

CUBRID SQL/Programming

—

작성년월일: 2009년 3월

목차

1. 인터페이스 소개
2. 데이터베이스 생성
3. 스키마 다루기
4. 데이터 다루기
5. 연산자와 함수

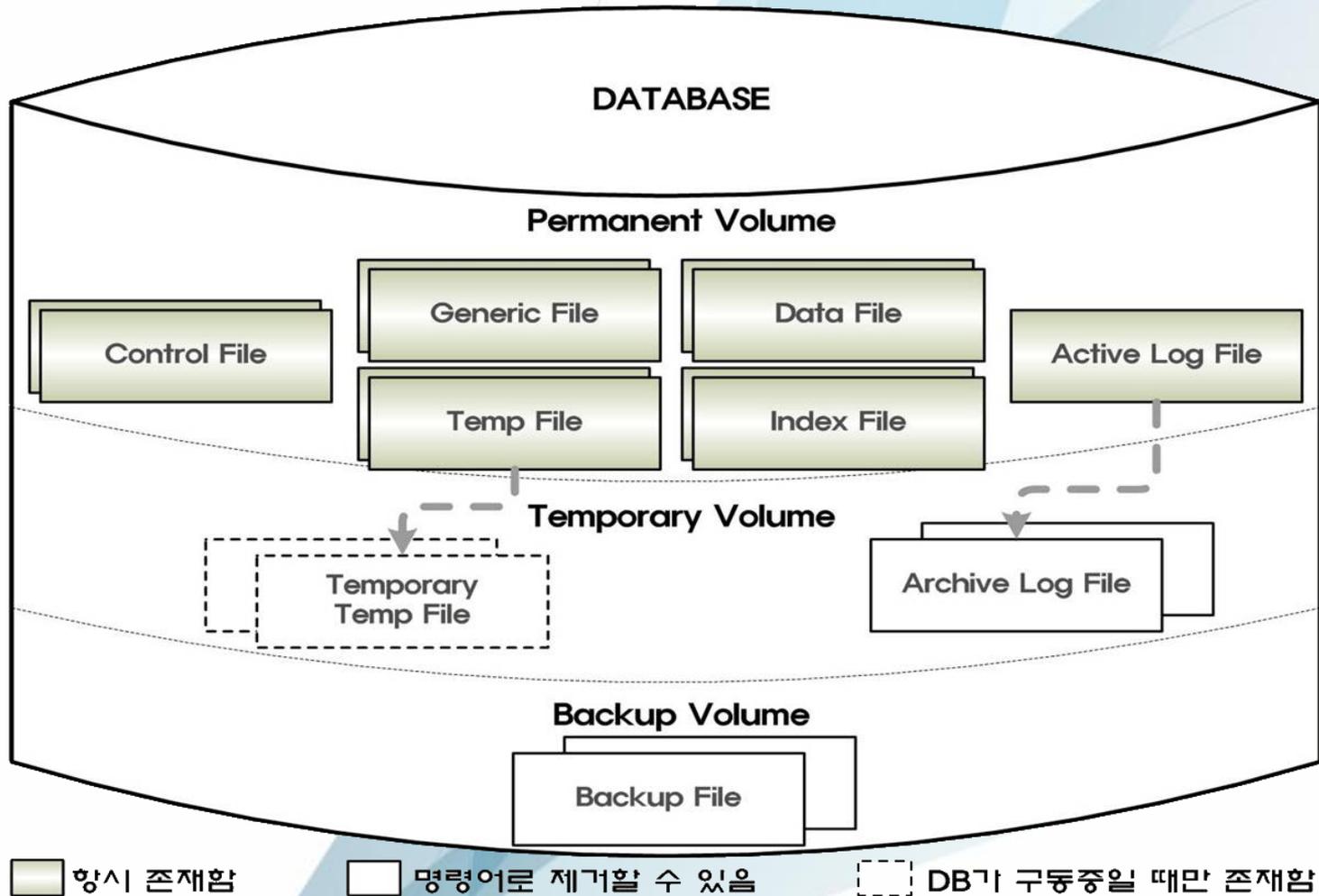
1. 인터페이스 소개

1.1 CUBRID 개발 인터페이스

- JDBC
 - JDBC 3.0 스펙을 기준으로 개발된 type 3 JDBC Driver 제공
 - Driver Manager, Data Source, XA, Prepared Statement Pooling 등 지원
 - iBATIS 사용 가능
- JSP (Java Stored Procedure)
 - Server-side JDBC와 유사한 개념
 - JDBC로 작성된 자바 프로그램을 저장 프로시저로 등록해서 사용 가능
- ODBC/OLE DB
 - ODBC 3.x 스펙을 기준으로 Core, Level 1, Level 2 API 일부를 지원하는 ODBC Driver 제공
 - Windows에서 많이 사용되는 OLE DB 지원
- PHP
 - PHP database module
- CCI API
- C API
- ESQL/C

2. 데이터베이스 생성

2.1 데이터베이스 볼륨 구조



2.2 데이터베이스 생성 실습 1

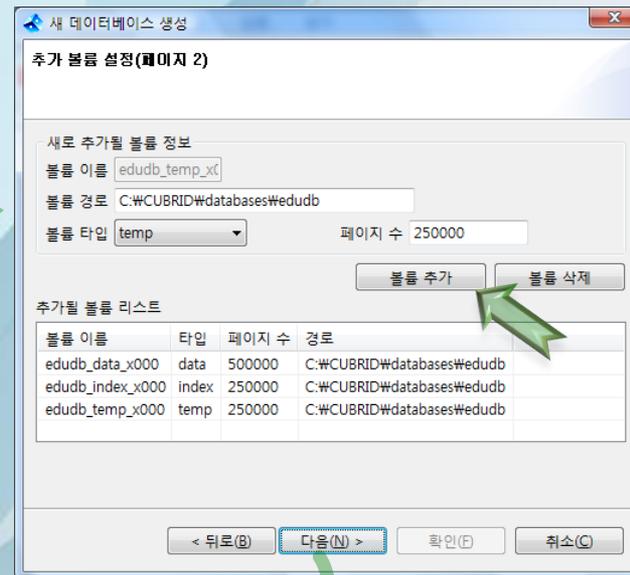
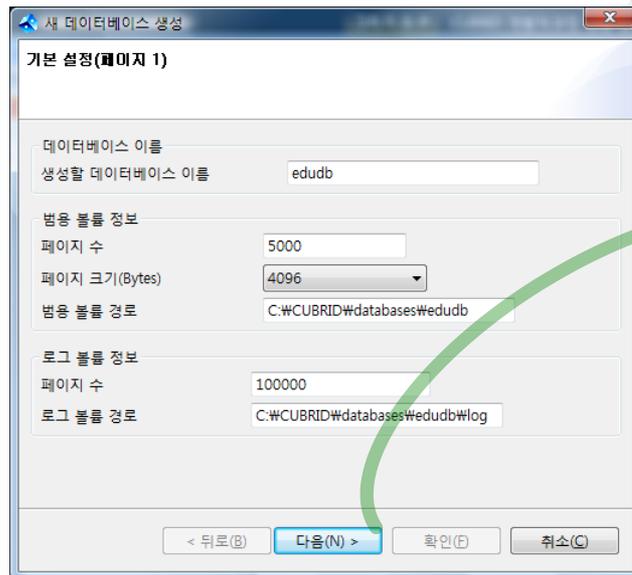
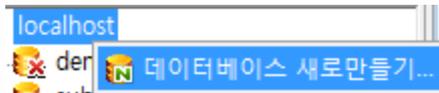
1. 명령어를 이용하여 edudb라는 데이터베이스 생성하기

- 데이터베이스 생성 위치: /database/edudb
- 초기 범용 볼륨: 5000pages
- 데이터 볼륨: 500,000 pages
- 인덱스 볼륨: 250,000 pages
- 템프 볼륨: 250,000 pages

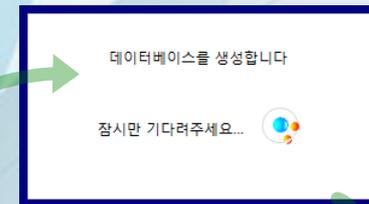
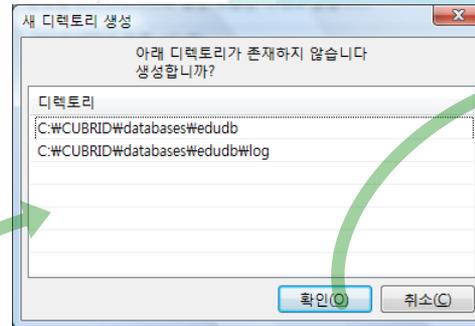
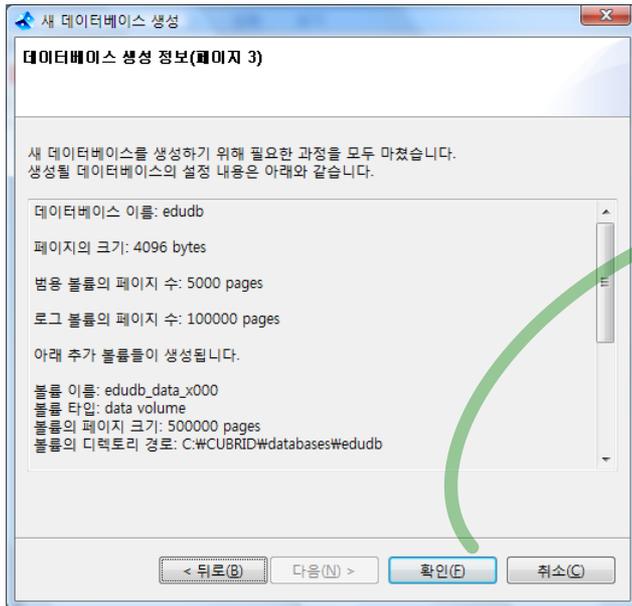
```
% cubrid createdb
% cubrid createdb -p 5000 -F /database/edudb -l 100000 -L /database/edudb edudb
% cubrid addvoldb -S -p data -F /database/edudb edudb 500000
% cubrid addvoldb -S -p index -F /database/edudb edudb 250000
% cubrid addvoldb -S -p temp -F /database/edudb edudb 250000
```

2.2 데이터베이스 생성 실습 2

2. 매니저를 이용하여 edudb라는 데이터베이스 생성하기



2.3 데이터베이스 생성 실습 3



3. 스키마 다루기

3.1 스키마 생성

- class (table)
 - 개수 무제한
 - 이름에 한글, 영문자, 숫자, _, #, % 사용 가능, 첫글자는 문자(한글, 영문)여야 함.
 - 이름의 최대 길이는 255자
- attribute (column)
 - class 당 최대 6,400 개 생성 가능
 - 이름에 한글, 영문자, 숫자, _, #, % 사용 가능, 첫글자는 문자(한글, 영문)여야 함.
 - 이름의 최대 길이는 255자
- index
 - class 당 최대 6,400 개 생성 가능
 - 이름지정 가능
- 제약조건
 - NULL
 - NOT NULL
 - UNIQUE
 - 중복된 값 허용하지 않음
 - default 와 같이 선언 불가
 - PRIMARY KEY
 - FOREIGN KEY

3.2 데이터 타입 1

구분	자료형	형식 예제	데이터
Character Strings	CHAR(n)	char(100)	'example'
	VARCHAR(n)	varchar(100)	'경기도 성남시 분당구 서현동'
	NCHAR(n)	nchar(100)	N'퍼스트타워13층'
	NCHAR VARYING	nchar varying(100)	N'엔에이치엔'
Bit Strings	BIT(n)	bit(4)	X'1A23'
	BIT VARYING(n)	bit varying(4)	B'001'
Numeric	NUMERIC(or DECIMAL)	numeric(8,4)	1234.5678
	INTEGER	integer or int	20080325
	SMALLINT	smallint	3000
	FLOAT	float	34.567982
	DOUBLE PRECISION	double	2.877e+33
	MONETARY	monetary	\$799.95 / 4579

3.2 데이터 타입 2

구분	자료형	형식 예제	데이터
Date-time	DATE	date	'10/14/2009'
	TIME	time	'08:20:14 PM', '20:20:14', '9:20 am'
	TIMESTAMP	timestamp	'1:03 pm 1/10/2009'
Collections	SET	set char(5)	{'aaa', 'bbbb', 'cccc'}
	MULTISET	multiset char(5)	{'aaa', 'bbbb', 'cccc'}
	LIST or SEQUENCE	sequence integer	{20, 30, 40, 50}

3.3 테이블 생성

- 테이블 생성 구문

```
CREATE TABLE 테이블 이름(하위 테이블)  
  [ (애트리뷰트 정의절 [{, 애트리뷰트 정의절 }]);
```

애트리뷰트 정의절 :
 애트리뷰트_이름 데이터_타입 [DEFAULT 값] [제약 사항]

제약 사항 :
 PRIMARY KEY
 FOREIGN KEY
 NOT NULL
 UNIQUE
 NULL

- 각 테이블 이름은 해당 데이터베이스에서 유일해야 함
- 테이블 내에서 애트리뷰트의 이름은 유일해야 함

3.3 테이블 생성 실습 1

```
CREATE TABLE my_company
```

테이블 이름

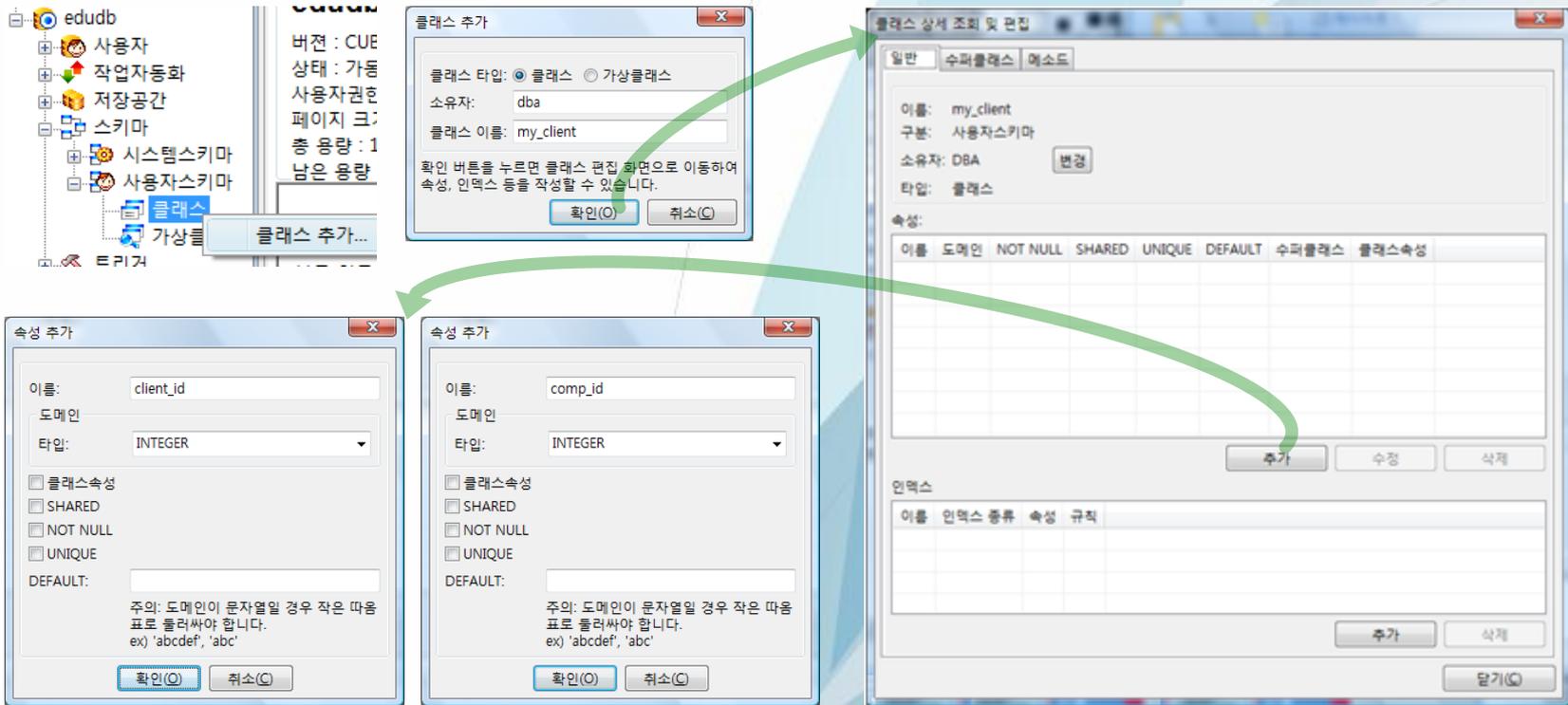
```
(  
  comp_id int not null unique,  
  comp_name varchar(100) default 'CUBRID'  
) ;
```

어트리뷰트
이름

데이터 타입

제약사항

3.3 테이블 생성 실습 2



속성:

이름	도메인	NOT NULL	SHARED	UNIQUE	DEFAULT
client_id	integer				
comp_id	integer				

3.4 테이블 변경

- DBA나 테이블 소유자, 인증을 받은 사용자는 **ALTER** 구문을 사용하여 테이블을 변경할 수 있다.

ALTER TABLE 테이블 이름 변경사항;

변경 사항:

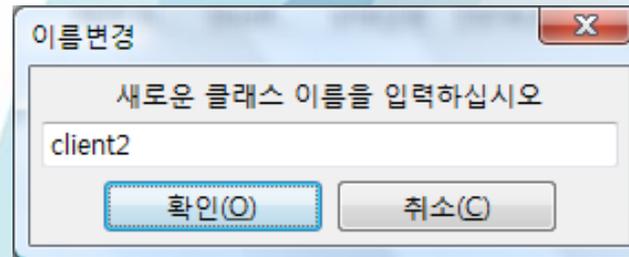
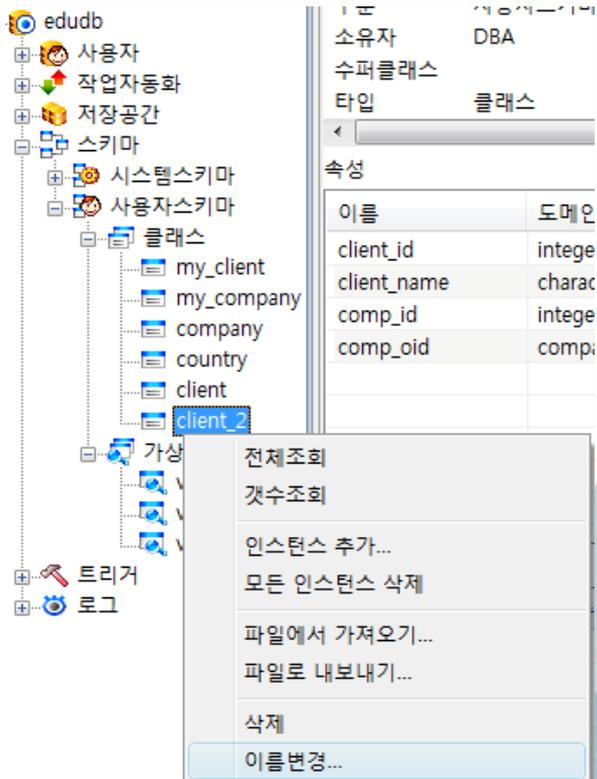
ADD	애트리뷰트 추가
DROP	애트리뷰트 제거
CHANGE	초기값 변경
RENAME	애트리뷰트이름 변경

3.5 테이블 이름 변경 실습

1. SQL 구문으로 변경하기

```
RENMAE TABLE client2 AS client_2
```

2. CUBRID 매니저로 변경하기

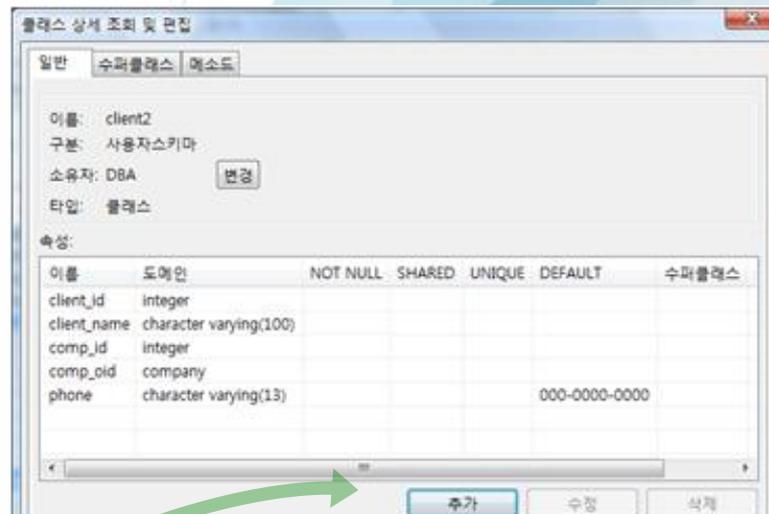
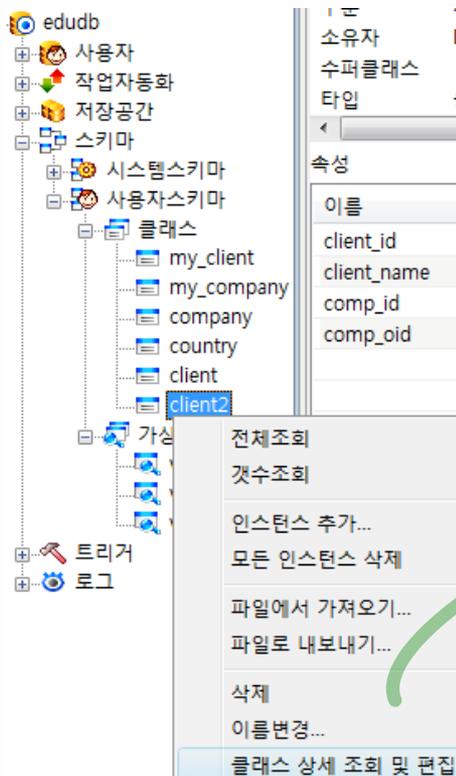


3.6 애트리뷰트 추가

1. SQL 구문으로 변경하기

```
ALTER TABLE client2 ADD ATTRIBUTE phone varchar(13) default '000-0000-0000'
```

2. CUBRID 매니저로 변경하기



속성:

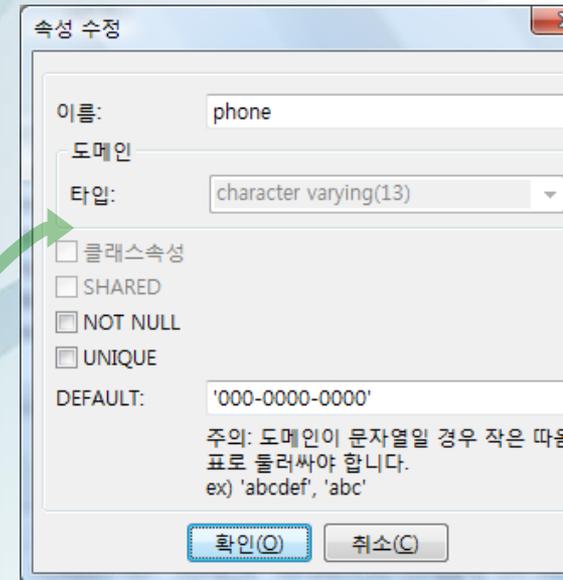
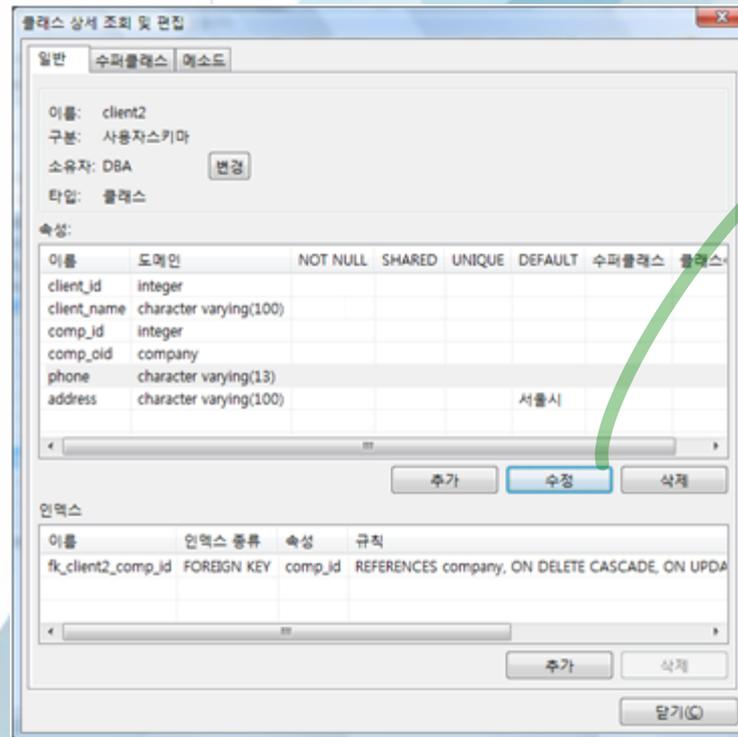
이름	도메인	NOT NULL	SHARED	UNIQUE	DEFAULT
client_id	integer				
client_name	character varying(100)	○			
comp_id	integer				
comp_oid	company				
phone	character varying(13)				000-0000-0000
address	character varying(100)				서울시

3.7 애트리뷰트 이름 변경

1. SQL 구문으로 변경하기

```
ALTER TABLE client2 RENAME ATTRIBUTE phone AS celluar_phone
```

2. CUBRID 매니저로 변경하기

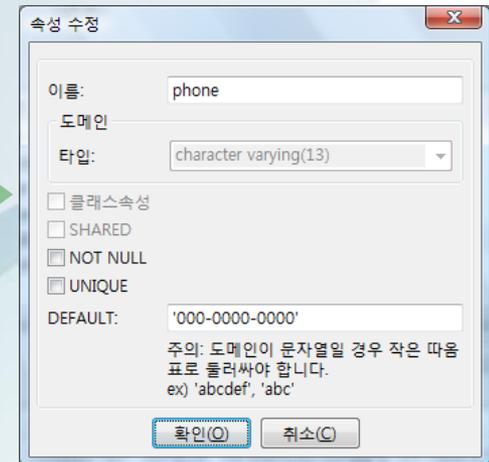
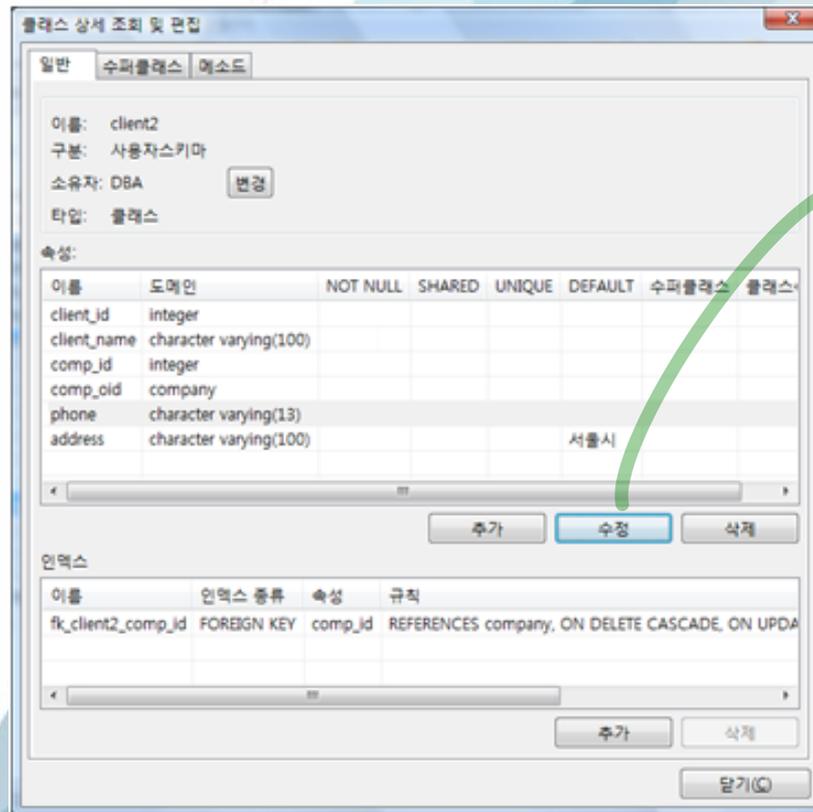
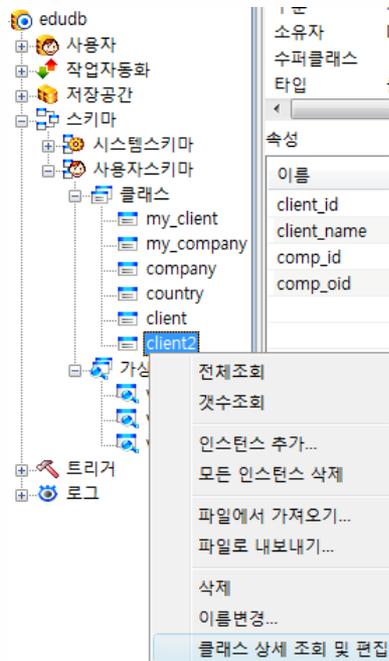


3.8 애트리뷰트 기본값 변경

1. SQL 구문으로 변경하기

```
ALTER TABLE client2 CHANGE phone default '0000-0000'  
ALTER TABLE client2 CHANGE phone default NULL
```

2. CUBRID 매니저로 변경하기



3.9 애트리뷰트 삭제

1. SQL 구문으로 변경하기

```
ALTER TABLE client2 DROP ATTRIBUTE address
```

2. CUBRID 매니저로 변경하기

The screenshot shows the CUBRID Manager interface. On the left, the 'edudb' database tree is expanded to show the 'client2' table. A context menu is open over 'client2', with '클래스 상세 조회 및 편집' (Class Detail View and Edit) selected. The main window displays the '클래스 상세 조회 및 편집' dialog for the 'client2' class. The '속성' (Attributes) table is visible, listing attributes: 'client_id' (integer), 'client_name' (character varying(100)), 'comp_id' (integer), 'comp_oid' (company), and 'address' (character varying(100)). A green arrow points from the '삭제' (Delete) button in the dialog to a warning dialog box that asks '삭제하시겠습니까? address' (Are you sure you want to delete address?). The warning dialog has '예(Y)' (Yes) and '아니오(N)' (No) buttons.

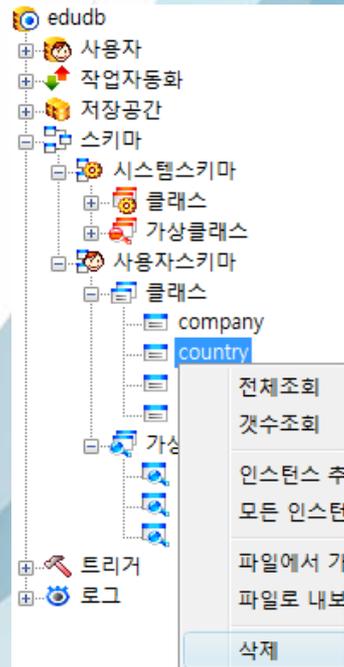
3.10 테이블 제거

```
DROP TABLE 테이블이름 [ { , 테이블이름 }... ]  
DROP ONLY 테이블이름  
DROP ALL 테이블이름 [ EXCEPT 테이블이름 ]
```

1. SQL 구문으로 제거하기

```
drop my_company  
drop table country
```

2. CUBRID 매니저로 제거하기



4. 데이터 다루기

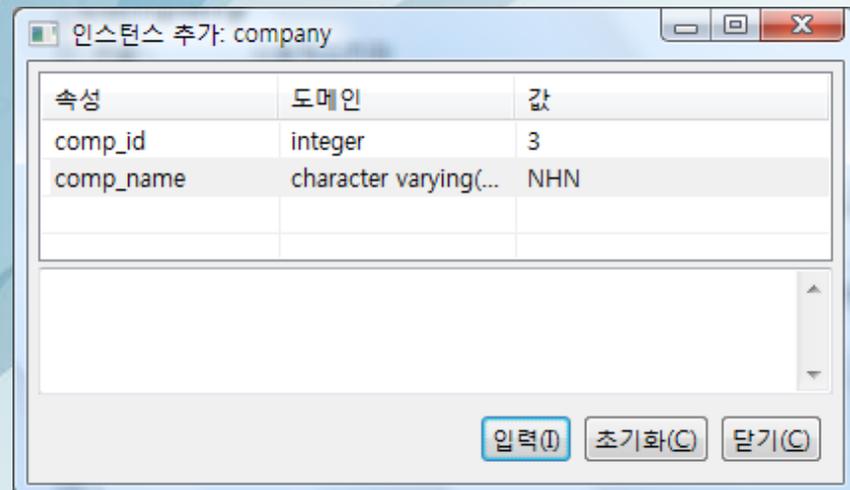
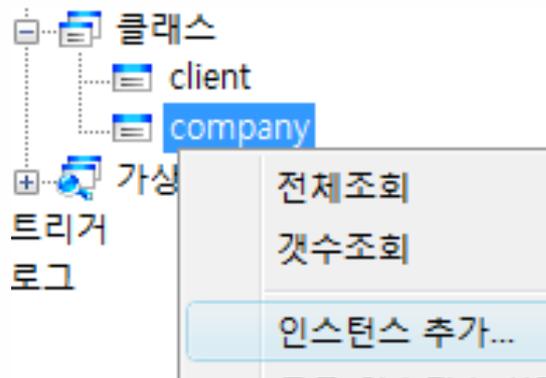
4.1 데이터 입력 1

```
INSERT INTO 테이블이름 VALUES (값1, 값2, ...)  
INSERT INTO 테이블이름 (애틀리뷰트1, 애틀리뷰트2, ...) VALUES (값1, 값2, ...);
```

1. SQL 구문으로 입력하기

```
INSERT INTO company VALUES (1, '큐브리드')  
INSERT INTO company(comp_id, comp_name) VALUES (3, 'NHN')
```

2. CUBRID 매니저로 입력하기



4.1 데이터 입력 2

- sub-query 결과를 이용한 입력

```
INSERT INTO client(client_id, comp_id, client_name)
VALUES (1, (select comp_id from company where comp_id =
1), '홍길동')
```

- sub-query 를 이용한 입력

```
INSERT INTO client(client_id, comp_id, client_name)
select 2, comp_id, '홍길순' from company where comp_id = 1
```

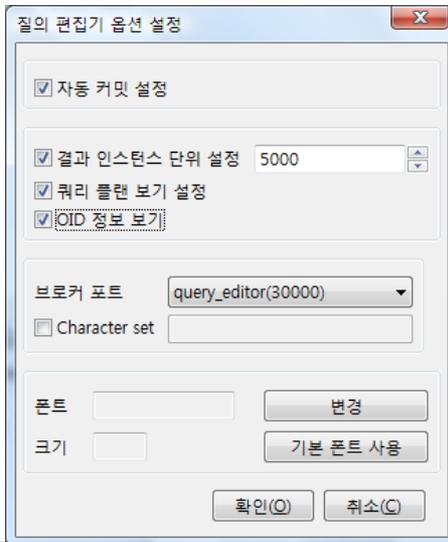
4.2 데이터 수정

UPDATE 테이블이름 **SET** 변경사항 **WHERE** 검색조건

1. SQL 구문으로 수정하기

```
UPDATE client SET client_name = '홍길자' where client_id = 2
UPDATE client SET (comp_id, client_name) = (select comp_id,
'홍길순' from company where comp_id= 1) where client_id = 2
```

2. CUBRID 매니저로 수정하기(옵션 "OID정보보기 선택"시 가능)



```
select * from client
```

결과1

OID	client_id	comp_id	client_name	email	phone	address	co
@370 1 1	1	1	홍길동	kildong@cubrid.com	NULL	NULL	@:
@370 2 1	2	1	홍길순	NULL	NULL	NULL	@:

4.3 데이터 삭제

```
DELETE FROM 테이블이름 WHERE 검색조건
```

1. SQL 구문으로 삭제하기

```
DELETE FROM company where comp_id = 3  
DELETE FROM company
```

2. CUBRID 매니저로 삭제하기

결과1

OID	client_id	comp_id	client_name	email	phone	address	co
@370 1 1	1	1	홍길동	kildong@cubrid.com	NULL	NULL	@:
@370 2 1	2	1	홍기스	NULL	NULL	NULL	@:

클립보드에 복사

삭제

? 1건의 인스턴스를 삭제 하시겠습니까?

예(Y)

아니요(N)

4.4 데이터 검색 1

```
SELECT 애트리뷰트명 FROM 테이블이름 WHERE 검색조건
```

1. SQL 구문으로 검색하기

```
SELECT * FROM nation  
SELECT name, capital FROM nation where code = 'KOR'  
SELECT athlete_code, medal, score FROM record
```

4.4 데이터 검색 2

1. SELECT -FROM절

```
SELECT DISTINCT ( or UNIQUE) FROM 테이블이름  
SELECT *, 테이블이름.*, alias.*
```

```
SELECT DISTINCT event_code from record where host_year=2004
```

```
SELECT sports, record.* from record, event  
where host_year=2004 and code=event_code
```

4.4 데이터 검색 4

2. WHERE 절

SELECT 애트리뷰트명 **FROM** 테이블이름 **WHERE** 검색조건

검색조건:

비교 연산자:

NOT, AND, OR, = <> < > >= <=

Predicates:

BETWEEN, IN, LIKE, NULL

```
select name, capital from nation where code = 'KOR'
select name, medal from record where host_year between 2000
and 2004
select name, medal, score from record where host_year = 2004
and event_code in (select code from event where sports =
'Swimming')
select name, gender, players from event where sports =
'Wrestling' and name like '%48kg'
select * from olympic where slogan is null;
```

4.4 데이터 검색 5

3. ORDER BY 절

ORDER BY *애트리뷰트명* [**ASC** | **DESC**]

ASC : 오름차순 (default)

DESC : 내림차순

```
select name, capital from nation order by name  
select name, capital from nation order by continent desc, name
```

4. GROUP BY 절

GROUP BY *애트리뷰트명* [**HAVING** 조건]

```
select sports from event where gender = 'M' group by sports  
select sports from event where gender = 'M' group by sports  
having count(*) > 1
```

5. 연산자와 함수

5.1 일반 연산자

연산자	의미
비교연산자	= → equal to <> → not equal to < → greater than > → less than >= → greater than or equal to <= → less than or equal to
산술연산자	+ , - , * , /
병합연산자	+ or <pre>SELECT capital ', ' name FROM nation WHERE name = 'Korea'</pre> 결과) Seoul, Korea
집합연산자	UNION → 두 질의의 결과를 합침. 중복 제거 O) UNION ALL → 두 질의의 결과를 합침. 중복 제거 X) DIFFERENCE → 앞부분 질의 결과에서 뒷부분 질의 결과를 제거 INTERSECTION → 두 질의에 공통적으로 들어간 결과

5.2 타입 변환 연산자

연산자	의미
CAST	<p>애트리뷰트 타입을 변환</p> <pre>SELECT name, CAST(name as char(40)) FROM nation</pre> <p>(결과) 'Serbia' 'Serbia'</p>
to_char	<p>시간/날짜/숫자형 데이터를 문자형으로 변환</p> <pre>SELECT TO_CHAR(TIMESTAMP '10:20:30 AM 12/25/2008', 'MONTH dd yyyy') FROM db_root;</pre> <p>(결과) DECEMBER 25 2008</p>
to_date	<p>문자형 데이터를 날짜형으로 변환</p> <pre>SELECT TO_DATE('9/6/1976') FROM db_root;</pre> <p>결과값: 1976-09-06</p>
to_time	<p>문자형 데이터를 시간형으로 변환</p> <pre>SELECT TO_TIME('10:30:20 AM') FROM db_root;</pre> <p>결과값: 10:30:20</p>
to_timestamp	<p>문자형 데이터를 timestamp 형으로 변환</p> <pre>SELECT TO_TIMESTAMP('10:30:20 AM 12/25/2008') FROM db_root;</pre> <p>결과값: 2008-12-25 10:30:20</p>
to_number	<p>문자형 데이터를 숫자형으로 변환</p> <pre>SELECT TO_NUMBER('-1234') FROM db_root;</pre> <p>결과값: -1234</p>

5.3 DATE/TIME 함수

연산자	의미
extract()	날짜나 시간 형의 데이터에서 특정 부분(연,월,일,시,분,초)의 값을 꺼냄 <pre>SELECT EXTRACT (MONTH FROM TIMESTAMP '07/31/2008 8:15:00 pm') FROM db_root;</pre>
add_months()	주어진 날짜에 달을 더한 후의 날짜 <pre>SELECT ADD_MONTHS (DATE '2/29/1980',12) FROM db_root;</pre>
last_day()	주어진 날짜의 달의 마지막 날짜 <pre>SELECT LAST_DAY (DATE '1/01/2000') FROM db_root;</pre>
months_between()	주어진 날짜간의 개월수 <pre>SELECT MONTHS_BETWEEN (DATE '8/15/2008',DATE '8/15/2006') FROM db_root;</pre>
sys_date sysdate current_date	현재 날짜 <pre>SELECT SYS_DATE FROM db_root;</pre>
sys_time systime current_time	현재 시간 <pre>SELECT SYS_TIME FROM db_root;</pre>
sys_timestamp systimestamp current_timestamp	현재 타임스탬프 <pre>SELECT SYS_TIMESTAMP FROM db_root;</pre>

5.4 숫자 함수 1

연산자	의미
mod()	주어진 수를 두 번째 인자로 나눈 나머지 <pre>SELECT MOD(10.945, 0.4) FROM db_root;</pre>
rand(), drand()	임의의 정수 난수값 0~1사이구간에서 질의에 대해 임의의 실수 난수값 발생 결과가 여러 개일 경우 동일한 난수값 <pre>SELECT DRAND() FROM db_root;</pre>
random(), drandom()	임의의 정수 난수값 0~1사이구간에서 질의결과에 대해 임의의 실수 난수값 발생 결과가 여러 개일 경우 서로 다른 난수값 <pre>SELECT DRANDOM() FROM db_root;</pre>
greatest()	주어진 값들 중 가장 큰 수
least()	주어진 값들 중 가장 큰 수
power()	제곱값
sign()	주어진 숫자의 부호
abs()	절대값

5.4 숫자 함수 2

연산자	의미
ceil()	주어진 값보다 크거나 같은 정수 <code>SELECT CEIL(-10.145) FROM db_root;</code>
floor()	주어진 값보다 작거나 같은 정수 <code>SELECT FLOOR(-10.145) FROM db_root;</code>
round()	주어진 수를 소수점 기준 두번째 인자 자리수까지 남기고 그 다음 자리에서 반올림 <code>SELECT ROUND(15.456, 1) FROM db_root;</code>
trunc()	주어진 수를 소수점 기준 두번째 인자 자리수까지 남기고 그 다음 자리부터는 버림 <code>SELECT TRUNC(235.1538, 1) FROM db_root;</code>

5.5 집계 함수 1

연산자	의미
count()	그루핑된 레코드의 개수 <pre>SELECT sum(gold), max(gold), min(gold), avg(gold), variance(gold), stddev(gold), count(*) from participant where host_year = 1988</pre>
min()	그루핑된 레코드들에 대하여 주어진 인자의 최소값
max()	그루핑된 레코드들에 대하여 주어진 인자의 최대값
sum()	그루핑된 레코드들에 대하여 주어진 인자의 합계
avg()	그루핑된 레코드들에 대하여 주어진 인자의 평균값
stddev()	그루핑된 레코드들에 대하여 주어진 인자의 표준편차값
variance()	그루핑된 레코드들에 대하여 주어진 인자의 분산

5.6 문자열 함수 1

연산자	의미
length()	문자열 길이를 구함 (byte 단위) <pre>SELECT LENGTH('Cubrid') FROM db_root;</pre>
position()	주어진 문자열에서 다른 문자열의 시작위치를 구함 <pre>SELECT POSITION ('of' IN 'United States of America') FROM db_root;</pre>
instr()	position() 과 동일하나 문자열 찾는 위치 지정 가능 <pre>SELECT INSTR('United States of America', 'United', 1) FROM db_root;</pre>
substring()	주어진 문자열을 특정 위치에서 주어진 개수만큼 잘라 넘겨줌 <pre>SELECT SUBSTRING('United States of America' FROM 9 FOR 5) FROM db_root;</pre>
substr()	substring() 과 동일하나, 위치나 개수가 ≤ 0 일 경우 동작방식이 다 름 <pre>SELECT SUBSTR('United States of America', -6, 2) FROM db_root;</pre>

5.6 문자열 함수 2

연산자	의미
trim(), ltrim(), rtrim()	문자열의 양쪽, 왼쪽, 오른쪽에 대하여 trim 문자열을 제거
lpad(), rpad()	문자열의 왼쪽, 오른쪽에 문자열 추가
stddev()	그룹핑된 레코드들에 대하여 주어진 인자의 표준편차값
upper(), lower()	문자열을 대문자, 소문자로 변환
replace()	주어진 문자열에서 검색문자열을 찾아 치환문자열이 있는 경우 치환하고, 없는 경우 삭제
translate()	주어진 문자열에서 검색문자 각각에 대해 치환문자 각각의 문자로 치환하고, 대응하는 문자가 없는 경우 삭제

5.7 조건/비교 함수 1

연산자	의미
case	주어진 조건식에 따라 결과 값을 결정 <pre>SELECT name, CASE WHEN gender = 'M' THEN 'MAN' WHEN gender = 'W' THEN 'WOMAN' END as GENDER FROM athlete;</pre>
decode()	주어진 조건값에 따라서 결과 값을 결정 <pre>SELECT DECODE(LENGTH('Korea'), 4, 'length: 4', 5, 'length: 5', 6, 'length: 6', 'length: unknown') FROM db_root;</pre>
coalesce	주어진 값이 NULL 인 경우 다른 값으로 변환 <pre>SELECT host_year, COALESCE(mascot, 'Not Exist') FROM olympic;</pre>
nvl	coalesce 와 동일 <pre>SELECT host_year, NVL(mascot, 'Not Exist') FROM olympic;</pre>

5.7 조건/비교 함수 2

nvl2	주어진 값이 NULL 이 아니면 첫번째 인자값, NULL 이면 두번째 인자값 <pre>SELECT host_year, NVL2(mascot, 'Exist', 'Not Exist') FROM olympic;</pre>
nullif()	주어진 두개의 값이 같으면 NULL, 다른 경우 첫번째 인자 <pre>SELECT host_year, NULLIF(mascot, 'HODORI') FROM olympic;</pre>

5.8 ROWNUM 함수

질의 수행결과 생성되는 결과 레코드에 대하여 일련번호 부여하는 함수
order by, group by 전에 생성되므로 order by, group by 사용시 결과 주의
임의 번째의 결과 선택 가능

```
SELECT *  
  FROM (SELECT nation_code FROM participant  
        WHERE host_year = 1988  
        ORDER BY gold DESC, silver DESC, bronze DESC)  
 AS T  
WHERE ROWNUM <5;
```

Question.

The background of the slide is an abstract composition of overlapping, semi-transparent blue shapes. These shapes, which vary in shades from light sky blue to a deeper cerulean, are primarily triangular and polygonal in form. They are arranged in a way that creates a sense of depth and movement, with some shapes appearing to recede into the background while others come forward. The overall effect is a modern, clean, and professional aesthetic.