



DataBase

DataBase의 정의

- Data의 집합체
- Data의 추가, 수정, 삭제 가능
- Data의 정렬 및 열람가능

다양한 DataBase

ORACLE®



Microsoft®
SQL Server®

IBM

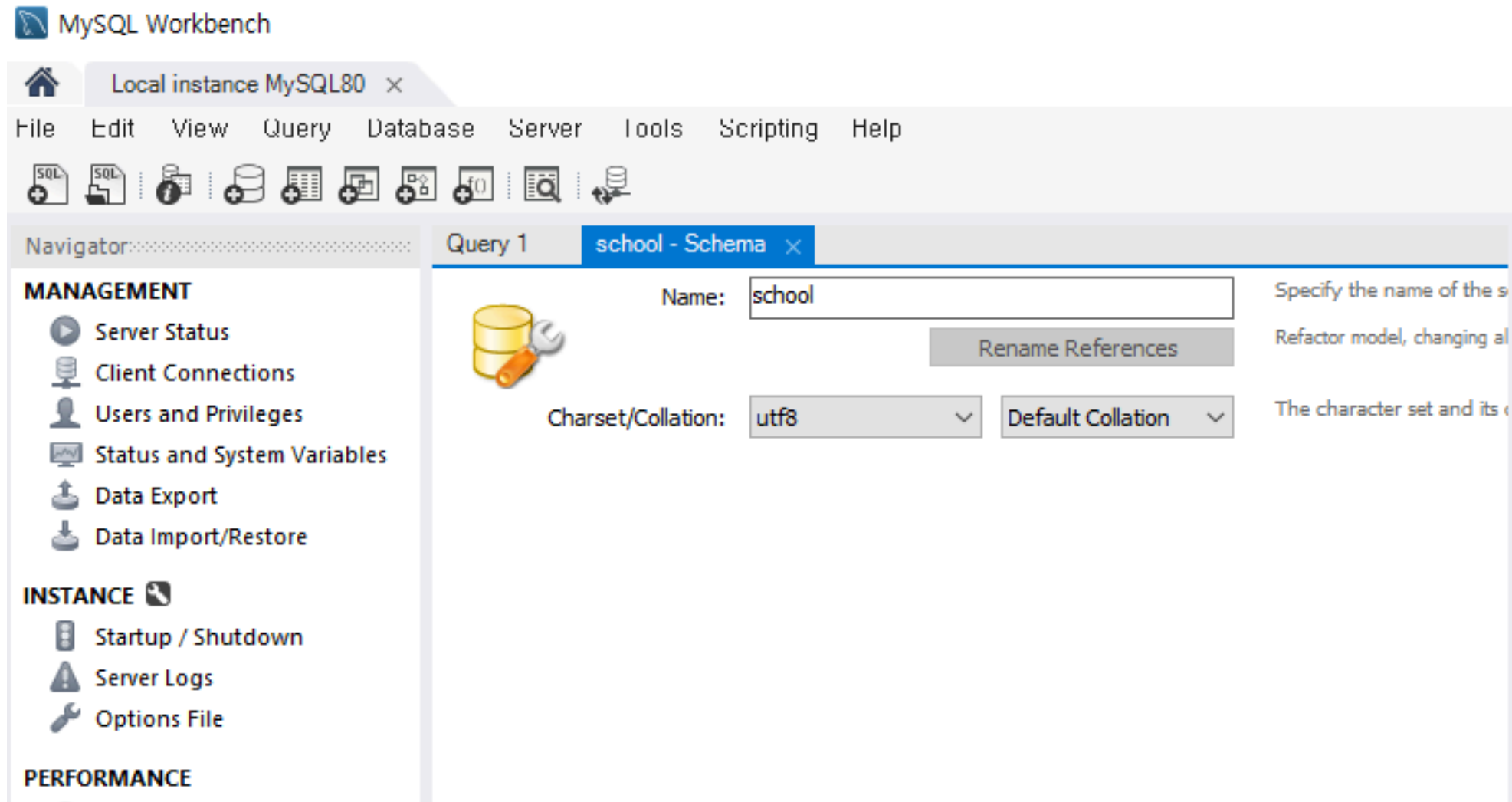
DB2

SYBASE®

MySQL®

SQLite

Schema 생성



SQL의 이해

- SQL-Structured Query Language
- 인간의 언어와 흡사(영어)
- ISO에서 표준제정
- SQL-92(SQL2)
- 대부분의 DBMS는 SQL-92를 지원.

테이블구성

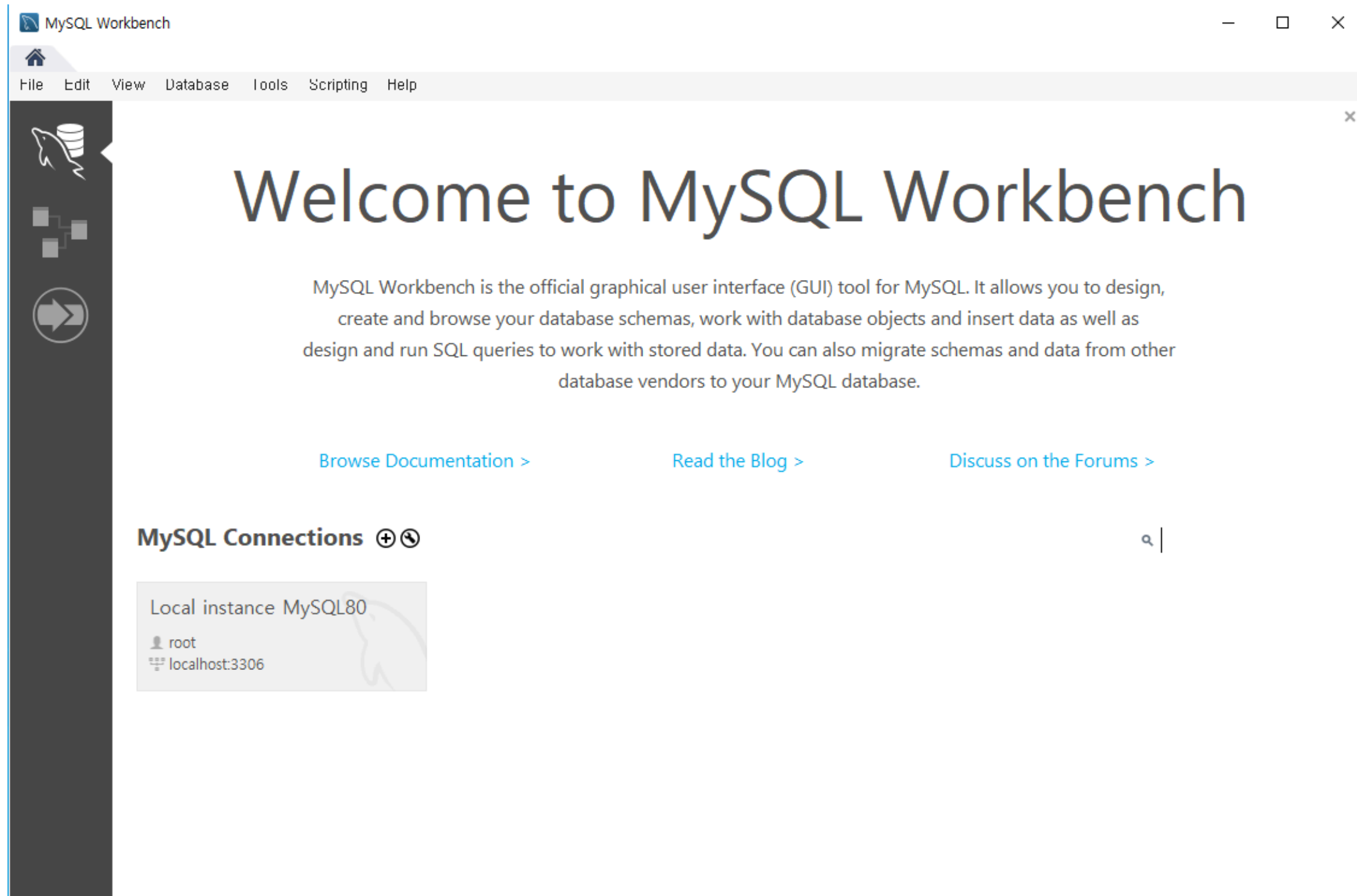
학생정보 테이블구조

학번을 나타내는 필드(Field), 필드명은 ID					
성명을 나타내는 필드(Field), 필드명은 Name			학년을 나타내는 필드, 필드명은 Grade		
ID	Name	Major	Grade	Gender	
1001	현빈	정보통신	1	남	학번1001에 대한 Record
1002	길라임	스턴트학	1	여	학번1002에 대한 Record
1003	홍길동	도술학	2	남	
1004	이명박	토목공학	4	?	

MySQL의 주요 데이터형

- INT
- FLOAT, DOUBLE
- CHAR(n) : 1~255의 고정길이
- VARCHAR(n): 1~255의 가변길이
- TEXT : ~65535길이의 문자데이터
- DATETIME

MySQL Workbench



테이블 생성





Table Name:






Schema:

Charset/Collation:

Engine:

Comments:



Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI
 id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
 name	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 major	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 grade	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 gender	CHAR(1)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

<

>

Column Name:

Data Type:

Charset/Collation:

Default:

Comments:

Storage: ☐ Virtual ☐ Stored

SQL CURD

- Create : INSERT
- Retrieve(Read) : SELECT
- Update : UPDATE
- Delete : DELETE

INSERT

INSERT INTO 테이블명(컬럼명1, 컬럼명2...) VALUES(값1, 값2....)

INSERT INTO students(name, major, grade, gender)
VALUES('RM', '정보통신', 1, '남')

INSERT INTO students(name, major)
VALUES('RM', '정보통신')

INSERT INTO students
VALUES('RM', '정보통신', 1, '남')

INSERT

```
INSERT INTO students(name, major, grade, gender)
VALUES('RM', '정보통신', 1, '남')
```

```
INSERT INTO students(name, major)
VALUES('제이홉', '컴퓨터공학')
```

```
INSERT INTO students
VALUES(3, '슈가', '영문학', 1, '남');
```

```
INSERT INTO students(id, name, major, grade, gender)
VALUES(4, '정국', '토목공학', 4, '남');
```

SELECT

SELECT (컬럼명1, 컬럼명2...) FROM 테이블명

SELECT id, name, major, grade, gender FROM students

SELECT * FROM students

SELECT id, name, major FROM students

SELECT * FROM students WHERE grade>2

WHERE 조건문

- =, <, >, <=, >=, <>, != 등의 비교연산자지원
- 필드명 연산자 값 (grade>2, gender='여')
- 두개이상의 조건절- AND로 연결
- grade>2 AND gender='여'
- LIKE 조건문
- BETWEEN 조건문

WHERE 조건문

```
SELECT * FROM students WHERE grade>2
```

```
SELECT * FROM students  
WHERE grade>2 AND gender='남'
```

```
SELECT * FROM students WHERE major='정보통신'
```

```
SELECT * FROM students  
WHERE grade>2 AND gender='남' AND major='정보통신'
```

LIKE 조건문

필드명 LIKE '정보%'

필드명 LIKE '%통신'

필드명 LIKE '현_'

필드명 LIKE '_빈'

LIKE 조건문

SELECT * FROM students WHERE major LIKE '정보%'

SELECT * FROM students WHERE major LIKE '%통신'

SELECT * FROM students WHERE name LIKE '현_'

SELECT * FROM students WHERE name LIKE '_빈'

BETWEEN 조건문

WHERE grade $2 \geq$ AND grade ≤ 3

WHERE grade BETWEEN 2 AND 3

SELECT * FROM students

WHERE grade BETWEEN 2 AND 3

IN 조건문

WHERE grade IN (2,3)

WHERE major IN ('정보통신', '컴퓨터공학')

SELECT * FROM students WHERE grade IN (2,3)

SELECT * FROM students
WHERE major IN ('정보통신', '컴퓨터공학')

NOT 연산자

WHERE not Grade = 2

WHERE NOT major IN ('정보통신', '토목공학')

WHERE NOT grade BETWEEN 2 AND 3

WHERE NOT major LIKE '정보%'

NOT 연산자

```
SELECT * FROM students WHERE NOT grade = 2
```

```
SELECT * FROM students  
WHERE NOT major IN ('정보통신', '컴퓨터공학')
```

```
SELECT * FROM Students  
WHERE NOT BETWEEN 2 AND 3
```

```
SELECT * FROM Students  
WHERE NOT Major LIKE '정보%'
```

UPDATE

UPDATE 테이블 SET 필드명=값

UPDATE 테이블 SET 필드명1=값1, 필드명2=값2...

UPDATE 테이블 SET 필드명=값 WHERE 조건식

UPDATE 테이블 SET 필드명1=값1, 필드명2=값2... 조건식

UPDATE

UPDATE students SET grade=1 Where name='RM'

UPDATE 테이블 SET 필드명1=값1, 필드명2=값2...

UPDATE 테이블 SET 필드명=값 WHERE 조건식

UPDATE 테이블 SET 필드명1=값1, 필드명2=값2... 조건식

DELETE

DELETE FROM 테이블 WHERE 조건식

DELETE FROM students WHERE grade = 1

DELETE FROM students WHERE major LIKE '컴퓨터%'

DELETE FROM students
WHERE grade BETWEEN 2 AND 3

Primary Key

- 중복되지않는 유일한 값(Unique)
- Null을 가질 수 없음
- 한테이블에 하나만 가질 수 있음

Primary Key

```
CREATE TABLE students (  
  id int PRIMARY KEY NOT NULL AUTO_INCREMENT,  
  name varchar(50) NOT NULL ,  
  major varchar(50) NOT NULL ,  
  grade int,  
  gender char(1))
```

INSERT

```
INSERT INTO students(id, name, major, grade, gender)
VALUES(1,'RM','정보통신',1, '남' );
```

```
INSERT INTO students(id, name, gender)
VALUES(2,'제이홉','여' );
```

```
INSERT INTO students
VALUES(3,'슈가','도술학',3, '남' );
```

```
INSERT INTO students(id, name, major, grade, gender)
VALUES(4,'정국','토목공학',4, '?' );
```

관계형 데이터베이스

학번	성명	전공	학년	성별
1	RM	정보통신	1	남
2	제이홉	컴퓨터공학	2	여
3	슈가	도술학	3	남
4	정국	토목공학	4	남

관계형 데이터베이스

학번	성명	전공	학년	성별	차량번호	차종
1	RM	정보통신	1	남	5387	그랜저
2	제이홉	컴퓨터공학	2	여	2011	소나타
3	슈가	도술학	3	남		
4	정국	토목공학	4	남		

관계형 데이터베이스

학번	성명	전공	학년	성별
1	RM	정보통신	1	남
2	제이홉	컴퓨터공학	2	여
3	슈가	도술학	3	남
4	정국	토목공학	4	남

차량번호	차종
5387	그랜저
2011	소나타

관계형 데이터베이스

학번	성명	전공	학년	성별
1	RM	정보통신	1	남
2	제이홉	컴퓨터공학	2	여
3	슈가	도술학	3	남
4	정국	토목공학	4	남

차량번호	차종	학번
5387	그랜저	1
2011	소나타	2

관계형 데이터베이스

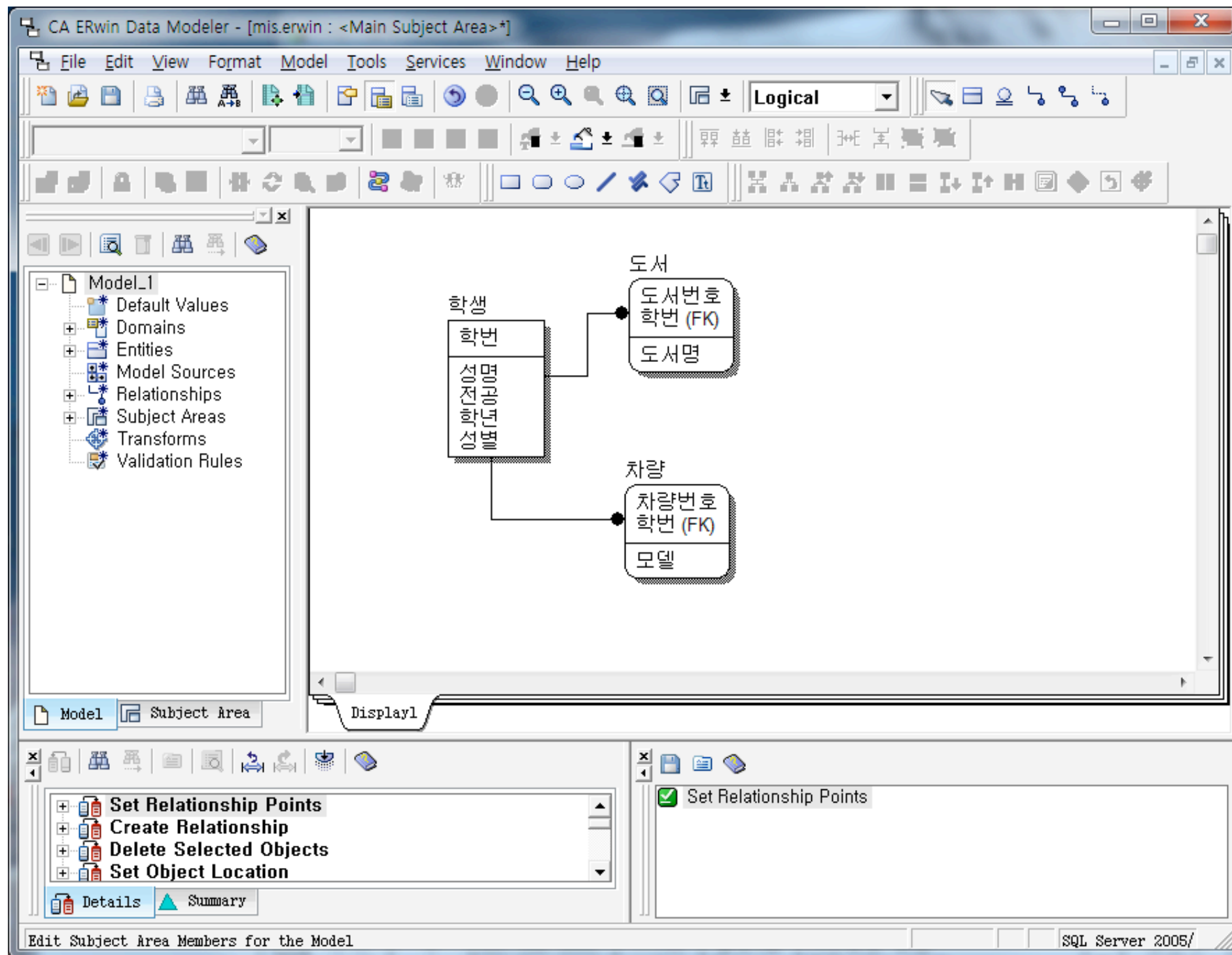
학번	성명	전공	학년	성별	도서번호	도서명
1	RM	정보통신	1	남	1234	android Progrademming
2	제이홉	컴퓨터공학	2	여	1235	스턴트개론
3	슈가	도술학	3	남	1237	슈가전
4	정국	토목공학	4	남		
5	RM	정보통신	1	남	1238	Core Data
6	RM	정보통신	1	남	1239	Cocoa Framework

관계형 데이터베이스

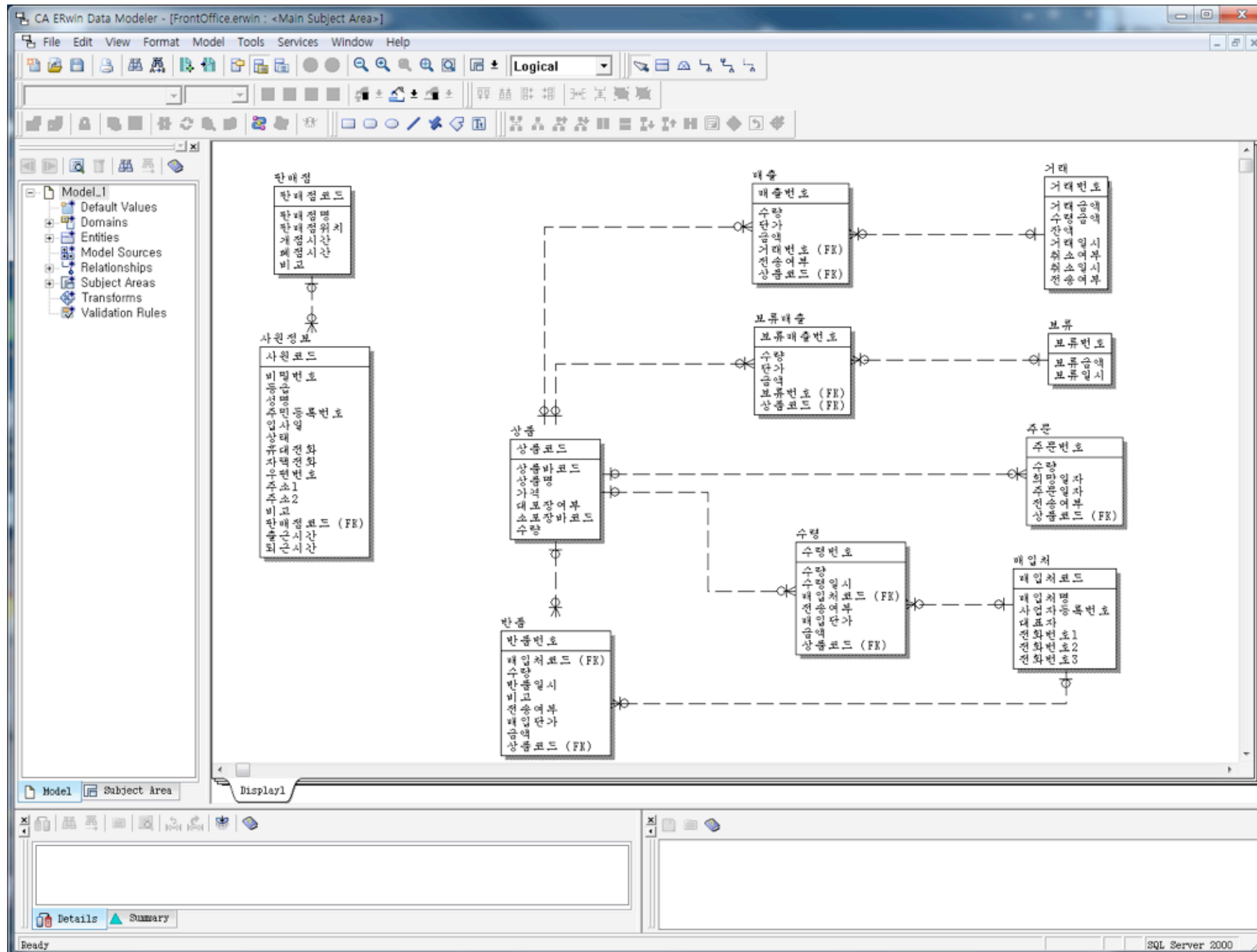
학번	성명	전공	학년	성별
1	RM	정보통신	1	남
2	제이홉	컴퓨터공학	2	여
3	슈가	도술학	3	남
4	정국	토목공학	4	남

도서 번호	도서명	학번
1234	android Progrademming	1
1235	스턴트개론	2
1237	슈가전	3
1238	Core Data	1
1239	Cocoa Framework	1

관계형 데이터베이스



관계형 데이터베이스



FOREIGN KEY

```
CREATE TABLE cars (  
no varchar(30) PRIMARY KEY NOT NULL UNIQUE ,  
model varchar(30) NOT NULL ,  
students_id int NOT NULL,  
FOREIGN KEY(students_id) REFERENCES students(id)  
)
```

FOREIGN KEY

```
CREATE TABLE books (  
no varchar(30) PRIMARY KEY NOT NULL UNIQUE ,  
title varchar(50) NOT NULL ,  
students_id int NOT NULL,  
FOREIGN KEY(students_id) REFERENCES students(id)  
)
```

FOREIGN KEY 제약

```
INSERT INTO cars(no, model, students_id)  
VALUES('5387','그랜저',1 );
```

```
INSERT INTO cars(no, model, students_id)  
VALUES('2011','소나타',2);
```

```
INSERT INTO cars(no, model, students_id)  
VALUES('5389','그랜저',5 );
```

FOREIGN KEY 제약

```
INSERT INTO books(no, title, students_id)
VALUES('1234', 'android Progrademming', 1);
```

```
INSERT INTO books(no, title, students_id)
VALUES('1235', '데이터베이스개론', 2);
```

```
INSERT INTO books(no, title, students_id)
VALUES('1237', '슈가전', 3);
```

```
INSERT INTO books(no, title, students_id)
VALUES('2001', '정의란 무엇인가?', 5);
```

INNER JOIN

학번	성명	전공	학년	성별	차량번호	모델
1	RM	정보통신	1	남	5387	그랜저
2	제이홉	컴퓨터공학	2	여	2011	소나타

SELECT 테이블명.컬럼명1, 테이블명.컬럼명2....

FROM 테이블1 INNER JOIN 테이블2

ON join조건문

INNER JOIN

```
SELECT id, name, major, no, model  
FROM students INNER JOIN cars  
ON students.id=cars.students_id
```

```
SELECT students.id, students.name,  
students.major, cars.no, cars.model  
FROM students INNER JOIN cars  
ON students.id=cars.students_id
```

```
SELECT students.id, students.name,  
students.major, cars.no, cars.model  
FROM students INNER JOIN cars  
ON students.id=cars.students_id WHERE grade >=2
```

INNER JOIN

```
SELECT s.id, s.name, s.major, c.no, c.model  
FROM students AS s INNER JOIN cars AS c  
WHERE s.id=c.students_id AND s.grade <4
```

```
SELECT s.id, s.name, s.major, c.no, c.model  
FROM students s INNER JOIN cars c  
WHERE s.id=c.students_id AND s.grade <4
```

OUTER JOIN

학번	성명	전공	학년	성별	차량번호	차종
1	RM	정보통신	1	남	5387	그랜저
2	제이홉	컴퓨터공학	2	여	2011	소나타
3	슈가	도술학	3	남		
4	정국	토목공학	4	남		

OUTER JOIN

```
SELECT 테이블명.컬럼명1, 테이블명.컬럼명2....  
FROM 테이블1 LEFT OUTER JOIN 테이블2  
ON Join조건문
```

```
SELECT s.id, s.name, s.major, c.no, c.model  
FROM students s LEFT OUTER JOIN cars c  
ON s.id=c.students_id;
```

```
SELECT s.id, s.name, s.major, b.no, b.title  
FROM students s LEFT OUTER JOIN books b  
ON s.id=b.students_id;
```

테이블 수정

테이블명 수정

```
ALTER TABLE students rename TO student;
```

컬럼추가

```
ALTER TABLE students ADD addr varchar(50);
```

트랜잭션

BEGIN;

DELETE cars(no, model, students_id) VALUES('5387','그랜저',1);

INSERT INTO cars VALUES('53875','그랜저2',1);

ROLLBACK;

BEGIN;

DELETE cars(no, model, students_id) VALUES('5387','그랜저',1);

INSERT INTO cars VALUES('53875','그랜저2',1);

COMMIT;