

데이터베이스 실습 5주차 강의

SQL

데이터 정의 (DDL) : 테이블 생성, 변경, 삭제 / 뷰 생성, 삭제

뷰 : 카테고리를 줄여줄 때 사용 (보안코딩 할 때 사용)

데이터 조작 (DML) : 데이터 검색, 삽입, 수정, 삭제

데이터 제어 (DCL) -> 아이디 마다 권한 부여

SQL -> 비절차적 데이터 언어

대화식 SQL : SQL Developer 사용

삽입 SQL : 프로그래밍 언어 안에 삽입

!중간고사! SQL의 분류, 테이블이 나오도록 명령어 작성

데이터 정의어 (DDL) : 테이블을 생성하고 변경·삭제하는 기능을 제공

데이터 조작어 (DML) : 테이블에 새 데이터를 삽입하거나, 테이블에 저장된 데이터를 수정·삭제·검색하는 기능을 제공

데이터 제어어 (DCL) : 보안을 위해 데이터에 대한 접근 및 사용 권한을 사용자별로 부여하거나 취소하는 기능을 제공

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

위의 테이블을 보여주고 이 테이블이 나오도록 명령어 작성 (PPT 참고 - 여러 개 있음)

테이블 생성 : CREATE TABLE

테이블 변경 : ALTER TABLE

테이블 삭제 : DROP TABLE

!중간고사! 테이블 생성 : CREATE TABLE 문 구조

```
CREATE TABLE 테이블_이름 (  
    ❶ 속성_이름 데이터_타입 [NOT NULL] [DEFAULT 기본_값]  
    ❷ [PRIMARY KEY (속성_리스트)]  
    ❸ [UNIQUE (속성_리스트)]  
    ❹ [FOREIGN KEY (속성_리스트) REFERENCES 테이블_이름(속성_리스트)]  
        [ON DELETE 옵션] [ON UPDATE 옵션]  
    ❺ [CONSTRAINT 이름] [CHECK(조건)]  
);
```

[]의 내용은 생략이 가능 → 안해도 된다.

SQL 문은 세미콜론(;)으로 문장의 끝을 표시

SQL 문은 대소문자를 구분하지 않음

!중간고사! 테이블 생성 : CREATE TABLE 문 정의

- ① : 테이블을 구성하는 각 속성의 이름, 데이터 타입, 기본 제약 사항 정의
- ② : 기본키 정의 -> 반드시 값이 있어야함 [구분자가 필요]
- ③ : 대체키 정의
- ④ : 외래키 정의 -> 다른 테이블에서 쓸 수 있으니 주의해야 함
- ⑤ : 데이터 무결성을 위한 제약조건 정의 -> 특정 필드 값에 lock 걸 수 있다

NOT NULL : 속성이 널 값을 허용하지 않음을 의미하는 키워드

예) 고객아이디 VARCHAR(20) NOT NULL

!중간고사! 테이블 생성 : CREATE TABLE 문

키의 정의

FOREIGN KEY : 동사형 테이블에서 사용

삭제와 수정 할 때 문제가 생김 -> 동사형 테이블을 만들 때는 외래키를 지정

삭제 처리 옵션 →

ON DELETE NO ACTION : 튜플을 삭제하지 못하게 함

ON DELETE CASCADE : 관련 튜플을 같이 삭제

ON DELETE SET NULL : 지워진 외래키 값을 NULL로 변경

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

예) FOREIGN KEY(소속부서) REFERENCES 부서(부서번호)

ON DELETE CASCADE ON UPDATE CASCADE

소속부서 테이블은 부서라는 테이블과 연결되어 있다.

소속부서가 삭제되거나 부서라는 테이블에 있는 부서번호가 삭제되면 같이 지워라

소속부서가 수정되거나 부서라는 테이블에 있는 부서번호가 수정되면 같이 수정해라

데이터 무결성 제약조건의 정의

CHECK : 특정 속성에 대한 제약조건을 지정

예) CHECK(재고량 \geq 0 AND 재고량 \leq 10000)

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)  
);
```

테이블 변경 : ALTER TABLE 문

새로운 속성 추가

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

기존 속성 삭제

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

!중간고사! 테이블 변경 : ALTER TABLE 문

옵션이 더 추가될 예정

예제 7-5

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

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

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);
```

데이터 검색 : SELECT 문

기본 검색

```
SELECT [ ALL | DISTINCT ] 속성_리스트  
FROM   테이블_리스트;
```

ALL : 기본값 / DISTINCT : 중복을 허용하지 않음

AS 키워드 → 보안코딩 할 때 많이 씀

결과 테이블에서 속성의 이름을 바꾸어 출력 가능

예제 7-16

제품 테이블에서 제품명과 단가를 검색하되, 단가를 가격이라는 새 이름으로 출력해보자.

```
▶▶ SELECT   제품명, 단가 AS 가격  
FROM       제품;
```

결과 테이블

	제품명	가격
1	그냥만두	4500
2	매운쫄면	5500
3	콩떡파이	2600
4	맛난초콜릿	2500
5	얼큰라면	1200
6	통통우동	1550
7	달콤비스킷	1500