SQL 분류

데이터 정의어 (DDL) Definition

- 테이블을 생성하고 변경·제거

데이터 제어어 (DCL) Control

- 보안을 위해 데이터에 대한 접근 및 사용권한을 사용자별로 부여 or 취소
  → DB 관리자가 주로 사용

데이터 조작어 (DML) Manipulation

- 테이블에 새 데이터 삽입 or 테이블에 저장된 데이터 수정·삭제·검색

DDL(Definition) : create. alter. drop

- create

테이블을 생성할때 어떤 속성을 정할지 중요하다

속성 이름, 데이터 타입, [NOT NULL], [DEFAULT 값)

기본 키

외래키

제약조건

(e.g.) char(n)과 varchar(n) 차이

길이가 n인 고정길이의 문자열

최대길이가 n인 가변길이의 문자열

기본키

· 테이블 당 "반드시" 하나

PRIMARY KEY (속성이름) → 여러 개로 지정 가능

대체키

· 후보키 중에 기본키로 선택되지 않은키 → 유일성 만족, 기본키와 다르게 NULL가능

UNIQUE KEY (속성이름)

외래키

FOREIGN KEY (속성) REFERENCES 참조테이블(참조속성)

"데이터 무결성" 제약조건 정의

- "CHECK" 키워드 사용하여 특정 속성에 대한 제약조건 지정
  - 새로운 튜플 삽입 or 수정을 할 때에도 만족해야함

- CONSTRAINT CHK_AGE CHECK(나이 >= 18)
  - CONSTRAINT: 해당 키워드를 사용하여
  - CHK_AGE: 이 제약조건에 대해
  - CHECK(나이 >= 18): 체크를 함

왜?

→ 제약조건을 나중에 수정하거나 제거할 때 식별 용이

ALTER

· 테이블에 새로운 속성 추가

ALTER TABLE 이름 ADD [create table 속성할 때와 같음]

· 기존 속성 삭제

ALTER TABLE 이름 DROP COLUMN 속성 이름

· 새로운 제약조건 추가

ALTER TABLE 이름 ADD CONSTRAINT 제약조건이름 CHECK (조건)

· 제약조건 삭제

ALTER TABLE 이름 DROP CONSTRAINT 제약조건 이름

· 테이블 삭제 **DROP**

DROP TABLE 이름

SQL 데이터 조작 기능 (DML)

- SELECT. INSERT. UPDATE. DELETE

IS NULL : NULL 값을 가지는지 파악

SELECT 컬럼명들 FROM 테이블명 WHERE 조건들 ORDER BY 정렬할 컬럼명들

,로 구분

*: 전체

distinct: 중복제거

AND. OR 구분

LIKE

문자열을 비교하는 조건에만 사용

- 'A%' : A로 시작하는 문자열
- '%A' : A로 끝나는 "
- '%A%' : A를 포함하는 "
- 'A__' : A로 시작하는 3글자"
- '__A%' : 세 번째 글자가 A인 "

ASC: 오름차순 DESC: 내림차순

집계 함수 -> where 절사용X. select 와 having 에서 사용가능

count. max. min => 모든 데이터에서 가능

sum. avg => 숫자 데이터에서 가능

null 값 포함X

그룹별 검색 : 특정 속성의 값이 같은 튜플을 모아 그룹을 만들고, 그룹별로 검색하기 위해 사용

SELECT GROUP BY에서 사용한 칼럼명들, 집계함수

-> GROUP BY가 없다면 테이블 전체를 하나의 그룹으로 본 것

FROM 테이블이름

WHERE 조건들

GROUP BY 칼럼명들 (, 로 구분)

Ex) "주문제품별" "수량"

-주문제품끼리 그룹을 지어 각 수량의 총합을 구하면 된다

HAVING Group by 에서 해당하는 조건들 (조건문)

WHERE	vs	HAVING
집계함수 사용X		사용 O

서브 쿼리 : select 안에 select

서브쿼리는
- 하나의 행으로 결과 반환 : 비교 연산자 사용가능
  (단일 행)
- 여러 행으로 결과 반환 : 비교 연산자 사용불가
  (다중 행)
  - in
  - not in
  - exist
  - not exist
  - ALL → 비교연산자와 함께 사용
  - ANY, SOME

데이터 삽입

1. insert를 이용하여 직접 삽입

2. 부속질의문을 이용해 튜플 삽입

1. insert into 테이블(속성이름) values(——)

2. insert into 테이블(속성이름) select문

데이터 수정

update 테이블

set 속성 = 값 ..

[where 조건];

데이터 삭제

delete 테이블

from 테이블 이름

[where 조건];

→ where 조건을 만족하는 튜플만을 삭제

뷰

- 다른 테이블을 기반으로 만들어진 가상 테이블
  - ↳ 데이터를 실제로 저장하지 않고 있음
- 논리적으로만 존재.

뷰 생성

create view 뷰이름(속성들) ← 생략시 select문 속성 이름 그대로 사용

as select → 생성하고자 하는 뷰의 정의. order by 사용X

[option] ← 생성한 뷰에 삽입이나 수정 연산할 때,
select문에서 제시한 뷰의 정의 조건을 위반하면 수행X. = 제약조건

· 뷰를 생성하고 데이터를 삽입.수정.삭제를 할수 있지만 "제한적'
포함 .원래 테이블에 영향을 미친다.

· 삽입·삭제·수정이 불가능한 경우
- 원래 테이블의 기본키를 포함하지 않은 뷰
- 기존테이블에 있던 내용이 아닌 집계함수로 새로계산된 내용포함될 경우

· 뷰의 장점
. 질의문을 좀 더 쉽게 작성 → 사용자가 where절 없이 데이터 검색 가능
- 데이터의 보안 유지에 도움 → 여러 사용자 각기 맞는 뷰를 정의하고 권한을 설정
데이터를 좀 더 편리하게 관리

삽입 SQL

· 응용프로그램(C. C++. java)에 삽입하여 사용 가능 → 이때는 삽입 SQL이라는 걸 나타내기 위해 EXEC를 붙임

커서

· 여러 개 행을 반환하는 select문을 삽입SQL로 사용할 경우

가리키는 "포인터" 역할

결과 테이블을 반환하거나 행 하나만 반환할 경우 커서 필요x