

</SQL>

D A T A B A S E

Chapter 07

데이터베이스 언어 SQL

Database Language SQL

대전대학교 컴퓨터공학과

CONTENTS

01

SQL의 소개

SQL의 개념, 역사, 분류
(DDL, DML, DCL)

02

SQL을 이용한 데이터 정의

CREATE TABLE, ALTER TABLE,
DROP TABLE

03

SQL을 이용한 데이터 조작

SELECT, INSERT, UPDATE,
DELETE

04

뷰

가상 테이블의 생성, 활용, 삭제

05

삽입 SQL

Embedded SQL과 커서

SQL — Structured Query Language

비절차적 데이터 언어

사용자가 처리를 원하는 데이터가 무엇인지만 제시하고, 어떻게 처리해야 하는지는 언급할 필요가 없음

SQL의 역사

SEQUEL (Structured English QUery Language): 1974년, IBM 연구소에서 개발한 연구용 관계 DBMS인 SYSTEM R을 위한 언어

많은 회사들이 자체 RDBMS의 질의어를 만들어 사용함

ANSI, ISO에서 RDBMS의 최초 표준 SQL (SQL-86, SQL1)을 제정함

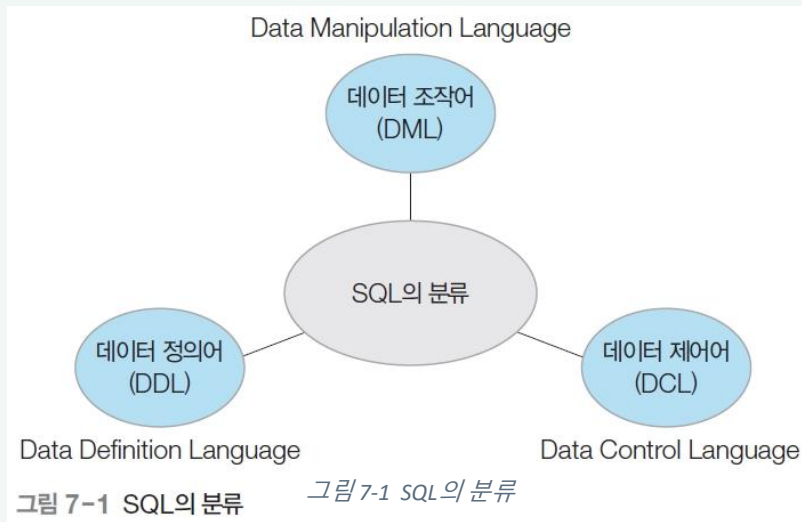
SQL1 → SQL2 (SQL-92) → SQL3 (SQL:1999)

SQL의 분류

DDL: 테이블을 생성, 변경 및 삭제하는 기능을 제공함

DML: 테이블에 데이터를 삽입, 수정, 삭제 및 검색하는 기능을 제공함

DCL: 데이터의 무결성, 보안, 권한 제어 및 회복하는 기능을 제공함



판매 데이터베이스

고객, 제품 및 주문 테이블 (릴레이션)

고객 테이블

고객아이디	고객이름	나이	등급	직업	적립금
apple	정소화	20	gold	학생	1000
banana	김선우	25	vip	간호사	2500
carrot	고명석	28	gold	교사	4500
orange	김용욱	22	silver	학생	0
melon	성원용	35	gold	회사원	5000
peach	오형준	NULL	silver	의사	300
pear	채광주	31	silver	회사원	500

제품 테이블

제품번호	제품명	재고량	단가	제조업체
p01	그냥만두	5000	4500	대한식품
p02	매운졸면	2500	5500	민국푸드
p03	콩떡파이	3600	2600	한빛제과
p04	맛난초콜릿	1250	2500	한빛제과
p05	얼큰라면	2200	1200	대한식품
p06	통통우동	1000	1550	민국푸드
p07	달콤비스킷	1650	1500	한빛제과

주문 테이블

주문번호	주문고객	주문제품	수량	배송지	주문일자
o01	apple	p03	10	서울시 마포구	2022-01-01
o02	melon	p01	5	인천시 계양구	2022-01-10
o03	banana	p06	45	경기도 부천시	2022-01-11
o04	carrot	p02	8	부산시 금정구	2022-02-01
o05	melon	p06	36	경기도 용인시	2022-02-20
o06	banana	p01	19	충청북도 보은군	2022-03-02
o07	apple	p03	22	서울시 영등포구	2022-03-15
o08	pear	p02	50	강원도 춘천시	2022-04-10
o09	banana	p04	15	전라남도 목포시	2022-04-11
o10	carrot	p03	20	경기도 안양시	2022-05-22

그림 7-2 질의에 사용할 판매 데이터베이스 : 고객, 제품, 주문 테이블

SQL의 데이터 정의 기능

데이터 정의 기능: 테이블 생성 / 변경 / 삭제



테이블의 생성 — CREATE TABLE

기본 규칙

SQL문에서 대소문자 구분하지 않음 | 생성 시 기본적으로 Null 값 허용 | 대체키는 여러 개 존재할 수 있음

```
CREATE TABLE 테이블_이름 (
```

- ① 속성_이름 데이터_타입 [NOT NULL] [DEFAULT 기본_값]
- ② [PRIMARY KEY (속성_리스트)]
- ③ [UNIQUE (속성_리스트)]
- ④ [FOREIGN KEY (속성_리스트) REFERENCES 테이블_이름(속성_리스트)]
[ON DELETE 옵션] [ON UPDATE 옵션]
- ⑤ [CONSTRAINT 이름] [CHECK(조건)]

```
);
```

표준 SQL의 속성의 대표적인 데이터 타입

CREATE TABLE

표 7-1 속성의 데이터 타입

데이터 타입	의미
INT 또는 INTEGER	정수
SMALLINT	INT보다 작은 정수
CHAR(n) 또는 CHARACTER(n)	길이가 n인 고정 길이의 문자열
VARCHAR(n) 또는 CHARACTER VARYING(n)	최대 길이가 n인 가변 길이의 문자열
NUMERIC(p, s) 또는 DECIMAL(p, s)	고정 소수점 실수 p는 소수점을 제외한 전체 숫자의 길이, s는 소수점 이하 숫자의 길이
FLOAT(n)	길이가 n인 부동 소수점 실수
REAL	부동 소수점 실수
DATE	연, 월, 일로 표현되는 날짜
TIME	시, 분, 초로 표현되는 시간
DATETIME	날짜와 시간

NOT NULL · DEFAULT 설정

CREATE TABLE

고객아이디 VARCHAR(20) NOT NULL

적립금 INT DEFAULT 0

학점 CHAR(2) DEFAULT 'P'

학점 CHAR(2) DEFAULT 'p' ← 대소문자 구분 주의!

PRIMARY KEY · UNIQUE 제약조건

CREATE TABLE

PRIMARY KEY(고객아이디)

단일 속성 기본키

PRIMARY KEY(주문고객, 주문제품)

복합 기본키 (2개 이상의 속성)

UNIQUE(고객이름)

대체키 (널 값을 가질 수 있음, 여러 개 지정해도 됨) | 외래키는 UNIQUE로 지정된 대체키를 참조하도록 정의할 수 있음



그림 7-4 외래키를 통해 관계를 맺고 있는 2개의 테이블

CREATE TABLE — 외래키 삭제 옵션

ON DELETE NO ACTION: 부서 테이블의 튜플을 삭제하지 못함

ON DELETE CASCADE: 관련 튜플을 함께 삭제함

ON DELETE SET NULL: 관련 튜플의 외래키 값을 NULL로 변경함

ON DELETE SET DEFAULT: 관련 튜플의 외래키 값을 미리 지정한 기본 값으로 변경함



NULL 혹은 기본값으로 변경

실습 ① ON DELETE NO ACTION (기본값)

STEP 1 테이블 생성 + 데이터 삽입 (먼저 실행!)

```
DROP TABLE IF EXISTS 직원1;
DROP TABLE IF EXISTS 부서1;
CREATE TABLE 부서1 (부서번호 CHAR(3) PRIMARY KEY, 부서명 VARCHAR(20));
CREATE TABLE 직원1 (직원번호 CHAR(3) PRIMARY KEY, 이름 VARCHAR(10),
    소속부서 CHAR(3), FOREIGN KEY(소속부서) REFERENCES 부서1(부서번호));
INSERT INTO 부서1 VALUES ('D01', '개발팀'), ('D02', '기획팀');
INSERT INTO 직원1 VALUES ('E01', '김철수', 'D01'), ('E02', '이영희', 'D01');
```

STEP 2 부서 삭제 시도 (따로 실행!)

```
DELETE FROM 부서1 WHERE 부서번호 = 'D01';
```

X 예상 결과: 에러!

옵션 생략 = NO ACTION → 직원이 D01을 참조 중이므로 삭제 거부!

STEP 3 확인 (따로 실행!)

```
SELECT * FROM 부서1;    -- D01, D02 둘 다 그대로
SELECT * FROM 직원1;   -- E01, E02 그대로
```

실습 ② ON DELETE CASCADE

STEP 1 테이블 생성 + 데이터 삽입 (먼저 실행!)

```
DROP TABLE IF EXISTS 직원2;  
DROP TABLE IF EXISTS 부서2;  
CREATE TABLE 부서2 (부서번호 CHAR(3) PRIMARY KEY, 부서명 VARCHAR(20));  
CREATE TABLE 직원2 (직원번호 CHAR(3) PRIMARY KEY, 이름 VARCHAR(10),  
    소속부서 CHAR(3), FOREIGN KEY(소속부서) REFERENCES 부서2(부서번호)  
    ON DELETE CASCADE);  
INSERT INTO 부서2 VALUES ('D01', '개발팀'), ('D02', '기획팀');  
INSERT INTO 직원2 VALUES ('E01', '김철수', 'D01'), ('E02', '이영희', 'D01'),  
    ('E03', '박민수', 'D02');
```

STEP 2 부서 삭제 (따로 실행!)

```
DELETE FROM 부서2 WHERE 부서번호 = 'D01';
```

▶ 예상 결과

CASCADE → 부서 D01 삭제 + 직원 E01, E02도 연쇄 삭제!

STEP 3 확인 (따로 실행!) – E03만 남아있는지 확인

```
SELECT * FROM 부서2;    -- D02만 남음  
SELECT * FROM 직원2;   -- E03만 남음! E01, E02 삭제됨
```

실습 ③ ON DELETE SET NULL

STEP 1 테이블 생성 + 데이터 삽입 (먼저 실행!)

```
DROP TABLE IF EXISTS 직원3;
DROP TABLE IF EXISTS 부서3;
CREATE TABLE 부서3 (부서번호 CHAR(3) PRIMARY KEY, 부서명 VARCHAR(20));
CREATE TABLE 직원3 (직원번호 CHAR(3) PRIMARY KEY, 이름 VARCHAR(10),
    소속부서 CHAR(3), FOREIGN KEY(소속부서) REFERENCES 부서3(부서번호)
    ON DELETE SET NULL);
INSERT INTO 부서3 VALUES ('D01', '개발팀'), ('D02', '기획팀');
INSERT INTO 직원3 VALUES ('E01', '김철수', 'D01'), ('E02', '이영희', 'D01');
```

STEP 2 부서 삭제 (따로 실행!)

```
DELETE FROM 부서3 WHERE 부서번호 = 'D01';
```

▶ 예상 결과

SET NULL → 직원은 남고, 소속부서만 NULL로 변경!

STEP 3 확인 – 직원이 남아있되 소속부서가 NULL인지 확인!

```
SELECT * FROM 직원3;
-- E01 김철수   소속부서: NULL
-- E02 이영희   소속부서: NULL
```

CREATE TABLE — 외래키 수정 옵션

ON UPDATE NO ACTION: 부서 테이블의 튜플을 변경하지 못함

ON UPDATE CASCADE: 관련 튜플을 함께 변경함

ON UPDATE SET NULL: 관련 튜플의 외래키 값을 NULL로 변경함

ON UPDATE SET DEFAULT: 관련 튜플의 외래키 값을 미리 지정한 기본 값으로 변경함



실습 ④ ON UPDATE CASCADE

STEP 1 테이블 생성 + 데이터 삽입 (먼저 실행!)

```
DROP TABLE IF EXISTS 직원4;
DROP TABLE IF EXISTS 부서4;
CREATE TABLE 부서4 (부서번호 CHAR(3) PRIMARY KEY, 부서명 VARCHAR(20));
CREATE TABLE 직원4 (직원번호 CHAR(3) PRIMARY KEY, 이름 VARCHAR(10),
    소속부서 CHAR(3), FOREIGN KEY(소속부서) REFERENCES 부서4(부서번호)
    ON UPDATE CASCADE);
INSERT INTO 부서4 VALUES ('D01', '개발팀');
INSERT INTO 직원4 VALUES ('E01', '김철수', 'D01');
```

STEP 2 부서번호 변경 (따로 실행!)

```
UPDATE 부서4 SET 부서번호 = 'D10' WHERE 부서번호 = 'D01';
```

▶ 예상 결과

CASCADE → 부서 D01→D10, 직원 소속부서도 D01→D10 자동 변경!

STEP 3 확인 – 직원 소속부서도 D10으로 바뀌었는지 확인!

```
SELECT * FROM 부서4;    -- D10 (D01에서 변경됨)
SELECT * FROM 직원4;   -- E01 소속부서: D10 (자동!)
```

실습 ⑤ 자식 쪽 변경은 옵션과 무관!

STEP 1 테이블 생성 (CASCADE 설정) + 데이터 삽입

```
DROP TABLE IF EXISTS 직원5; DROP TABLE IF EXISTS 부서5;
CREATE TABLE 부서5 (부서번호 CHAR(3) PRIMARY KEY, 부서명 VARCHAR(20));
CREATE TABLE 직원5 (직원번호 CHAR(3) PRIMARY KEY, 이름 VARCHAR(10),
    소속부서 CHAR(3), FOREIGN KEY(소속부서) REFERENCES 부서5(부서번호)
    ON DELETE CASCADE ON UPDATE CASCADE);
INSERT INTO 부서5 VALUES ('D01', '개발팀'), ('D02', '기획팀');
INSERT INTO 직원5 VALUES ('E01', '김철수', 'D01'), ('E02', '이영희', 'D02');
```

STEP 2 자식 튜플 삭제 (따로 실행!) – 그냥 됨!

```
DELETE FROM 직원5 WHERE 직원번호 = 'E02';
SELECT * FROM 직원5; -- E02만 사라짐, 부서 무관
```

STEP 3 자식 FK를 존재하는 값으로 수정 – 성공!

```
UPDATE 직원5 SET 소속부서 = 'D02' WHERE 직원번호 = 'E01';
SELECT * FROM 직원5; -- E01 소속부서: D02
```

STEP 4 자식 FK를 없는 값으로 수정 – 에러!

```
UPDATE 직원5 SET 소속부서 = 'D99' WHERE 직원번호 = 'E01';
```

X 예상 결과: 에러!

D99가 부서5에 없음 → 참조 무결성 위반 → 에러! (CASCADE와 무관)

CREATE TABLE — 외래키 예제

FOREIGN KEY(소속부서) REFERENCES 부서(부서번호) ON DELETE CASCADE ON UPDATE CASCADE

관련 튜플을 함께 삭제/수정함

FOREIGN KEY(소속부서) REFERENCES 부서(부서번호)

옵션 생략 시 → ON DELETE NO ACTION과 ON UPDATE NO ACTION이 자동으로 선택됨

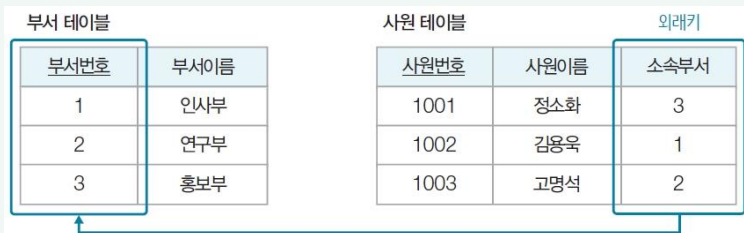


그림 7-4 외래키를 통해 관계를 맺고 있는 2개의 테이블

실습 ⑥ 같은 부모, 자식별 다른 정책

STEP 1 부서 + 정규직(SET NULL) + 인턴(CASCADE) 생성

```
DROP TABLE IF EXISTS 인턴6; DROP TABLE IF EXISTS 정규직6;
DROP TABLE IF EXISTS 부서6;
CREATE TABLE 부서6 (부서번호 CHAR(3) PRIMARY KEY, 부서명 VARCHAR(20));
CREATE TABLE 정규직6 (직원번호 CHAR(3) PRIMARY KEY, 이름 VARCHAR(10),
    소속부서 CHAR(3), FOREIGN KEY(소속부서) REFERENCES 부서6(부서번호)
    ON DELETE SET NULL);
CREATE TABLE 인턴6 (인턴번호 CHAR(3) PRIMARY KEY, 이름 VARCHAR(10),
    소속부서 CHAR(3), FOREIGN KEY(소속부서) REFERENCES 부서6(부서번호)
    ON DELETE CASCADE);
INSERT INTO 부서6 VALUES ('D01', '개발팀');
INSERT INTO 정규직6 VALUES ('F01', '김정규', 'D01');
INSERT INTO 인턴6 VALUES ('I01', '박인턴', 'D01');
```

STEP 2 부서 삭제 (따로 실행!)

```
DELETE FROM 부서6 WHERE 부서번호 = 'D01';
```

STEP 3 정규직 vs 인턴 결과 비교!

```
SELECT * FROM 정규직6; -- F01 남아있고, 소속부서=NULL
SELECT * FROM 인턴6; -- I01 삭제됨!
```

▶ 예상 결과

정규직 → SET NULL: 소속만 NULL (보존)

인턴 → CASCADE: 투플 삭제! 같은 부모여도 자식마다 독립적 옵션!

CREATE TABLE — 데이터 무결성 제약조건

CHECK(재고량>=0 AND 재고량<=10000)

재고량은 0개 이상, 10,000개 이하로 유지되어야 함

CONSTRAINT CHK_CPY CHECK(제조업체 = '한빛제과')

데이터 무결성 제약조건에 CHK_CPY라는 고유 이름을 부여함
(다른 테이블에는 없어야 하며, 유일하게 부여되어야 함)

고객 테이블

고객아이디	고객이름	나이	등급	직업	적립금
apple	정소화	20	gold	학생	1000
banana	김선우	25	vip	간호사	2500
carrot	고명석	28	gold	교사	4500
orange	김용욱	22	silver	학생	0
melon	성원용	35	gold	회사원	5000
peach	오형준	NULL	silver	의사	300
pear	채광주	31	silver	회사원	500

제품 테이블

제품번호	제품명	재고량	단가	제조업체
p01	그냥만두	5000	4500	대한식품
p02	매운짬면	2500	5500	민국푸드
p03	쿵떡파이	3600	2600	한빛제과
p04	맛난초콜릿	1250	2500	한빛제과
p05	얼큰라면	2200	1200	대한식품
p06	통통우동	1000	1550	민국푸드
p07	달콤비스킷	1650	1500	한빛제과

CREATE TABLE 테이블 생성의 예

예제 7-1: 고객 테이블 생성

예제 7-1

고객 테이블은 고객아이디, 고객이름, 나이, 등급, 직업, 적립금 속성으로 구성되고, 고객아이디 속성이 기본키다. 고객이름과 등급 속성은 값을 반드시 입력해야 하고, 적립금 속성은 값을 입력하지 않으면 0이 기본으로 입력되도록 고객 테이블을 생성해보자.

```
▶▶ CREATE TABLE 고객 (  
    고객아이디 VARCHAR(20) NOT NULL,  
    고객이름   VARCHAR(10) NOT NULL,  
    나이       INT,  
    등급      VARCHAR(10) NOT NULL,  
    직업       VARCHAR(20),  
    적립금    INT DEFAULT 0,  
    PRIMARY KEY(고객아이디)  
);
```

CREATE TABLE 테이블 생성의 예

예제 7-2: 제품 테이블 생성

예제 7-2

제품 테이블은 제품번호, 제품명, 재고량, 단가, 제조업체 속성으로 구성되고, 제품번호 속성이 기본키다. 재고량이 항상 0개 이상 10,000개 이하를 유지하도록 제품 테이블을 생성해보자.

```
▶▶ CREATE TABLE 제품 (  
    제품번호 CHAR(3) NOT NULL,  
    제품명 VARCHAR(20),  
    재고량 INT,  
    단가 INT,  
    제조업체 VARCHAR(20),  
    PRIMARY KEY(제품번호),  
    CHECK (재고량 >= 0 AND 재고량 <=10000)  
);
```

CREATE TABLE 테이블 생성의 예

예제 7-3: 주문 테이블 생성

예제 7-3

주문 테이블은 주문번호, 주문고객, 주문제품, 수량, 배송지, 주문일자 속성으로 구성되고, 주문번호 속성이 기본키다. 주문고객 속성이 고객 테이블의 고객아이디 속성을 참조하는 외래키이고, 주문제품 속성이 제품 테이블의 제품번호 속성을 참조하는 외래키가 되도록 주문 테이블을 생성해보자.

```
▶▶ CREATE TABLE 주문 (  
    주문번호 CHAR(3) NOT NULL,  
    주문고객 VARCHAR(20),  
    주문제품 CHAR(3),  
    수량 INT,  
    배송지 VARCHAR(30),  
    주문일자 DATE, MS SQL에서는 주문일자 속성의 데이터 타입을 DATETIME로 지정함  
    PRIMARY KEY(주문번호),  
    FOREIGN KEY(주문고객) REFERENCES 고객(고객아이디),  
    FOREIGN KEY(주문제품) REFERENCES 제품(제품번호)  
);
```

CREATE TABLE 테이블 생성의 예

예제 7-4: 배송업체 테이블 생성

예제 7-4

배송업체 테이블은 업체번호, 업체명, 주소, 전화번호 속성으로 구성되고 업체번호 속성이 기본키다. 배송업체 테이블을 생성해보자.

```
▶▶ CREATE TABLE 배송업체 (  
    업체번호 CHAR(3) NOT NULL,  
    업체명 VARCHAR(20),  
    주소 VARCHAR(100),  
    전화번호 VARCHAR(20),  
    PRIMARY KEY(업체번호)  
);
```

ALTER TABLE — 속성 추가 기본 형식

테이블 변경 — ALTER TABLE

```
ALTER TABLE 테이블_이름  
ADD 속성_이름 데이터_타입 [NOT NULL] [DEFAULT 기본_값];
```

예제 7-5

[예제 7-1]에서 생성한 고객 테이블에 가입날짜 속성을 추가해보자.

```
▶▶ ALTER TABLE 고객 ADD 가입날짜 DATE;
```

ALTER TABLE — 속성 삭제 기본 형식

테이블 변경 — ALTER TABLE

```
ALTER TABLE 테이블_이름 DROP COLUMN 속성_이름;
```

예제 7-6

[예제 7-5]에서 추가한 고객 테이블의 가입날짜 속성을 삭제해보자.

```
▶▶ ALTER TABLE 고객 DROP COLUMN 가입날짜;
```

ALTER TABLE — 제약조건 추가 기본 형식

테이블 변경 — ALTER TABLE

```
ALTER TABLE 테이블_이름 ADD CONSTRAINT 제약조건_이름 제약조건_내용;
```

예제 7-7

고객 테이블에 20세 이상의 고객만 가입할 수 있다는 데이터 무결성 제약조건을 추가해보자.

```
▶▶ ALTER TABLE 고객 ADD CONSTRAINT CHK_AGE CHECK(나이 >= 20);
```

ALTER TABLE — 제약조건 삭제 기본 형식

테이블 변경 — ALTER TABLE

```
ALTER TABLE 테이블_이름 DROP CONSTRAINT 제약조건_이름;
```

예제 7-8

[예제 7-7]에서 추가한 고객 테이블에 20세 이상의 고객만 가입할 수 있다는 데이터 무결성 제약조건을 삭제해보자.

```
▶▶ ALTER TABLE 고객 DROP CONSTRAINT CHK_AGE;
```

테이블 삭제 — DROP TABLE

테이블 삭제 기본 형식

```
DROP TABLE 테이블_이름;
```

예제 7-9

배송업체 테이블을 삭제해보자.

```
▶▶ DROP TABLE 배송업체;
```

Wrap-Up

배운 것들을 정리해봅시다.

SQL의 소개

비절차적 데이터 언어 | DDL · DML · DCL 세 가지 분류 | 1974 SEQUEL → SQL 표준화

CREATE TABLE

테이블 생성 | 데이터 타입 | NOT NULL · DEFAULT | PRIMARY KEY · UNIQUE · FOREIGN KEY · CHECK

ALTER TABLE

속성 추가(ADD) · 삭제(DROP COLUMN) | 제약조건 추가(ADD CONSTRAINT) · 삭제(DROP CONSTRAINT)

DROP TABLE

테이블 완전 삭제 | 외래키로 참조 중이면 삭제 불가

Next Class

SQL을 이용한 데이터 조작

SELECT, INSERT, UPDATE, DELETE 등
데이터 조작어(DML)를 상세하게 학습합니다.



`</SQL>`