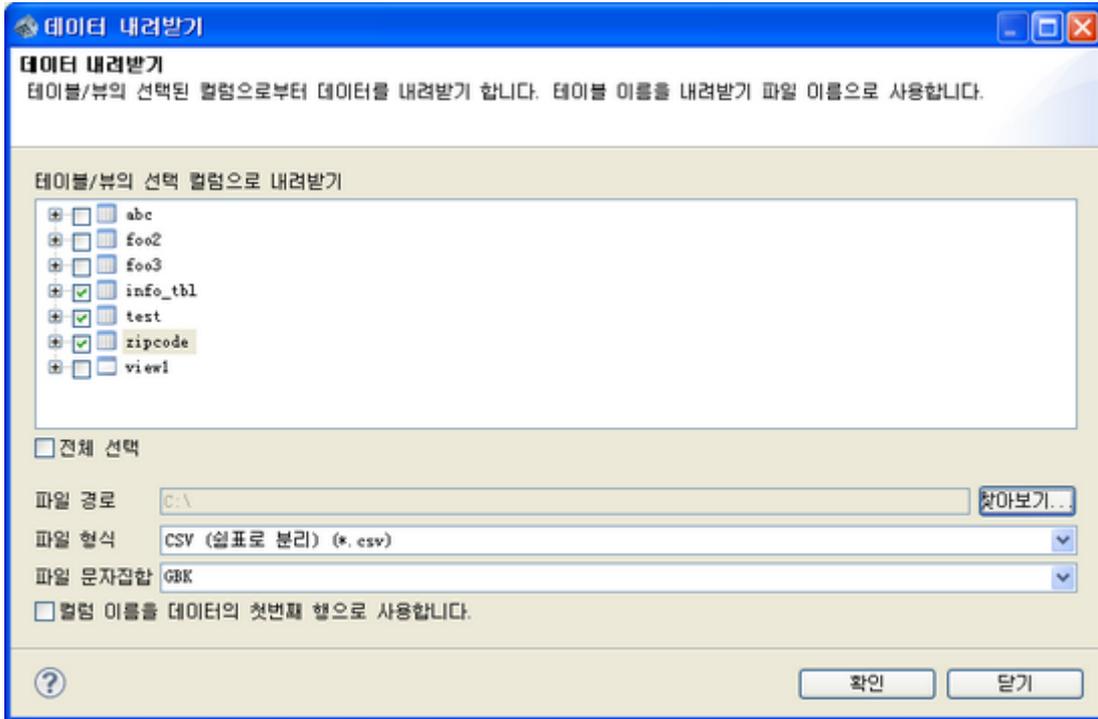
엑셀 또는 CSV 파일의 데이터를 가져와 데이터베이스에 저장할 수 있다. [파일 문자집합] 값을 설정하여 파일로부터 가져올 데이터의 문자 집합을 설정할 수 있다. [JDBC 문자집합]에는 데이터베이스에 저장될 데이터의 문자 집합이 표시되며, 탐색 트리에서 해당 데이터베이스를 마우스 오른쪽 버튼 클릭하여 [속성]을 선택하면 값을 변경할 수 있다.

데이터를 가져오는 중 오류가 발생했을 때 처리 방법을 [무시와 오류 값은 NULL로 설정]과 [정지] 중에서 선택할 수 있다.



## 데이터 내려받기

하나 또는 여러 개의 테이블을 선택하여 엑셀(.xls), CSV, SQL, 큐브리드 로드(.obs) 형식으로 내려 받을 수 있다. 파일명은 테이블 이름으로 지정되며, [파일 경로]에 지정한 경로에 저장된다. 사용자는 [파일 문자집합]에서 파일에 저장할 데이터의 문자집합을 선택할 수 있다.



### 테이블 삭제

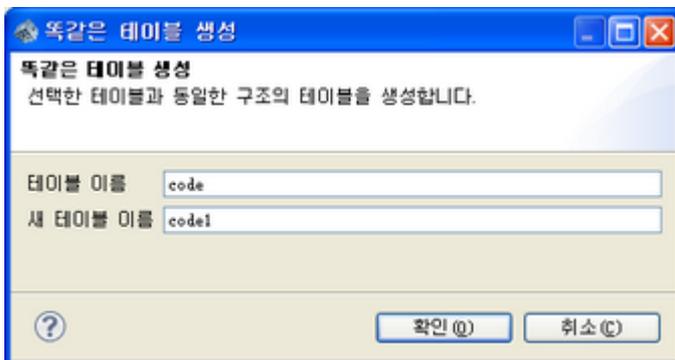
해당 테이블을 삭제한다. **DROP TABLE** 문과 동일하다.

### 테이블 이름 변경

현재 테이블의 이름을 새로운 이름으로 변경한다. **RENAME TABLE** 문과 동일하다.

### 동일 구조 테이블 추가

이미 존재하는 테이블과 동일한 스키마 구조를 가지는 빈 테이블을 생성하며, **CREATE TABLE LIKE** 문이 수행된 것과 같다. 자세한 내용은 CUBRID 온라인 매뉴얼의 "CREATE TABLE LIKE"를 참고한다.

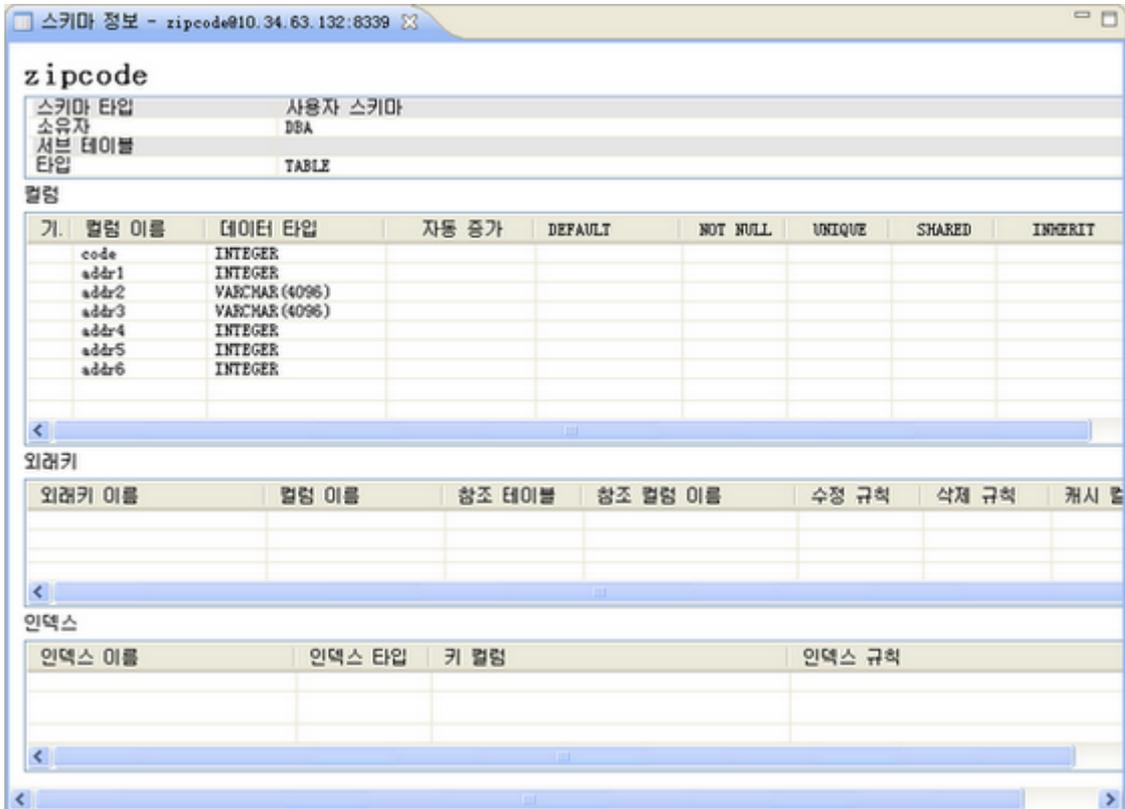


### 테이블 편집

[테이블 추가]의 모든 기능을 테이블 편집에서도 사용할 수 있다. 단, 테이블 내의 컬럼 순서는 조정할 수 없다.

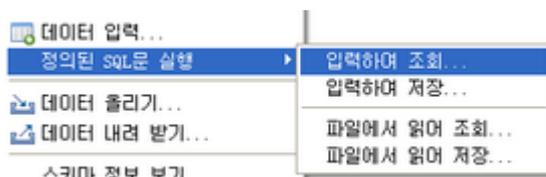
### 테이블 정보 보기

해당 테이블을 더블 클릭하면 해당 테이블의 스키마 구성 정보를 조회해 볼 수 있다.

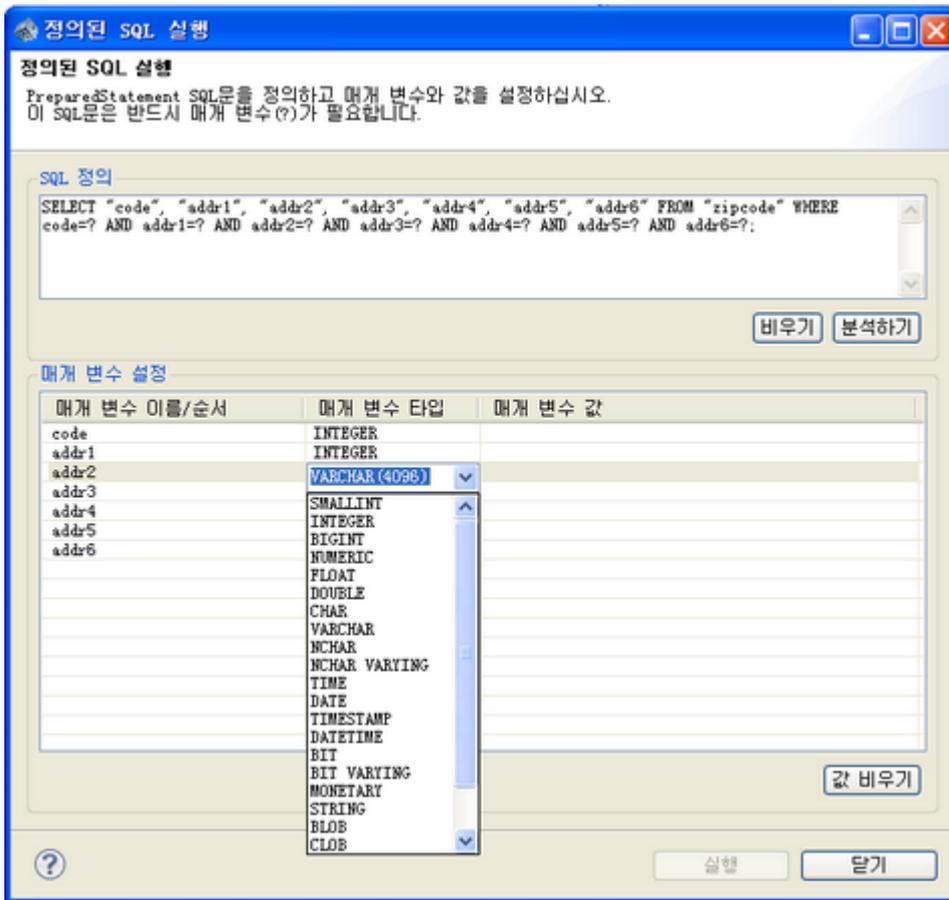


### 정의된 SQL 문 실행

[정의된 SQL문 실행]을 선택하면, SQL 레벨에서 특정 질의문을 미리 저장하고 실행 시마다 매개 변수 값을 설정할 수 있는 prepared statement 기능을 사용할 수 있다.

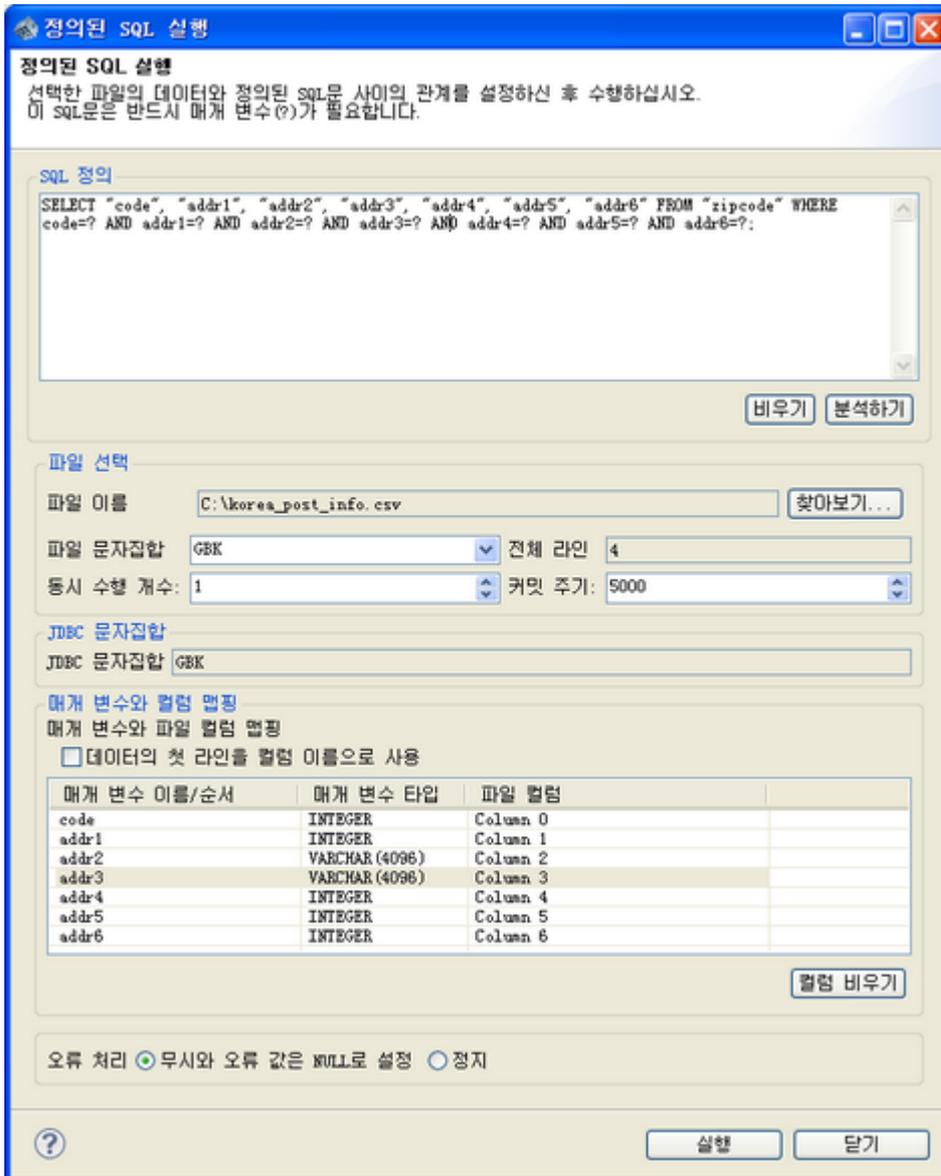


[입력하여 조회]를 선택하면 정의된 prepared statement 문의 매개 변수 값을 입력하여 지정한 조건식을 만족하는 **SELECT** 문을 실행할 수 있다.

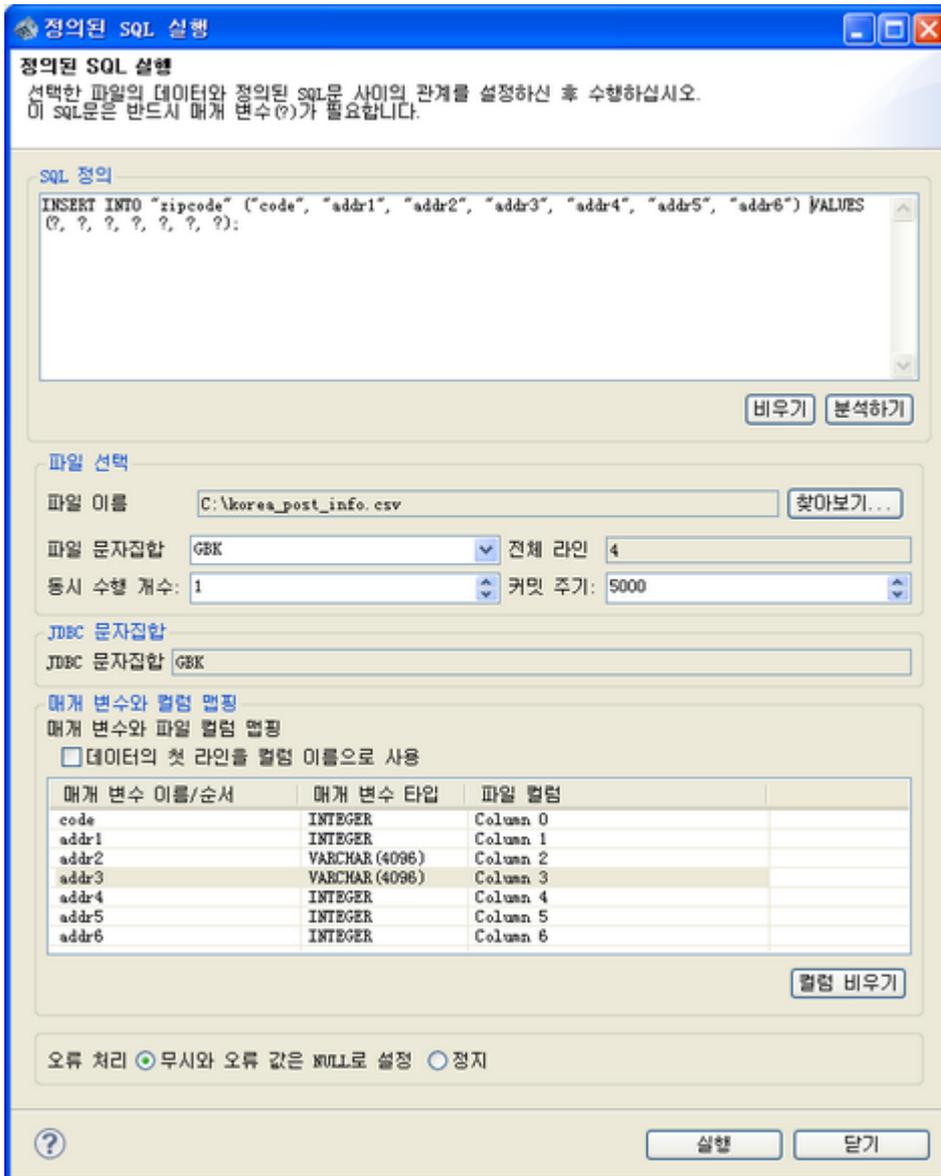


[입력하여 저장]를 선택하면 정의된 prepared statement 문의 매개 변수 값을 입력하여 **INSERT** 문을 실행할 수 있다.



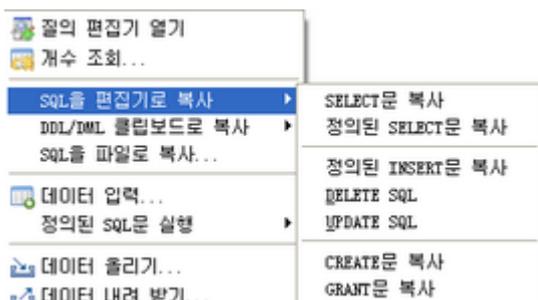


[파일에서 읽어 저장]은 정의된 prepared statement 문에 여러 개의 매개 변수 값을 입력하면서 반복적으로 해당 **INSERT** 문을 실행할 때 사용할 수 있다. 매개 변수 값이 저장되는 파일은 엑셀(xls) 또는 CSV 형식이며, 매개 변수 값이 저장된 파일의 문자 집합, 동시 수행할 스레드 개수, 그리고 커밋 주기를 설정할 수 있다. JDBC 문자 집합은 탐색 트리에서 해당 데이터베이스를 마우스 오른쪽 버튼 클릭하여 [속성]을 선택하면 변경할 수 있다.



### SQL 을 편집기로 복사

선택한 테이블의 DDL과 DML을 현재의 편집기로 복사하는 기능이다. 열려 있는 편집기가 없으면, 새로 편집기를 열어 DDL과 DML을 보여 준다. [SQL을 편집기로 복사]의 하위 메뉴로 SQL문의 형식을 지정한 후에 필요에 따라 수정할 수 있다. **INSERT, SELECT, CREATE, UPDATE, DELETE, GRANT** 문이 포함된다.

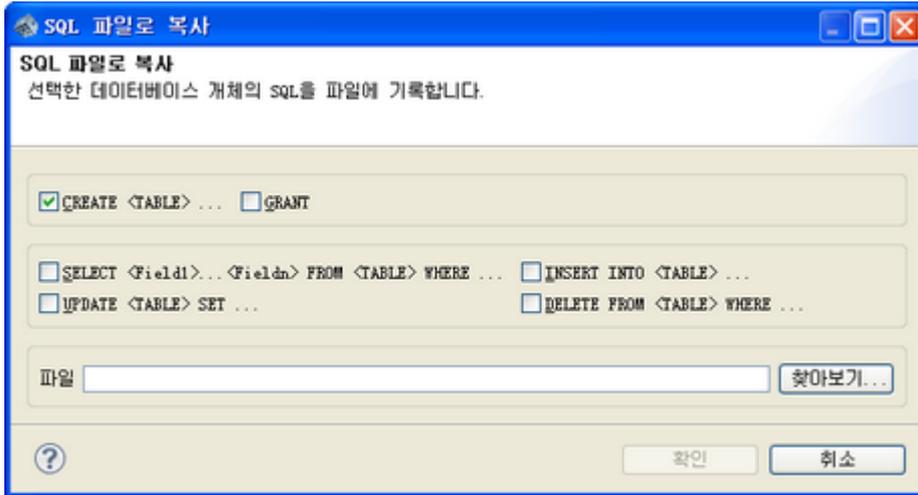


### DDL/DML 클립보드로 복사

[SQL을 편집기로 복사]와 유사하며, 선택한 테이블의 DDL과 DML을 복사한다. 차이점은 편집기가 아니라 클립보드에 복사한다는 것이다.

### SQL 을 파일로 복사

선택한 테이블의 DDL과 DML을 특정 파일로 복사할 수 있다.



## 뷰

데이터베이스에 로그인하면, 접근할 수 있는 뷰와 시스템 뷰를 탐색 트리에서 볼 수 있다.

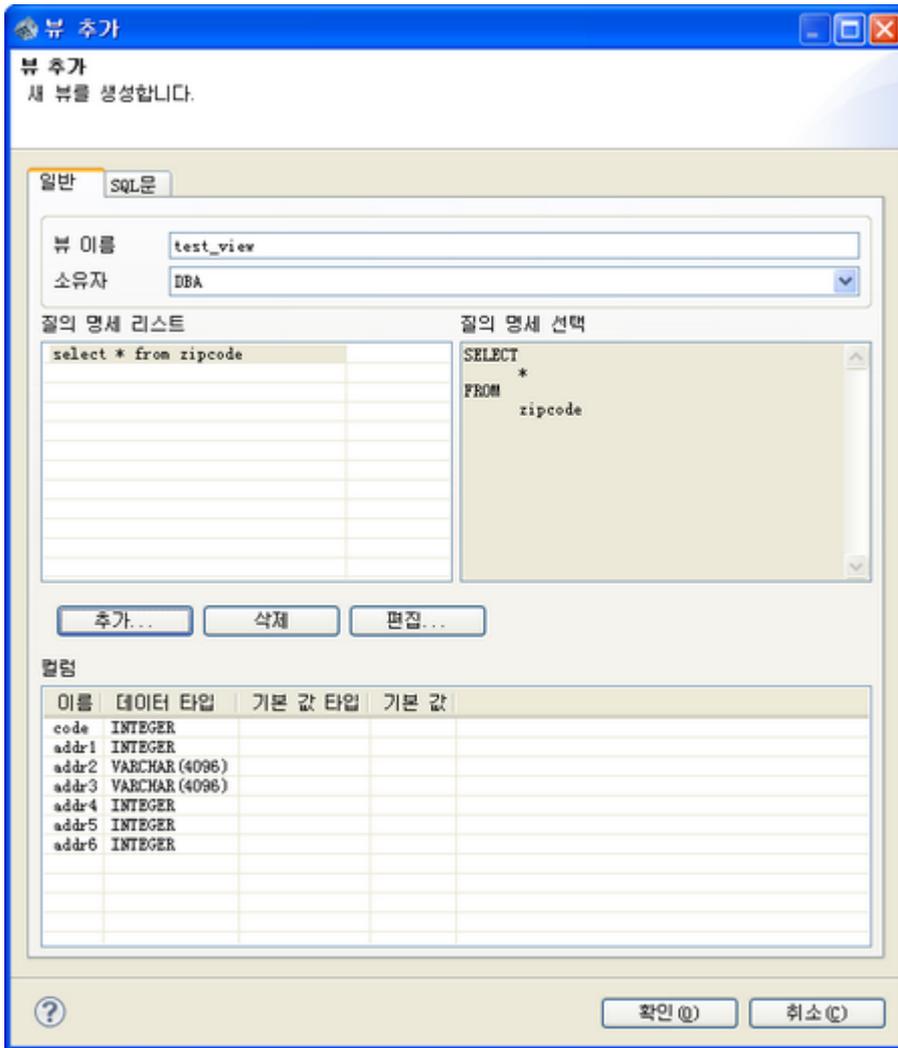


### 뷰 추가

탐색 트리의 [뷰]를 마우스 오른쪽 클릭해서 [뷰 추가]를 선택한다.



뷰 이름과 소유자를 설정한 후, [추가] 버튼을 클릭하고 새로 생성할 뷰에 대한 질의를 작성한다. 뷰 추가 마법사를 통해 작성한 내용은 [SQL문] 탭에서 SQL 구문으로 확인해 볼 수 있다.



### 전체 조회

탐색 트리의 각 뷰를 마우스 오른쪽 버튼 클릭하여 [전체 조회]를 선택하면 수행할 수 있다. 또는 [질의 편집기]가 열려 있을 때 질의 편집기의 편집 창 또는 결과 창에 해당 뷰를 끌어서 놓아서 수행할 수 있다. 수행하면 새로운 질의 편집기가 열리면서 전체 데이터를 조회한다.

### 개수 조회

해당 뷰의 전체 데이터 개수를 조회한다. 다음 구문과 같은 기능을 수행한다.

```
SELECT COUNT(*) FROM view
```

### 데이터 내려 받기

데이터 내려 받기 기능을 통해서 해당 뷰의 모든 데이터를 엑셀, CSV, SQL, 큐브리드 로드 형식으로 받을 수 있다.

### 뷰 삭제

해당 뷰를 삭제한다. **DROP VIEW** 문과 동일하다.

### 뷰 이름 변경

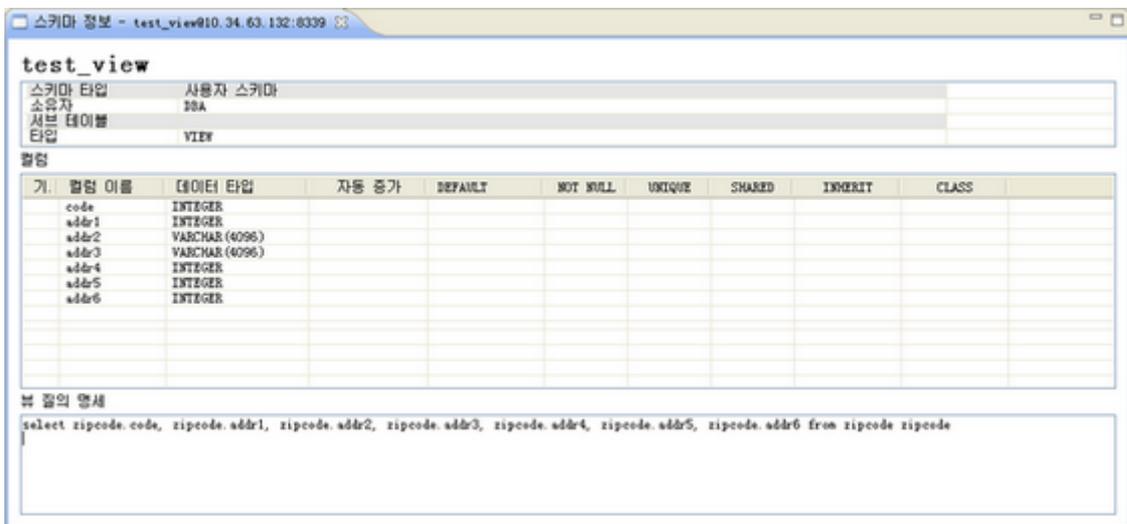
현재 뷰의 이름을 새로운 이름으로 변경한다. **RENAME VIEW** 문과 동일하다.

### 뷰 편집

[뷰 추가]의 모든 기능을 뷰 편집에서도 사용할 수 있다.

### 뷰 정보 보기

해당 뷰를 더블 클릭하면 해당 뷰의 스키마 구성 정보를 조회해 볼 수 있다.

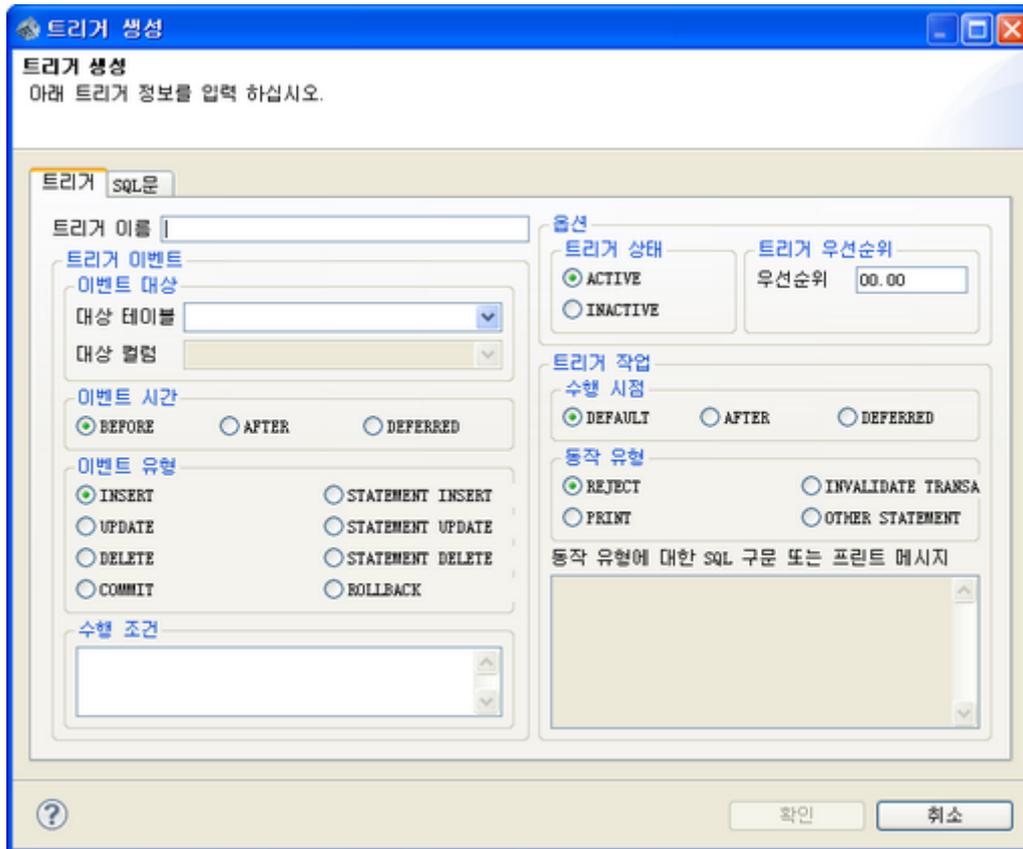


## 트리거

데이터베이스에 로그인하면, 접근할 수 있는 트리거를 탐색 트리에서 볼 수 있다.

### 트리거 추가

탐색 트리의 [트리거]를 마우스 오른쪽 버튼 클릭해서 [트리거 추가]를 선택한다.



- **트리거 이름:** 추가할 트리거의 이름을 입력한다.
- **대상 테이블/컬럼:** 이벤트의 대상 테이블과 컬럼에 대한 정보를 입력한다.
- **이벤트 시간:** 트리거의 조건을 검사하는 시점을 선택한다. **BEFORE**, **AFTER**, **DEFERRED** 중에서 선택할 수 있다.
- **이벤트 유형:** 발생할 이벤트의 형식을 선택한다. 이벤트 형식에는 **INSERT**, **DELETE**, **UPDATE**, **STATEMENT INSERT**, **STATEMENT DELETE**, **STATEMENT UPDATE**, **COMMIT**, **ROLLBACK**이 있다.
- **수행 조건:** 트리거가 동작하기 위한 조건을 입력한다.
- **수행 시점:** 트리거가 동작하는 시점을 지정한다. **DEFAULT**를 선택하면 트리거 조건 확인 시점을 기준으로 동작한다.
- **동작 유형:** 트리거 동작 유형을 선택한다. 트리거 동작 유형이 **PRINT**, **OTHER**, **STATEMENT**인 경우, 하단의 [동작 유형에 대한 SQL 구문 또는 프린트 메시지]에 부가 정보를 입력할 수 있다.
- **트리거 상태:** 추가할 트리거를 활성화할 것인지 아니면 비활성화할 것인지를 지정할 수 있다.
- **트리거 우선순위:** 트리거의 우선 순위를 지정할 수 있다. 우선 순위 값은 **FLOAT** 값으로 0.00에서 9999.99까지 지정할 수 있다.

## 트리거 삭제

탐색 트리에서 해당 트리거를 마우스 오른쪽 버튼 클릭해서 [트리거 삭제]를 선택한다.

### 트리거 편집

탐색 트리에서 해당 트리거를 마우스 오른쪽 버튼 클릭해서 [트리거 편집]을 선택한다. [우선순위]의 값만 편집할 수 있다.

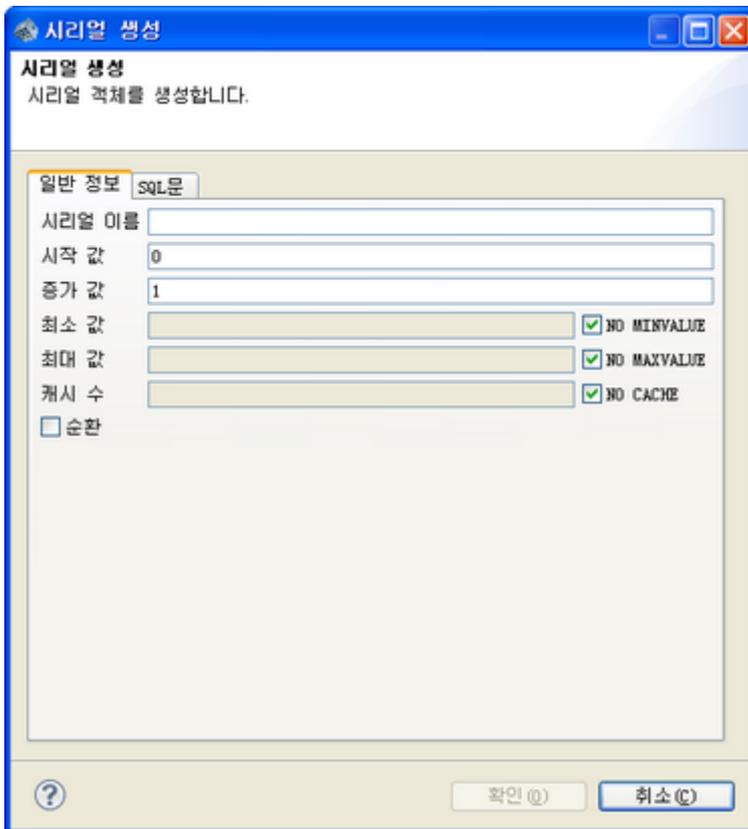
## 시리얼

데이터베이스에 로그인하면, 접근할 수 있는 시리얼을 탐색 트리에서 볼 수 있다.

### 시리얼 추가

탐색 트리의 [시리얼]을 마우스 오른쪽 버튼 클릭해서 [시리얼 추가]를 선택한다.

시리얼 이름은 데이터베이스 내에서 유일해야 한다.



### 시리얼 삭제

탐색 트리에서 해당 시리얼을 마우스 오른쪽 버튼 클릭해서 [시리얼 삭제]를 선택한다.

### 시리얼 편집

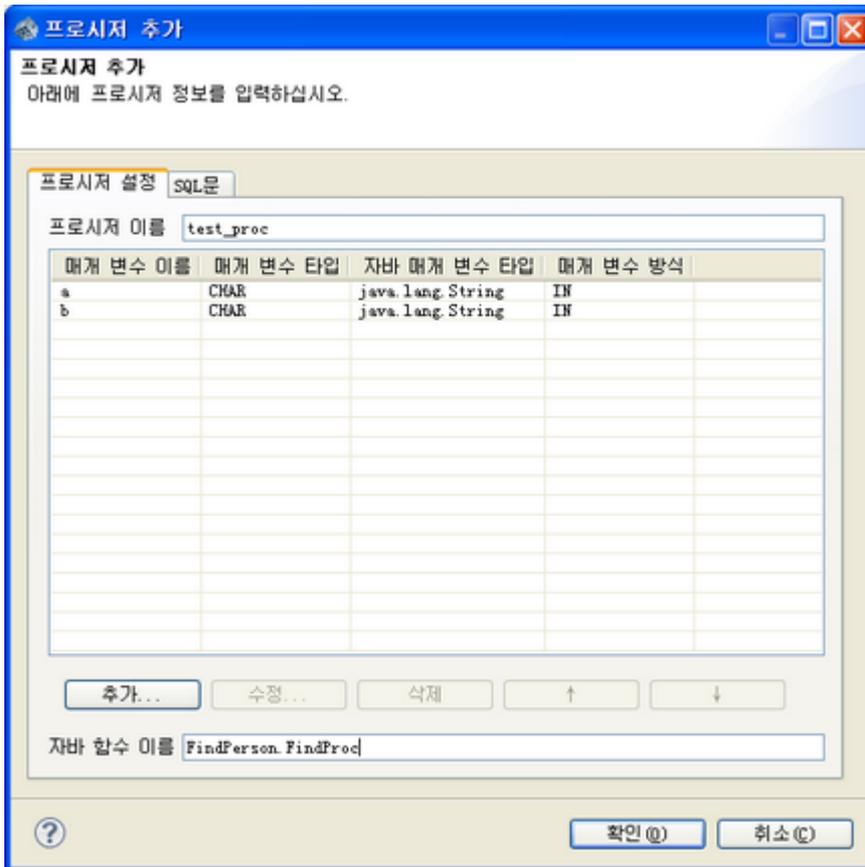
탐색 트리에서 해당 시리얼을 마우스 오른쪽 버튼 클릭해서 [시리얼 편집]을 선택한다.



### 프로시저 추가

Java로 작성한 프로시저를 데이터베이스 서버에서 **loadjava** 명령을 통해 등록한 후, 해당 Java 프로시저를 사용하기 위한 데이터베이스 프로시저를 추가하는 기능이다.

탐색 트리의 [저장 프로시저]를 마우스 오른쪽 버튼 클릭해서 [프로시저 추가]를 선택한다.



선택한 SQL 타입과 호환이 가능한 Java 타입만 보여 주어 사용자가 타입 매핑 시 실수하지 않도록 지원한다.

자세한 내용은 CUBRID 온라인 매뉴얼의 "Java 저장 함수/프로시저"를 참고한다.

위 프로시저 추가 마법사를 통해 작성된 구문은 다음과 같다.

```
CREATE PROCEDURE "test_proc" ("a" CHAR,"b" CHAR)
AS LANGUAGE JAVA
NAME 'FindPerson.FindProc(java.lang.String,java.lang.Integer)'
```

### 함수/프로시저 삭제

탐색 트리에서 특정 함수 또는 프로시저를 마우스 오른쪽 버튼 클릭하여 [함수 삭제] 또는 [프로시저 삭제]를 선택하면 수행할 수 있다.

### 함수/프로시저 편집

탐색 트리에서 특정 함수 또는 프로시저를 마우스 오른쪽 버튼 클릭하여 [함수 편집] 또는 [프로시저 편집]을 선택하면 수행할 수 있다. 기능은 함수/프로시저 추가 시와 같다.

## 작업 자동화

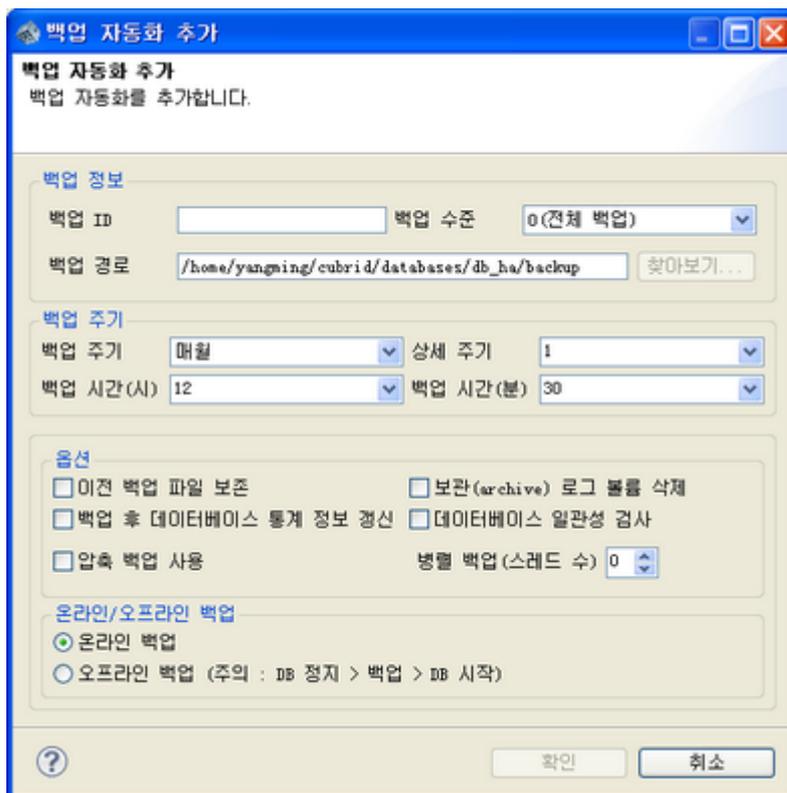
### 백업 자동화

데이터베이스에 로그인하면, 탐색 트리의 [작업 자동화] 아래의 [백업 자동화] 메뉴를 볼 수 있다. [백업 자동화]를 마우스 오른쪽 버튼 클릭하면 [백업 자동화 추가] 메뉴와 [백업 자동화 수행 로그] 메뉴가 나타난다.



### 백업 자동화 추가

CUBRID 매니저 서버를 통해서 주기적으로 백업을 수행하고자 할 경우 백업 자동화 기능을 사용하며, [백업 자동화 추가] 메뉴에서 관련 정보를 설정한다. 백업 자동화 기능은 매니저 서버가 구동 중인 상태에서 **dba** 사용자가 설정할 수 있으며, 해당 데이터베이스의 구동/중지 상태와는 무관하다. 매니저 서버가 구동 중인 상태에서만 설정된 정보에 따라 백업 작업이 자동으로 수행된다.



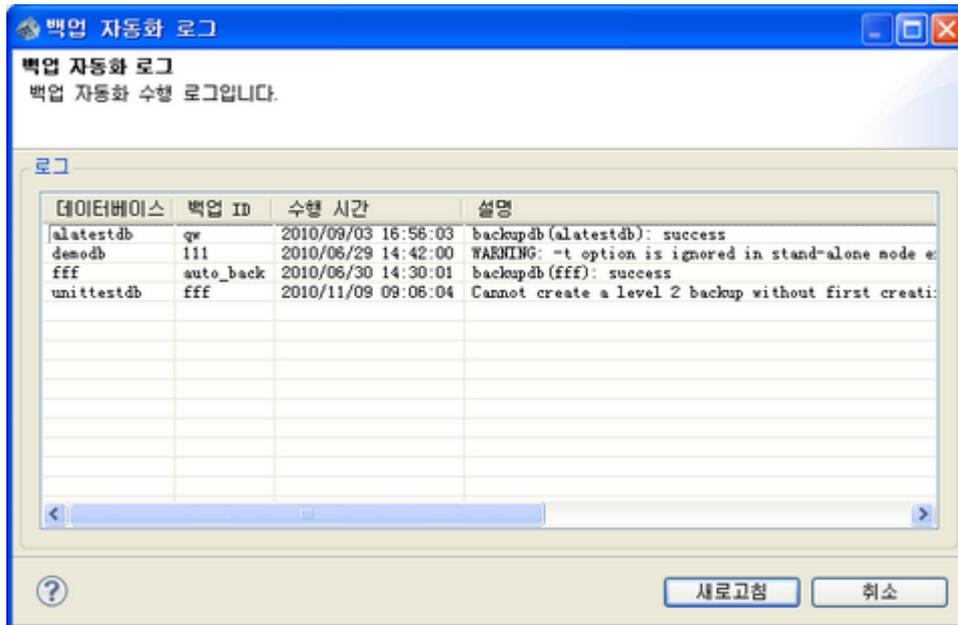
- **백업 ID:** 백업 작업의 이름을 설정한다. 백업 계획이 여러 개일 수 있으므로, 하나의 데이터베이스 내에서는 유일해야 한다.
- **백업 수준:** 0, 1, 2 중에서 선택할 수 있다. 0레벨은 전체 백업을 의미하고, 1레벨은 0레벨 백업 이후의 변경 내역만 백업하는 1차 증분 백업(incremental backup)을 의미하며, 2레벨은 1레벨 백업 이후의 변경

내역만 백업하는 2차 증분 백업을 의미한다. 즉, 0레벨 전체 백업본이 존재할 때에만 1 또는 2레벨 백업 작업이 자동 수행된다.

- **백업 경로:** 백업 볼륨이 저장된 디렉터리를 설정한다.
- **백업 주기:** 백업을 수행할 주기를 매월, 매주, 매일, 특정일 중에서 선택할 수 있다.
- **상세 주기:** 선택한 백업 주기를 상세하게 설정할 수 있다.
- **백업 시간:** 자동 백업을 수행할 시간을 입력한다. 시간과 분을 각각 입력한다.
- **옵션:** 자동으로 수행할 백업 작업의 옵션을 지정할 수 있다. 자세한 정보는 CUBRID 온라인 매뉴얼의 "데이터베이스 백업"을 참고한다.
  - **이전 백업 파일 보존:** 현재 데이터베이스의 기존 백업 볼륨 파일을 **database\_directory/backupold** 디렉터리에 저장하는 옵션이다.
  - **보관 로그 볼륨 삭제:** 백업 시, 보관 로그 볼륨들을 삭제한다. 해당 데이터베이스 서버가 복제 마스터 서버로 설정되어 있는 상태에서 보관 로그 볼륨 삭제를 수행하면, 자동으로 복제에 영향이 없는 부분까지만 삭제한다.
  - **백업 후 데이터베이스 통계 정보 갱신:** 백업 수행 후, 통계 정보를 갱신한다.
  - **데이터베이스 일관성 검사:** 백업 시, 데이터베이스의 일관성 검사를 수행한다.
  - **압축 백업 사용:** 백업 시, 압축을 사용한다.
  - **병렬 백업(스레드 수):** 백업 시, 몇 개의 스레드를 동시에 이용할 것인지를 설정할 수 있다. 최대값은 CPU 개수만큼 설정하는 것이 적절하며, 기본값은 0이다. 기본값으로 설정된 경우 시스템에 의해 자동으로 스레드 개수가 결정된다.
- **온라인 백업:** 데이터베이스가 구동 중인 상태에서만 자동 백업을 실행한다. 중지 상태에서는 데이터베이스를 백업하지 않고 에러 로그만 기록한다.
- **오프라인 백업:** 데이터베이스가 중단 상태일 때만 자동 백업을 실행한다. 데이터베이스가 구동 중이면 데이터베이스를 강제 중단하고 자동 백업을 수행한 후, 다시 데이터베이스를 구동한다.

### 백업 자동화 수행 로그

[백업 자동화 수행 로그] 메뉴에서 자동으로 수행한 백업 작업에 대한 로그를 확인할 수 있다. 백업 대상 데이터베이스, 백업 ID, 백업 수행 시간 등의 정보가 기록된다.

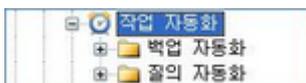


### 백업 자동화 편집 / 삭제

[백업 자동화 추가] 메뉴로 백업 ID가 every\_mon\_lv1이라는 백업 작업을 등록하면, 탐색 트리의 [백업 자동화] 항목 아래에 <every\_mon\_lv1> 항목이 생긴다. 해당 항목을 마우스 오른쪽 버튼 클릭하면 [백업 자동화 편집] 메뉴와 [백업 자동화 삭제] 메뉴가 나타나며, 등록된 백업 자동화 작업을 편집 또는 삭제할 수 있다.

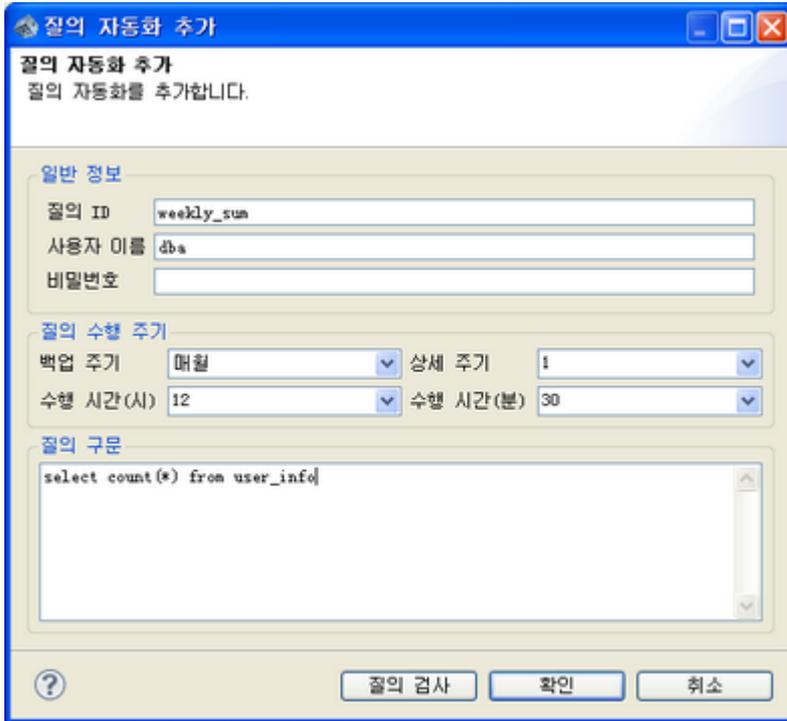
### 질의 자동화

데이터베이스에 로그인하면, 탐색 트리의 [작업 자동화] 아래의 [질의 자동화] 메뉴를 볼 수 있다. [질의 자동화]를 마우스 오른쪽 버튼 클릭하면 [질의 자동화 추가] 메뉴와 [질의 자동화 수행 로그] 메뉴가 나타난다.



### 질의 자동화 추가

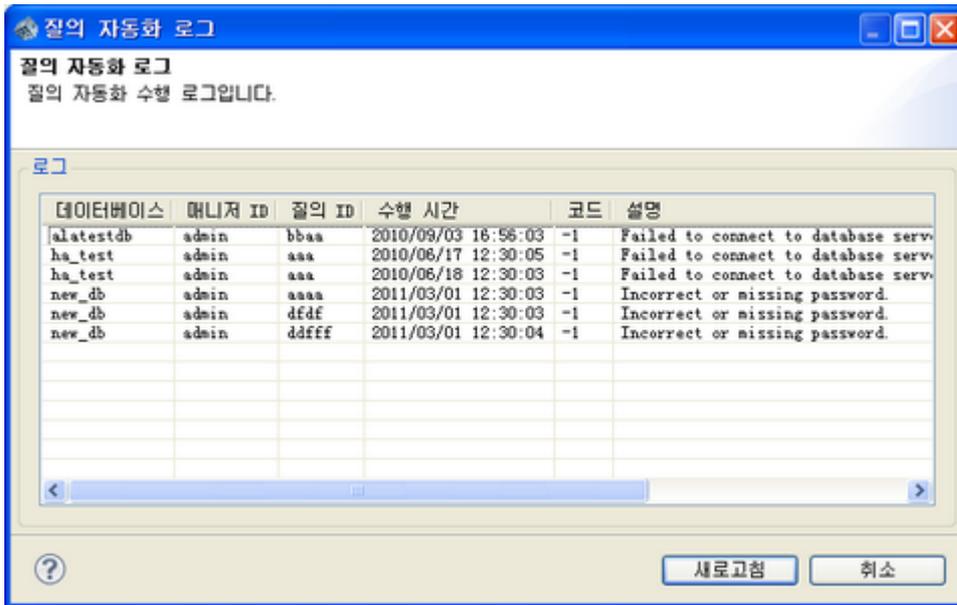
CUBRID 매니저 서버를 통해서 주기적으로 질의를 수행하고자 할 경우 질의 자동화 기능을 사용한다. [질의 자동화 추가] 메뉴에서 관련 정보를 설정할 수 있다. 질의 자동화 기능은 매니저 서버가 구동 중인 상태에서 **dba** 사용자가 설정할 수 있으며, 해당 데이터베이스의 구동/중지 상태와는 무관하다. 매니저 서버가 구동 중인 상태에서만 설정된 정보에 따라 질의 작업이 자동으로 수행된다.



- **질의 ID:** 질의 작업의 이름을 설정한다. 질의 작업이 여러 개일 수 있으므로, 하나의 데이터베이스 내에서는 유일해야 한다.
- **사용자 이름/비밀번호:** 등록된 질의를 자동으로 수행할 데이터베이스 사용자 정보를 비밀번호와 함께 입력한다. 사용자 정보가 틀리면 해당 질의를 자동으로 수행할 수 없으므로, 사용자 비밀번호가 변경되는 경우 [질의 자동화 편집] 메뉴에서 기존의 사용자 정보를 변경해야 한다.
- **실행 주기:** 질의를 자동 실행할 주기를 매월, 매주, 매일, 특정일 중에서 선택할 수 있다.
- **상세 주기:** 선택한 실행 주기를 상세하게 설정할 수 있다.
- **실행 시간:** 질의를 자동 실행할 시간을 입력한다. 시간과 분을 각각 입력한다.
- **질의 구문:** 자동 실행할 질의문을 입력한다. 등록된 질의문은 설정된 시간마다 주기적으로 자동 실행되지만, 실행 결과는 기록되지 않음을 유의한다. 자동 실행되는 질의의 결과는 기록되지 않으므로, 위 그림처럼 **SELECT** 문을 자동 실행하는 것은 아무 의미가 없다. 그러나, 통계 정보 갱신(**UPDATE STATISTICS**), 데이터 갱신 또는 삭제를 주기적으로 수행하고자 하면 질의 자동화 기능이 유용하다. 자동 실행한 질의에 대한 로그는 **\$CUBRID/log/manager/auto\_execquery.log** 파일에 기록된다.
- **질의 검사:** 자동 실행할 질의문을 등록하기 전에 해당 질의의 오류를 미리 검사할 수 있는 기능이다. 해당 질의를 실제 수행하지는 않고, 쿼리 플랜을 생성하여 오류를 판단한다. **csql** 인터프리터에서 **optimization level 514** 모드로 수행하는 것과 동일하다.

### 질의 자동화 로그

[질의 자동화 수행 로그] 메뉴에서 자동으로 수행된 질의에 관한 로그를 확인할 수 있다. 대상 데이터베이스, 매니저 사용자 ID, 질의 ID, 질의 수행 시간, 에러 코드(성공:0, 실패:-1) 등의 정보가 기록된다.

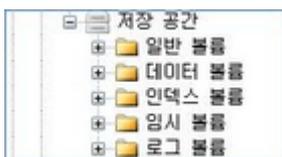


### 질의 자동화 편집 / 삭제

[질의 자동화 추가] 메뉴로 질의 ID가 every\_mon\_delete인 질의 작업을 등록하면, 탐색 트리의 [질의 자동화] 항목 아래에 <every\_mon\_delete> 항목이 생긴다. 해당 항목을 마우스 오른쪽 버튼 클릭하면 [질의 자동화 편집] 메뉴와 [질의 자동화 삭제] 메뉴가 나타나며, 등록된 질의 자동화 작업을 편집 또는 삭제할 수 있다.

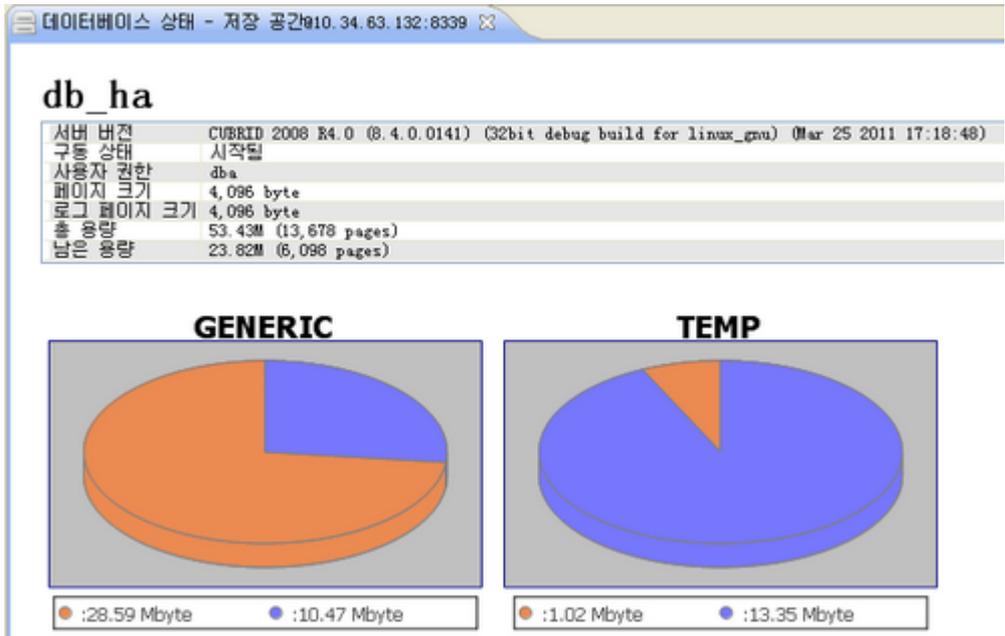
## 저장 공간

데이터베이스에 로그인하면, 접근할 수 있는 저장 공간을 탐색 트리에서 볼 수 있다. 데이터베이스 볼륨 상태를 분석하거나, 자동 볼륨 추가 설정 및 볼륨 추가를 수행할 수 있다.

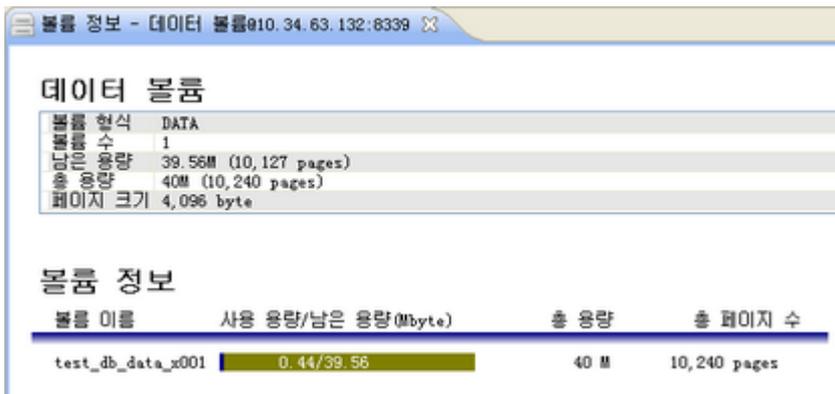


### 데이터베이스 공간 정보

데이터베이스 공간 정보를 확인하려면, 탐색 트리에서 해당 데이터베이스 이름을 더블 클릭하거나, [저장 공간]을 마우스 오른쪽 버튼 클릭해서 [데이터베이스 공간 정보]를 선택한다.



탐색 트리에서 [저장 공간]의 하위 노드를 더블 클릭하면 해당 노드를 구성하는 볼륨 정보를 출력한다.



### 자동 볼륨 추가 설정

**자동 볼륨 추가**

데이터베이스의 자동 볼륨 추가 정보를 설정합니다.

**볼륨 형식 : 데이터**

볼륨 자동 추가 기능 사용

여유 공간 비율 (%) : 15

볼륨 크기 (Mbyte) : 2048,000    확장될 페이지 수 : 524288

**볼륨 형식 : 인덱스**

볼륨 자동 추가 기능 사용

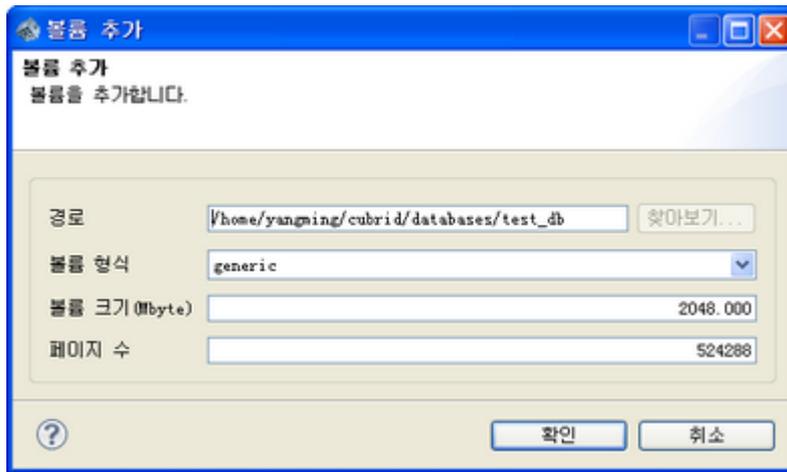
여유 공간 비율 (%) : 15

볼륨 크기 (Mbyte) : 2048,000    확장될 페이지 수 : 524288

확인    취소

- **블룸 형식:** 데이터 블룸 또는 인덱스 블룸에 대해서만 자동 추가 기능을 사용할 수 있다.
- **블룸 자동 추가 기능 사용:** 해당 형식의 블룸에 대해 자동 추가 기능을 사용한다.
- **여유 공간 비율:** [여유 공간 비율]에 지정한 값과 남은 블룸이 같을 때, 자동으로 블룸을 추가한다는 의미이다. 예를 들어, 여유 공간 비율이 5%이면, 기존 블룸의 여유 공간이 5% 남았을 때, 자동으로 데이터 블룸을 추가한다. 최소 값은 5%이고 최대 값은 30%이다.
- **블룸 크기:** 자동으로 추가할 블룸의 크기를 설정한다.

## 블룸 추가



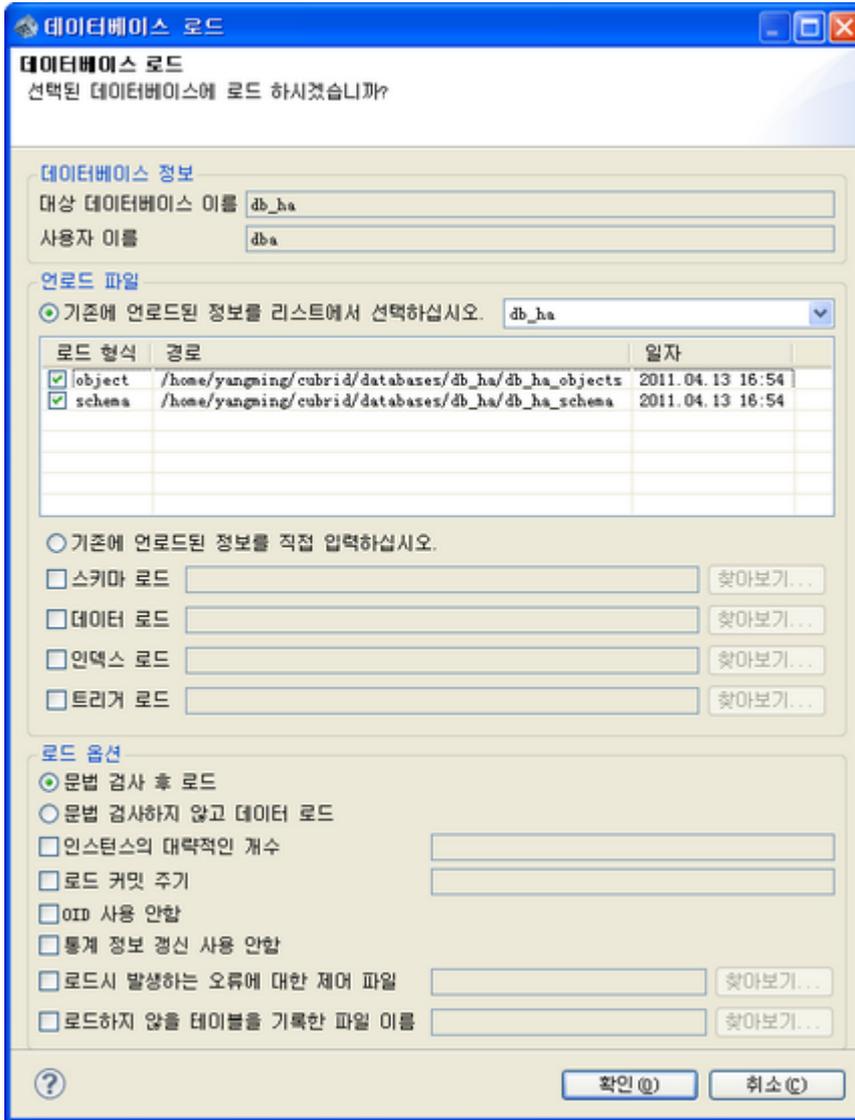
- **경로:** 추가할 블룸이 저장되는 디렉터리를 입력한다. 기본값은 데이터베이스 블룸이 생성되는 디렉터리이다.
- **블룸 형식:** 추가할 블룸의 타입을 정한다. data, generic, index, temp 중 하나를 선택할 수 있다.
- **블룸 크기:** 추가할 블룸의 크기를 입력한다. 입력 단위는 Mbyte이다.
- **페이지 수:** 블룸 크기에서 입력한 Mbyte를 해당 데이터베이스 페이지 크기로 자동 환산하여 보여 준다.

## 데이터베이스 로드

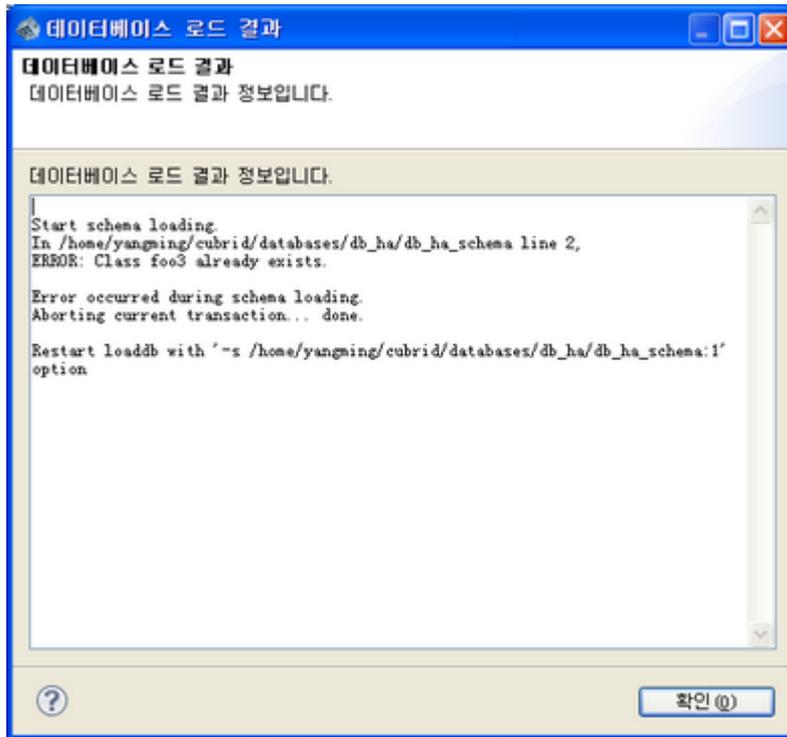
언로드한 데이터를 선택한 데이터베이스에 로드하려면, 해당 데이터베이스에 로그인한 후 다음 중 하나를 실행한다.

- 툴바에서 [데이터베이스 로드 - 해당 데이터베이스를 마우스 오른쪽 버튼 클릭하고 [데이터베이스 로드]를 선택한다.
- 메뉴에서 [동작] > [데이터베이스 로드]를 선택한다.

데이터베이스 로드 작업은 데이터베이스 서버가 중단된 상태에서만 수행할 수 있으므로 구동 중에는 [데이터베이스 복구] 메뉴가 비활성화된다.



- 데이터베이스 정보:** 언로드한 데이터를 로드할 대상 데이터베이스의 이름과 데이터베이스에 로그인한 사용자 권한이 제시된다. [사용자 이름]에는 현재 로그인한 사용자가 아닌 다른 사용자를 입력할 수 없다.
- 언로드 파일:** 언로드된 데이터의 정보를 입력한다. 리스트에서 선택하거나 언로드된 파일이 위치하는 경로를 직접 입력할 수 있다.  
 CUBRID 매니저 클라이언트와 CUBRID 매니저 서버가 다른 기기에서 구동된다면, 언로드된 스키마, 데이터, 인덱스, 트리거 파일은 CUBRID 매니저 서버 기기에 존재해야 한다.  
 기존에 CUBRID 매니저를 통해서 언로드한 정보가 있으면, 언로드 정보를 이용하여 로드할 수 있고, 없다면 기존에 언로드한 정보를 직접 입력하여 로드한다.
- 로드 옵션:** 문법 검사 후 데이터를 로드할지, 문법 검사 없이 데이터를 로드할지, 문법 검사만 수행할지 등을 선택할 수 있다. 자세한 내용은 CUBRID 온라인 매뉴얼의 "데이터베이스 가져오기"를 참조한다.  
 [데이터베이스 로드] 대화 상자에서 옵션을 선택한 후, [확인] 버튼을 클릭하면 데이터베이스 로드 작업의 수행 결과를 확인할 수 있다.

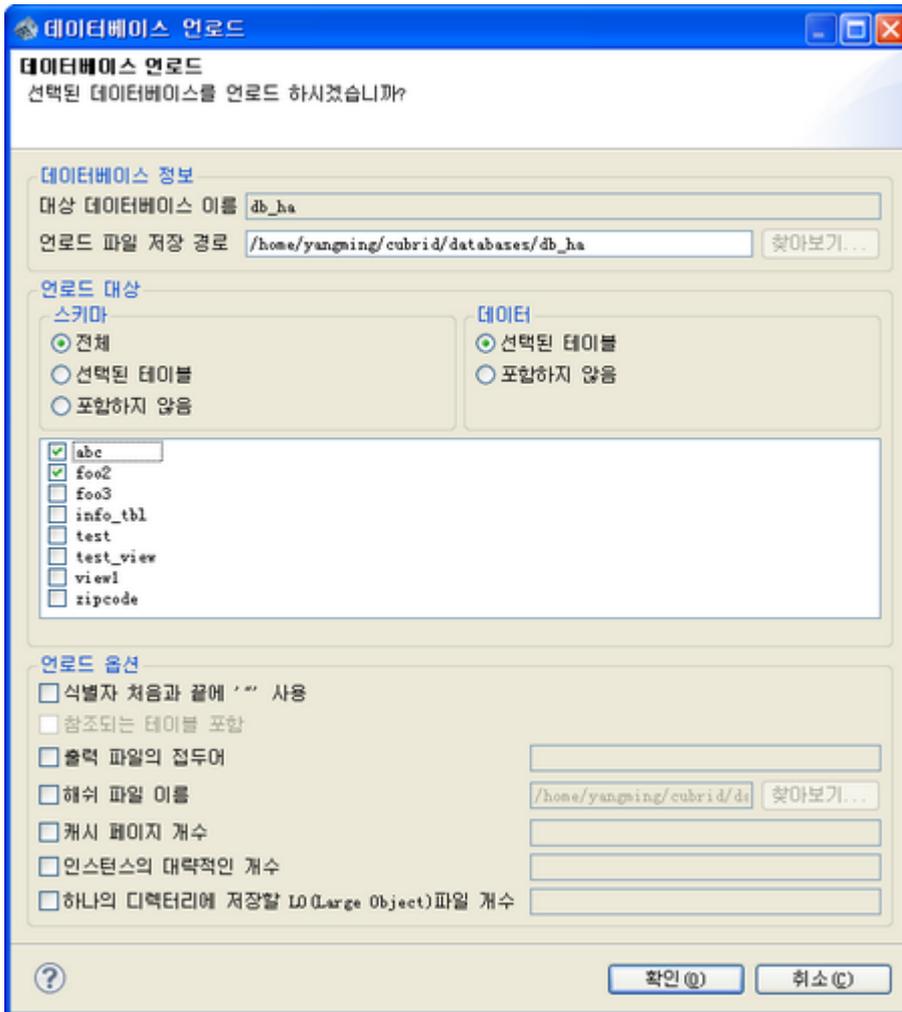


## 데이터베이스 언로드

선택한 데이터베이스의 데이터를 언로드하려면, 해당 데이터베이스에 로그인한 후 다음 중 하나를 실행한다.

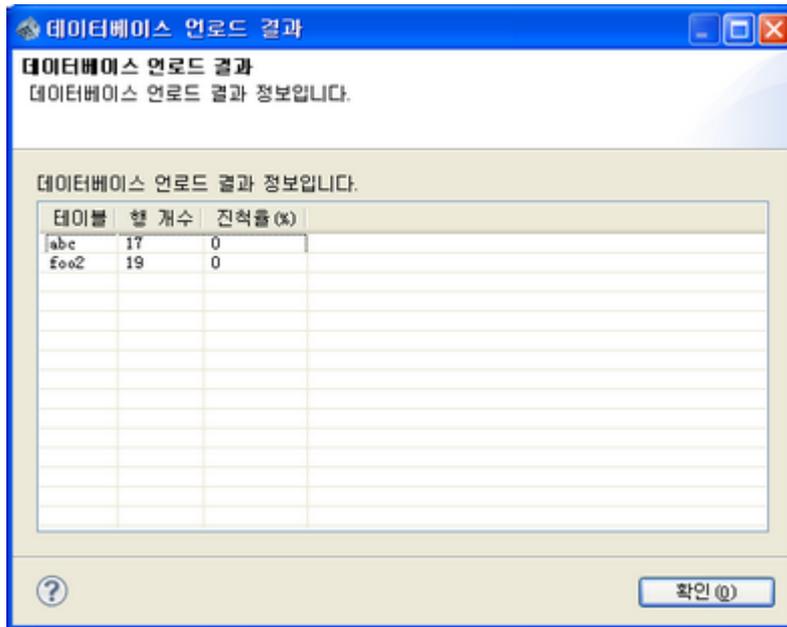
- 툴바에서 [데이터베이스 언로드 - 해당 데이터베이스를 마우스 오른쪽 버튼 클릭하고 [데이터베이스 언로드]를 선택한다.
- 메뉴에서 [동작] > [데이터베이스 언로드]를 선택한다.

데이터베이스 언로드 작업은 데이터베이스 서버가 구동 중일 때에도 수행할 수 있다.



- **데이터베이스 정보:** 언로드할 데이터베이스 이름이 출력되며, 언로드 작업 후 생성될 파일이 저장될 디렉터리를 지정한다.
- **언로드 대상:** 언로드할 데이터베이스의 스키마와 데이터를 선택한다.
- **언로드 옵션:** 식별자를 문자열 처리(" ")할지, 한 디렉터리에 LO 파일을 몇 개씩 저장할지 등을 선택할 수 있다. 자세한 내용은 CUBRID 온라인 매뉴얼의 "데이터베이스 내보내기"를 참조한다.

[데이터베이스 언로드] 대화 상자에서 옵션을 선택한 후, [확인] 버튼을 클릭하면 데이터베이스 언로드 작업의 수행 결과를 확인할 수 있다.



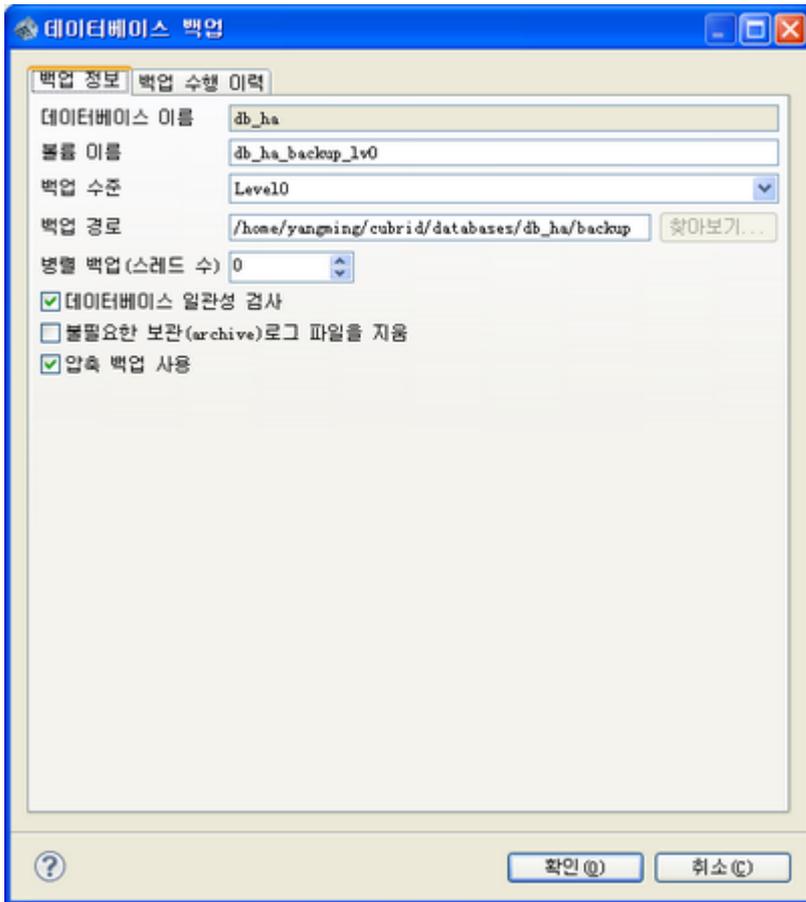
## 데이터베이스 백업

데이터베이스의 데이터를 백업하려면, 해당 데이터베이스에 로그인한 후 다음 중 하나를 실행한다.

- 툴바에서 [데이터베이스 백업]  을 클릭한다.
- 해당 데이터베이스를 마우스 오른쪽 버튼 클릭하고 [데이터베이스 백업]을 선택한다.
- 메뉴에서 [동작] > [데이터베이스 백업]을 선택한다.

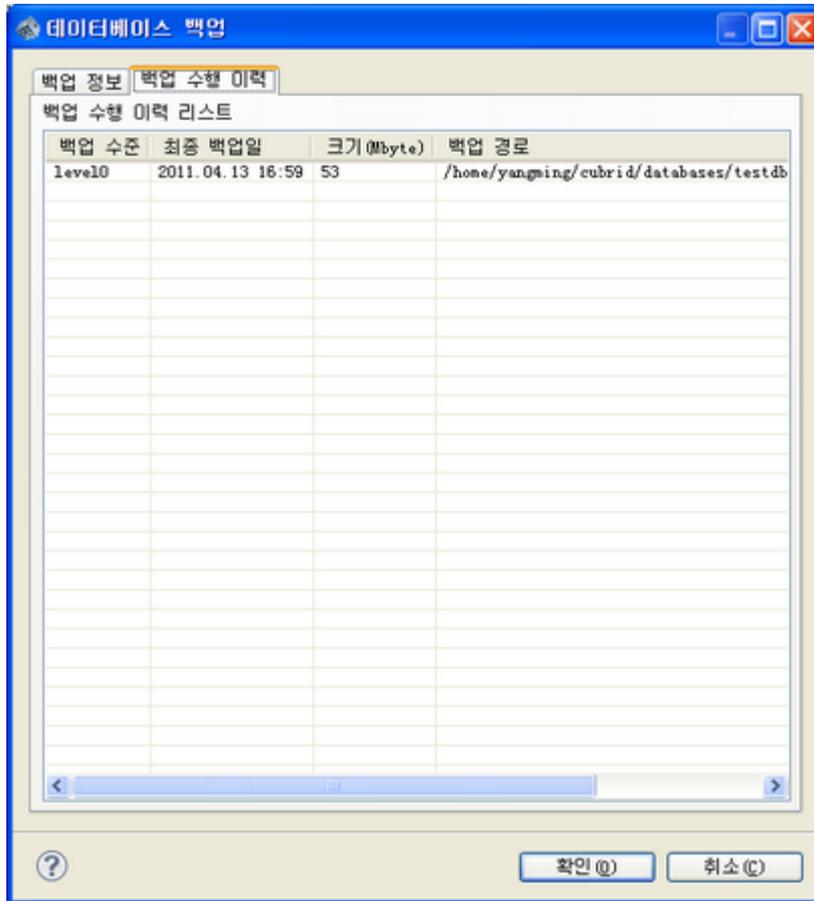
백업 작업은 데이터베이스 서버가 구동 중일 때에도 수행할 수 있다

데이터베이스 백업에 대한 자세한 사항은 CUBRID 온라인 매뉴얼의 "데이터베이스 백업"을 참고한다.



- **데이터베이스 이름:** 백업할 데이터베이스의 이름이다.
- **볼륨 이름:** 백업 볼륨의 이름을 지정한다. 기본값은 "데이터베이스 이름\_backup\_백업 레벨"의 형식이지만, 사용자가 수정할 수 있다.
- **백업 수준:** 백업 수준을 선택한다. Level0은 전체 백업을 의미하고, Level1은 기존에 Level0으로 백업이 수행된 경우, 그 이후에 변경된 부분에 대해서만 백업을 수행하는 것이다. Level2는 기존에 Level1로 백업이 수행된 경우, 그 이후에 변경된 부분에 대해서만 백업을 수행하는 것이다.  
예를 들어, 기존 백업 정보에 Level0 백업 레벨로 백업된 볼륨이 표시된다면, 관리자는 Level0 또는 Level1만 선택할 수 있다. 또한 기존에 백업이 수행된 적이 없다면 관리자는 Level0만 선택할 수 있다.
- **백업 경로:** 백업 파일이 저장될 디렉터리 경로를 지정한다. 기본값은 **\$CUBRID/databases/[DBNAME]/backup** 디렉터리로 지정된다.
- **데이터베이스 일관성 검사:** 백업할 데이터베이스의 일관성 검사를 먼저 수행한 후 데이터베이스를 백업한다. 선택하는 것을 권장한다.
- **불필요한 보관 로그 파일을 지움:** 데이터베이스를 복구할 때 필요하지 않은 보관 로그 파일들을 삭제한다. 백업 파일의 오류로 데이터베이스 복구가 원활하게 진행되지 않을 경우, 보관 로그 파일이 필요할 수 있으므로 주의한다. 해당 데이터베이스 서버가 복제 환경에서 마스터 서버로 설정되어 있는 경우에 이 옵션을 선택하면, 자동으로 복제 로그 유지 옵션이 체크된다.
- **병렬 백업(스레드 수):** 데이터베이스 백업을 수행할 때 동시에 이용할 스레드 수를 설정한다. 최대값은 CPU 개수만큼 설정하는 것이 적절하다. 기본값은 0이며, 이 경우 자동으로 스레드 수가 결정된다.
- **압축 백업 사용:** 데이터베이스 백업을 수행할 때 압축을 사용한다. 선택하는 것을 권장한다.

대상 데이터베이스에 대해 이전에 백업을 수행한 적이 있다면, 백업 수행 이력을 [백업 수행 이력] 탭에서 확인할 수 있다. 백업 수준, 최종 백업일, 백업 파일의 크기, 백업 파일의 경로에 관한 정보를 확인할 수 있으며, 현재 데이터베이스 볼륨이 위치한 디스크의 여유 공간도 확인할 수 있다.



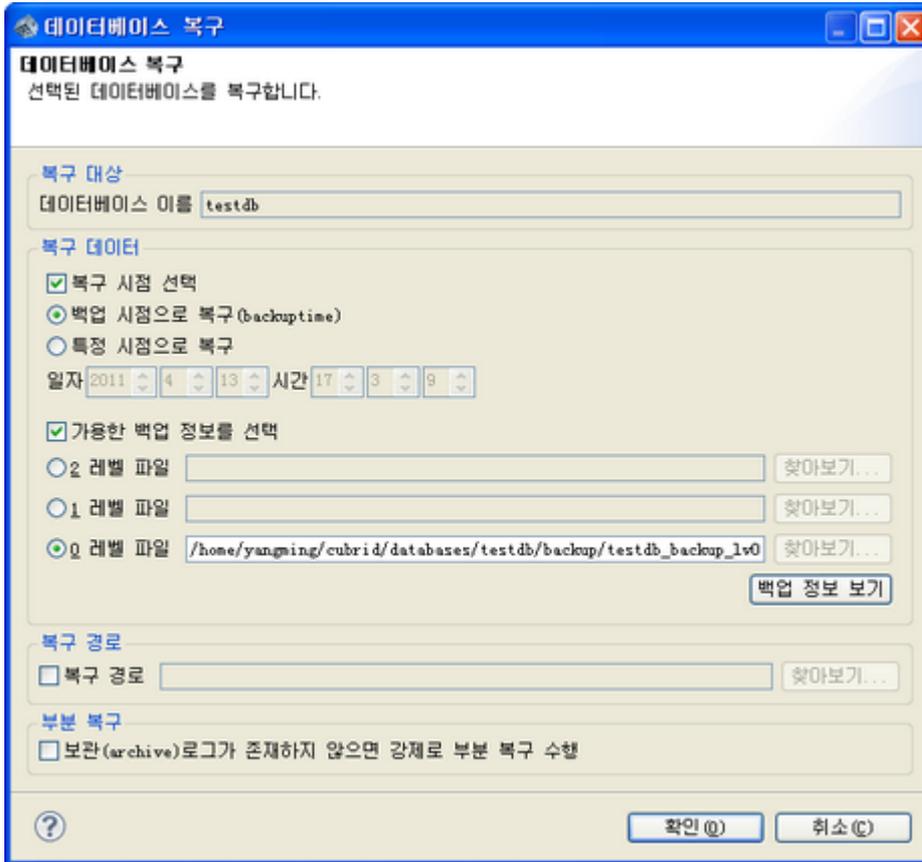
## 데이터베이스 복구

데이터베이스의 데이터를 복구하려면, 해당 데이터베이스에 로그인한 후 다음 중 하나를 실행한다.

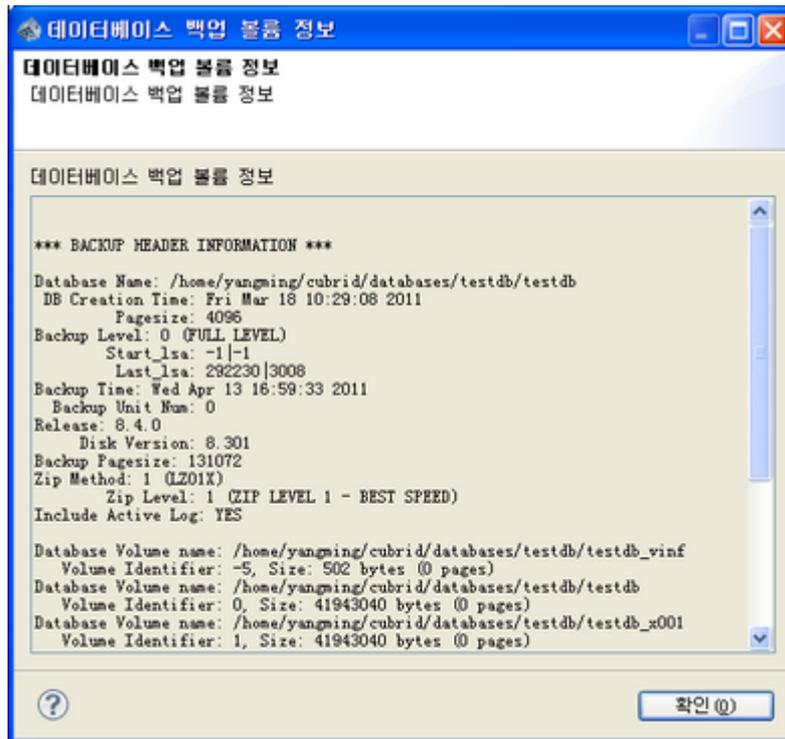
- 툴바에서 [데이터베이스 복구] 아이콘을 클릭한다.
- 해당 데이터베이스를 마우스 오른쪽 버튼 클릭하고 [데이터베이스 복구]를 선택한다.
- 메뉴에서 [동작] > [데이터베이스 복구]를 선택한다.

데이터베이스 로드 작업은 데이터베이스 서버가 중단된 상태에서만 수행할 수 있으므로 구동 중에는 [데이터베이스 복구] 메뉴가 비활성화된다.

데이터베이스 복구에 대한 자세한 사항은 CUBRID 온라인 매뉴얼의 "데이터베이스 복구"를 참조한다.



- **데이터베이스 이름:** 복구할 대상이 되는 데이터베이스 이름이다.
- **복구 일자 및 시간:** 어느 시점으로 데이터베이스를 복구할지 지정한다. [백업 시점으로 복구]를 선택하면 복구 명령어에 **backuptime** 키워드를 사용하여 복구하며, 이는 백업이 완료된 시점에서의 복구를 의미한다. [특정 시점으로 복구]를 선택하면 사용자가 원하는 날짜와 시간을 입력할 수 있다.
- **가용한 백업 정보:** 복구할 대상이 되는 데이터베이스가 어떤 백업 레벨로 백업되었는지 확인하여, 원하는 복구 레벨을 선택할 수 있다. 표시되는 파일 경로는 해당 레벨로 백업된 파일이 위치하는 디렉터리 경로이다.
- **보관 로그가 존재하지 않으면 강제로 부분 복구 수행:** 완전한 로그 정보가 없으면 부분 복구를 수행한다. 즉, 데이터베이스를 복구할 때, 백업 시점 이후에 생성된 보관 로그나 활동 로그가 없어도 복구를 수행할 수 있다.
- **사용자가 지정한 경로에 복구:** 데이터베이스 위치 정보 파일(**databases.txt**)에 지정된 경로로 대상 데이터베이스를 복구한다.
- **백업 정보 보기:** 선택한 백업 수준으로 백업된 파일의 정보를 보여준다.



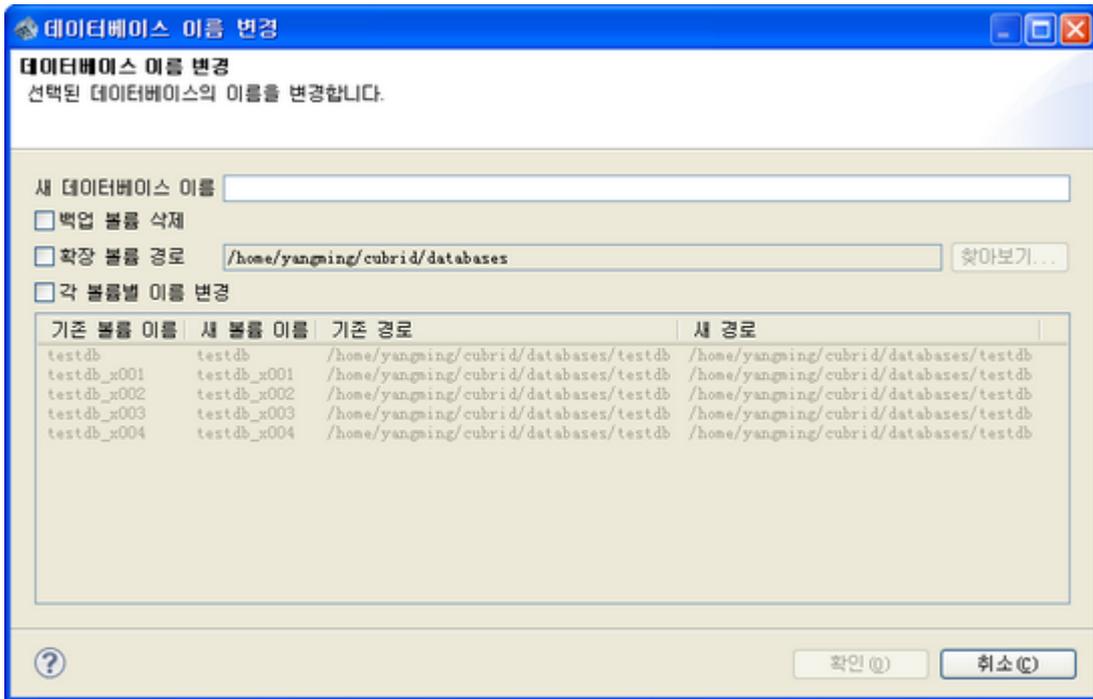
## 데이터베이스 이름 변경

데이터베이스의 이름을 변경하려면, 해당 데이터베이스에 로그인한 후 다음 중 하나를 실행한다.

- 해당 데이터베이스를 마우스 오른쪽 버튼 클릭하고 [데이터베이스 이름 변경]을 선택한다.
- 메뉴에서 [동작] > [데이터베이스 이름 변경]을 선택한다.

데이터베이스 이름 변경 작업은 데이터베이스 서버가 중단된 상태에서만 수행할 수 있으므로 구동 중에는 [데이터베이스 이름 변경] 메뉴가 비활성화된다.

데이터베이스 이름 변경에 대한 자세한 사항은 CUBRID 온라인 매뉴얼의 "데이터베이스 변경"을 참조한다.



- **새 데이터베이스 이름:** 이름을 변경하고자 하는 데이터베이스의 새 이름을 입력한다.
- **백업 볼륨 삭제:** 이름 변경 전의 데이터베이스의 백업 볼륨을 삭제한다.
- **확장 볼륨 경로:** 새 이름의 데이터베이스에서 추가될 볼륨들이 저장되는 경로를 지정하는 옵션이다.
- **각 볼륨별 이름 변경:** 변경 전의 데이터베이스 볼륨이 여러 개인 경우, 각각의 데이터베이스 볼륨의 이름을 변경하고, 각 데이터베이스 볼륨이 저장되는 새로운 디렉터리 경로를 지정할 수 있다.

## 데이터베이스 복사

데이터베이스를 복사하려면, 해당 데이터베이스에 로그인한 후 다음 중 하나를 실행한다.

- 해당 데이터베이스를 마우스 오른쪽 버튼 클릭하고 [데이터베이스 복사]를 선택한다.
- 메뉴에서 [동작] > [데이터베이스 복사]를 선택한다.

데이터베이스 복사 작업은 데이터베이스 서버가 중단된 상태에서만 수행할 수 있으므로 구동 중에는 [데이터베이스 복사] 메뉴가 비활성화된다.



- **원본 데이터베이스:** 복사할 데이터베이스의 이름, 데이터베이스 볼륨이 저장된 경로 및 로그 파일이 저장된 경로가 표시된다.
- **대상 데이터베이스:** 복사하면 생성될 데이터베이스의 이름, 데이터베이스 볼륨이 저장될 경로, 확장 볼륨이 저장될 경로 및 로그 파일이 저장될 경로를 입력한다.
- **개별 볼륨 복사 설정:** 원본 데이터베이스의 볼륨이 여러 개인 경우, 각 데이터베이스 볼륨의 이름을 변경하고, 각 데이터베이스 볼륨이 복사될 새로운 디렉터리 경로를 지정할 수 있다.
- **동일한 파일 덮어쓰기:** 대상 데이터베이스와 이름이 동일한 데이터베이스가 이미 존재하면 덮어쓴다.
- **원본 데이터베이스 삭제:** 데이터베이스 복사 후, 원본 데이터베이스를 삭제한다.

## 데이터베이스 최적화

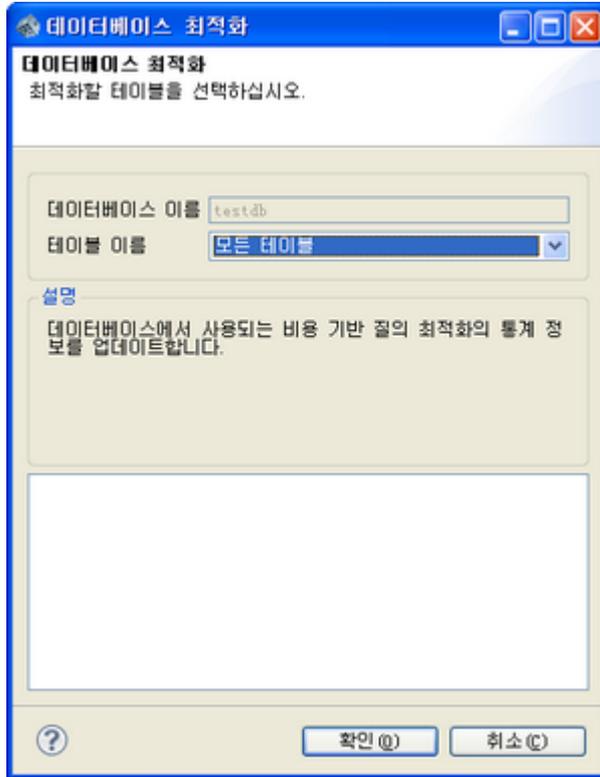
데이터베이스를 최적화하려면, 해당 데이터베이스에 로그인한 후 다음 중 하나를 실행한다.

- 툴바에서 [데이터베이스 최적화 93

- 메뉴에서 [동작] > [데이터베이스 최적화]를 선택한다.

[데이터베이스 최적화] 대화 상자에서 모든 테이블 또는 일부 테이블에 대해 최적화를 수행할 수 있다.

최적화할 대상을 선택하고 [확인]을 클릭하면 데이터베이스 최적화가 진행된다.

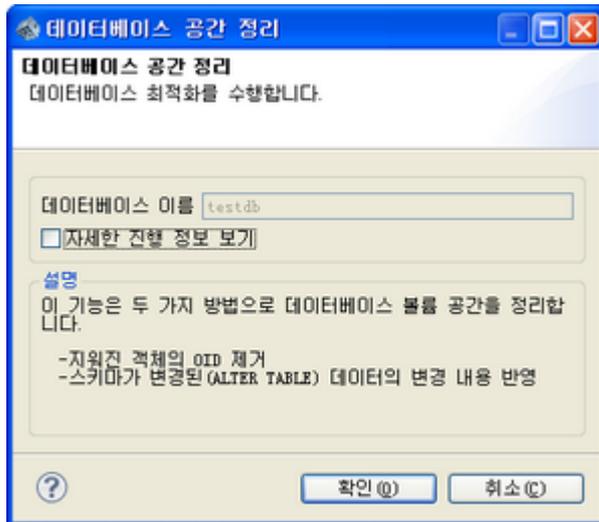


## 데이터베이스 공간 정리

데이터베이스의 공간을 정리하려면, 해당 데이터베이스에 로그인한 후 다음 중 하나를 실행한다.

- 해당 데이터베이스를 마우스 오른쪽 버튼 클릭하고 [데이터베이스 공간 정리]를 선택한다.
- 메뉴에서 [동작] > [데이터베이스 공간 정리]를 선택한다.

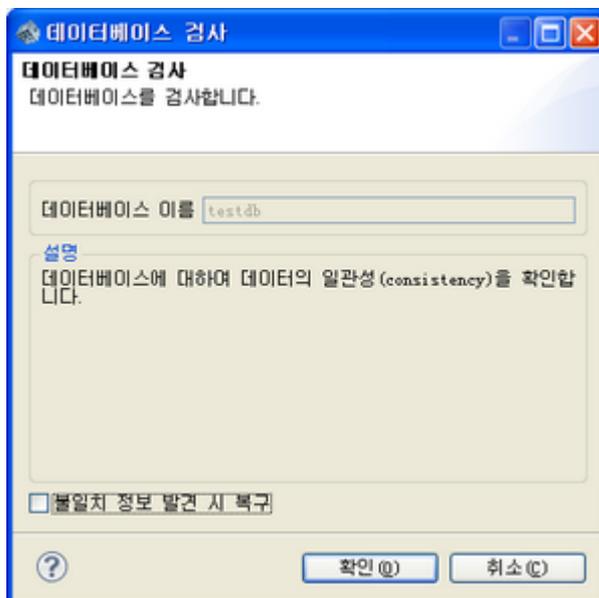
데이터베이스 공간 정리 작업은 데이터베이스 서버가 중단된 상태에서만 수행할 수 있으므로 구동 중에는 [데이터베이스 공간 정리] 메뉴가 비활성화된다. [자세한 진행 정보 보기]를 선택하면, 공간 정리 작업 완료 후 작업에 대한 자세한 정보가 출력된다.



## 데이터베이스 검사

데이터베이스를 검사하려면, 해당 데이터베이스에 로그인한 후 다음 중 하나를 실행한다.

- 툴바에서 [데이터베이스 검사] 를 클릭한다.
- 해당 데이터베이스를 마우스 오른쪽 버튼 클릭하고 [데이터베이스 검사]를 선택한다.
- 메뉴에서 [동작] > [데이터베이스 검사]를 선택한다.



## 데이터베이스 잠금 정보

데이터베이스의 잠금 정보를 보려면, 해당 데이터베이스에 로그인한 후 다음 중 하나를 실행한다.

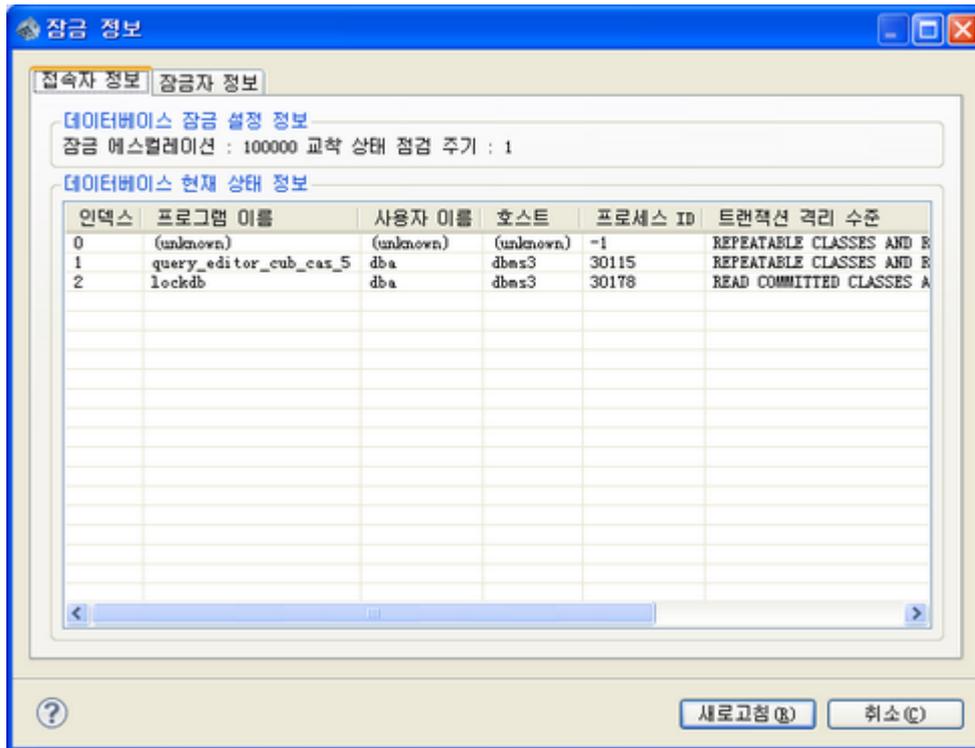
- 툴바에서 [데이터베이스 잠금 정보] 를 클릭한다.
- 해당 데이터베이스를 마우스 오른쪽 버튼 클릭하고 [데이터베이스 잠금 정보]를 선택한다.

- 메뉴에서 [동작] > [데이터베이스 잠금 정보]를 선택한다.

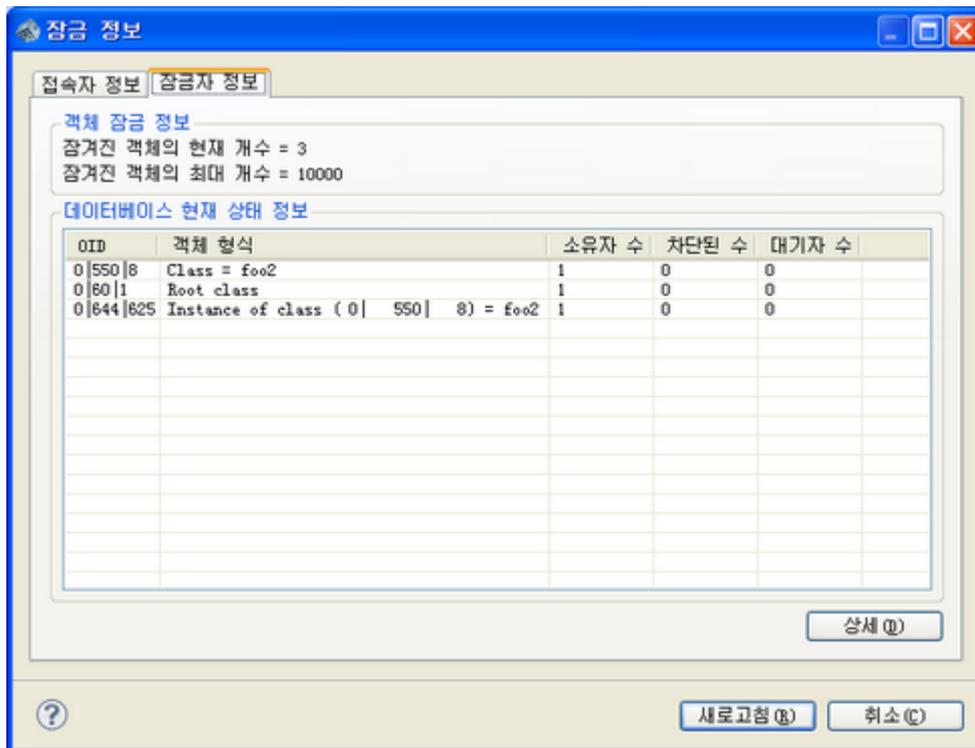
이 기능은 데이터베이스 서버가 구동 중일 때에만 수행할 수 있으며, 서버가 구동하지 않을 때에는 해당 메뉴가 비활성화된다.

[잠금 정보] 대화 상자는 두 개의 탭으로 이루어져 있다.

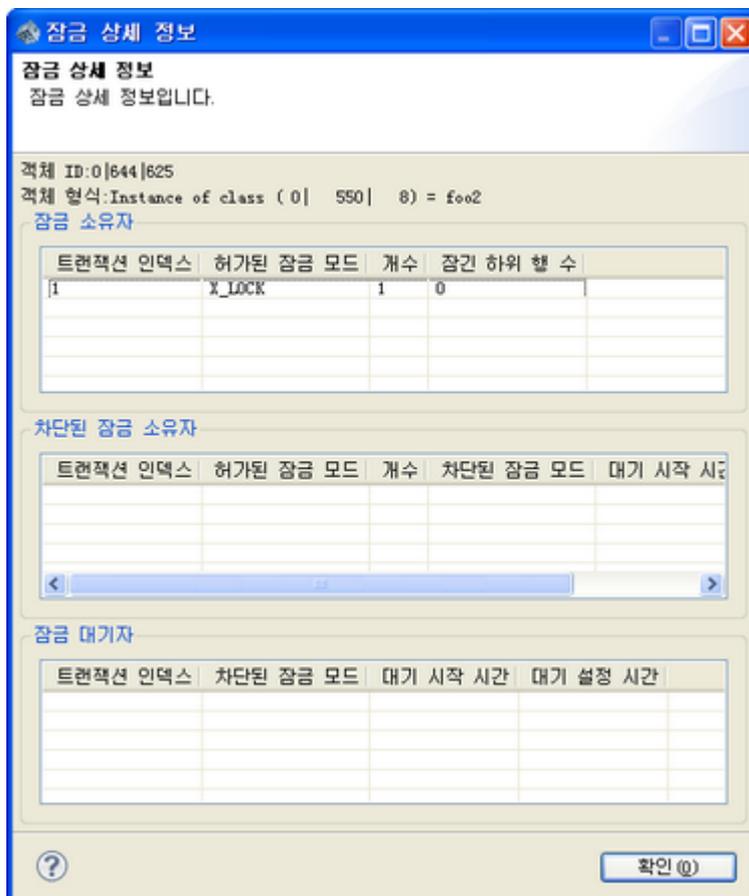
[접속자 정보] 탭은 현재 데이터베이스에 접속한 클라이언트의 정보를 제공한다.



[잠금자 정보] 탭에서는 데이터베이스 객체에 대한 잠금 정보를 확인할 수 있다.



[잠금자 정보] 탭에서 [상세]를 클릭하면 잠금에 대한 상세 정보를 볼 수 있다.

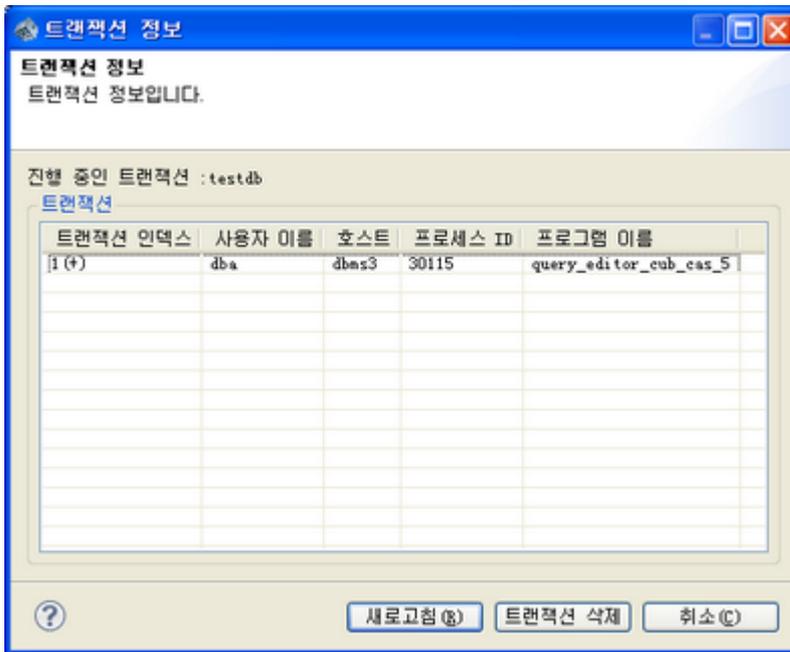


## 데이터베이스 트랜잭션 정보

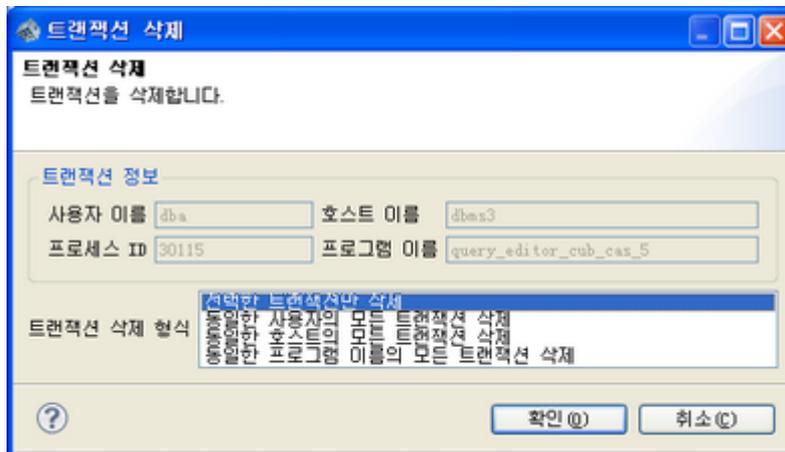
데이터베이스의 트랜잭션 정보를 보려면, 해당 데이터베이스에 로그인한 후 다음 중 하나를 실행한다.

- 툴바에서 [데이터베이스 트랜잭션 정보]를 클릭한다.
- 해당 데이터베이스를 마우스 오른쪽 버튼 클릭하고 [데이터베이스 트랜잭션 정보]를 선택한다.
- 메뉴에서 [동작] > [데이터베이스 트랜잭션 정보]를 선택한다.

이 기능은 데이터베이스 서버가 구동 중일 때에만 수행할 수 있으며, 서버가 구동하지 않을 때에는 해당 메뉴가 비활성화된다.



- **새로고침:** CUBRID 매니저 서버를 통해 현재 진행 중인 트랜잭션 정보를 다시 수집하여 보여 준다.
- **트랜잭션 삭제:** 트랜잭션 리스트에서 선택한 트랜잭션을 강제로 종료한다. 트랜잭션 삭제는 네 가지 옵션 중 하나를 선택할 수 있다.

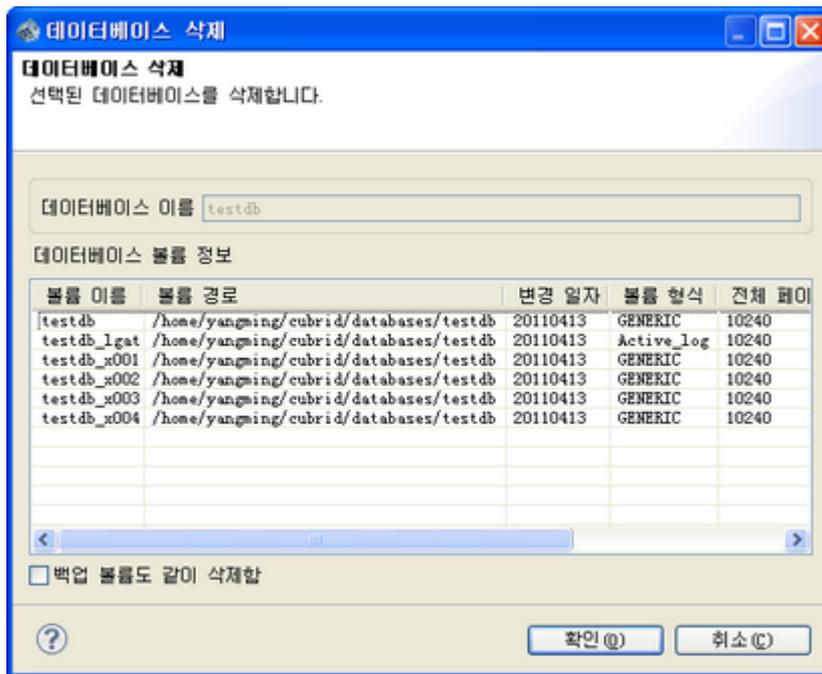


## 데이터베이스 삭제

데이터베이스의 데이터를 복구하려면, 해당 데이터베이스에 로그인한 후 다음 중 하나를 실행한다.

- 해당 데이터베이스를 마우스 오른쪽 버튼 클릭하고 [데이터베이스 삭제]를 선택한다.
- 메뉴에서 [동작] > [데이터베이스 삭제]를 선택한다.

데이터베이스 삭제 작업은 데이터베이스 서버가 중단된 상태에서만 수행할 수 있으므로 구동 중에는 [데이터베이스 삭제] 메뉴가 비활성화된다.

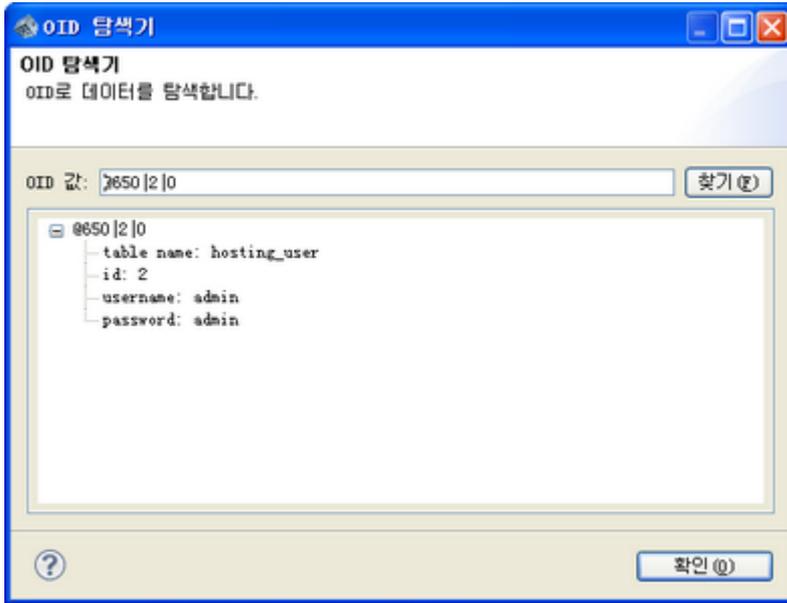


[확인]을 클릭하면 **dba** 비밀번호를 확인한 후 삭제한다. **dba** 이외의 사용자는 삭제할 수 없다.

## OID 탐색기

OID로 데이터를 탐색할 때 사용한다.

해당 데이터베이스 또는 질의 편집기의 결과 창에서 마우스 오른쪽 버튼 클릭하고 [OID 탐색기]를 선택하여 실행할 수 있다. 이 기능은 데이터베이스 서버가 구동 중일 때에만 수행할 수 있으며, 서버가 구동하지 않을 때에는 해당 메뉴가 비활성화된다.

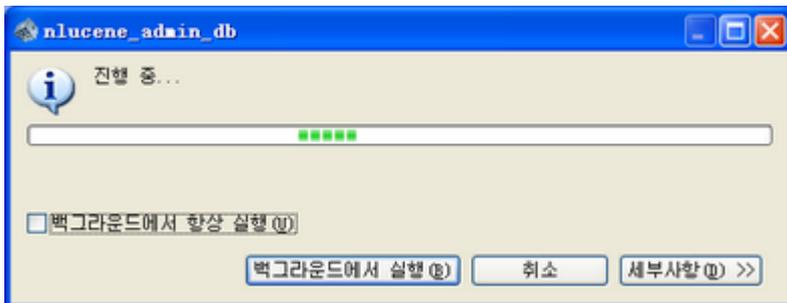


## 다중 호스트 환경에서의 백그라운드 운영

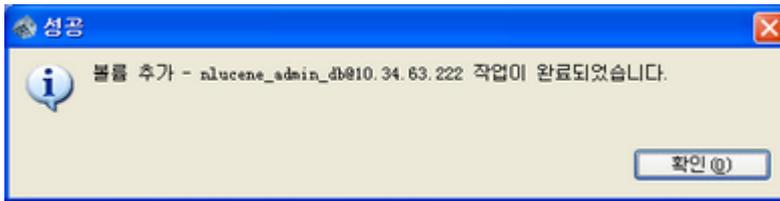
[호스트 추가]를 이용하여 여러 호스트의 데이터베이스를 관리하는 경우, 데이터베이스 운영 작업이 백그라운드에서 실행되도록 설정할 수 있다. 예를 들어, A 호스트의 데이터베이스를 백업하는 도중에 B 호스트의 데이터베이스에 질의를 실행하고자 한다면, 시간이 오래 걸리는 데이터베이스 백업 작업이 백그라운드에서 실행되도록 설정할 수 있다. 다음은 백그라운드에서 실행할 수 있는 데이터베이스 운영 작업이다.

- **데이터베이스 관련:** 데이터베이스 생성, 데이터베이스 언로드/로드, 데이터베이스 백업/복구, 데이터베이스 이름 변경, 데이터베이스 복사, 데이터베이스 최적화, 데이터베이스 공간 정리, 데이터베이스 검사
- **테이블 관련:** 데이터 올리기/내려 받기, 모든 데이터 삭제, 제약 조건을 **NULL**에서 **NOT NULL**로 변경하면서 데이터 업데이트
- **볼륨 관련:** 볼륨 추가

볼륨을 추가하면 다음과 같은 창이 나타난다. 이 창에서 [백그라운드에서 실행]을 선택할 수 있다.



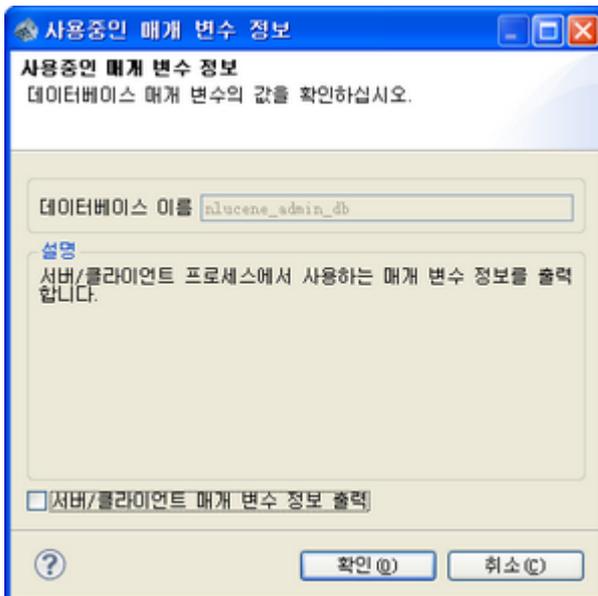
해당 운영 작업이 실행 완료되면, 다음과 같은 작업 완료 창이 나타난다.



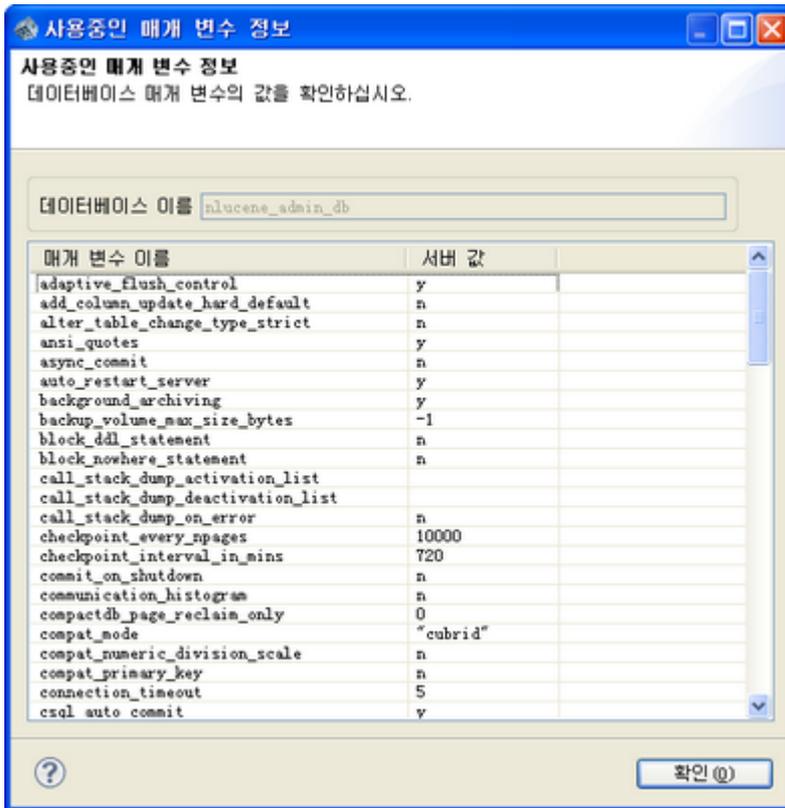
운영 작업이 백그라운드로 실행 중인 상태에서 관리자가 해당 데이터베이스를 종료하거나 해당 데이터베이스나 호스트로부터 로그아웃하면, 백그라운드로 수행 중인 작업이 있음을 알리는 창이 나타난다.

## 사용 중인 매개 변수 정보

해당 데이터베이스에서 사용하는 매개 변수(parameter) 정보를 출력한다. **cubrid paramdump** 유틸리티 실행 시 출력되는 항목을 매니저에서 확인할 수 있다. 유틸리티에 관한 자세한 내용은 CUBRID 온라인 매뉴얼의 "서버/클라이언트에서 사용하는 파라미터 출력"을 참고한다.



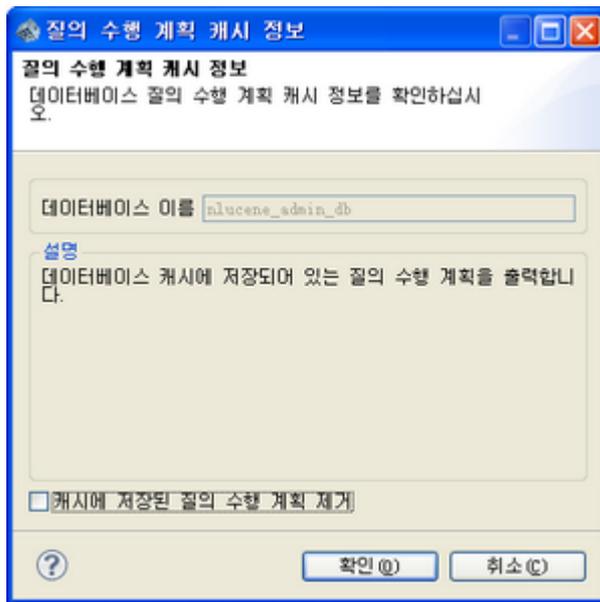
[서버/클라이언트 매개 변수 정보 출력]을 선택하면 서버에 적용된 파라미터 값과 클라이언트에 적용된 파라미터 값을 모두 확인할 수 있고, 선택을 해제하면 서버에 적용된 파라미터 값만 확인할 수 있다. [확인]을 클릭하면 아래와 같이 서버/클라이언트 파라미터 값을 확인할 수 있다.



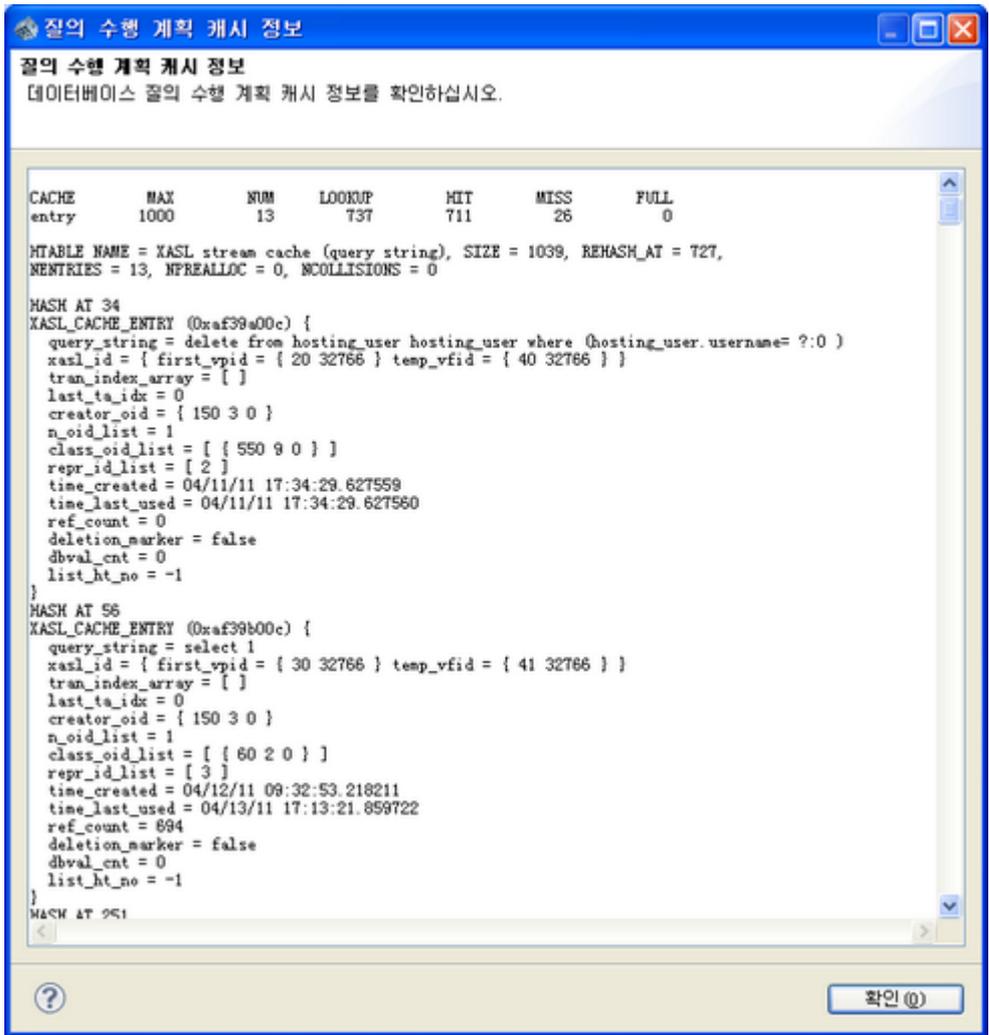
## 질의 수행 계획 캐시 정보

해당 데이터베이스 서버에 캐시 저장되어 있는 질의 수행 계획에 관한 정보를 출력한다. **cubrid plandump** 유틸리티 실행 시 출력되는 항목을 매니저에서 확인할 수 있다. 유틸리티에 관한 상세한 설명은 CUBRID 온라인 매뉴얼의 "질의 수행 계획 캐시 확인"을 참고한다.

이 기능은 데이터베이스 서버 구동 중에만 수행할 수 있으므로, 데이터베이스 서버가 구동 중일 때만 [질의 수행 계획 캐시 정보] 메뉴가 활성화된다.



[캐시에 저장된 질의 수행 계획 제거]를 선택하면, 수집한 데이터베이스 질의 수행 계획 캐시 정보를 출력한 후 이를 초기화한다. [확인] 버튼을 누르면 아래와 같이 캐시에 저장된 질의 수행 계획 정보를 확인할 수 있다.



## 브로커

### 브로커 구조

브로커는 다양한 인터페이스(ODBC, OLEDB, JDBC, PHP 등)가 데이터베이스에 접속할 수 있도록 연결해주는 연결자로서 다양한 기능을 가진다. 자세한 내용은 온라인 매뉴얼의 관리자 안내서를 참고한다.

브로커는 설정된 각 브로커 이름과 해당 브로커의 SQL 로그로 구성된다.



### 개별 브로커 기능

브로커는 다수의 개별 브로커로 구성될 수 있으며, 개별 브로커는 유일한 브로커 이름, 포트, 공유 메모리 ID로 설정해야 한다.



각 브로커의 상태 보기와 속성 편집이 가능하며, 해당 브로커를 시작/정지할 수 있다.

시작된 브로커는 탐색 트리에서  아이콘으로 표시되고, 정지 상태의 브로커는  아이콘으로 표시된다.

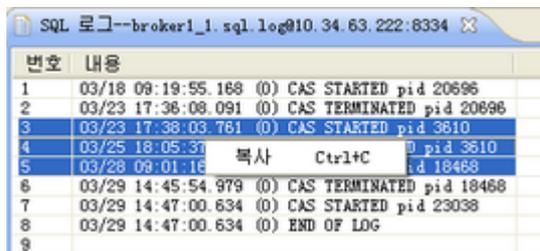
### SQL 로그

해당 브로커의 SQL\_LOG 매개 변수 값이 ON이면 해당 브로커를 통해서 수행된 모든 질의는 로그 파일에 기록된다. 이 로그 파일을 CUBRID 매니저에서 보고 분석 및 재실행해 볼 수 있다.



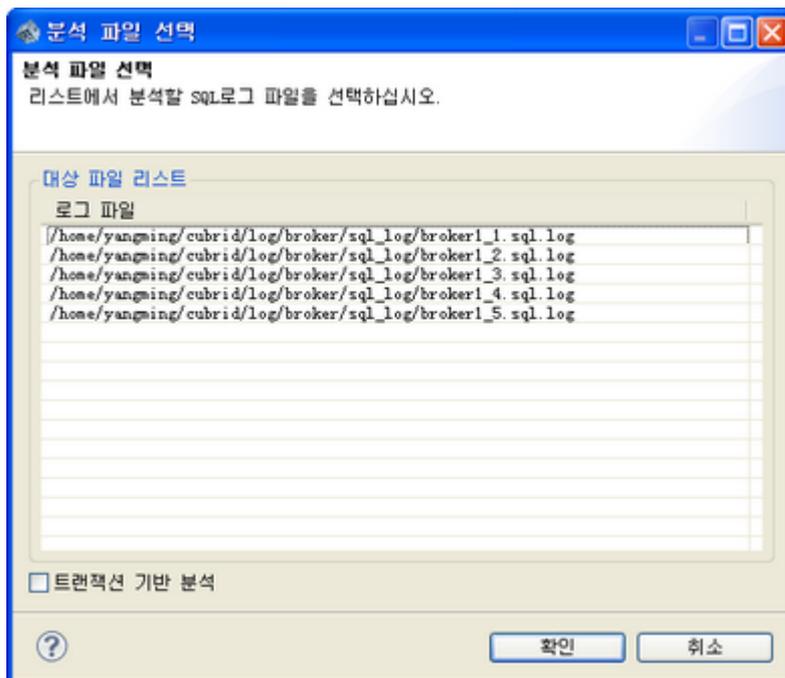
**로그 보기**

선택한 SQL 로그 파일에 기록된 SQL 로그를 100라인씩 읽어 와서 볼 수 있는 기능으로, 로그 정보 중 특정 범위를 선택하여 복사할 수 있는 기능을 제공한다.

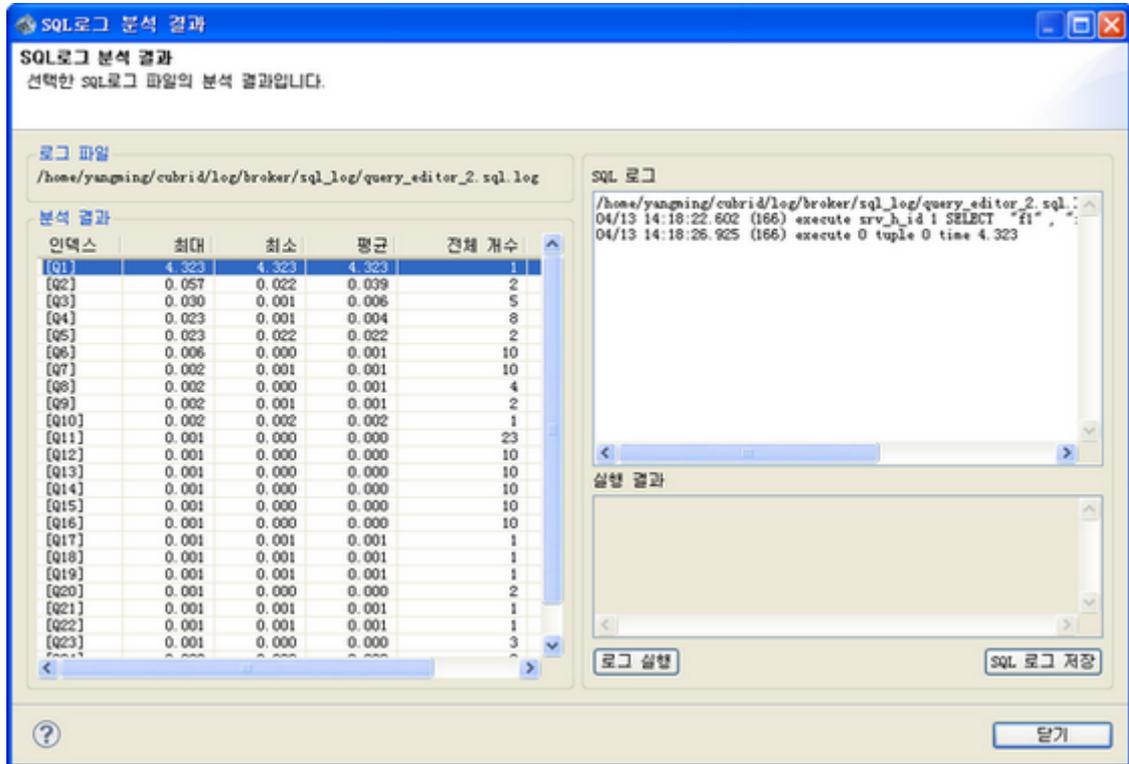


**로그 분석**

[로그 분석]을 선택하면, 어떤 브로커 SQL 로그를 분석할 것인지 선택할 수 있는 [분석 파일 선택] 대화 상자가 나타난다. [트랜잭션 기반 분석]을 선택하면 트랜잭션 단위로 로그 분석을 수행하고, 체크하지 않으면 각 질의마다 로그 분석을 수행한다.



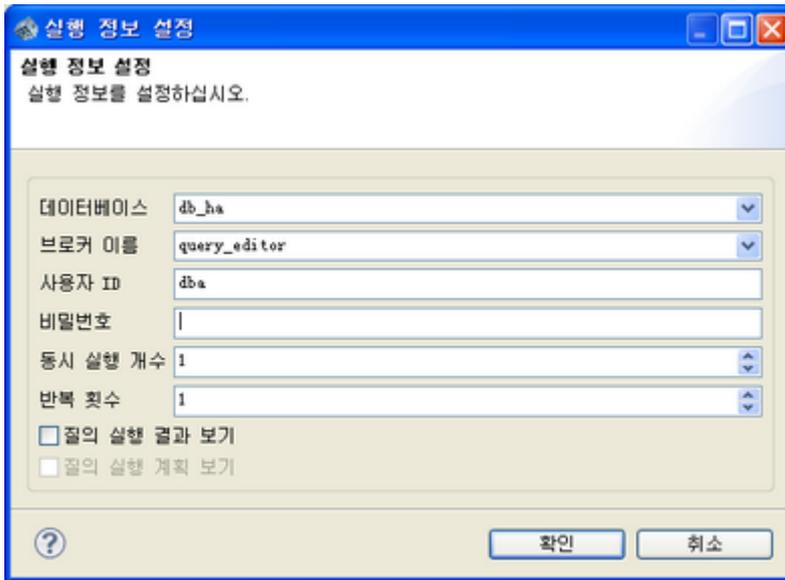
[분석 대상 파일 선택] 대화 상자에서 분석하고자 하는 SQL 로그 파일을 선택하고 [확인]을 클릭하면, 선택한 로그 파일을 분석한 결과를 보여주는 [SQL 로그 분석] 대화 상자가 나타난다.



- **로그 파일:** 분석 대상이 되는 브로커 SQL 로그 파일의 이름 및 디렉터리 경로를 표시한다.
- **분석 결과:** 로그 분석 결과를 보여준다. [트랜잭션 기반 분석]을 선택했다면 각 트랜잭션의 수행 시간이 표시되고, 선택하지 않았다면 각 질의별 분석 정보(전체 수행 횟수, 에러 횟수, 최고 수행 시간, 최저 수행 시간, 평균 수행 시간)가 표시된다. 결과 분석 목록의 컬럼을 클릭하면, 선택된 컬럼 값을 기준으로 분석 결과가 정렬된다.
- **SQL 로그:** 분석 결과에 관한 로그 내용을 보여준다.
- **실행 결과:** 로그 실행 결과를 보여준다.
- **로그 실행:** 로그 내용에 있는 SQL 로그를 재실행한다. 로그 질의를 변경하고 재실행하면서 질의를 튜닝하고 오류를 수정할 수 있다.
- **SQL 로그 저장:** 로그 내용을 파일로 저장한다.

### 로그 재실행

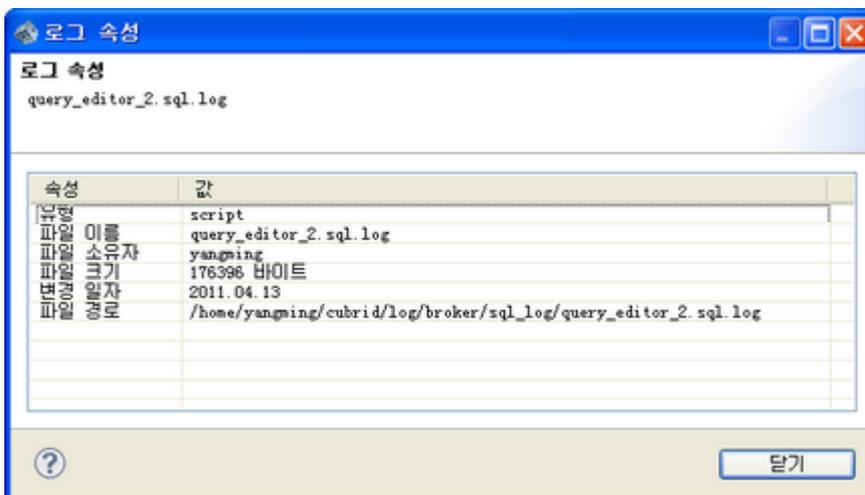
[로그 재실행] 메뉴를 선택하면, 다음과 같이 로그 재실행을 위한 환경을 설정할 수 있는 [재실행 정보] 대화 상자가 나타난다.



- **데이터베이스:** 로그를 재실행할 데이터베이스를 선택한다.
- **브로커 이름:** 로그를 재실행할 브로커를 선택한다.
- **사용자 ID/비밀번호:** 로그를 재실행할 데이터베이스의 사용자 ID와 비밀번호를 입력한다.
- **동시 실행 개수:** 동시에 실행할 로그 질의를 개수를 지정한다. 로그가 재실행될 때, 이 개수만큼의 스레드가 생성되어 같은 질의가 동시에 실행된다. 다중 사용자 환경에서 질의가 어떻게 실행되는지 확인할 때 유용한 기능이다.
- **반복 횟수:** 질의의 반복 실행 횟수를 지정한다.
- **질의 실행 결과 보기:** 질의 실행 결과를 확인할 수 있다.
- **질의 실행 계획 보기:** 이 옵션은 [질의 실행 결과 보기]를 선택한 경우에만 유효하다.

### 로그 속성

해당 로그 파일의 정보를 제공한다.



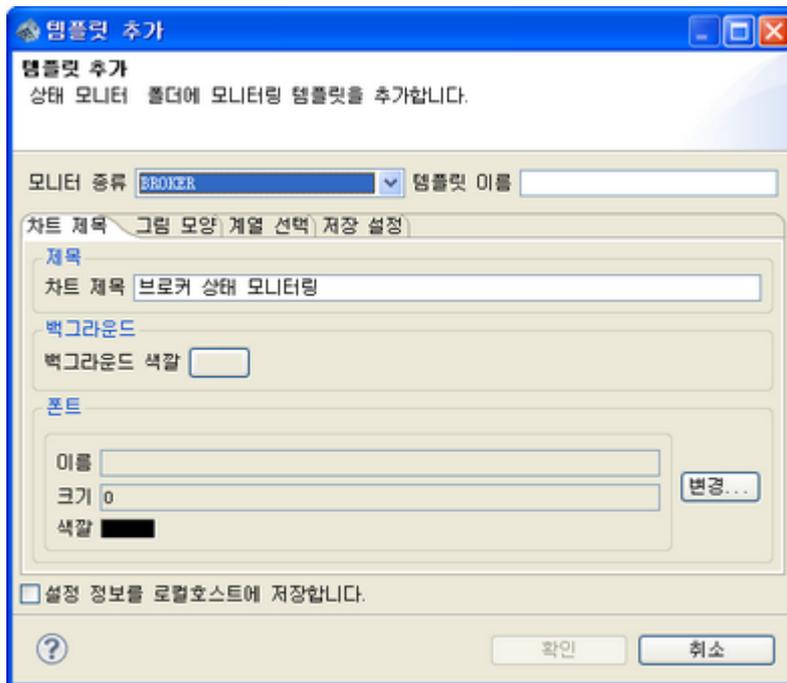
## 상태 모니터

상태 모니터는 다음과 같은 기능을 제공한다.

### 상태 모니터 추가

탐색 트리에서 [상태 모니터]를 마우스 오른쪽 버튼 클릭하고 [상태 모니터 추가]를 선택하면 아래와 같은 [템플릿 추가] 대화 상자가 나타나며, 모니터링 대상, 모니터링 항목 및 모니터링 차트 속성을 선택하여 모니터링 템플릿을 생성할 수 있다. 단, 3.0 미만 버전의 서버에 접속하는 경우 상태 모니터 추가창의 기능과 설정 방법이 다를 수 있다.

- [모니터 종류]: [DATABASE] 또는 [BROKER] 중 하나의 모니터링 대상을 선택할 수 있다.
- [템플릿 이름]: 생성할 상태 모니터링 템플릿의 이름을 입력한다.
- [차트 제목] 탭: 차트 상단에 출력될 제목, 백그라운드 색상, 폰트 등을 설정할 수 있다.
- [그림 모양] 탭: 차트 제목, 가로/세로 축 및 눈금선 등의 색상을 설정할 수 있다.
- [계열 선택] 탭: 모니터링 항목을 선택할 수 있다.



### 데이터베이스 모니터링 항목

[모니터 종류]로 [DATABASE]를 선택하면, [계열 선택] 탭에서 원하는 데이터베이스 모니터링 항목을 선택하여 템플릿에 추가할 수 있다.

데이터베이스 모니터링은 **cubrid statdump** 유틸리티를 이용해 CUBRID 데이터베이스 서버가 실행한 통계 정보를 수집하여 보여 주며, 데이터 페이지 버퍼 적중률을 비롯하여 로그, 페이지, 인덱스, 질의, 트랜잭션과

같은 항목을 모니터링한다. 데이터베이스 모니터링 항목에 관한 보다 자세한 내용은 CUBRID 온라인 매뉴얼의 "데이터베이스 서버 실행 통계 정보 출력"을 참고한다.

### 브로커 모니터링 항목

[모니터 종류]로 [BROKER]를 선택하면, [계열 선택] 탭에서 원하는 브로커 모니터링 항목을 선택하여 템플릿에 추가할 수 있다.

브로커 모니터링 항목은 SESSION, ACTIVE\_SESSION, ERR\_Q, LONG\_Q, LONG\_T, QPS, RPS, TPS가 있다. 자세한 내용은 CUBRID 온라인 매뉴얼의 "브로커 상태 확인"을 참고한다. 브로커의 ACTIVE\_SESSION을 제외한 나머지 모니터링을 위한 샘플링 간격은 1초이다.

- **SESSION:** 브로커에 현재 연결된 응용 서버(CAS) 개수를 의미한다. 즉, IDLE 상태가 아닌 응용 서버의 개수이다. 이 값은 브로커 파라미터 MAX\_NUM\_APPL\_SERVER의 설정 값을 초과할 수 없다.
- **ACTIVE\_SESSION:** 브로커에 연결된 응용 서버(CAS) 중 현재 BUSY 상태인 응용 서버의 개수를 의미한다. 즉, 현재 트랜잭션을 수행 중인 세션의 개수를 의미한다.
- **ERR\_Q:** 에러가 발생한 질의 수이다.
- **LONG\_Q:** 장기 수행된 질의 수이다.
- **LONG\_T:** 장기 수행된 트랜잭션 수이다.
- **QPS:** 브로커에서 처리한 초당 질의 수이다.
- **RPS:** 브로커에 들어온 초당 요청 횟수이다.
- **TPS:** 브로커에서 처리한 초당 트랜잭션 수이다.

### 템플릿 정보

- **이름:** 새롭게 생성할 상태 모니터링 템플릿의 이름을 입력한다.
- **설명:** 생성할 템플릿에 관한 설명을 입력한다.
- **샘플링 간격(초):** 대상 객체를 모니터링할 주기를 초 단위로 지정한다.
- **대상 데이터베이스:** 모니터링 대상 객체가 데이터베이스 서버와 관련된 경우 활성화되며, 대상 데이터베이스를 선택한다.

### 상태 모니터 편집

편집하려는 상태 모니터 템플릿을 마우스 오른쪽 버튼 클릭하고 [상태 모니터 편집]을 선택하면, [상태 모니터 추가]와 동일한 인터페이스에서 편집할 수 있다.

### 상태 모니터 삭제

등록된 상태 모니터 템플릿을 삭제하려면, 해당 상태 모니터 템플릿을 마우스 오른쪽 버튼 클릭하고 [상태 모니터 삭제]를 선택한다.

## 상태 모니터 수행

수행하려는 상태 모니터 템플릿을 마우스 오른쪽 버튼 클릭하고 [상태 모니터 수행]을 선택하면, 선택한 템플릿에 따라 데이터베이스 서버와 브로커의 상태를 모니터링할 수 있는 상태 모니터링 창이 나타난다. 상태 모니터는 각 수집 항목별로 별도의 차트를 보여 주며, 현재 값, 최소 값, 최대 값 그리고 평균 값으로 구성되어 있다.

상태 모니터는 별도의 뷰로 구성되어 있어 다른 뷰 인터페이스와 동시에 볼 수 있으며, CUBRID 매니저 외부로 끌어내어 볼 수도 있으므로, 다중 모니터 환경에서 작업과 상태 모니터링을 동시에 할 때 편리하다.

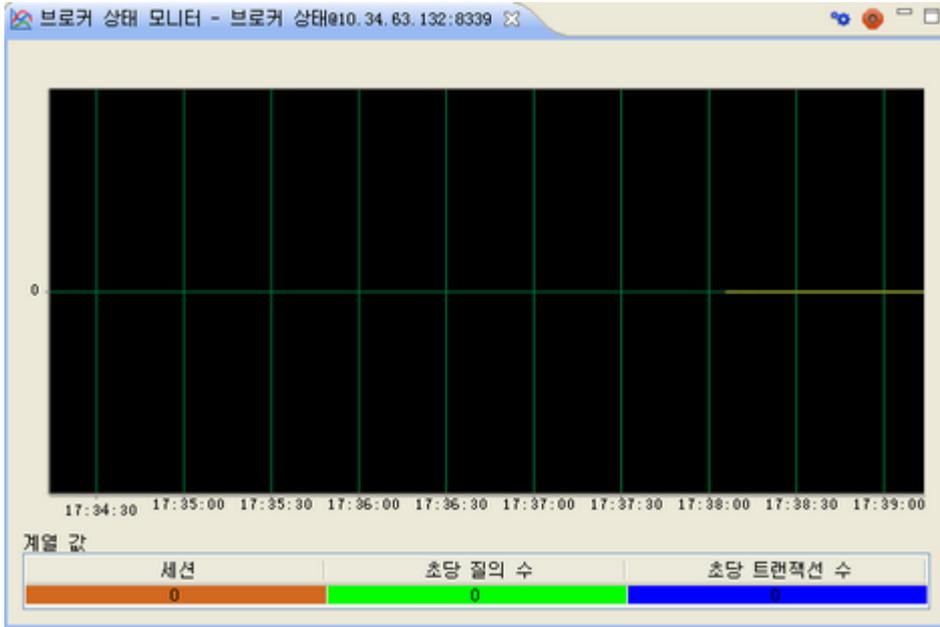


## 통합 상태 모니터

통합 상태 모니터링은 브로커 및 데이터베이스 모니터링 항목을 하나의 차트로 통합하여 확인할 수 있는 기능이다. CUBRID 2008 R2.2 이상 버전에서 지원한다.

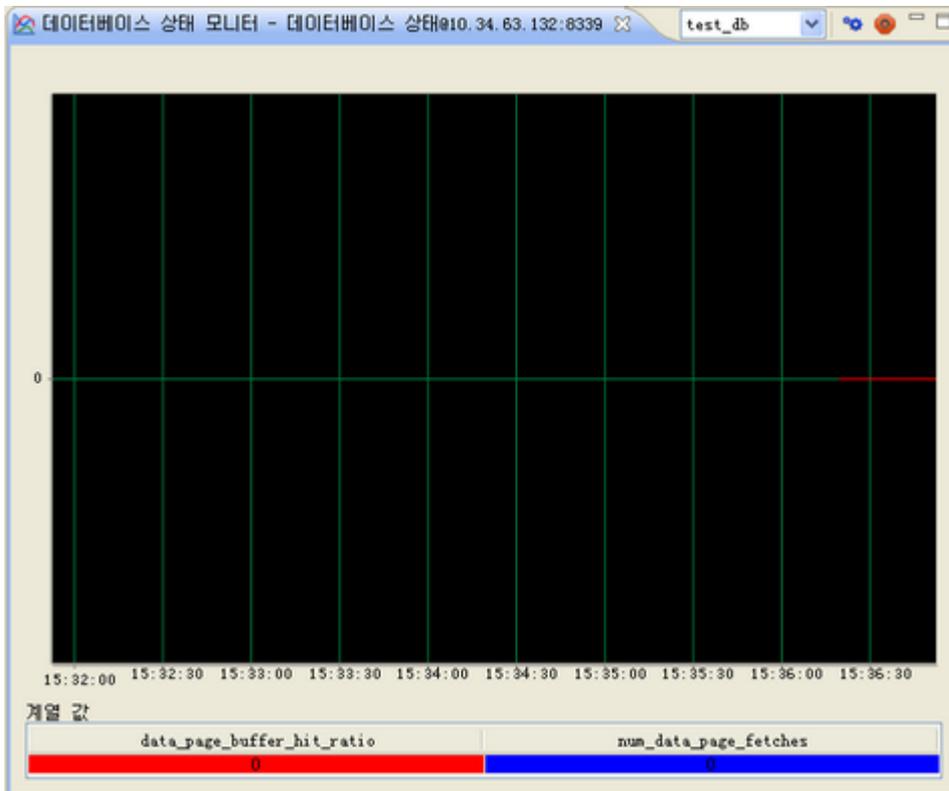
### 브로커 상태 모니터

하나의 차트에서 브로커 관련 항목을 통합 모니터링할 수 있다. 브로커 상태 모니터링 창의 상단 바에 있는 [설정 



### 데이터베이스 상태 모니터

하나의 차트에서 데이터베이스 관련 항목을 통합 모니터링할 수 있다. 모니터링할 수 있는 항목은 **cubrid statdump** 유틸리티 실행 시 출력되는 항목과 같다. 데이터베이스 상태 모니터링 창의 상단 바에 있는 [설정]을 클릭해서 모니터링 항목과 차트 속성을 설정할 수 있다.



## 차트 속성 설정

차트 설정 마법사를 이용해서 차트 제목, 그림 모양, 모니터링 항목을 설정할 수 있다.

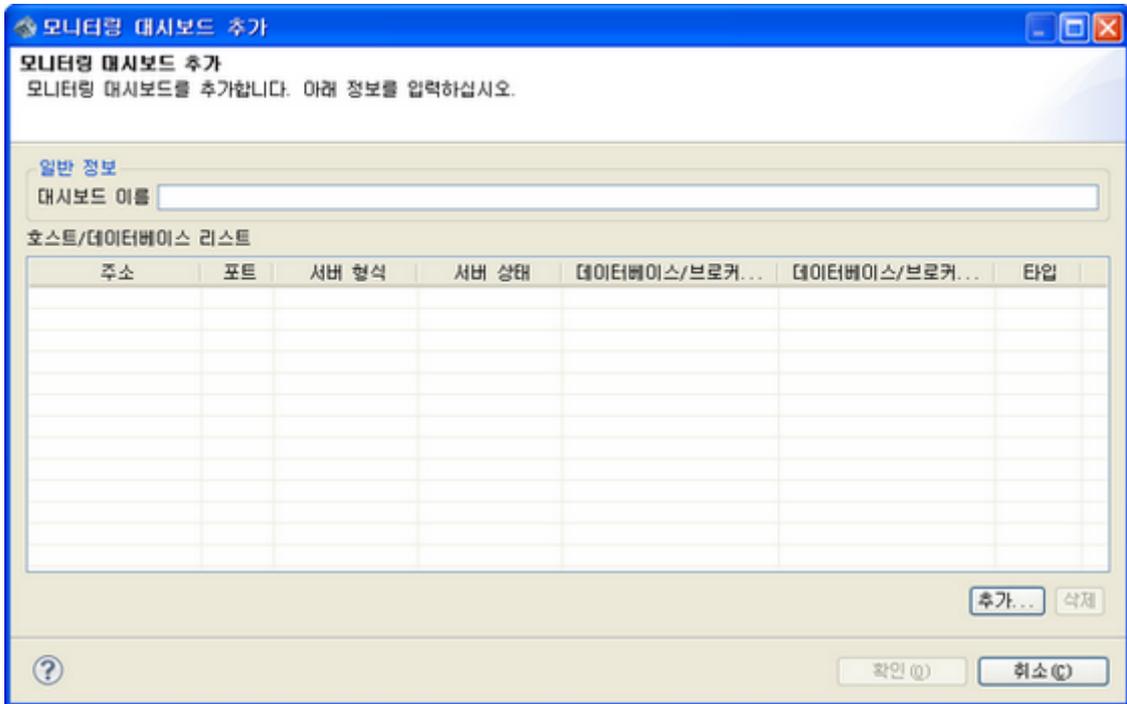
# 대시보드

## 대시보드 추가

메뉴에서 [도구] > [대시보드 탐색기 열기]를 선택하면, [호스트] 탭 옆에 [대시보드] 탭이 나타난다.



[대시보드] 탭에서 마우스 오른쪽 버튼을 클릭하여 [대시보드 추가]를 선택하면, [모니터링 대시보드 추가] 대화 상자가 나타난다. [대시보드 이름]에 대시보드 이름을 지정하고 [추가]를 클릭한다.



### 1 단계: 호스트 정보 설정

모니터링할 데이터베이스 서버의 호스트 정보를 입력한다. 이때, 해당 호스트에는 CUBRID 매니저 서버가 구동 중이어야 한다.

호스트/데이터베이스 추가

호스트 정보 설정  
호스트 정보를 입력하십시오.

호스트 정보

별칭 이름 10.34.64.35

주소 10.34.64.35

포트 8007

사용자 이름 admin

비밀번호 \*\*\*\*

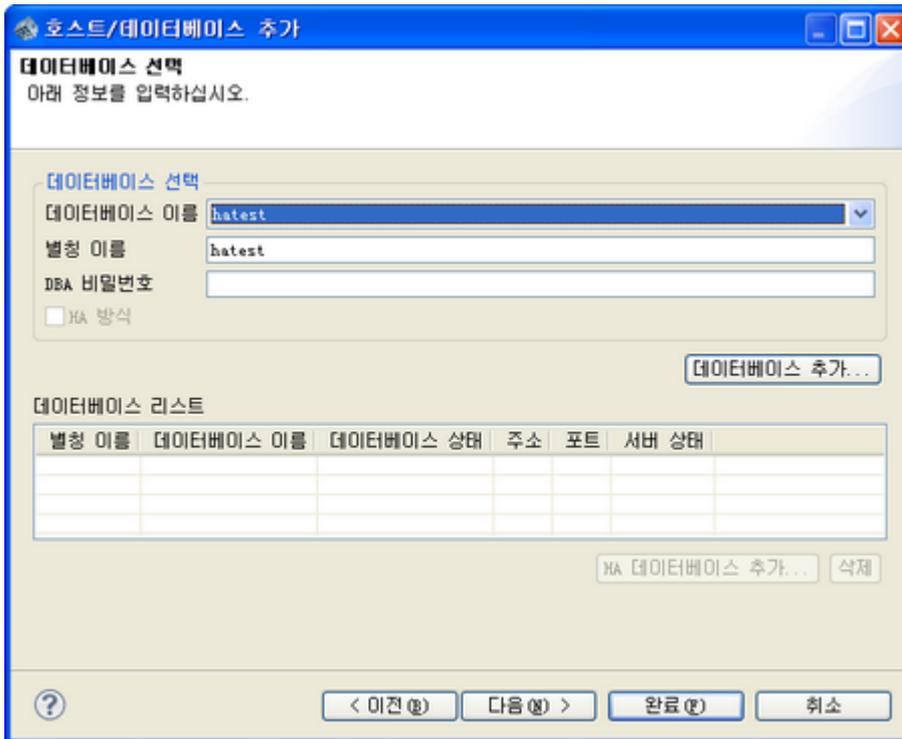
연결 시험...

< 이전 > 다음 > 완료 취소

- **별칭 이름:** 해당 호스트 정보에 부여하는 별칭을 입력한다. 생략하면 호스트 주소가 사용된다.
- **주소:** 해당 호스트의 IP 주소를 입력한다.
- **포트:** 해당 호스트에 설치된 CUBRID 매니저 서버의 **cm\_port** 값을 입력한다. **cm\_port** 값은 `$CUBRID/conf/cm.conf` 파일에서 확인할 수 있다.
- **비밀번호:** CUBRID 매니저 관리자(admin)의 비밀번호를 입력한다.
- **연결 시험:** 입력된 호스트 정보로 접속이 가능한지 검증한다. 성공했을 때만 다음 단계로 이동할 수 있다.

## 2 단계: 데이터베이스 선택

해당 호스트에 설치된 데이터베이스 중 모니터링하고자 하는 데이터베이스를 선택한다.

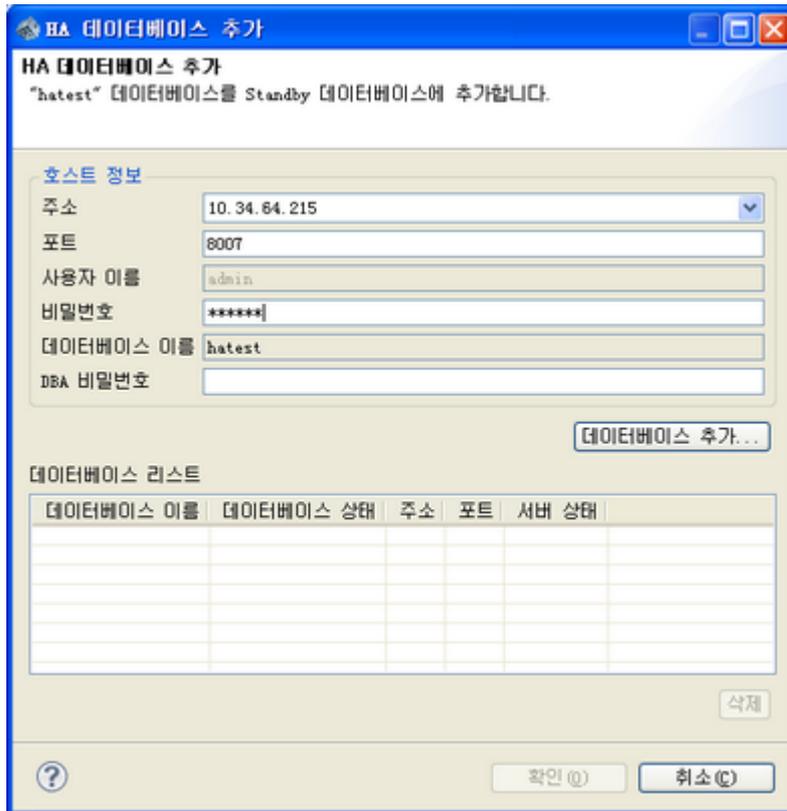


- **별칭 이름:** 해당 데이터베이스 정보에 부여하는 별칭을 입력한다. 생략하면 데이터베이스 이름이 사용된다.
- **DBA 비밀번호:** 해당 데이터베이스의 DBA 사용자 비밀번호를 입력한다.
- **HA 방식:** 해당 데이터베이스가 HA 구성(ha\_mode=on)되어 있으면 자동으로 선택된다.
- **데이터베이스 추가:** 입력된 데이터베이스 정보로 해당 데이터베이스에 접속이 가능한지 검증한 후 데이터베이스 리스트에 추가한다. 해당 데이터베이스가 HA 구성되어 있으면, HA 구성된 나머지 데이터베이스를 검출하여 추가할 수 있는 창이 출력된다([2-1단계: HA 데이터베이스 추가](#) 참고).
- **HA 데이터베이스 추가:** 해당 데이터베이스가 HA 구성된 경우, HA 구성된 나머지 데이터베이스를 추가할 수 있다.
- **삭제:** 데이터베이스 리스트에 등록된 정보를 삭제한다.

### 2-1 단계: HA 데이터베이스 추가

2단계에서 [HA 데이터베이스 추가]를 클릭하면 HA 구성된 나머지 노드를 추가할 수 있다. standby 노드를 추가할지 확인하는 창에서 [예]를 클릭하면 [HA 데이터베이스 추가] 대화 상자에서 standby 노드의 호스트 정보와 데이터베이스 정보를 입력할 수 있다.

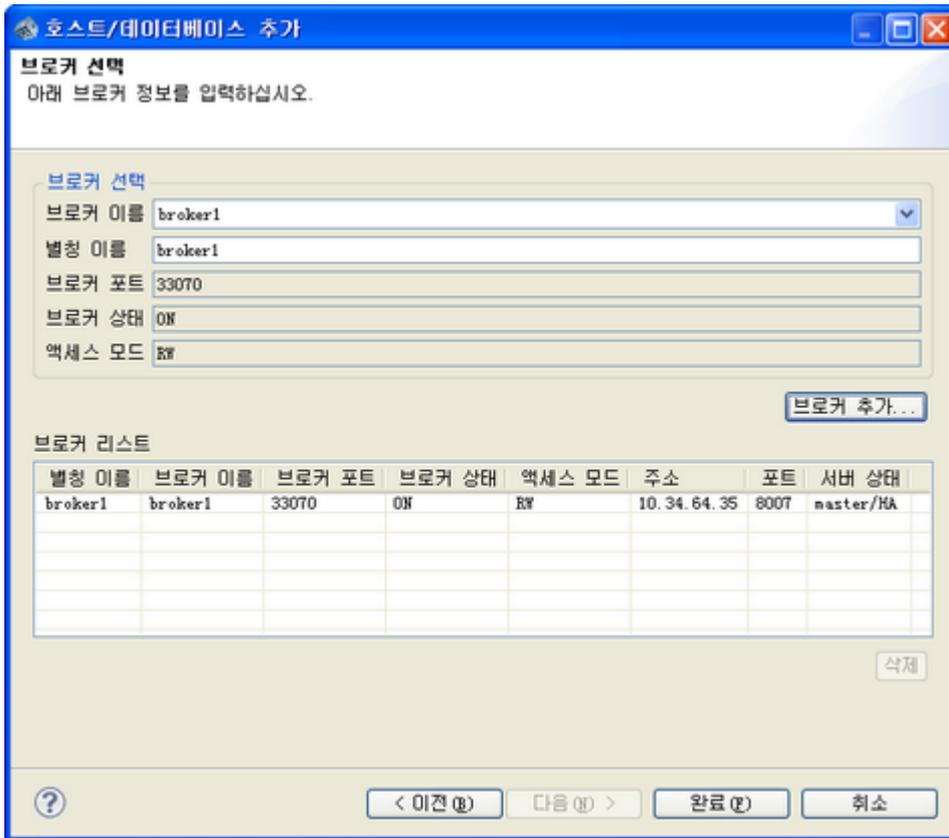




- **포트:** 해당 호스트에 설치된 CUBRID 매니저 서버의 **cm\_port** 값을 입력한다. **cm\_port** 값은 **\$CUBRID/conf/cm.conf** 파일에서 확인할 수 있다.
- **비밀번호:** CUBRID 매니저 관리자(admin)의 비밀번호를 입력한다.
- **DBA 비밀번호:** 해당 데이터베이스의 DBA 사용자 비밀번호를 입력한다.
- **데이터베이스 추가:** 입력된 데이터베이스 정보로 해당 데이터베이스에 접속이 가능한지 검증한 후 데이터베이스 리스트에 추가한다.
- **삭제:** 데이터베이스 리스트에 등록된 정보를 삭제한다.

### 3 단계: 브로커 선택

해당 호스트에서 구동 중인 브로커를 선택하여 모니터링 대상으로 추가할 수 있다.

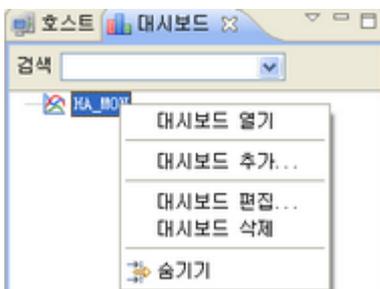


- **별칭 이름:** 해당 브로커 이름의 별칭을 입력한다. 생략하면 브로커 이름이 사용된다.
- **브로커 추가:** 해당 브로커를 브로커 리스트에 추가한다.
- **삭제:** 브로커 리스트에 등록된 정보를 삭제한다.

참고 대시보드는 CUBRID 2008 R3.1 이상 버전에서 지원한다.

## 대시보드 관리

개별 대시보드를 마우스 오른쪽 버튼 클릭하면 대시보드를 관리할 수 있는 바로가기 메뉴가 나타난다.



### 대시보드 열기

해당 대시보드의 대상을 모니터링할 수 있는 대시보드 창을 연다.

## 대시보드 편집

해당 대시보드의 호스트, 데이터베이스, 브로커 정보를 편집한다.

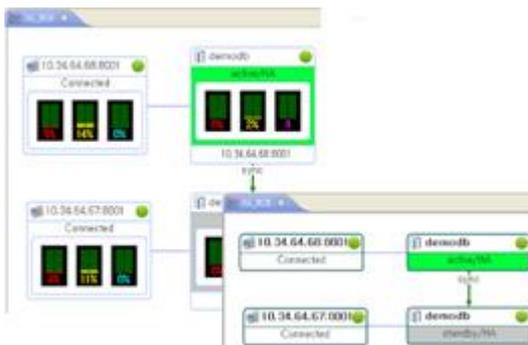
## 대시보드 삭제

해당 대시보드를 삭제한다.

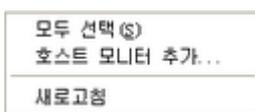
## 대시보드 창

### 대시보드 창 관리

마우스나 키보드 방향 키를 사용하여 대시보드 창에 표시된 대상 객체의 위치를 사용자가 조정할 수 있다. 조정된 위치 정보는 CUBRID 매니저 클라이언트가 설치된 로컬 디렉터리에 저장되어, 항상 마지막으로 저장된 위치 정보에 따라 객체를 표시한다. <Ctrl> 키를 누른 상태에서 마우스 휠을 움직이거나 <+> 키 또는 <-> 키를 누르면 화면의 배율을 조절할 수 있다.



대시보드 창에서 마우스 오른쪽 버튼 클릭하면 다음과 같은 바로가기 메뉴가 나타난다.



- **모두 선택:** 대시보드 창에 표시된 객체 전부를 선택하여 위치를 이동할 수 있다.
- **호스트 모니터 추가:** 해당 대시보드에 모니터링하고자 하는 새로운 호스트 정보를 추가할 수 있다.
- **새로 고침:** 해당 대시보드를 수동으로 새로 읽어들인다.

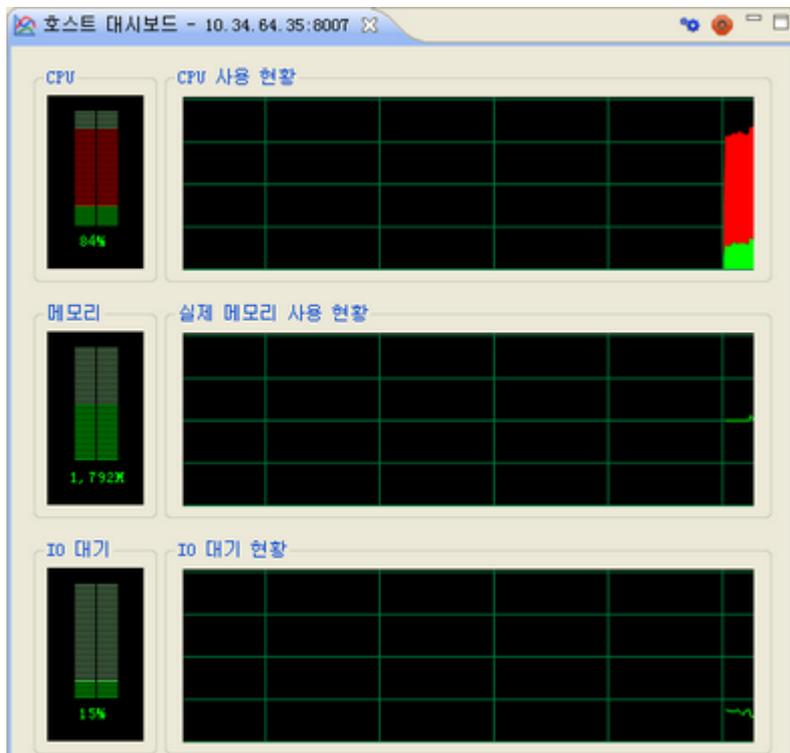
### 호스트 모니터

데이터베이스 또는 브로커가 구동 중인 호스트를 모니터링하며, 해당 호스트의 CPU, MEMORY, IO WAIT 항목의 현재 값을 퍼센트 단위로 보여준다. 호스트 모니터를 마우스 오른쪽 버튼 클릭하면 다음과 상세 기능을 설정할 수 있는 바로가기 메뉴가 나타난다.

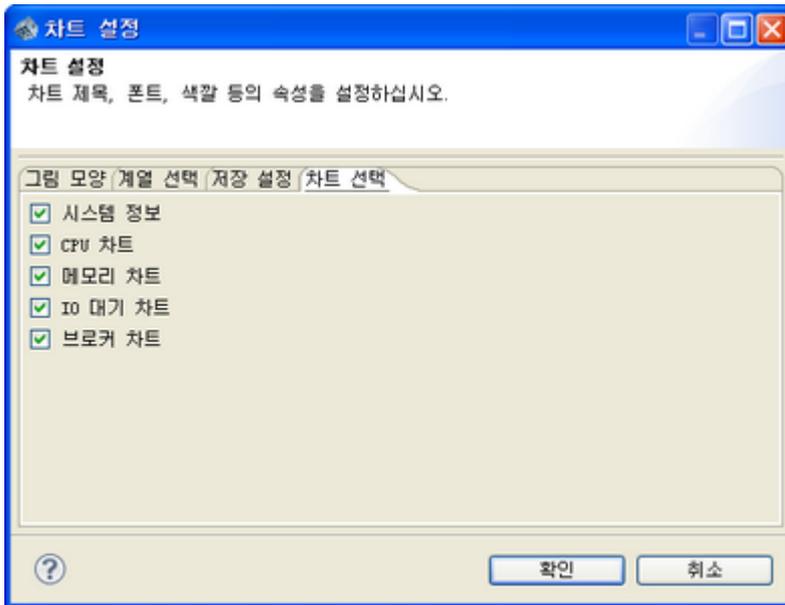


### 상세 정보 보기

[상세 정보 보기]는 해당 호스트의 CPU, MEMORY, IO WAIT 상태의 시간에 따른 변화를 보여준다.



오른쪽 위의 차트 설정(⚙)을 클릭하면 차트 속성을 설정할 수 있다.



- **그림 모양:** 호스트 모니터 차트의 배경색과 가로 세로 눈금선의 색상을 설정한다.
- **계열 선택:** 호스트 모니터 차트에 보여줄 항목을 선택하고 해당 항목 그래프의 색상 및 선 굵기를 설정한다.
- **저장 설정:** 기록 기능을 실행했을 때 기록한 내용을 저장할 디렉토리를 설정한다.
- **차트 선택:** 모니터링할 차트를 선택한다.

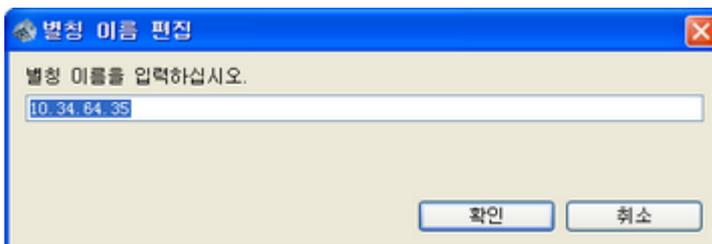
오른쪽 위의 기록 시작(●)을 클릭하면 현재 모니터링 중인 모든 수치 데이터를 로컬에 저장하여 이후 차트에서 다시 확인할 수 있다. 기록을 중지하려면 기록 중지(●)를 클릭하면 된다.

### 삭제하기

모니터링 대상 호스트 정보 및 이에 포함된 데이터베이스 정보, 브로커 정보를 삭제한다.

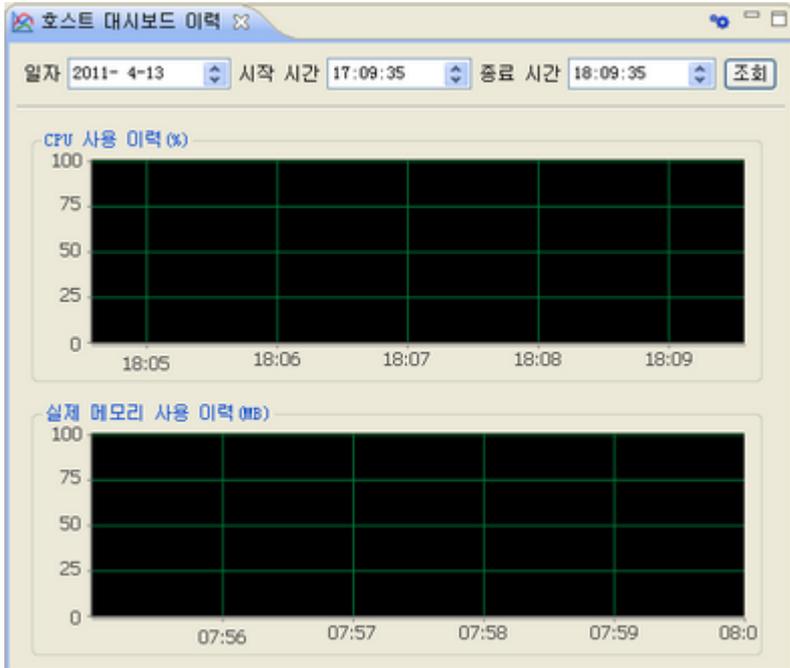
### 별칭 이름 편집

해당 호스트 모니터 창에 표시되는 별칭을 수정할 수 있다. 기본값으로 해당 호스트의 IP 주소가 표시되며, 호스트 모니터에서 운영자가 쉽게 파악할 수 있는 서비스 이름으로 편집할 수 있다.



### 모니터링 기록 정보 보기

기록한 모니터링 정보를 확인할 수 있는 기능이다. 일자, 시간을 입력하여 특정 시간 대의 항목 값 추이를 확인할 수 있다.



### 데이터베이스 추가

해당 호스트에 구동 중인 데이터베이스를 직접 추가한다.

### 브로커 추가

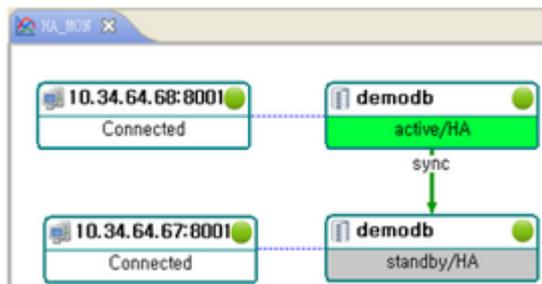
해당 호스트에서 구동 중인 브로커를 직접 추가한다.

### 호스트 감추기

해당 호스트 모니터가 모니터링 대시보드에 표시되지 않도록 한다.

### 최소화하기

해당 호스트 모니터 창을 최소화한다. 호스트 별칭 및 접속 상태만 표시된다.



### 새로고침

해당 호스트 모니터를 수동으로 새로 읽어들인다.

## 데이터베이스 모니터

운영 중인 데이터베이스 서버를 모니터링하며, 해당 데이터베이스 서버의 CPU, MEMORY, HA 지연(msec) 항목의 현재 값을 보여준다. 데이터베이스 모니터를 마우스 오른쪽 버튼 클릭하면 다음과 상세 기능을 설정할 수 있는 바로가기 메뉴가 나타난다.

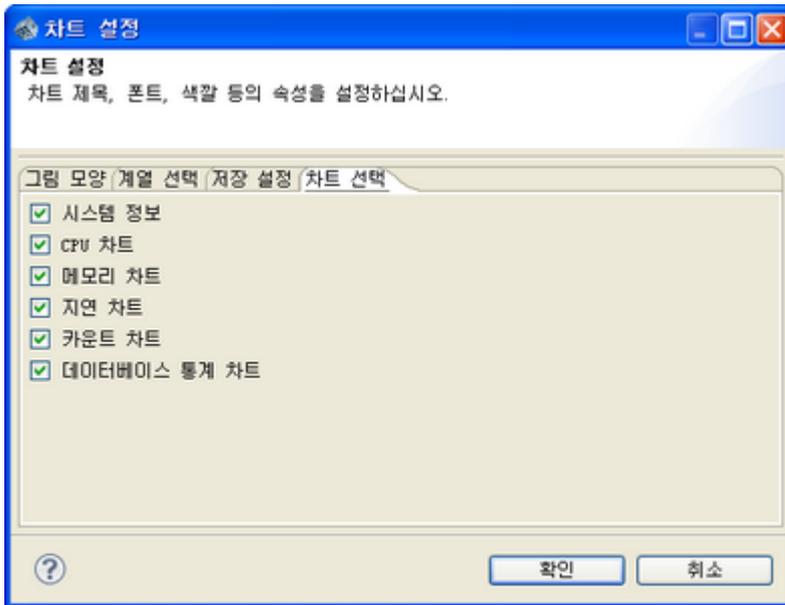


## 상세 정보 보기

[상세 정보 보기]는 해당 데이터베이스 서버의 CPU 사용 정보, 메모리 정보, HA 지연 정보 항목의 시간에 따른 변화를 보여준다.



오른쪽 위의 차트 설정(⚙)을 클릭하면 차트 속성을 설정할 수 있다.



- **그림 모양:** 데이터베이스 모니터 차트의 배경색과 가로 세로 눈금선의 색상을 설정한다.
- **계열 선택:** 데이터베이스 모니터 차트에 보여줄 항목을 선택하고 해당 항목 그래프의 색상 및 선 굵기를 설정한다. 기본값으로 **data\_page\_buffer\_hit\_ratio**와 **num\_data\_page\_fetches**가 설정되어 있으며, **cubrid statdump**에서 보여 주는 모든 항목을 설정할 수 있다. 단, 동시에 다른 터미널에서 **cubrid statdump** 명령을 실행하면 부정확한 값이 출력될 수 있다.
- **저장 설정:** 기록 기능을 실행했을 때 기록한 내용을 저장할 디렉토리를 설정한다.
- **차트 선택:** 모니터링할 차트를 선택한다.

오른쪽 위의 기록 시작(  )을 클릭하면 현재 모니터링 중인 모든 수치 데이터를 로컬에 저장하여 이후 차트에서 다시 확인할 수 있다. 기록을 중지하려면 기록 중지(  )를 클릭하면 된다.

### 삭제하기

호스트 정보에서 해당 데이터베이스 정보를 삭제한다.

### 별칭 이름 편집

해당 데이터베이스 모니터 창에 표시되는 별칭을 수정할 수 있다. 기본값으로 해당 데이터베이스의 이름이 표시되며, 데이터베이스 모니터에서 운영자가 쉽게 파악할 수 있는 다른 이름으로 편집할 수 있다.

### 모니터링 기록 정보 보기

기록한 모니터링 정보를 확인할 수 있는 기능이다. 일자, 시간을 입력하여 특정 시간 대의 항목 값 추이를 확인한다.

### HA 적용 로그 보기

HA 구성된 standby 데이터베이스 서버에서 활성화되는 기능이며, **cubrid applylogdb** 명령 수행 시 발생한 로그를 확인한다.

### HA 복사 로그 보기

HA 구성된 active 데이터베이스 서버에서 활성화되는 기능이며, **cubrid copylogdb** 명령 수행 시 발생한 로그를 확인한다.

### 데이터베이스 로그 보기

해당 데이터베이스의 오류 로그 파일을 선택하여 확인한다.

### 호스트 보기

해당 데이터베이스 서버가 구동 중인 호스트 정보를 확인한다.

### 최소화하기

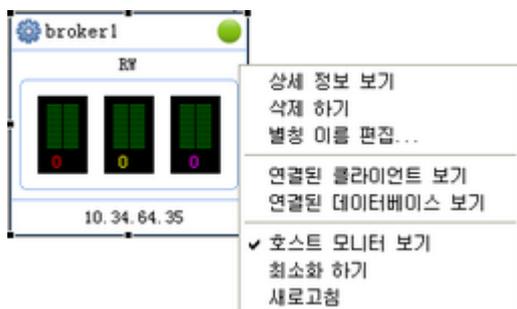
해당 데이터베이스 모니터 창을 최소화한다. 데이터베이스 별칭 및 연결 정보만 표시된다.

### 새로고침

해당 데이터베이스 모니터를 수동으로 새로 읽어들인다.

## 브로커 모니터

운영 중인 브로커를 모니터링하며, 해당 브로커의 session 수, active session 수, TPS 항목의 현재 값을 보여준다. 브로커 모니터를 마우스 오른쪽 버튼 클릭하면 다음과 상세 기능을 설정할 수 있는 바로가기 메뉴가 나타난다.



### 상세 정보 보기

[상세 정보 보기]는 해당 브로커에 대한 항목 값을 보여준다.

PID	PORT	JOB QUEUE	AUTO ABD A...	SQL LOG MODE	SESSION TI...	KEEP CONNE...	ACCESS MODE	ACTIVE_SES...	SESSION	TPS
12233	33070	0	ON	ALL	300	AUTO	BY	0	0	0

ID	PID	QPS	LQS	PFSIZE	STATUS	DB
1	21730	190029	82	46484	IDLE	
2	27127	60011	22	51900	IDLE	hatest
3	12236	1	4	51016	IDLE	hatest
4	12237	0	1	49908	IDLE	hatest
5	12238	0	1	49908	IDLE	hatest

### 삭제하기

호스트 정보에서 해당 브로커 정보를 삭제한다.

### 별칭 이름 편집

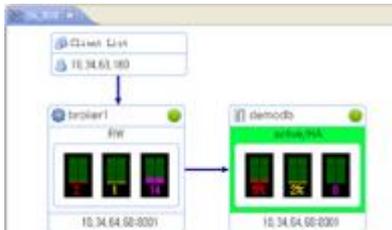
해당 브로커 모니터 창에 표시되는 별칭을 수정할 수 있다. 기본값으로 해당 브로커의 이름이 표시되며, 운영자가 쉽게 파악할 수 있는 다른 이름으로 편집할 수 있다.

### 모니터링 기록 정보 보기

기록한 모니터링 정보를 확인할 수 있는 기능이다. 일자, 시간을 입력하여 특정 시간 대의 항목 값 추이를 확인한다.

### 연결된 클라이언트 보기

해당 브로커에 접속한 응용 클라이언트의 정보를 표시하는 창을 활성화시킨다. 접속한 응용 클라이언트의 IP 주소가 표시된다.



### 연결된 데이터베이스 보기

해당 브로커를 통해 접속하는 데이터베이스 정보를 확인한다. 해당 정보는 브로커 모니터링 항목 중 DB 항목 값을 기준으로 표시된다.

### 호스트 보기

해당 브로커가 구동 중인 호스트 정보를 확인한다.

### 최소화하기

해당 브로커 모니터 창을 최소화한다. 브로커 별칭 및 연결 정보만 표시된다.

### 새로고침

해당 브로커 모니터를 수동으로 새로 읽어들인다.

## 로그

로그는 브로커 로그, 매니저 로그, 데이터베이스 로그로 구성되어 있다. 로그 정보를 구성하는 것은 접근 로그, 오류 로그 그리고 관리 로그로 분류해 볼 수 있다. 각 로그는 다음과 같이 구성된다.



### 브로커 로그

#### 접근 로그

접근 로그 파일은 응용 클라이언트 접속에 관한 정보를 기록하며, "broker\_name.access"의 이름으로 저장된 것을 분석하여 출력한다. 또한, 브로커 환경 설정 파일에서 **LOG\_BACKUP** 매개 변수가 "ON"으로 설정된 경우, 브로커의 구동이 정상적으로 종료되면 접속 로그 파일에 종료된 날짜와 시간 정보가 추가되어 로그 파일이 저장된다.

번호	인덱스	접속 IP	시작 시간	종료 시간	소요 시간	프로세스 ID	오류 정보
1	1	10.34.63.108	2011/04/01 14:18:55	2011/04/01 14:18:55	0:0:0	32442	
2	1	10.34.63.108	2011/04/01 14:18:56	2011/04/01 14:18:56	0:0:0	32442	
3	1	10.34.63.123	2011/04/01 16:07:44	2011/04/01 16:07:45	0:0:1	32442	
4	1	10.34.63.135	2011/04/01 17:04:26	2011/04/01 17:04:26	0:0:0	32442	
5	1	10.34.63.135	2011/04/01 17:04:26	2011/04/01 17:04:26	0:0:0	32442	
6	1	10.34.63.135	2011/04/01 17:04:28	2011/04/01 17:04:28	0:0:0	32442	
7	2	10.34.63.135	2011/04/01 17:05:02	2011/04/01 17:05:02	0:0:0	32443	
8	2	10.34.63.135	2011/04/01 17:06:19	2011/04/01 17:06:20	0:0:1	32443	
9	1	10.34.63.135	2011/04/01 17:04:33	2011/04/01 17:06:22	0:1:49	32442	
10	1	10.34.63.135	2011/04/01 17:32:56	2011/04/01 17:37:56	0:5:0	32442	
11	1	10.34.63.123	2011/04/02 09:41:12	2011/04/02 09:41:40	0:0:28	32442	
12	1	10.34.63.135	2011/04/02 10:50:38	2011/04/02 10:50:39	0:0:1	32442	
13	1	10.34.63.135	2011/04/02 10:50:39	2011/04/02 10:50:39	0:0:0	32442	
14	2	10.34.63.135	2011/04/02 10:50:45	2011/04/02 10:50:45	0:0:0	32443	
15	2	10.34.63.135	2011/04/02 10:51:06	2011/04/02 10:51:06	0:0:0	32443	

#### 오류 로그

오류 로그 파일은 응용 클라이언트의 요청을 처리하는 도중에 발생한 오류에 관한 정보를 기록하며, "broker\_name\_app\_server\_num.err"의 이름으로 저장된다.

번호	시간	오류 형식	오류 코드	트랜잭션 ID	오류 ID	오류 메시지
1	04/11/11 11:13:36.019	DEBUG				file ../src/t
2	04/11/11 11:13:36.020	DEBUG				file ../src/t
3	04/11/11 11:13:36.020	DEBUG				file ../src/t
4	04/11/11 11:49:46.342	DEBUG				file ../src/t
5	04/11/11 11:49:46.342	DEBUG				file ../src/t
6	04/11/11 11:49:46.343	DEBUG				file ../src/t
7	04/11/11 13:26:16.973	DEBUG				file ../src/t
8	04/11/11 13:26:16.974	DEBUG				file ../src/t
9	04/11/11 13:26:16.975	DEBUG				file ../src/t
10	04/11/11 14:04:06.012	DEBUG				file ../src/t
11	04/11/11 14:04:06.012	DEBUG				file ../src/t
12	04/11/11 14:04:06.012	DEBUG				file ../src/t
13	04/11/11 14:05:03.250	DEBUG				file ../src/t
14	04/11/11 14:05:03.250	DEBUG				file ../src/t
15	04/11/11 14:05:03.251	DEBUG				file ../src/t

다음은 오류 로그의 예와 설명이다.

```
Time: 02/04/09 13:45:17.687 - SYNTAX ERROR *** ERROR CODE = -493, Tran = 1, EID = 38
Syntax: Unknown class "unknown_tbl". select * from unknown_tbl
```

- **Time: 02/04/09 13:45:17.687** : 오류 발생 시각
- **SYNTAX ERROR** : 오류의 종류(SYNTAX ERROR, ERROR 등)
- **\*\*\* ERROR CODE = -493** : 에러 코드
- **Tran = 1** : 트랜잭션 ID. -1은 트랜잭션 ID를 할당받지 못한 경우.
- **EID = 38** : 오류 ID. SQL 문 처리 중 오류가 발생한 경우, 서버와 클라이언트 오류 로그가 관련이 있는 SQL 로그를 찾을 때 사용함.

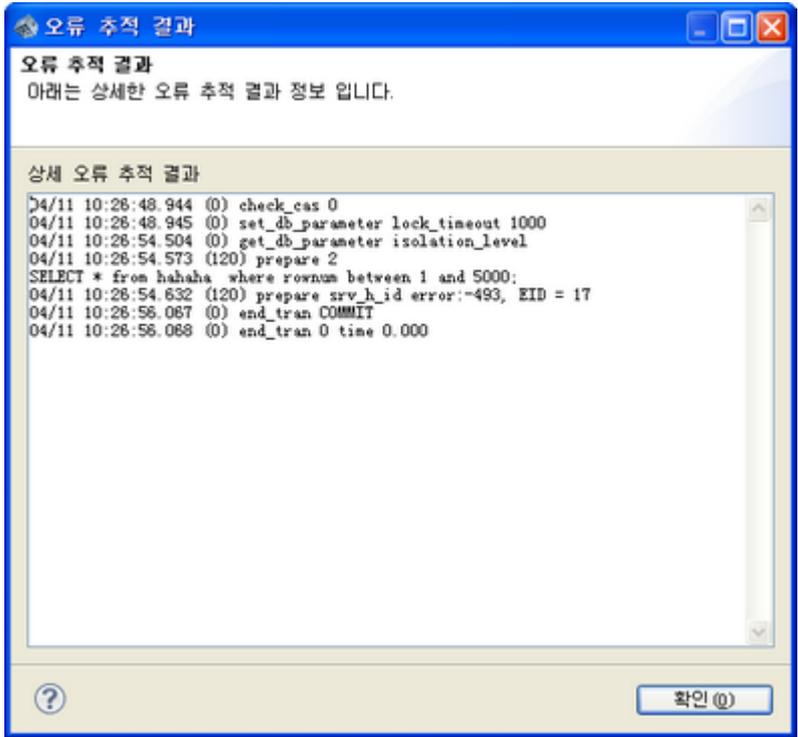
### 오류 로그 추적

SYNTAX ERROR 타입의 브로커 오류 로그는 추적할 수 있다.

번호	시간	오류 형식	오류 코드	트랜잭션 ID	오류 ID	오류 메시지
1	04/11/11 10:26:48.928	DEBUG				file ../src/tran
2	04/11/11 10:26:48.928	DEBUG				file ../src/tran
3	04/11/11 10:26:48.929	DEBUG				file ../src/tran
4	04/11/11 10:26:54.573	SYNTAX ERROR	-493	1	17	Syntax: Unknown

복사 Ctrl+C  
 오류 추적

추적할 오류 로그를 마우스 오른쪽 버튼 클릭하여 [오류 추적]을 선택하면 다음과 같은 오류 추적 결과 대화상자가 나타난다.



### 관리 로그

서비스 구동 및 정지에 관한 이력을 관리한다.

번호	시간	상태
1	2010/09/06 09:49:52	start
2	2010/09/06 09:52:06	stop
3	2010/09/06 09:52:11	start

### 매니저 로그

#### 접근 로그

접근 로그 파일은 매니저 접속에 관한 정보를 기록하며, 매니저 사용자 계정, 작업 수행 내역, 시간 정보를 확인할 수 있다.

#### 오류 로그

오류 로그 파일은 매니저 접속에 대해 발생한 오류에 관한 정보를 기록한다.

### 데이터베이스 로그

#### 관리 로그

데이터베이스 서버 구동에 관한 오류 정보를 기록하며, 이는 <database\_name>\_<date>\_<time>.err 파일 내용을 출력한다.

# CUBRID 매니저 관리자 안내서

이 장에서는 CUBRID 매니저 사용자 관리 방법을 제공하여 사용자를 지원하는 CLI 도구인 CUBRID 매니저 관리자를 사용하는 방법을 설명한다. 이 장에서 설명하는 주요 내용은 다음과 같다.

- CUBRID 매니저 관리자 유틸리티 사용법
- CUBRID 매니저 관리자 유틸리티 구문
- DBMT 사용자
- DBMT 사용자 추가
- DBMT 사용자 삭제
- DBMT 사용자 정보 출력
- DBMT 사용자 권한 변경
- DBMT 사용자 비밀번호 변경
- DBMT 사용자의 데이터베이스 정보 추가
- DBMT 사용자의 데이터베이스 정보 삭제
- DBMT 사용자의 데이터베이스 정보 변경

## CUBRID 매니저 관리자 유틸리티

---

CUBRID 매니저 관리자 유틸리티는 CUBRID 매니저 서비스 관리 기능을 제공한다. 아직까지는 DBMT 사용자(데이터베이스 관리 사용자, CUBRID에서는 CUBRID 매니저 사용자를 나타냄) 정보에 대한 작업만 제공한다. 관리 유틸리티는 다음과 같다.

- DBMT 사용자 추가/삭제/정보 출력 유틸리티
- **cm\_admin adduser**
- **cm\_admin deluser**
- **cm\_admin viewuser**
- DBMT 사용자 변경 유틸리티: DBMT 사용자 권한 정보 및 해당 사용자의 데이터베이스 정보를 변경한다.
- **cm\_admin changeuserauth**
- **cm\_admin changeuserpwd**
- **cm\_admin adddbinfo**
- **cm\_admin changedbinfo**
- **cm\_admin deldbinfo**

프롬프트에서 **cm\_admin**을 입력하면 다음과 같은 정보가 표시된다.

```
cmserver utility, version R0.1
usage: cm_admin <utility-name> [args]
Available Utilities:
  adduser
  deluser
  viewuser
  changeuserauth
  changeuserpwd
adddbinfo
  deldbinfo
  changedbinfo
```

## CUBRID 매니저 관리자 유틸리티

---

다음은 CUBRID 매니저 관리자 유틸리티 구문 사용법이다.

```
cm_admin utility_name
utility_name:
  adduser [option] <dbmtuser-name> <dbmtuser-password> --- DBMT 사용자 추가
  deluser <dbmtuser-name> --- DBMT 사용자 삭제
  viewuser [dbmtuser-name] --- DBMT 사용자 정보 출력
  changeuserauth [option] <dbmtuser-name> --- DBMT 사용자 권한 변경
  changeuserpwd [option] <dbmtuser-name> --- DBMT 사용자 비밀번호 변경
  adddbinfo [option] <dbmtuser-name> <database-name> --- DBMT 사용자의 데이터베이스 정보 추가
  deldbinfo <dbmtuser-name> <database-name> --- DBMT 사용자의 데이터베이스 정보 삭제
  changedbinfo [option] <database-name> number-of-pages --- DBMT 사용자의 데이터베이스 정보
  변경
```

## DBMT 사용자

---

DBMT 사용자 정보는 다음과 같은 정보로 구성된다.

- DBMT 사용자 권한: 다음과 같은 권한 정보를 포함한다.
- Unicas(브로커) 권한
- 데이터베이스 생성 권한. 현재는 **admin** 사용자만 이 권한을 가질 수 있다.
- 상태 모니터링 권한
- 데이터베이스 정보: DBMT 사용자가 사용할 수 있는 데이터베이스
- DBMT 사용자 비밀번호

CUBRID 매니저의 기본 사용자는 모든 관리 권한을 가진 **admin** 사용자이며 기본 비밀번호는 admin이다.

## DBMT 사용자 추가

### 설명

**cm\_admin adduser** 유틸리티는 특정 권한과 데이터베이스 정보를 갖는 DBMT 사용자를 생성한다. unicas 권한, 데이터베이스 생성 권한 및 상태 모니터링 권한 등을 DBMT 사용자에게 부여할 수 있다.

### 구문

```
cm_admin adduser options dbmtuser-name dbmtuser-password
options :
[{-u | --unicas} authority]
[{-c | --dbcreate} authority]
[{-m | --monitor} authority]
[{-d | --dbinfo} database-info]
```

- **cm\_admin** : CUBRID 매니저를 관리하는 통합 유틸리티
- **adduser** : 새 DBMT 사용자를 생성하는 명령어
- **options** : **-u**, **-c**, **-m**, **-d** 옵션이 지원된다. 자세한 정보는 해당 옵션의 설명 및 예제를 참조한다.
- **dbmtuser-name** : 생성할 DBMT 사용자의 고유한 이름을 지정한다. DBMT 사용자 이름은 4자 이상이어야 한다. 지정한 **dbmtuser-name**이 기존 **dbmtuser-name**과 같으면 **cm\_admin**은 DBMT 사용자 생성을 중지한다.
- **dbmtuser-password** : DBMT 사용자의 비밀번호이다. 비밀번호는 4자 이상이어야 한다.

### 옵션

옵션	설명
-u	생성할 DBMT 사용자의 unicas 권한을 지정한다.
--unicas	사용할 수 있는 값: admin, none, monitor. 기본값: <b>none</b>
-c	생성할 DBMT 사용자의 데이터베이스 생성 권한을 지정한다.
--dbcreate	사용할 수 있는 값: none, admin, 기본값: <b>none</b>
-m	생성할 DBMT 사용자의 모니터링 권한을 지정한다.
--monitor	사용할 수 있는 값: admin, none, monitor. 기본값: <b>none</b>
-d	생성할 DBMT 사용자의 데이터베이스 정보를 지정한다.
--dbinfo	DBINFO 는 "<dbname>;<uid>;<broker_ip>,<broker_port>"의 형식으로 지정해야 한다.

### Unicas 권한(-u)

다음은 이름이 "testdbmt"이고 비밀번호가 "testdbmtpwd"인 DBMT 사용자를 생성하고 unicas 권한을 monitor로 설정하는 예이다.

```
cm_admin adduser -u monitor testdbmt testdbmtpwd
```

### 데이터베이스 생성 권한(-c)

다음은 이름이 "testdbmt"이고 비밀번호가 "testdbmtpwd"인 DBMT 사용자를 생성하고 데이터베이스 생성 권한을 admin으로 설정하는 예이다.

```
cm_admin adduser -c admin testdbmt testdbmtpwd
```

### 상태 모니터링 권한(-m)

다음은 이름이 "testdbmt"이고 비밀번호가 "testdbmtpwd"인 DBMT 사용자를 생성하고 상태 모니터링 권한을 admin으로 설정하는 예이다.

```
cm_admin adduser -m admin testdbmt testdbmtpwd
```

### 데이터베이스 정보 추가(-d)

다음은 이름이 "testdbmt"인 DBMT 사용자에게 "testdb;dba;localhost,30000"이라는 데이터베이스 정보를 추가하는 예이다.

```
cm_admin adduser -d "testdb;dba;localhost,30000" testdbmt
```

## DBMT 사용자 삭제

---

### 설명

**cm\_admin deluser** 유틸리티는 지정한 DBMT 사용자 이름을 기준으로 DBMT 사용자를 삭제한다.

### 구문

```
cm_admin deluser dbmtuser-name dbmtuser-password
```

- **cm\_admin** : CUBRID 매니저를 관리하는 통합 유틸리티
- **deluser** : 기존 DBMT 사용자를 삭제하는 명령어
- *dbmtuser-name* : 삭제할 DBMT 사용자 이름

### 예제

다음은 이름이 "testdbmt"인 DBMT 사용자를 삭제하는 예이다.

```
cm_admin deluser testdbmt
```

## DBMT 사용자 정보 출력

---

### 설명

**cm\_admin viewuser** 유틸리티는 지정한 DBMT 사용자 이름을 기준으로 DBMT 사용자를 삭제한다.

### 구문

```
cm_admin viewuser dbmtuser-name dbmtuser-password
```

- **cm\_admin** : CUBRID 매니저를 관리하는 통합 유틸리티
- **viewuser** : DBMT 사용자의 권한 및 데이터베이스 정보를 출력하는 명령어
- *dbmtuser-name* : DBMT 사용자 이름. 이 값을 입력하면 해당 사용자의 정보만 출력하고, 생략하면 모든 기존 DBMT 사용자 정보를 표시한다.

### 예제

다음은 이름이 "testdbmt"인 DBMT 사용자의 정보를 출력하는 예이다.

```
cm_admin viewuser testdbmt
```

다음과 같이 정보가 출력된다.

```
DBMT USER: testdbmt
Auth info:
  unicas: none
  dbcreate: none
  statusmonitorauth: none
DB info:
=====
====
DBNAME                               UID                               BROKER
INFO
=====
====
testdb                                dba                               localhost,30000
```

## DBMT 사용자 권한 변경

### 설명

`cm_admin changeuserauth` 유틸리티는 DBMT 사용자의 권한을 변경한다.

### 구문

```
cm_admin changeuserauthoptions dbmtuser-name
options :
[{-u | --unicas} authority]
[{-c | --dbcreate} authority]
[{-m | --monitor} authority]
```

- **cm\_admin** : CUBRID 매니저를 관리하는 통합 유틸리티
- **changeuserauth** : DBMT 사용자의 권한 정보를 변경하는 명령어
- *options* : **-u**, **-c**, **-m** 옵션이 지원된다. 자세한 정보는 해당 옵션의 설명 및 예제를 참조한다.
- *dbmtuser-name* : 권한을 변경할 DBMT 사용자의 이름

### 옵션

옵션	설명
-u	DBMT 사용자의 unicas 권한을 지정한다.
--unicas	사용할 수 있는 값: admin, none, monitor
-c	DBMT 사용자의 데이터베이스 생성 권한을 지정한다.
--dbcreate	사용할 수 있는 값: none, admin
-m	DBMT 사용자의 모니터링 권한을 지정한다.
--monitor	사용할 수 있는 값: admin, none, monitor

### Unicas 권한(-u)

다음은 이름이 "testdbmt"인 DBMT 사용자의 unicas 권한을 monitor로 변경하는 예이다.

```
cm_admin changeuserauth -u monitor testdbmt
```

### 데이터베이스 생성 권한(-c)

다음은 이름이 "testdbmt"인 DBMT 사용자의 데이터베이스 생성 권한을 admin으로 변경하는 예이다.

```
cm_admin changeuserauth -c admin testdbmt
```

### 상태 모니터링 권한(-m)

다음은 이름이 "testdbmt"인 DBMT 사용자의 상태 모니터링 권한을 admin으로 변경하는 예이다.

```
cm_admin changeuserauth -m admin testdbmt
```

## DBMT 사용자 비밀번호 변경

### 설명

**cm\_admin changeuserpwd** 유틸리티는 DBMT 사용자의 비밀번호를 변경한다.

### 구문

```
cm_admin changeuserpwd options dbmtuser-name
options :
[{-o | --oldpass} oldpassword]
[{-n | --newpass} newpassword]
[--adminpass adminpassword]
```

- **cm\_admin** : CUBRID 매니저를 관리하는 통합 유틸리티
- **changeuserpwd** : DBMT 사용자의 비밀번호를 변경하는 명령어
- *options* : **-o**, **-n**, **--adminpass** 옵션이 지원된다. **--oldpass** 옵션이나 **--adminpass** 옵션 중 하나는 사용해야 한다. 자세한 정보는 해당 옵션의 설명 및 예제를 참조한다.
- *dbmtuser-name* : 비밀번호를 변경할 DBMT 사용자의 이름이다.

### 옵션

옵션	설명
-o --oldpass	DBMT 사용자의 이전 비밀번호를 지정한다. 이 옵션이나 <b>--adminpass</b> 옵션 중 하나는 사용해야 한다.
-n --newpass	DBMT 사용자의 새 비밀번호를 지정한다.
-- adminpass	admin 사용자의 비밀번호를 지정한다. 이 옵션이나 <b>--oldpass</b> 옵션 중 하나는 사용해야 한다.

#### 이전 비밀번호(-o)와 새 비밀번호(-n)

다음은 이름이 "testdbmt"인 DBMT 사용자의 비밀번호를 변경하는 예이다. **--oldpass** 옵션이나 **--adminpass** 옵션 중 하나는 사용해야 한다.

```
cm_admin changeuserpwd -o old_password -n new_password testdbmt
```

#### 이전 비밀번호(-o)와 새 비밀번호(-n)

다음은 관리자 비밀번호를 사용하여 이름이 "testdbmt"인 DBMT 사용자의 비밀번호를 변경하는 예이다. **--oldpass** 옵션이나 **--adminpass** 옵션 중 하나는 사용해야 한다.

```
cm_admin changeuserauth --adminpass admin_password -n new_password testdbmt
```

## DBMT 사용자의 데이터베이스 정보 추가

### 설명

**cm\_admin adddbinfo** 유틸리티는 데이터베이스 정보(데이터베이스 이름, UID, 브로커 IP 및 브로커 포트)를 DBMT 사용자에게 추가한다.

### 구문

```
cm_admin adddbinfo options dbmtuser-name database-name
options :
[{-u | --uid} userid]
[{-h | --host} broker-ip]
[{-p | --port} broker-port]
```

- **cm\_admin** : CUBRID 매니저를 관리하는 통합 유틸리티
- **adddbinfo** : DBMT 사용자에게 데이터베이스 정보를 추가하는 명령어
- **options** : **-u**, **-h**, **-p** 옵션이 지원된다. 자세한 정보는 해당 옵션의 설명 및 예제를 참조한다.
- **dbmtuser-name** : DBMT 사용자 이름
- **database-name** : 추가할 데이터베이스 이름

### 옵션

#### 옵션 설명

**-u** 데이터베이스의 사용자 ID 를 지정한다.

**--uid** 기본값: **dba**

**-h** 클라이언트가 데이터베이스에 접속할 때 사용하는 브로커의 호스트를 지정한다.

**--host** 기본값: **localhost**

**-p** 클라이언트가 데이터베이스에 접속할 때 사용하는 브로커의 포트 번호를 지정한다.

**--port** 기본값: **30000**

#### 데이터베이스 사용자 ID(-u)

다음은 이름이 "testdbmt"인 DBMT 사용자에게 이름이 "testdb"이고 사용자 ID가 "uid"인 데이터베이스를 추가하는 예이다.

```
cm_admin adddbinfo -u uid testdbmt testdb
```

#### 브로커의 호스트 IP(-h)

다음은 이름이 "testdbmt"인 DBMT 사용자에게 이름이 "testdb"이고 호스트 IP가 "127.0.0.1"인 데이터베이스를 추가하는 예이다.

```
cm_admin adddbinfo -u uid testdbmt testdb
```

#### 브로커의 포트(-p)

다음은 이름이 "testdbmt"인 DBMT 사용자에게 이름이 "testdb"이고 브로커 포트가 "33000"인 데이터베이스를 추가하는 예이다.

```
cm_admin adddbinfo -h 127.0.0.1 testdbmt testdb
```

#### 기본값으로 데이터베이스 생성

다음은 이름이 "testdbmt"인 DBMT 사용자에게 이름이 "testdb"이고 기본값을 사용하는 데이터베이스를 추가하는 예이다.

```
cm_admin adddbinfo -p 33000 testdbmt testdb
```

## DBMT 사용자의 데이터베이스 정보 삭제

---

### 설명

**cm\_admin deldbinfo** 유틸리티는 지정한 DBMT 사용자의 데이터베이스 정보를 삭제한다.

### 구문

```
cm_admin deldbinfo options dbmtuser-name database-name
```

- **cm\_admin** : CUBRID 매니저를 관리하는 통합 유틸리티
- **deldbinfo** : DBMT 사용자의 데이터베이스 정보를 삭제하는 명령어
- *dbmtuser-name* : DBMT 사용자 이름
- *database-name* : 삭제할 데이터베이스 이름

### 예제

다음은 이름이 "testdbmt"인 DBMT 사용자에게서 이름이 "testdb"인 데이터베이스 정보를 삭제하는 예이다.

```
cm_admin deldbinfo testdbmt testdb
```

## DBMT 사용자의 데이터베이스 정보 변경

### 설명

**cm\_admin changedbinfo** 유틸리티는 지정한 DBMT 사용자의 데이터베이스 정보를 변경한다.

### 구문

```
cm_admin changedbinfo options dbmtuser-name database-name
options :
[{-u | --uid} userid]
[{-h | --host} broker-ip]
[{-p | --port} broker-port]
```

- **cm\_admin** : CUBRID 매니저를 관리하는 통합 유틸리티
- **changedbinfo** : DBMT 사용자의 데이터베이스 정보를 변경하는 명령어
- *options* : **-u**, **-h**, **-p** 옵션이 지원된다. 자세한 정보는 해당 옵션의 설명 및 예제를 참조한다.
- *dbmtuser-name* : DBMT 사용자 이름
- *database-name* : 변경할 데이터베이스 이름

### 옵션

#### 옵션 설명

**-u** 데이터베이스의 사용자 ID 를 지정한다.

**--uid**

**-h** 클라이언트가 데이터베이스에 접속할 때 사용하는 브로커의 호스트를 지정한다.

**--host**

**-p** 클라이언트가 데이터베이스에 접속할 때 사용하는 브로커의 포트 번호를 지정한다.

**--port**

#### 데이터베이스 사용자 ID(-u)

다음은 이름이 "testdbmt"인 DBMT 사용자의 "testdb" 데이터베이스에서 사용자 ID 정보를 "uid"로 업데이트하는 예이다.

```
cm_admin changedbinfo -u uid testdbmt testdb
```

#### 브로커의 호스트 IP(-h)

다음은 이름이 "testdbmt"인 DBMT 사용자의 "testdb" 데이터베이스에서 호스트 IP 정보를 "10.34.63.132"로 업데이트하는 예이다.

```
cm_admin changedbinfo -h 10.34.63.132 testdbmt testdb
```

#### 브로커의 포트(-p)

다음은 이름이 "testdbmt"인 DBMT 사용자의 "testdb" 데이터베이스에서 브로커 포트 정보를 "33000"로 업데이트하는 예이다.

```
cm_admin changedbinfo -p 33000 testdbmt testdb
```