

1. 개요 - 1.1 SQL 역사

- ❖ SQL 질의어의 시조 - SEQUEL
- ❖ 1976년 IBM 에서 기능을 보완하여 SEQUEL 2로 발전.
 - DBMS를 제공하는 회사에서 각기 상이한 형태의 질의어를 제공하였기 때문에 사용자가 다른 DBMS를 이용하기 위해서는 그 시스템의 질의어를 새롭게 익혀야하는 문제점이 있었음.
- ❖ 1986년 ANSI 와 ISO가 공동으로 SQL의 표준화 작업을 진행
- ❖ 현재 표준화 작업은 계속 진행되고 있음.
- ❖ 본 책에서는 1992년 채택된 SQL/92를 기준으로 함.

SQL 역사

SQUARE (Specifying Queries as Relational Expressions)

SEQUEL (Structured English QUery Language)

- 초기 SQL은 SEQUEL로 명명
- [Cham74] D. Chamberlin, R. Boyce, "SEQUEL ; A Structured English Query Language", in SIGMOD, 1974.

SEQUEL 2

- SQL의 기초
- 구현 : IBM Almaden Research Center
- [Cham76] D. Chamberlin, et al., "SEQUEL2 ; A Unified Approach to Data Definition, Manipulation, and Control", IBM Journal of Research and Development, 20:6, November, 1976.

SQL 1

- ANSI와 ISO에서 표준안 제안
- [ANSI86] ANSI, "The Database Language SQL, Document ANSI X3.135", 1986.



1. 개요 - 1.2 SQL 구성

- ❖ 데이터 정의어(**DDL**) : 데이터가 저장되는 테이블(릴레이션)을 생성, 변경, 삭제하는 기능을 제공하며, 테이블에 저장된 데이터를 빨리 접근하기 위한 인덱스(index)를 정의하고, 뷰(view)를 정의할 수 있는 기능을 포함.
- ❖ 데이터 조작어(**DML**) : 테이블에 데이터를 삽입하고, 저장 데이터 중 조건에 맞는 것들을 검색하며, 불필요한 데이터를 삭제하는 기능을 제공.
 - 대화식 **DML** : 사용자가 직접 데이터베이스에 접근하여 이용.
 - 삽입 **DML** : 개발자들이 응용 프로그램 제작 시에 이용.

SQL 질의어 구성

SQL

데이터 정의어 (DDL : Data Definition Language)

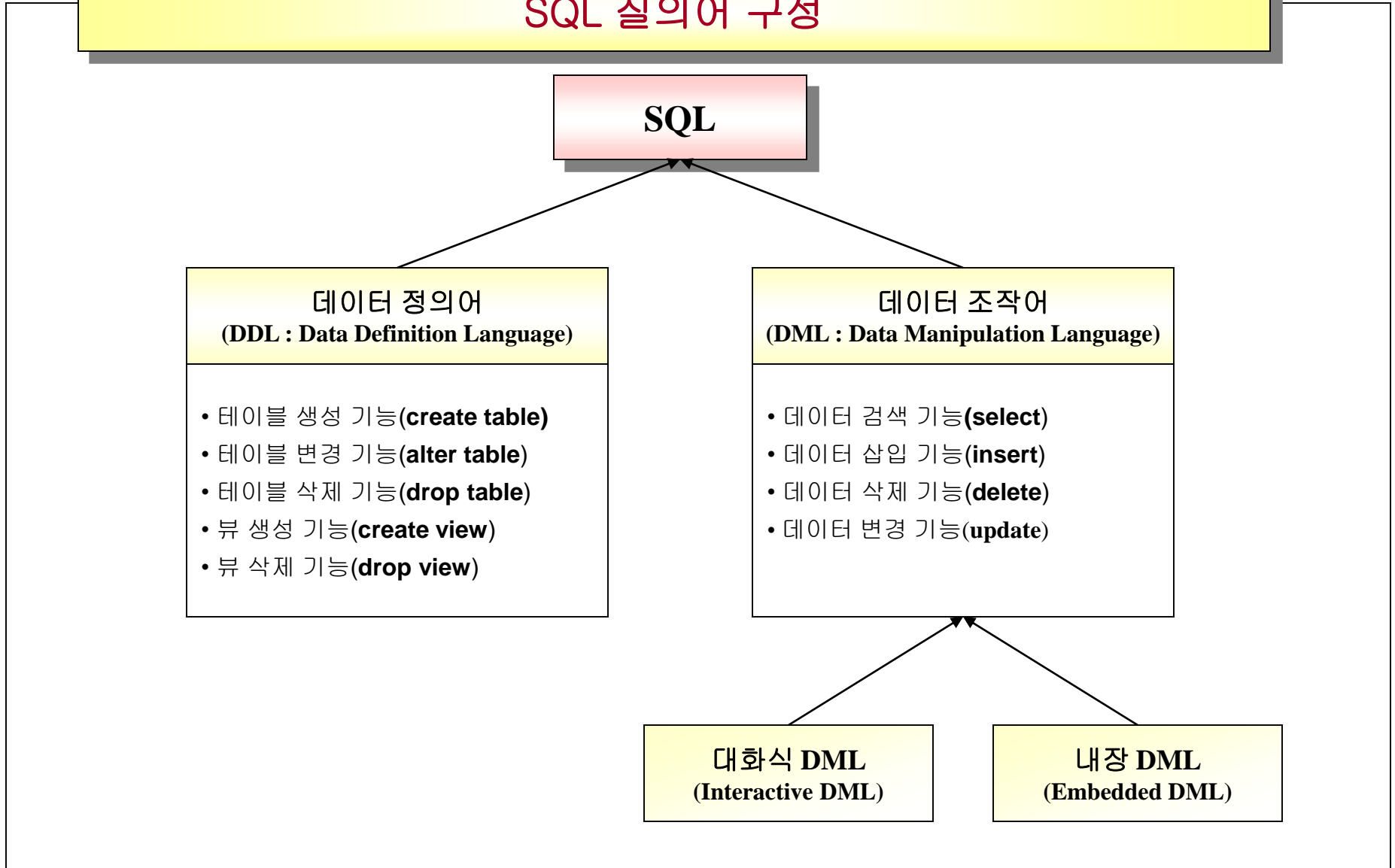
- 테이블 생성 기능(**create table**)
- 테이블 변경 기능(**alter table**)
- 테이블 삭제 기능(**drop table**)
- 뷰 생성 기능(**create view**)
- 뷰 삭제 기능(**drop view**)

데이터 조작용어 (DML : Data Manipulation Language)

- 데이터 검색 기능(**select**)
- 데이터 삽입 기능(**insert**)
- 데이터 삭제 기능(**delete**)
- 데이터 변경 기능(**update**)

대화식 DML
(Interactive DML)

내장 DML
(Embedded DML)



예제 데이터베이스의 항목 설명

학과(DEPARTMENT)	: 학과에 대한 정보를 포함하는 테이블
학과코드(dept_no)	: 학과를 유일하게 식별할 수 있는 속성 값
학과명(dept_name)	: 학과 이름
학과장(chief)	: 학과장을 맡고 있는 교수의 교수번호
조교(assistant)	: 학과의 조교 이름
전화(dept_tel)	: 학과의 전화번호
위치(location)	: 학과가 위치한 건물의 방 번호
교수(PROFESSOR)	: 교수에 대한 정보를 포함하는 테이블
교수번호(p_no)	: 교수를 유일하게 식별할 수 있는 속성 값
이름(p_name)	: 교수의 이름
직급(grade)	: 교수의 직급
소속학과(dept_no)	: 교수가 소속된 학과의 학과코드
연구실(room_no)	: 교수의 연구실이 위치한 건물의 방 번호
전화(p_tel)	: 교수 연구실의 전화번호
주소(p_address)	: 교수의 집 주소
생년월일(p_birth)	: 교수의 생년월일
입사일(join_date)	: 교수의 입사일
호봉(pay_step)	: 교수의 호봉

예제 데이터베이스의 항목 설명

강좌(COURSE)	: 강좌에 대한 정보를 포함하는 테이블
강좌번호(c_no)	: 강좌를 유일하게 식별할 수 있는 속성 값
강좌이름(c_name)	: 강좌의 이름
시수(times)	: 강좌의 시수
학점(point)	: 강좌의 학점
필수/선택(R/O)	: 강좌의 필수·선택 여부
학생(STUDENT)	: 학생에 대한 정보를 포함하는 테이블
학번(s_no)	: 학생을 유일하게 식별할 수 있는 속성 값
이름(s_name)	: 학생의 이름
학과(dept)	: 학생이 소속된 학과의 이름
학년(year)	: 학생의 학년
전화(s_tel)	: 학생의 집 전화번호
주소(s_address)	: 학생의 집 주소
지도교수(advisor)	: 학생의 지도교수를 가리키는 교수번호
생년월일(s_birth)	: 학생의 생년월일

예제 데이터베이스의 항목 설명

강의(LECTURE)	: 강의에 대한 정보를 포함하는 테이블
강좌번호(c_no)	: 강의가 개설된 강좌번호
교수번호(p_no)	: 강의가 개설된 강좌를 담당하는 교수번호
요일(day_week)	: 강의 요일
시작시간(start_time)	: 강의 시작 시간
강의실(room_no)	: 강의가 진행되는 강의실 번호
등록(ENROLLMENT)	: 등록에 대한 정보를 포함하는 테이블
강좌번호(c_no)	: 학생이 강좌를 등록한 번호
학번(s_no)	: 강좌를 등록한 학생의 번호
성적(grade)	: 해당 강좌의 수강 성적
점수(merits)	: 해당 강좌의 수강 점수

예제 데이터베이스(1/2)

학과 (DEPARTMENT)

학과코드 dept_no	학과명 dept_name	학과장 chief	조교 assistant	전화 dept_tel	위치 location
001	컴퓨터공학과	0001	이지은	3333	T101
002	정보과학과	0015	김영미	3334	T201
003	정보통신공학과	0022	박순애	3335	T301
004	정보산업공학과	0033	이미선	3336	T401
005	전자공학과	0045	이춘애	3337	T501
006	문헌정보학과	0051	김기범	3338	T601
007	전기공학과	0063	박준용	3339	T701

교수 (PROFESSOR)

교수번호 p_no	이름 p_name	직급 grade	소속학과 dept_no	연구실 room_no	전화 p_tel	주소 p_address	생년월일 p_birth	입사일 join_date	호봉 pay_step
0001	김수철	교수	001	T102	4102	서울 ...	19520401	00880301	35
0002	이기문	교수	001	T103	4103	null	19540605	00890901	33
0003	박동진	부교수	003	T302	4302	서울 ...	19610105	00910301	26
0004	이재무	교수	003	C303	4303	서울 ...	19501102	00910901	31
0005	고병오	조교수	004	C402	4402	서울 ...	19631207	00910901	06
0006	오영덕	교수	004	T403	4403	null	19580201	00920301	30
0007	오성균	교수	004	T404	4404	서울 ...	19540405	00930301	30
0008	안우영	조교수	002	T202	4202	서울 ...	19651102	00930901	19
0009	이강우	부교수	001	T104	4104	일산 ...	19600205	00940301	22
0010	김홍일	조교수	001	T105	4105	서울 ...	10660305	00940301	20
0011	이순미	조교수	002	C203	4203	서울 ...	19641108	00950301	21
0012	최의인	부교수	002	T204	4204	서울 ...	19590209	00950301	24
0013	박선주	조교수	003	T304	4304	의정부 ...	19650201	00960301	08
0014	홍명희	교수	003	T305	4305	서울 ...	19531212	00960301	33
0015	김충훈	조교수	004	T404	4404	서울 ...	19661215	00960301	16
0016	김병근	조교수	005	C502	4502	서울 ...	19651213	00970301	12
0017	최종민	조교수	005	T503	4503	서울 ...	19661103	00970901	13
0018	신판섭	조교수	006	T602	4602	수원 ...	19631202	00980301	12
0019	서진형	조교수	006	T603	4603	성남 ...	19621205	00980301	11
0020	안인순	부교수	007	T702	4702	인천 ...	19621225	00980301	25

예제 데이터베이스(2/2)

학생 (STUDENT)

학번 s_no	이름 s_name	학과 dept	학년 year	전화 s_tel	주소 s_address	지도교수 advisor	생년월일 s_birth
9911001	정 화우	컴퓨터공학과	2	023355519	서울...	0001	19800505
9912002	김정길	정보학과	2	03463355411	경기도...	0008	19801215
9911003	황재우	컴퓨터공학과	2	0325491115	인천...	0002	19801119
9813009	이수자	정보통신공학과	3	0428601144	대전...	0003	19791205
9811015	이길년	컴퓨터공학과	3	023386634	서울...	0009	19791130
9714016	박창홍	정보산업학과	4	024444555	서울...	0005	19781025
9815011	김대웅	전자공학전공	3	024555505	서울...	0016	19790504
9914001	이기성	정보산업학과	2	023385055	서울...	0006	19801008
9914002	이미자	정보산업학과	2	028754440	서울...	0007	19800407
9716001	김을수	컴퓨터공학과	4	029234676	서울...	0018	19781207
9716004	김영수	컴퓨터공학과	4	029879922	서울...	0019	19781212
9917001	인소람	전기학과	2	024474213	서울...	0020	19801104
9717005	박창홍	전기학과	4	026436294	서울...	0020	19781105
9917051	정기정	전기학과	2	028901234	서울...	0020	19801025
9612035	다현수	정보통신공학과	3	023332345	서울...	0008	19791106
9913045	신미영	정보통신공학과	2	023201673	서울...	0004	19800507
9817066	김창우	전기학과	3	025404683	서울...	0020	19790403
9715011	한미라	전자공학전공	4	0325401288	인천...	0017	19780407
9611035	양미라	컴퓨터공학과	4	03445421345	경기도...	0009	19780905
9913045	박을수	정보통신공학과	2	024876822	서울...	0013	19801008
9713041	한은희	정보통신공학과	4	023879090	서울...	0014	19781005
9812041	이재환	정보학과	3	022745678	서울...	0011	19790506
9915011	나기홍	전자공학과	2	028709876	서울...	0016	19800530

강의 (LECTURE)

강좌번호 c_no	교수번호 p_no	요일 day_week	시작시간 start_time	강의실 room_no
0001	0001	mon	1	T447
0001	0002	tue	5	T448
0001	0005	wed	1	T337
0001	0006	thu	1	T337
0002	0003	mon	4	T447
0002	0004	tue	1	T448
0003	0006	wed	3	T337
0003	0007	thu	7	T337
0004	0008	mon	6	T447
0004	0009	tue	3	T448
0004	0010	wed	1	T337
0005	0012	thu	2	T337
0006	0013	mon	3	T447
0008	0020	mon	7	T448

강좌 (COURSE)

강좌번호 c_no	강좌이름 c_name	시수 times	학점 point	필수/선택 R/O
0001	정보사회와 컴퓨터	3	3	R
0002	데이터베이스	3	3	R
0003	네트워크	3	3	R
0004	C언어	3	3	O
0005	자료구조	3	3	R
0006	컴퓨터구조	3	3	R
0007	운영체제	3	3	R
0008	시뮬레이션	2	2	O

등록 (ENROLLMENT)

강좌번호 c_no	학번 s_no	성적 grade	점수 merits
0001	9911001	A+	95
0001	9912002	B+	82
0001	9911003	A0	90
0001	9813009	C+	75
0001	9811015	D+	65
0001	9714016	A0	91
0002	9815011	B+	86
0002	9914001	C0	70
0002	9914002	D0	60
0002	9716001	A0	90
0002	9716004	A0	90
0003	9917001	B+	85
0003	9717005	B+	86
0003	9917051	C+	76
0003	9612035	D+	67
0004	9913045	C+	78
0004	9817066	D+	69
0004	9715011	D+	68
0004	9611035	F	40
0005	9913045	B+	86
0005	9713041	B+	87
0005	9812041	C0	70
0005	9915011	C0	71
0006	9911001	B+	87
0006	9912002	B+	88
0006	9911003	B0	81
0006	9813009	A+	96
0008	9811015	A0	90
0008	9716004	C+	76
0008	9917001	C+	78
0008	9717005	B0	82

2. 데이터 정의 기능

SQL 질의어에서 지원하는 대표적인 데이터 정의 기능은 테이블생성 (create), 변경(alter), 삭제(drop) 기능이다.

❖ 2.1 테이블 생성 [create table]

- 명령어 : create table
- 구문형식

```
create table 테이블_이름  
  (열_이름 데이터_유형 [not null].....  
   [primary key(열_이름_리스트)]  
   [check(조건식)] );
```

2. 데이터 정의 기능

❖ 2.1.1 무결성 제약 검사

- ① **not null – not null** 로 표기된 열_이름은 속성 값으로 널(null)값을 가질 수 없음을 나타냄.

dept_no integer not null

- ② **primary key – primary key** 로 지정한 열_이름_리스트는 해당 테이블을 유일하게 식별할 수 있는 기본키임을 나타냄

primary key(c_no, p_no)

- ③ **check – check** 절은 테이블의 특정 열(속성)이 지켜져야 할 조건을 지정.

check(1 <= year <= 4)

2.1.2 데이터 유형 요약

유 형	의 미
char(n)	사용자가 지정한 길이(n) 만큼의 고정 길이 문자열을 정의하는 것으로, character 로 표기해도 된다.
varchar(n)	사용자가 지정한 최대 길이(n)만큼의 가변 길이 문자열을 정의하는 것으로, character varying 으로 표기해도 된다.
int	정수형 데이터를 정의하는 것으로, integer 로 표기해도 된다. 일반적으로 정수형 데이터의 길이는 사용 컴퓨터 기종마다 상이하기 때문에 컴퓨터 기종에 의존적이다.
smallint	Small integer 또는 short integer 를 의미하는 것으로, 데이터 길이는 컴퓨터 기종에 의존적이다.
numeric(p,d)	사용자가 정의한 고정 소수점 수(fixed point number)로서, p 는 소수점을 기준으로 왼쪽 부분에 위치하는 숫자의 길이(부호 포함)를 가리키며, q 는 소수점을 기준으로 오른쪽 부분에 위치하는 숫자의 길이를 가리킨다. 만약 데이터 유형이 numeric(4,3) 으로 정의되었다면, 333.202는 저장이 가능하나, 4444.1 이나 333.4444 등은 저장이 불가능하다.

2.1.2 데이터 유형 요약

유 형	의 미
real	부동 소수점 수(floating point number)를 의미하는 것으로, 데이터 길이는 컴퓨터 기종에 의존적이다.
double precision	배정도(double precision) 부동 소수점 수를 의미하는 것으로, 데이터 길이는 컴퓨터 기종에 의존적이다.
float(n)	.사용자가 지정한 길이(n) 만큼의 부동 소수점 수를 의미한다.
date	년, 월, 일을 포함하는 날짜 유형이다.
time	시, 분, 초를 포함하는 시간 유형이다.

2.1.2 데이터 유형 요약

예제 1

<그림 5-4> 예제 데이터베이스의 학과 (department) 릴레이션을 생성하는 SQL 질의어를 작성하라. 단, 기본키는 dept_no이고, 기본키 값은 널 값을 가질 수 없다.

create table department

 (dept_no **integer** **not null,**

 dept_name **char(14),**

 chief **integer,**

 assistant **char(8),**

 dept_tel **integer,**

 location **char(4),**

primary key (dept_no));

2.1.2 데이터 유형 요약

예제 2

<그림 5-4> 예제 데이터베이스의 교수(**PROFESSOR**) 릴레이션을 생성하는 SQL 질의어를 작성하라. 단, 기본키는 **P_NO** 이고, 기본키 값은 널 값을 가질 수 없다.

create table professor

(p_no	integer not null,
p_name	char(10),
grade	char(6),
dept_no	integer,
room_no	char(4),
p_tel	integer,
p_address	char(30),
p_birth	date,
join_date	date,
pay_step	integer,
primary key (p_no));	

2.1.2 데이터 유형 요약

예제 3

<그림 5-4> 예제 데이터베이스의 교수(**PROFESSOR**) 릴레이션을 생성하는 SQL 질의어를 작성하라. 단, 기본키는 **P_NO** 이고, 기본키 값은 널 값을 가질 수 없다. 또한, 시수(times)는 1부터 5이내의 값을 가져야 한다.

```
create table course
  (c_no           integer           not null,
   c_name        char(20),
   times         char(1),
   point         char(1),
   R_O           char(1),
   primary key (c_no),
   check (1 <= times and times <= 5 ));
```


2.1.2 데이터 유형 요약

예제 4

<그림 5-4> 예제 데이터베이스의 학생(STUDENT) 릴레이션을 생성하는 SQL 질의어를 작성하라. 단, 기본키는 s_no이고, 기본키 값은 널 값을 가질 수 없다,

create table student

(s_no	integer	not null,
s_name	char(10),	
dept	char(14),	
year	integer,	
s_tel	integer,	
s_address	char(30),	
advisor	integer,	
s_birth	date,	
primary key	(s_no)) ;	

2.1.2 데이터 유형 요약

예제 5

<그림 5-4> 예제 데이터베이스의 강의(LECTURE)릴레이션을 생성하는 SQL 질의어를 작성하라. 단 기본키는 c_no와 p_no이고, 기본키 값은 널 값을 가질 수 없다. 또한, start_times은 1부터 9이내의 값을 가져야 한다.

create table lecture

(c_no integer not null,

p_no integer not null,

day_week char(3),

start_time char(1),

room_no char(4),

primary key (c_no, p_no),

check (1 <= start_time and start_time <= 9));

2.1.2 데이터 유형 요약

예제 6

<그림 5-4> 예제 데이터베이스의 등록(ENROLLMENT)릴레이션을 생성하는 SQL 질의어를 작성하라. 단, 기본키는 c_no 와 s_no이고, 기본키 값은 널 값을 가질 수 없다.

create table enrollment

(c_no	integer	not null,
s_no	integer	not null,
grade	char(2),	
merits	integer,	
primary key	(c_no, s_no));	

2.2 테이블 제거와 변경

❖ 2.2.1 테이블 제거 (drop table)

- 명령어 : **drop table**
- 구문형식

```
drop table 테이블_이름 cascade | restrict ;
```

2.2 테이블 제거와 변경

예제 7

<그림 5-4> 예제 데이터베이스의 학과(DEPARTMENT) 릴레이션을 제거하는 SQL 질의어를 작성하라. 단 학과 릴레이션을 참조하고 있는 다른 릴레이션도 함께 제거하라.

```
drop table department cascade ;
```

예제 8.

<그림 5-4> 예제 데이터베이스의 학과 (DEPARTMENT) 릴레이션을 제거하는 SQL 질의어를 작성하라. 단, 학과 릴레이션을 다른 릴레이션이 참고하고 있을 경우 제거 명령을 실행하지 않도록 하라.

```
drop table department restrict ;
```

2.2 테이블 제거와 변경

❖ 2.2.2 테이블 변경 (alter table)

- 명령어 : alter table
- 구문형식

<속성 추가>

```
alter table 테이블_이름 add 열_이름 데이터_유형
```

<속성 제거>

```
alter table 테이블_이름 drop 열_이름 cascade | restrict ;
```

2.2 테이블 제거와 변경

예제 9

<그림 5-4> 예제 데이터베이스의 학과 (DEPARTMENT) 릴레이션에 교수수(nbrOfFaculty) 속성을 추가하는 SQL 질의어를 작성하라. 단, 교수수 (nbrOfFaculty) 속성의 데이터 유형은 integer 로 지정하라.

```
alter table department add nbrOfFaculty integer;
```

예제 10

<그림 5-4> 예제 데이터베이스의 학과 (DEPARTMENT) 릴레이션에서 전화(dept_tel) 속성을 제거하는 SQL 질의어를 작성하라. 단, 학과 릴레이션의 전화 속성을 다른 릴레이션이 참조하고 있을 경우 함께 제거하라.

```
drop table department drop dept_tel cascade ;
```

3. 데이터 조작 기능

SQL 질의어에서 지원하는 대표적인 데이터 조작 기능은 크게 기존의 데이터에 데이터를 삽입(insert)하고, 테이블에서 원하는 데이터를 검색(select)하며, 필요한 데이터를 삭제(delete)하거나 변경된 데이터 값을 갱신(update)하는 기능이다.

❖ 3.1 데이터 검색[select]

- 명령어 : select
- 구문형식

```
select [all | distinct ] 속성_리스트 [as 제목_리스트 ]  
from 릴레이션_리스트  
[where 조건]  
[group by 속성_리스트  
[having 조건] ]  
[order by 속성_리스트 [ asc | desc ] ] ;
```


select 명령에서 이용가능한 명령어 요약

명령어	의미
all	중복을 포함한 속성 결과 검색 (아무런 표시를 하지 않을 경우 기본 값으로 all 을 지니게 된다.
distinct	중복을 제거한 속성 결과 검색
as	검색결과에의 제목을 지정하는 명령
where	조건을 지정할 수 있는 명령
group by	그룹별 검색 기능을 수행하는 명령
having	그룹별 검색 기능에 조건을 추가한 명령
order by	정렬 기능을 수행하는 명령
asc	Order by 명령 사용 시에 오름차순 명령을 지정하는 것으로, ascending 의 약자이다. 만약 아무런 표시를 하지 않을 경우 기본 값으로 asc 값을 가지게 된다.
desc	Order by 명령 사용 시에 내림차순 명령을 지정하는 것으로, descending 의 약자이다.

3.1.1 하나의 릴레이션에서의 검색

예제 11 릴레이션의 모든 내용 검색

<그림 5-4> 예제 데이터베이스의 학과 (DEPARTMENT) 릴레이션에 등록된 모든 정보를 검색하라.

select *

from department ;

결과 :

학과코드 dept_no	학과명 dept_name	학과장 chief	조교 assistant	전화 dept_tel	위치 location
001	컴퓨터공학과	0001	이지은	3333	T101
002	정보과학과	0015	김영미	3334	T201
003	정보통신공학과	0022	박순애	3335	T301
004	정보산업공학과	0033	이미선	3336	T401
005	전자공학과	0045	이춘애	3337	T501
006	문헌정보학과	0051	김기범	3338	T601
007	전기공학과	0063	박준용	3339	T701

3.1.1 하나의 릴레이션에서의 검색

예제 12 특정 속성 검색

<그림 5-4> 예제 데이터베이스의 등록(ENROLLMENT) 릴레이션에 등록된 강좌번호(c_no)를 검색하라.

```
select c_no  
from enrollment;
```

결 과 :

강좌번호 c_no
0001
0001
0001
0001
0001
0001
0002
0002
0002
0002
0002
0002
0003
0003
0003
0003
0004
0004
0004
0004
0005
0005
0005
0005
0006
0006
0006
0006
0008
0008
0008
0008

3.1.1 하나의 릴레이션에서의 검색

예제 13 중복을 제거한 특정 속성 검색

<그림 5-4> 예제 데이터베이스의 등록(ENROLLMENT) 릴레이션에 등록된 강좌번호를 검색하라.

```
select distinct c_no  
from enrollment;
```

결과

강좌번호 c_no
0001
0002
0003
0004
0005
0006
0008

3.1.2 조건을 지정한 검색

예제 14 **where** 절을 이용한 검색(1)

<그림 5-4> 예제 데이터베이스의 학생(STUDENT) 릴레이션에서
학번이 9911003인 학생의 지도교수를 검색하라.

```
select advisor  
from student  
where s_no='9911003';
```

결 과

지도교수 advisor
0002

3.1.2 조건을 지정한 검색

예제 15 where 절을 이용한 검색(2)

<그림 5-4> 예제 데이터베이스의 교수(**PROFESSOR**) 릴레이션에서 교수번호가 **0016**인 교수의 이름, 직급, 호봉을 검색하라.

```
select p_name, grade, pay_step  
from professor  
where p_no='0016';
```

결 과 :

이름 p_name	직급 grade	호봉 pay_step
김병곤	조교수	16

3.1.2 조건을 지정한 검색

예제 16 where 절을 이용한 검색(3)

<그림 5-4> 예제 데이터베이스의 강좌(COURSE) 릴레이션에서
강좌번호가 0002인 강좌의 강좌이름, 시수, 학점을 검색하라.

```
select c_name, times, point
from course
where c_no='0002';
```

결 과 :

강좌이름 c_name	시수 times	학점 point
데이터베이스	3	3

3.1.2 조건을 지정한 검색

<예제 17> **where** 절을 이용한 검색(4)

<그림 5-4> 예제 데이터베이스의 등록(ENROLLMENT) 릴레이션에서
학번이 **9911001**인 학생이 등록한 강좌번호를 검색하라.

```
select c_no  
from enrollment  
where s_no='9911001';
```

결 과 :

강좌번호 c_no
0001
0006

3.1.3 집단 함수를 이용한 검색

- ❖ SQL 질의어를 이용하여 데이터 검색시에 자주 사용하는 계산 기능들 (예 : 개수, 계산, 합산, 평균, 최대 최소값 계산 등) 집단함수 또는 열 함수로 제공한다.

집단함수 요약

집단 함수	의 미
COUNT	값의 개수
SUM	값의 총합
AVG	평균값
MAX	최대값
MIN	최소값

3.1.3 집단 함수를 이용한 검색

예제 18 집단 함수를 이용한 검색(1)

<그림 5-4> 예제 데이터베이스의 교수(**PROFESSOR**) 릴레이션에 등록된 교수 수는 얼마인지 검색하라.

```
select COUNT(*)
```

```
from professor;
```

결 과 : 20

예제 19 집단 함수를 이용한 검색(2)

<그림 5-4> 예제 데이터베이스의 등록(**ENROLLMENT**) 릴레이션에 등록된 강좌 수는 얼마인지 검색하라.

```
select COUNT(*)
```

```
from enrollment;
```

결 과 : 31

3.1.3 집단 함수를 이용한 검색

예제 20 집단 함수를 이용한 검색(3)

<그림 5-4> 예제 데이터베이스의 등록(ENROLLMENT) 릴레이션에 등록된 강좌 수는 얼마인지 검색하라. 단 중복을 제거하여 라.

```
select COUNT(distinct c_no)
from enrollment;
```

결 과 : 7

3.1.4 AS 를 이용한 검색

예제 21 as 명령을 이용한 검색(1)

<그림 5-4> 예제 데이터베이스의 교수(**PROFESSOR**) 릴레이션에 등록된 교수 수가 얼마인지 교수 수라는 제목으로 출력하라.

```
select COUNT(*) as 교수 수  
from professor;
```

예제 22 as 명령을 이용한 검색(2)

<그림 5-4> 예제 데이터베이스의 등록(**ENROLLMENT**) 릴레이션에 등록된 강좌 수가 얼마인지 강좌 수라는 제목으로 출력하라.
단 중복을 제거하여 계산하라.

```
select COUNT(distinct c_no) as 강좌 수  
from enrollment;
```

결 과 :

강좌 수

7

3.1.4 AS 를 이용한 검색

예제 23 as 명령을 이용한 검색(3)

<그림 5-4> 예제 데이터베이스의 강의(LECTURE) 릴레이션에서 교수 번호가 0005인 교수가 강의하는 과목의 수는 몇 개인지 강의 강좌 수라는 제목으로 출력하라. 단 중복을 제거하여 계산하라.

```
select COUNT(distinct c_nbr) as 강의 강좌 수
from lecture
where prof_nbr='0005';
```

결 과 :

강의강좌수

1

3.1.5 산술식을 이용한 검색

- ❖ SQL 질의어를 구성함에 있어 산술식을 이용해야 하는 경우에는 DBMS 에서 지원하는 연산자와 함수를 이용할 수 있다.

대표적인 함수

구 분	연 산 자	의 미
산술식 연산자	사칙 연산자	+(덧셈), -(뺄셈), *(곱셈), /(나눗셈)
수치 함수	abs(n) mod(n,b) sqrt(n)	n은 절대값 (단, n 은 수치형 데이터) n 을 b 로 나눈 나머지 값 (단, n 과 b는 정수형 데이터) n 의 루트값 (단, n 은 정수형 또는 실수형 데이터)
문자열 함수	length (str) Lower(str) Upper(str)	문자열 str 의 길이 문자열 str을 소문자로 변환 문자열 str 을 대문자로 변환

3.1.5 산술식을 이용한 검색

예제 24 산술식을 이용한 검색

<그림 5-4> 예제 데이터베이스의 등록(ENROLLMENT) 릴레이션에서
강좌번호 0002를 수강한 학생 중 학번이 9914002인 학생의 점수를 10점 더하여
여 검색하라. 단, 출력 결과를 최종점수라는 제목으로 출력하라.

```
select merits+10 as 최종점수
from enrollment
where c_nbr='0002' and s_nbr='9914002';
```

결 과 :

최종 점수

70

3.1.6 문자열 연산식을 이용한 검색 (like)

- ❖ SQL 질의어 작성 시에 문자열(string)을 포함한 조건식을 이용할 경우, like 명령어를 사용한다.

SQL 문자열 연산자

구 분	연 산 자
%(percent)	0 개 이상의 문자(문자열)를 매칭시키는 연산자(예, DOS 명령어의 * 와 동일한 의미).
- (underscore)	한 개의 문자를 매칭시키는 연산자에 (예, DOS 명령의 ? 와 동일한 의미).
\ (backslash)	특수 문자 (문자열 연산자) 를 일반 문자로 사용할 수 있도록 지정하는 연산자 예를 들어, 문자열 연산자인 %, _ , \ 를 일반문자로 사용하고자 할 때, \ 를 문자의 앞부분에 기입한다.

3.1.6 문자열 연산식을 이용한 검색 (like)

문자열 연산식 사용 예

구 분	연 산 자
“ 서울 % “	‘서울’로 시작하는 문자열 (‘서울’도 포함)
“% 직할시% ”	‘직할시’를 포함하는 문자열 (‘직할시’도 포함)
“%0301”	‘0301’로 끝나는 문자열 (‘0301’도 포함).
“ _ _ _ _ _ ”	5 개의 문자로 구성된 문자열
“ _ _ % ”	2개 이상의 문자로 구성된 문자열
“96 \ % % ”	‘96%’로 시작하는 문자열
“100 \\ % ”	‘100\’로 시작하는 문자열

3.1.6 문자열 연산식을 이용한 검색 (like)

예제 25 문자열 연산식을 이용한 검색(1)

<그림 5-4> 예제 데이터베이스의 교수(**PROFESSOR**) 릴레이션에서 '서울' 지역에 사는 교수들의 교수번호와 이름을 검색하라.

```
select p_no, p_name
from professor
where p_address like '서울%';
```

결 과 :

교수번호 p_no	이름 p_name
0001	김수철
0003	박동진
0004	이재무
0005	고병오
0007	오성균
0008	안우영
0010	김홍일
0011	이순미
0012	최의인
0014	홍명희
0015	김중훈
0016	김병곤
0017	최종민

3.1.6 문자열 연산식을 이용한 검색 (like)

예제 26 문자열 연산식을 이용한 검색(2)

<그림 5-4> 예제 데이터베이스의 교수(**PROFESSOR**) 릴레이션에서 연구실이 'C동'에 있는(즉, 연구실 번호가 C로 시작하는) 교수들의 교수번호와 이름을 검색하라.

```
select p_no, p_name
from professor
where room_no like 'C%';
```

또는

```
select p_no, p_name
from professor
where room_no like 'C____';
```

결 과 :

교수번호 p_no	이름 p_name
0004	이재무
0005	고병오
0011	이순미
0016	김병곤

3.1.6 문자열 연산식을 이용한 검색 (like)

예제 27 문자열 연산식을 이용한 검색(3)

<그림 5-4> 예제 데이터베이스의 학과(DEPARTMENT) 릴레이션에서 학과 이름에 '정보'가 포함되어 있는 학과들의 학과코드와 학과이름을 검색하라.

```
select dept_no, dept_name
from department
where dept_name like '%정보%';
```

결 과 :

학과코드 dept_no	학과명 dept_name
002	정보과학과
003	정보통신공학과
004	정보산업공학과
006	문헌정보학과

3.1.7 정렬 기능을 이용한 검색 (order by)

- ❖ **order by** 명령을 이용하여 사용자가 지정한 기준으로 검색결과를 출력할 수 있다.
 - 오름차순 : **order by** 명령만 추가.
 - 내림차순 : **desc** 를 추가.

3.1.7 정렬 기능을 이용한 검색 (order by)

예제 28 정렬 기능을 이용한 검색(1)

<그림 5-4> 예제 데이터베이스의 등록(ENROLLMENT) 릴레이션에서 학생의 점수를 오름차순으로 정렬한 강좌번호, 학번, 점수를 검색하라.

```
select c_no, s_no, merits
from enrollment
order by merits;
```

또는

```
select c_no, s_no, merits
from enrollment
order by merits asc;
```

결 과 :

강좌번호 c_no	학번 s_no	점수 merits
0004	9611035	40
0002	9914002	60
0001	9811015	65
0003	9612035	67
0004	9715011	68
0004	9817066	69
0005	9812041	70
0002	9914001	70
0005	9915011	71
0001	9813009	75
0008	9716004	76
0003	9917051	76
0008	9917001	78
0004	9913045	78
0006	9911003	81
0008	9717005	82
0001	9912002	82
0005	9917001	86
0003	9913045	86
0005	9717005	86
0002	9815011	86
0006	9911001	87
0005	9713041	87
0006	9912002	88
0008	9811015	90
0002	9716004	90
0002	9716001	90
0008	9911003	90
0001	9714016	91
0001	9911001	95
0006	9813009	96

3.1.7 정렬 기능을 이용한 검색 (order by)

예제 29 정렬 기능을 이용한 검색(2)

<그림 5-4> 예제 데이터베이스의 등록(ENROLLMENT) 릴레이션에서 학생의 점수를 내림차순으로 정렬한 강좌번호, 학번, 점수를 검색하라.

```
select c_no, s_no, merits
from enrollment
order by merits desc;
```

결 과 :

강좌번호 c_no	학번 s_no	점수 merits
0006	9813009	96
0001	9911001	95
0001	9714016	91
0001	9911003	90
0002	9716001	90
0002	9716004	90
0008	9811015	90
0006	9912002	88
0005	9713041	87
0006	9911001	87
0002	9815011	86
0003	9717005	86
0005	9913045	86
0003	9917001	85
0001	9912002	82
0008	9717005	82
0006	9911003	81
0004	9913045	78
0008	9917001	78
0003	9917051	76
0008	9716004	76
0001	9813009	75
0005	9915011	71
0002	9914001	70
0005	9812041	70
0004	9817066	69
0004	9715011	68
0003	9612035	67
0001	9811015	65
0002	9914002	60
0004	9611035	40

3.1.7 정렬 기능을 이용한 검색 (order by)

예제 30 정렬 기능을 이용한 검색(3)

<그림 5-4> 예제 데이터베이스의 교수(PROFESSOR) 릴레이션에서 교수의 호봉을 내림차순으로 정렬한 교수번호, 교수이름, 호봉을 검색하라.

```
select p_no, p_name, pay_step
from professor
order by pay_step desc;
```

결 과 :

교수번호 p_no	이름 p_name	호봉 pay_step
0001	김수철	35
0002	이기문	33
0014	홍명희	23
0004	이재무	31
0006	오영덕	30
0007	오성균	30
0003	박동진	26
0020	안인순	25
0012	최의인	24
0009	이강우	22
0011	이순미	21
0010	김홍일	20
0008	안우영	19
0015	김중훈	16
0017	최종민	13
0016	김병곤	12
0018	신판섭	12
0019	서진형	11
0013	박선주	08
0005	고병오	06

3.1.8 그룹별 검색 기능을 이용한 검색(group by)

- ❖ SQL 명령중 group by 는 지정된 속성(열)의 값이 일치하는 값들을 모아서 그룹을 만드는 역할을 수행한다.

예제 31 그룹별 검색 기능을 이용한 검색(1)

<그림 5-4> 예제 데이터베이스의 등록(ENROLLMENT) 릴레이션에서 개설된 강좌번호를 검색하라.

```
select c_no  
from enrollment  
group by c_no;
```

또는

```
select distinct c_no  
from enrollment
```

결 과 :

강좌번호 c_no
0001
0002
0003
0004
0005
0006
0008

3.1. 8 그룹별 검색 기능을 이용한 검색(group by)

예제 32 그룹별 검색 기능을 이용한 검색(2)

<그림 5-4> 예제 데이터베이스의 등록(ENROLLMENT) 릴레이션에서 개설된 강좌별로 등록한 학생의 수를 검색하라.

```
select c_no, count(s_no) as 등록학생수
from enrollment
group by c_no;
```

결 과 :

강좌번호 c_no	등록학생수
0001	6
0002	5
0003	4
0004	4
0005	4
0006	4
0008	4

3.1. 8 그룹별 검색 기능을 이용한 검색(group by)

예제 33 그룹별 검색 기능을 이용한 검색(3)

<그림 5-4> 예제 데이터베이스의 등록(ENROLLMENT) 릴레이션에서 개설된 강좌별로 등록한 학생의 수와 점수의 총합을 검색하라.

```
select c_no, count(s_no) as 등록학생수, sum(merits) as 총점
from enrollment
group by c_no;
```

결 과 :

강좌번호 c_no	등록학생수	총점
0001	6	498
0002	5	396
0003	4	314
0004	4	255
0005	4	314
0006	4	352
0008	4	326

3. 1. 8 그룹별 검색 기능을 이용한 검색(group by)

예제 34 그룹별 검색 기능을 이용한 검색(4)
<그림 5-4> 예제 데이터베이스의 학생(STUDENT) 릴레이션에서 학과를 기준으로 소속 학생들의 평균 학년을 검색하라.

```
select dept, avg(year) as 평균학년  
from student  
group by dept;
```

결 과 :

학과 dept	평균학년
컴퓨터공학과	2.75
정보과학과	2.67
정보통신공학과	2.75
정보산업공학과	2.67
전자공학과	3
문헌정보학과	4
전기공학과	2.75

3. 1. 8 그룹별 검색 기능을 이용한 검색(group by)

예제 35 그룹별 검색 기능을 이용한 검색(5)

<그림 5-4> 예제 데이터베이스의 학생(STUDENT) 릴레이션에서 학과를 기준으로 소속 학생들의 학생 수와 평균 학년을 검색하라.

```
select dept, count(s_no) as 학생수 avg(year) as 평균학년
from student
group by dept;
```

결 과 :

학과 dept	학생수	평균학년
컴퓨터공학과	4	2.75
정보과학과	3	2.67
정보통신공학과	4	2.75
정보산업공학과	3	2.67
전자공학과	3	3
문헌정보학과	2	4
전기공학과	4	2.75

3. 1. 9 그룹별 검색 기능에 조건을 추가한 검색(having)

- ❖ SQL 질의어에서 그룹별 검색 기능을 이용하기 위하여 **group by** 명령을 이용할 때, 검색 조건을 추가하기 위해서는 **having** 절을 이용한다.

예제 36 **having** 절을 이용한 검색

<그림 5-4> 예제 데이터베이스의 등록(ENROLLMENT) 릴레이션에서 5명 이상의 학생들이 수강한 과목의 최고점수와 최저점수 및 평균점수를 수강과목 기준으로 검색하라.

```
select c_no, max(merits) as 최고점수, min(merits) as 최저점수, avg(merits) as 평균점수
from enrollment
group by c_no
having count(*) >= 5;
```

결 과 :

강좌번호 c_no	최고점수	최저점수	평균점수
0001	95	65	83
0002	90	60	79.2

3. 1. 10 Null 값을 이용한 검색

- ❖ 아직 결정되지 않은 값(unknown value)을 의미하는 Null 값을 이용하여 SQL 질의어를 작성할 수 있으며, 이 때 사용하는 명령이 is [not] null 이다.

예제 37 Null 값을 이용한 검색(1)

<그림 5-4> 예제 데이터베이스의 교수(PROFESSOR) 릴레이션에서 주소가 Null 값인 교수의 번호와 이름을 검색하라.

```
select p_no, p_name  
from professor  
where p_address is null;
```

결 과 :

교수번호 p_no	이름 p_name
0002	이기문
0005	오영덕

3. 1. 10 Null 값을 이용한 검색

예제 38 Null 값을 이용한 검색(2)

<그림 5-4> 예제 데이터베이스의 학과(DEPARTMENT) 릴레이션에서 학과명이 Null 값이 아닌 학과코드를 검색하라.

```
select dept_no
from department
where dept_name is not null;
```

결 과 :

학과코드 dept_no
001
002
003
004
005
006
007

3. 1. 11 조인 연산을 이용한 검색

- ❖ 두 개 이상의 릴레이션을 조인하기 위해서는 동일한 도메인을 갖는 속성들을 연결(조인)하여야 하며, 이 때 사용하는 속성을 조인 속성(join attribute)라 한다.1

예제 39 조인 연산을 이용한 검색

<그림 5-4> 예제 데이터베이스의 학생(STUDENT) 릴레이션에서 학번이 9911001인 학생의 학번, 이름, 지도교수 이름을 검색하라.

```
select s_no, s_name, p_name as 지도교수
from student, professor
where s_no = '9911001' and advisor = p_no;
```

결 과 :

학번 s_no	이름 s_name	지도교수
9911001	정찬우	김수철

3. 1. 12 한정된 열이름을 사용한 검색

- ❖ 두 개 이상의 릴레이션에 조인한 경우 공통적으로도 동일한 이름의 속성들이 서로 다른 릴레이션에 존재할 수 있다. 이러한 경우 동일한 이름의 속성이 어느 릴레이션에 속하는 것인지를 나타내기 위하여 한정자 표기법을 사용한다.

예제 40 한정된 열이름을 사용한 검색(1)

<그림 5-4> 예제 데이터베이스의 강의(LECTURE) 릴레이션에서 개설된 강좌의 강좌번호, 강좌이름, 시수, 학점을 검색하라.

```
select distinct c_no, c_name, times, point
from course, lecture
where course.c_no = lecture.c_no;
```

결 과 :

강좌번호 c_no	강좌이름 c_name	시수 times	학점 point
0001	정보사회와 컴퓨터	3	3
0002	데이터베이스	3	3
0003	네트워크	3	3
0004	C언어	3	3
0005	자료구조	3	3
0006	컴퓨터구조	3	3
0008	시뮬레이션	2	2

3. 1. 12 한정된 열이름을 사용한 검색

예제 41 한정된 열이름을 사용한 검색(2)

<그림 5-4> 예제 데이터베이스의 등록(ENROLLMENT) 릴레이션에서 0001 강좌를 수강하는 학생들의 학번, 이름, 성적을 검색하라.

```
select enrollment.s_no, s_name, grade
from student, enrollment
where c_no = '0001' and enrollment.s_no = student.s_no;
```

결 과 :

학번 s_no	이름 s_name	성적 grade
9911001	정찬우	A+
9912002	김정길	B+
9911003	강재우	A0
9813009	이수자	C+
9811015	이길녀	D+
9714016	박찬중	A0

3. 1. 13 릴레이션 별칭(correlation name)을 사용한 검색

- ❖ 동일한 이름의 속성들이 서로 다른 릴레이션에 존재하여 한정자 표기법을 사용할 경우, 릴레이션의 전체이름을 사용하지 않고 별칭(correlation name)을 정의하여 사용할 수 있다. 릴레이션의 별칭을 사용할 경우 **from** 절에서 목표 릴레이션 이름 뒤에 별칭을 정의한다.

예제 42 릴레이션 별칭을 사용한 검색(1)

<그림 5-4> 예제 데이터베이스의 강의(LECTURE) 릴레이션에서 0001 강좌를 강의하는 교수들의 교수번호와 이름을 검색하라.

```
select p_no, p_name
from lecture l, professor p
where c_no='0001' and l.p_no=p.p_no;
```

결 과 :

교수번호 p_no	이름 p_name
0001	김수철
0002	이기문
0005	고병오

3. 1. 13 릴레이션 별칭(correlation name)을 사용한 검색

예제 43 릴레이션 별칭을 사용한 검색(2)

<그림 5-4> 예제 데이터베이스의 강의(LECTURE) 릴레이션에서 0001 강좌의 강좌번호와 강좌이름, 이 강좌를 강의하는 교수들의 교수번호와 이름을 검색하라.

```
select c_no, c_name, p_no, p_name
from course c, lecture l, professor p
where c_no='0001' and c.c_no=l.c_no and l.p_no=p.p_no;
```

결 과 :

강좌번호 c_no	강좌이름 c_name	교수번호 p_no	이름 p_name
0001	정보사회와 컴퓨터	0001	김수철
0001	정보사회와 컴퓨터	0002	이기문
0001	정보사회와 컴퓨터	0005	고병오

3. 1. 14 부속질의(subquery)를 이용한 검색

- ❖ 부속질의(subquery)란 다른 SQL 명령(예, select, insert, delete, update 등)의 where 절에 내포된(nested) 질의를 일컫는 것으로, 부속 질의에는 select 절과 from 절이 꼭 포함되어야 한다. 이와 같은 부속 질의는 상위 질의보다 먼저 실행되어야 하기 때문에 괄호(())로 묶여야 한다.

부속질의 작성 시 이용되는 키워드

키워드	의미
ALL	<ul style="list-style-type: none"> ▶ 부속질의에서 검색된 모든 결과를 만족할 경우 참(true) ▶ θ all (부속질의) θ는 {<, <=, =, <>, >, >=} 중 하나를 사용
ANY	<ul style="list-style-type: none"> ▶ 부속질의에서 검색된 결과 중 최소한 하나를 만족할 경우 참(true) ▶ θ any (부속질의) θ는 {<, <=, =, <>, >, >=} 중 하나를 사용
IN	<ul style="list-style-type: none"> ▶ 부속질의에서 검색된 결과에 대하여 멤버십 연산자(\in)를 수행 ▶ 형식 : [not] in (부속질의)
EXISTS	<ul style="list-style-type: none"> ▶ 부속질의에서 검색된 결과가 존재(nonempty)하는지를 검사 ▶ 형식 : [not] exists (부속질의)

3. 1. 14 부속질의(subquery)를 이용한 검색

예제 44 in 명령을 이용한 검색(1)

<그림 5-4> 예제 데이터베이스에서 학번이 9911051인 학생이 소속된 학과의 학과장 교수번호, 조교이름, 학과의 전화번호 등을 검색하라.

```
select chief, assistant, dept_tel
from department
where dept_name in (select
                    from student
                    where s_no='9911001');
```

또는

```
select chief, assistant, dept_tel
from department, student
where s_no='9911001' and dept_name=dept;
```

결 과 :

학과장 chief	조교 assistant	전화 dept_tel
0001	이지은	3333

3. 1. 14 부속질의(subquery)를 이용한 검색

예제 45 in 명령을 이용한 검색(2)

<그림 5-4> 예제 데이터베이스에서 0002 과목을 등록한 학생 중 성적이 A+와 A0인 학생의 학번, 성적, 점수를 검색하라.

```
select s_no, grade, merits
from enrollment
where c_no='0002' and grade in (A+, A0);
```

결 과 :

학번 s_no	성적 grade	점수 merits
9716001	A0	90
9716004	A0	90

3. 1. 14 부속질의(subquery)를 이용한 검색

예제 46 in 명령을 이용한 검색(3)

<그림 5-4> 예제 데이터베이스에서 0002 과목을 등록한 학생 중 성적이 A+와 A0인 학생의 학번, 이름, 성적, 점수를 검색하라.

```
select s_no, grade, merits
from enrollment e, student s
where c_no='0002' and e.s_no=s.s_no and grade in ('A+', 'A0');
```

결 과 :

학번 s_no	이름 s_name	성적 grade	점수 merits
9716001	김은수	A0	90
9716004	김현수	A0	90

3. 1. 14 부속질의(subquery)를 이용한 검색

예제 47 in 명령을 이용한 검색(4)

<그림 5-4> 예제 데이터베이스에서 교수의 연구실이 T동에 있지 않은 교수번호, 이름, 연구실 번호 등을 검색하라.

```
select p_no, p_name, room_no
from professor
where room_no not in ('T%');
```

또는

```
select p_no, p_name, room_no
from professor
where room_no <> 'T%';
```

결 과 :

교수번호 p_no	이름 p_name	연구실 room_no
0004	이재무	C303
0005	고병오	C402
0011	이순미	C203
0016	김병곤	C502

3. 1. 14 부속질의(subquery)를 이용한 검색

예제 48 exists 명령을 이용한 검색(1)

<그림 5-4> 예제 데이터베이스에서 0004 강좌를 등록한 학생의 이름과 학과를 검색하라.

```
select s_name, dept
from student
where exists      (select s_no
                   from enrollment
                   where s_no = student.s_no and c_no='0004');
```

결 과 :

이름 s_name	학과 dept
신미영	정보통신공학과
김찬우	전기공학과
양현석	전자공학과
양미라	컴퓨터공학과

3. 1. 14 부속질의(subquery)를 이용한 검색

예제 49 exists 명령을 이용한 검색(2)

<그림 5-4> 예제 데이터베이스에서 0004 강좌를 등록하지 않은 학생의 이름과 학과를 검색하라.

```
select s_name, dept
from student
where not exists (select s_no
                  from enrollment
                  where s_no = student.s_no and c_no='0004');
```

결 과 :

이름 s_name	학과 dept
정찬우	컴퓨터공학과
김정길	정보과학과
강재우	컴퓨터공학과
이수자	정보통신공학과
이길녀	컴퓨터공학과
박찬종	정보산업공학과
길태종	전자공학과
이기성	정보산업공학과
이미자	정보산업공학과
김은수	문헌정보학과
김현수	문헌정보학과
인소란	전기공학과
박창호	전기공학과
성기정	전기공학과
마현수	정보과학과
함경수	정보통신공학과
한근희	정보통신공학과
이재황	정보과학과
나기호	전자공학과

3. 1. 15 집합 연산을 이용한 검색

- ❖ SQL 질의어에서는 표준 집합 연산 (예, 결합, 교집합, 차집합 등) 을 이용하여 데이터베이스의 데이터를 처리할 수 있다.

집합 연산 명령어 요약

구 분	의 미
결 합	명령어 : union 두 개의 질의를 하나의 복합 질의로 결합 두 질의에서 행을 선택하여 중복된 행을 제거 중복된 행까지 결과를 얻기 위해서는 union all 이용
교집합	명령어 : intersection 두 개의 릴레이션에 모두 있는 행들을 포함하는 테이블 생성
합집합	명령어 : difference 첫 번째 릴레이션에는 있지만 두 번째 집합에는 없는 행들을 포함하는 테이블 생성.

3. 1. 15 집합연산을 이용한 검색

예제 50 union 명령을 이용한 검색

<그림 5-4> 예제 데이터베이스에서 조교수이거나 0004 강좌를 강의하는 교수의 교수번호를 검색하라.

```
select p_no
from professor
where grade='조교수'
union
select p_no
from lecture
where c_no='0004';
```

결 과 :

교수번호 p_no
0005
0008
0009
0010
0011
0013
0015
0016
0017
0018
0019

3. 2 데이터 삽입

- ❖ SQL 질의어에서 테이블에 데이터를 삽입하기 위해서 사용하는 명령어는 insert 이다.
 - 첫 번째 : 튜플을 직접 테이블(릴레이션)에 삽입하는 방법
 - 두 번째 : **select** 문을 이용한 부속 질의어를 사용하는 방법

<튜플의 직접 삽입>

```
insert  
into   테이블[(열_이름_리스트)]  
values (열값_리스트);
```

<부속 질의어를 이용한 삽입>

```
insert  
into   테이블[(열_이름_리스트)]  
select 문;
```

3. 2.1 튜플의 직접 삽입

예제 51

<그림 5-4> 예제 데이터베이스에서 학과(DEPARTMENT) 릴레이션에 다음과 같은 튜플 값을 삽입하는 SQL 질의어를 작성하라.

(학과코드 : 001, 학과명 : '컴퓨터공학과', 학과장 : 0001, 조교 : '이지은', 전화 : 3333, 위치 : 'T101')

insert

into department(dept_no, dept_name, chief, assistant, dept_tel, location)
values(001, '컴퓨터공학과', 0001, '이지은', 3333, 'T101');

또는

insert

into department
values(001, '컴퓨터공학과', 0001, '이지은', 3333, 'T101');

3. 2.1 튜플의 직접 삽입

예제 52

<그림 5-4> 예제 데이터베이스에서 강의(LECTURE) 릴레이션에 다음과 같은 튜플 값을 삽입하는 SQL 질의어를 작성하라.

(강좌번호 : 0001, 교수번호 : 0001, 요일 : 'mon', 강의실 : 'T447')

```
insert  
into   lecture(c_no, p_no, day_week, room_no)  
values      (0001, 0001, 'mon', 'T447');
```

3. 2. 2 부속 질의어를 이용한 삽입

예제 53

<그림 5-4> 예제 데이터베이스의 등록(ENROLLMENT) 릴레이션에서 0002 강좌를 수강한 학생의 학번, 성적, 점수를 검색하여 데이터베이스(DATABASE) 릴레이션에 삽입하는 SQL 질의어를 작성하라. 단, 데이터베이스(DATABASE) 릴레이션은 (s_no, grade, merits)로 구성된다고 가정한다.

```
insert  
into database(s_no, grade, merits)  
select s_no, grade, merits  
from student  
where c_no='0002';
```

또는

```
insert  
into database  
select s_no, grade, merits  
from student  
where c_no='0002';
```

3. 3 데이터 삭제(delete)

- ❖ SQL 질의어에서 테이블의 데이터를 삭제하기 위해서 사용하는 명령어는 delete 이다. 구문형식은 다음과 같다.

```
delete  
from      테이블  
[where    조건];
```

3. 3 .1 전체 튜플 삭제

예제 54

<그림 5-4> 예제 데이터베이스의 등록(ENROLLMENT) 릴레이션에 포함된 모든 튜플을 삭제하라.

```
delete  
from enrollment;
```

3. 3 .2 조건을 지정한 튜플 삭제

예제 55

<그림 5-4> 예제 데이터베이스의 등록(ENROLLMENT) 릴레이션에서 학번이 9717005인 튜플을 삭제하라.

```
delete  
from enrollment  
wheres_no='9717005';
```

3. 3 .3 부속 질의어를 이용한 튜플 삭제

예제 56

<그림 5-4> 예제 데이터베이스에서 컴퓨터공학과 학생 중 등록(ENROLLMENT) 릴레이션의 점수가 60점 미만인 학생의 튜플을 삭제하라.

delete

from enrollment

where merits < 60 **and** s_no in

(**select**

s_no

from

student

where

dept='컴퓨터공학과');

3. 4 데이터 갱신 [update]

- ❖ SQL 질의어에서 에 저장된 데이터의 내용을 갱신하기 위해서 사용하는 명령어는 update이며 구문형식은 아래와 같다.

update	테이블
set	열_이름=산술식+
[where	조건];

3. 4 .1 전체 튜플 변경

예제 57

<그림 5-4> 예제 데이터베이스에서 강의(LECTURE) 릴레이션에 포함된 모든 튜플들의 시작시간을 1교시로 변경하라.

```
update      lecture
set         start_time = 1;
```

3. 4 .2 조건을 지정한 튜플 변경

예제 58

<그림 5-4> 예제 데이터베이스의 교수(PROFESSOR) 릴레이션에서 교수번호가 0011인 교수의 전화번호를 4207로 변경하라.

```
update      professor
set         p_tel = 4207
wherep_no = '0011';
```

3. 4 .2 조건을 지정한 튜플 변경

예제 59

<그림 5-4> 예제 데이터베이스의 교수(PROFESSOR) 릴레이션에서 직급이 조교수인 교수들의 호봉을 1호봉씩 증가시켜라.

```
update      professor
set         pay_step = pay_step + 1
where      grade = '조교수';
```

예제 60

<그림 5-4> 예제 데이터베이스의 등록(ENROLLMENT) 릴레이션에서 0001 강좌를 수강한 학생들의 점수를 5점씩 감점시켜라.

```
update      enrollment
set         merits = merits - 5
where      c_no = '0001';
```


3. 4 .3 부속 질의어를 이용한 투플 변경

예제 61

<그림 5-4> 예제 데이터베이스에서 0001 강좌를 수강하는 정보통신공학과 학생들의 점수를 5점씩 증가시켜라.

```
update
set
where
```

```
enrollment
merits = merits + 5
c_no = '0001' and s_no in (select s_no
from student
where dept = '정보통신공학과');
```

4. 뷰 (View)

- ❖ 뷰(view)란 **create table** 명령으로 생성한 릴레이션(기본 테이블 : **base table**)으로부터 특정 조건에 맞는 내용들을 추출하여 생성한 가상의(**virtual**) 릴레이션(테이블)이다.

4.1 뷰의 생성

- ❖ SQL 질의어에서 뷰를 생성하기 위해서 사용하는 명령어는 **create view**이며 구문형식은 다음과 같다.

```
create view 뷰_이름[(열_이름_리스트)]  
as          select 문  
            [with check option];
```

4.1 뷰의 생성

예제 62

<그림 5-4> 예제 데이터베이스의 등록(ENROLLMENT) 릴레이션에서 0001 강좌를 수강하는 학생들에 대한 정보를 0001enroll 뷰로 정의하라.

```
create view      0001enroll(s_no, grade, merits)
as              select    s_no, grade, merits
                  from      enrollment
                  where     c_no = '0001'
                  with check option;
```

또는

```
create view      0001enroll
as              select    s_no, grade, merits
                  from      enrollment
                  where     c_no = '0001'
                  with check option;
```

결 과 :

학번 s_no	성적 grade	점수 merits
9911001	A+	95
9912002	B+	82
9911003	A0	90
9813009	C+	75
9811015	D+	65
9714016	A0	91

4.2 뷰의 제거

- ❖ SQL 질의어에서 뷰를 제거하기 위해서 사용하는 명령어는 drop view 이며 구문 형식은 다음과 같다.

```
drop view 뷰_이름 restrict | cascade;
```

예제 63

<예제 62>에서 생성한 0001enroll 뷰를 제거하는 SQL 질의어를 작성하라.
단, 0001enroll 뷰를 참조하고 있는 다른 뷰와 제약조건들도 함께 제거하라.

```
drop view 0001enroll cascade;
```

예제 64

<예제 62>에서 생성한 0001enroll 뷰를 제거하는 SQL 질의어를 작성하라.
단, 0001enroll 뷰를 다른 곳에서 참조할 경우 제거 명령을 실행하지 않도록 하라.

```
drop table department restrict;
```