

1

데이터베이스 언어 SQL 개요

01 데이터베이스 언어 SQL 개요

1 SQL의 개요

- SQL(Structured Query Language, 구조화된 질의어)는 관계형 데이터베이스 관리 시스템(RDBMS)의 데이터를 관리하기 위해 설계된 특수 목적의 프로그래밍 언어임
- RDBMS에서 자료의 검색과 관리, 데이터베이스 스키마 생성, 소멸, 수정, 데이터베이스 객체 접근 조정 관리를 위해 고안됨
- 대다수의 데이터베이스 관련 프로그램들이 SQL을 표준으로 채택하고 있음

※ 출처: <https://ko.wikipedia.org/wiki/SQL>

01 데이터베이스 언어 SQL 개요

2 SQL 명령 종류

데이터베이스 언어 SQL 문법의 종류는 세 가지로 구분됨

- 데이터 정의 언어
(DDL: Data Definition Language)
: 스카마(Schema) 관련 명령
- 데이터 조작 언어
(DML: Data Manipulation Language)
: 인스턴스(Instance) 관련 명령
- 데이터 제어 언어
(DCL Data Control Language)
: 보안, 트랜잭션 관리, 롤백 관련 명령

※ 출처: <https://ko.wikipedia.org/wiki/SQL>

01 데이터베이스 언어 SQL 개요

2 SQL 명령 종류

DDL 명령

- Create Table : 테이블 스키마 생성
- Alter Table : 테이블 스키마 변경
- Drop Table
 - : 내부 데이터 뿐만 아니라 테이블 자체도 삭제,
Rollback 되지 않음
- Truncate Table
 - : 내부 데이터는 전체 삭제하지만 테이블 스키마는
삭제하지 않음, Rollback 되지 않음

※ 출처: <https://ko.wikipedia.org/wiki/SQL>

01 데이터베이스 언어 SQL 개요

2 SQL 명령 종류

🔍 DML 명령

- Insert Into
- Update ~ Set
- Delete From
- Select ~ From ~ Where

※ 출처: <https://ko.wikipedia.org/wiki/SQL>

01 데이터베이스 언어 SQL 개요

2 SQL 명령 종류

🔍 DCL 명령

- GRANT : 사용자에게 특정 작업을 수행할 권한을 부여
- REVOKE : 사용자로부터 이미 준 권한을 박탈
- SET TRANSACTION : 트랜잭션 모드 설정(동시 트랜잭션 격리 수준 (ISOLATION MODE) 등))

※ 출처: <https://ko.wikipedia.org/wiki/SQL>

01 데이터베이스 언어 SQL 개요

2 SQL 명령 종류

🔍 DCL 명령

- BEGIN : 트랜잭션 시작
- COMMIT : 트랜잭션의 실행 확정
- ROLLBACK : 트랜잭션 취소
- SAVEPOINT : 무작위로 롤백 지점을 설정
- LOCK : 테이블 등의 자원을 독점 사용

※ 출처: <https://ko.wikipedia.org/wiki/SQL>

2

SQL 데이터 탑

02 SQL 데이터 타입

1 SQL 연산자

연산자	설명
=	같음
< > 또는 !=	같지 않음
>	보다 큼
<	보다 작음
>=	보다 크거나 같음
<=	보다 작거나 같음
BETWEEN	일정 범위 사이
LIKE	문자열 패턴 검색
IN	컬럼의 여러 가능한 값들을 지정

※ 출처: <https://ko.wikipedia.org/wiki/SQL>

02 SQL 데이터 타입

2 SQL 데이터 타입

- SQL 테이블에서 각 컬럼은 그것이 포함하는 값에 대한 자료형(Data Type)을 선언함, ANSI SQL은 다음과 같은 자료형을 포함하고 있음
- 문자열
 - CHARACTER(n) / CHAR(n)
: 고정 폭 n-문자열, 필요한 만큼 공백으로 채워짐
 - CHARACTER VARYING(n) / VARCHAR(n)
: 가변 폭 문자열, 최대 n개 문자의 크기를 갖음

※ 출처: <https://ko.wikipedia.org/wiki/SQL>

02 SQL 데이터 타입

2 SQL 데이터 타입



수

- INTEGER, SMALLINT
- FLOAT, REAL, DOUBLE PRECISION
- NUMERIC(Precision, Scale) /
DECIMAL(Precision, Scale)
: 예를 들어, 숫자 123.45 는 5 라는 Precision
(정밀도)과 2 라는 Scale(소수점 이하 자릿수)을
포함하고 있음
- SQL은 숫자나 날짜를 버림하는 TRUNC,
반올림하는 ROUND 함수를 제공함

※ 출처: <https://ko.wikipedia.org/wiki/SQL>

02 SQL 데이터 타입

2 SQL 데이터 타입

비트 열

- BIT(n) : n 비트 고정 크기
- BIT VARYING(n) : n 비트까지의 가변 크기

※ 출처: <https://ko.wikipedia.org/wiki/SQL>

02 SQL 데이터 타입

2 SQL 데이터 타입

날짜와 시간

- DATE : 날짜 값 (예; 2011-05-03)
- TIME : 시간 값 (예; 15:51:36)
- TIMESTAMP
: DATE와 TIME이 하나의 변수로 결합
(예, 2011-05-03 15:51:36)
- SQL은 TO_DATE, TO_TIME, TO_TIMESTAMP
함수를 제공함, 현재 데이터베이스 시스템의 날짜와
시간은 Now 함수나 시스템 변수 SYSDATE를 통해
얻을 수 있음

※ 출처: <https://ko.wikipedia.org/wiki/SQL>

3

SQL에서 기본 제약조건

03 SQL에서 기본 제약조건

1 제약조건의 명시

- 🔍 테이블을 생성(CREATE TABLE 명령)할 때
주로 명시함
- 🔍 속성에 대한 제약조건과 디폴트 값 명시
 - 속성에 null 값의 허용 여부 명시 : NOT NULL
 - 속성에 디폴트 값 지정 : DEFAULT <값>
 - 속성의 값을 특정한 값들로 한정
: CHECK (값의 범위)

03 SQL에서 기본 제약조건

1 제약조건의 명시

🔍 키와 참조 무결성 제약조건의 명시

- CREATE TABLE 명령에서 PRIMARY KEY 절은 테이블의 기본 키를 구성하는 하나 이상의 속성들을 명시
- UNIQUE 절은 대체키를 명시
- FOREIGN KEY 절에서는 참조 무결성을 명시함

03 SQL에서 기본 제약조건

1 제약조건의 명시

- 외래 키(Foreign key)를 정의할 때
참조 무결성 위반 시 취할 동작을 명시할 수 있음
 - 동작의 종류에는 CASCADE, SET NULL, SET DEFAULT 가 있으며
 - 위반의 종류를 나타내는 ON DELETE 나 ON UPDATE 와 함께 사용해야 함
 - 예제)
FOREIGN KEY (sid) REFERENCES Sailor(sid)
ON DELETE SET DEFAULT
ON UPDATE CASCADE

03 SQL에서 기본 제약조건

2 제약조건 명시의 예

- 🔍 Sailor 데이터베이스의 Relational Model 이 다음과 같을 때,

```
Sailor(sid, sname, rating, age) PK={sid}  
Boat(bid, bname, color) PK = {bid}  
Res(sid, bid, rdate) PK={sid, bid}, FK={sid}, {bid}
```

03 SQL에서 기본 제약조건

2 제약조건 명시의 예

도메인 제약조건 명시

```
CREATE TABLE Sailor(  
    sid int Primary Key,  
    sname varchar(20),  
    rating int,  
    age int  
) ;
```

03 SQL에서 기본 제약조건

2 제약조건 명시의 예

키 제약조건 명시

```
CREATE TABLE Boat(  
    bid int PRIMARY KEY,  
    bname varchar(20),  
    color nchar(5)  
) ;
```

03 SQL에서 기본 제약조건

2 제약조건 명시의 예

외래키 제약조건 명시

```
CREATE TABLE Res(  
    sid int, bid int,  
    rdate date default current_date,  
    CONSTRAINT PK_Res primary key (sid, bid),  
    CONSTRAINT FK_Res_Sailor  
        FOREIGN KEY (sid) REFERENCES Sailor(sid),  
    CONSTRAINT FK_Res_Boat  
        FOREIGN KEY (bid) REFERENCES Boat(bid)  
);
```

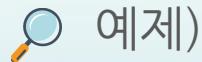
4

SQL에서 스키마 생성, 소멸, 변경문

1 스키마 생성문

- ▣ 새로운 기본 테이블을 생성하는 데 사용하며
테이블의 이름과 함께 각 속성과 데이터 유형을 기술함
- ▣ 데이터 유형
: INTEGER, FLOAT, DECIMAL(i,j), CHAR(n),
VARCHAR(n)
- ▣ 속성에 Primary Key, Foreign Key,
UNIQUE, NOT NULL 을 명시할 수 있음

1 스키마 생성문



```
CREATE TABLE DEPARTMENT (
    DNAME VARCHAR(10) NOT NULL,
    DNUMBER INTEGER,
    MGRSSN CHAR(9),
    MGRSTARTDATE CHAR(9),
    PRIMARY KEY (DNUMBER),
    UNIQUE (DNAME),
    FOREIGN KEY (MGRSSN) REFERENCES EMP
);
```

04 SQL에서 스키마 생성, 소멸, 변경문

2 스키마 소멸문

- 🔍 DROP TABLE 은 테이블과 관련
제약조건들을 모두 제거함
- 🔍 TRUNCATE TABLE 은 테이블 스키마는
유지하지만 모든 투플들을 삭제함
- 🔍 두 명령 모두 롤백으로 복구가 안 되는 것에 주의
- 🔍 예제)

DROP TABLE DEPENDENT;

TRUNCATE TABLE DEPENDENT ;

04 SQL에서 스키마 생성, 소멸, 변경문

3 스키마 변경문

- 🔍 테이블 이름의 변경, 속성의 추가 또는 제거, 속성 이름의 변경, 속성의 도메인 변경, 속성에 대한 제약조건의 추가 또는 삭제 명령
- 🔍 속성 추가
 - 테이블에 포함된 모든 투플들에 새로 추가된 속성에 대해 NULL 값이 설정되기 때문에 추가되는 속성에 대해 NOT NULL 제약조건을 사용할 수 없음
 - 예제)

ALTER TABLE EMP

ADD COLUMN JOB VARCHAR(12) ;

04 SQL에서 스키마 생성, 소멸, 변경문

3 스키마 변경문

🔍 속성 삭제

- 데이터가 있는 경우에도 삭제 가능
- 한 번의 명령에 하나의 속성만 제거 가능
- 예제)

ALTER TABLE Res

DROP COLUMN rdate ;

04 SQL에서 스키마 생성, 소멸, 변경문

3 스키마 변경문

🔍 속성 수정

- 속성의 데이터 타입 변경
- 기존 데이터가 훼손될 수 있기 때문에 속성의 크기를 늘릴 수는 있지만 줄일 수는 없음
- 속성의 디폴트 값, NOT NULL 제약조건의 변경
- 예제)

ALTER TABLE Sailor

MODIFY COLUMN sname varchar(50) ;

04 SQL에서 스키마 생성, 소멸, 변경문

3 스키마 변경문

🔍 속성 이름 변경

- 해당 속성과 관련된 제약조건에 대해서도 자동적으로 변경됨
- 예제)

ALTER TABLE Sailor

RENAME COLUMN sname to name ;

04 SQL에서 스키마 생성, 소멸, 변경문

3 스키마 변경문

🔍 제약조건 추가

- 테이블 생성 이후에 속성에 제약조건을 추가할 수 있도록 함
- 예제)

ALTER TABLE Sailor

ADD CONSTRAINT sname_unique

UNIQUE (sname) ;

04 SQL에서 스키마 생성, 소멸, 변경문

3 스키마 변경문

🔍 제약조건 삭제

- 테이블 생성 이후에 속성에 걸린 제약조건을 삭제할 수 있도록 함
- 제약조건 명을 알기 위해서는 제약조건의 조회 할 수 있어야 함 (제약조건을 조회하는 명령은 DBMS마다 다름)
- 예제)

ALTER TABLE Sailor

DROP CONSTRAINT sname_unique ;

5

상용 DBMS 상에서 작성 실습

05 상용 DBMS 상에서 작성 실습

1 실습 참고 사이트

- 🔍 Oracle 사이트 <https://www.oracle.com/kr/>
- 🔍 MySQL 사이트 <https://dev.mysql.com/>
- 🔍 MariaDB 사이트 <https://mariadb.org/>
- 🔍 PostgreSQL 사이트 <https://www.postgresql.org/>

- 🔍 DBA 커뮤니티 구루비 <http://www.gurubee.net/>
- 🔍 DB 사랑 넷 <http://database.sarang.net/>

- 🔍 소프트캠퍼스 오라클(Oracle) 데이터 베이스11g [youtube.com](https://www.youtube.com)
- 🔍 한빛미디어 이것이 MySQL이다 [youtube.com](https://www.youtube.com)