MYONGJI UNIVERSITY

SQL #3 - DDL & DML

MYONGJI UNIVERSITY

DDL

Data Definition Language

- Create Table, Drop Table , Truncate Table, Alter Table
- Data Type
- Constraint (NOT NULL, DEFAULT, CHECK, REFERENCE)

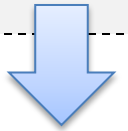
DDL 요약

- CREATE TABLE: 테이블 생성
- ALTER TABLE: 테이블 관련 변경
- DROP TABLE: 테이블 삭제
- RENAME: 이름 변경
- TRUNCATE: 테이블의 모든 데이터 삭제
- COMMENT: 테이블에 설명 추가

테이블 생성

- CREATE TABLE문 이용
- 테이블이름, 컬럼 이름, 데이터 타입 등 정의

```
CREATE TABLE book (
       bookno NUMBER(5),
       title VARCHAR2(50),
       author VARCHAR2(10),
       pubdate DATE
);
```



bookno	title	author	pubdate
1	토지	박경리	2005-03-12
2	슬램덩크	다케이코	2006-04-05
…	…	…	…

Subquery를 이용한 테이블 생성

- Subquery의 결과와 동일한 테이블 생성됨
- 질의 결과 레코드들이 포함됨
- NOT NULL 제약 조건 만 상속됨

```
CREATE TABLE empSALES
AS
      SELECT * FROM emp
      WHERE job = 'SALES';
```

Naming Rules

- 테이블, 컬럼, … 등의 이름 명명 규칙
  - 문자로 시작
  - 30자 이내
  - A-Z, a-z, 0-9, _, $, #
  - 같은 유저가 소유한 다른 Object의 이름과 겹치지 않아야 함 (다른 유저 소유의 object와는 같을 수도 있음)
  - 오라클 예약어는 사용할 수 없음

TABLE 종류

- User Tables:
  - Are a collection of tables created and maintained by the user
  - Contain user information

- Data Dictionary:
  - Is a collection of tables created and maintained by the Oracle Server
  - Contain database information

기본 데이터 타입

Data type	Description
VARCHAR2(size)	가변길이 문자열 (최대 4000byte)
CHAR(size)	고정길이 문자열 (최대 2000byte)
NUMBER(p,s)	가변길이 숫자. 전체 p자리 중 소수점 이하 s자리 (p:38, s:-84~127, 21Byte) 자리수 지정 없으면 NUMBER(38)
DATE	고정길이 날짜+시간, 7Byte

- 참고
  - VARCHAR2와 CHAR의 차이점
  - INT, FLOAT 등의 ANSI Type도 내부적으로 NUMBER(38)로 변환됨

Data type	Description
NCHAR(size)	national character set에 따라 결정되는 size 만큼의 고정길이 character data로 최대 2000byte까지 가능. 디폴트는 1 character.
NVARCHAR2 (size)	national character set에 따라 결정되는 size 만큼의 가변길이character data로 최대 4000 byte까지 가능하며 반드시 길이를 정해 주어야 함.
LONG	가변 길이character data로 최대 2 gigabyte까지 가능.
RAW (size)	가변 길이raw binary data로 최대 2000 까지 가능하며 반드시 길이를 주어야 함.
LONG RAW	가변길이 raw binary data로 최대2 gigabyte까지 가능.
BLOB	Binary data로 4 gigabyte까지 가능.
CLOB	Single-byte character data로 4 gigabyte까지 가능.
NCLOB	national character set까지 포함한 모든 character data로 4 gigabyte까지 가능.
BFILE	외부 화일로 저장된 binary data로 4 gigabyte까지 가능.
ROWID	Row의 물리적 주소를 나타내는 binary data로 extended rowid 는 10 byte, restricted rowid는 6 byte 길이.
TIMESTAMP	Date값을 미세한 초 단위까지 저장. NLS_TIMESTAMP_FORMAT 형식으로 처리.
INTERVAL YEAR TO MONTH	두 datetime 값의 차이에서 YEAR와 MONTH값 만을 저장.
INTERVAL DAY TO SECOND	두 datetime 값의 차이를 DAY, HOUR, MINUTE, SECOND 까지 저장.

ALTER TABLE

- 컬럼 추가
  - ALTER TABLE book **ADD** (pubs VARCHAR2(50));
- 컬럼 수정
  - ALTER TABLE book **MODIFY** (title VARCHAR2(100));
- 컬럼 삭제
  - ALTER TABLE book **DROP** author;
- UNUSED 컬럼
  - ALTER TABLE book **SET UNUSED** (author);
    ALTER TABLE book **DROP UNUSED COLUMNS**;

기타 테이블 관련 명령

- 테이블 삭제
  - **DROP TABLE** book;
- 데이터 삭제
  - **TRUNCATE TABLE** book;
- Comment
  - **COMMENT ON TABLE** book **IS** 'this is comment';
- RENAME
  - **RENAME** book **TO** article;

- 주의:
  - ROLLBACK의 대상이 아님!

제약조건

- Constraint
  - Database 테이블 레벨에서 특정한 규칙을 설정해둠
  - 예상치 못한 데이터의 손실이나 일관성을 어기는 데이터의 추가, 변경 등을 예방함
- 종류
  - **NOT NULL**
  - **UNIQUE**
  - **PRIMARY KEY**
  - **FOREIGN KEY**
  - **CHECK**

제약조건 정의

- Syntax

```
CREATE TABLE 테이블이름 (
        컬럼이름 datatype [DEFAULT 기본값] [컬럼제약조건],
        컬럼이름 datatype [DEFAULT 기본값] [컬럼제약조건],
        …
        [테이블 제약조건] …);
```

  - 컬럼 제약조건: [CONSTRAINT *이름*] constraint_type
  - 테이블제약조건: [CONSTRAINT *이름*] constraint_type(column,..)

- 주의
  - 제약조건에 이름을 부여하지 않으면 Oracle이 Sys-C*n*의 형태로 자동 부여

제약조건 (1/3)

- NOT NULL
  - NULL 값이 들어올 수 없음
  - 컬럼형태로만 제약조건 정의할 수 있음 (테이블 제약조건 불가)

```
CREATE TABLE book (
        bookno NUMBER(5) NOT NULL
);
```

- UNIQUE
  - 중복된 값을 허용하지 않음 (NULL은 들어올 수 있음)
  - 복합 컬럼에 대해서도 정의 가능
  - 자동적으로 인덱스 생성

```
CREATE TABLE book (
        bookno NUMBER(5) CONSTRAINT c_emp_u UNIQUE
);
```

제약조건 (2/3)

- PRIMARY KEY
  - NOT NULL + UNIQUE (인덱스 자동 생성)
  - 테이블 당 하나만 나올 수 있음
  - 복합 컬럼에 대해서 정의 가능 (순서 중요)

```
CREATE TABLE book (
        ssn1 NUMBER(6),
        ssn2 NUMBER(7),
        PRIMARY KEY (ssn1,ssn2)
);
```

- CHECK
  - 임의의 조건 검사. 조건식이 참이어야 변경 가능
  - 동일 테이블의 컬럼만 이용 가능

```
CREATE TABLE book (
        rate NUMBER CHECK (rate IN (1,2,3,4,5))
);
```

제약조건 (3/3)

- FOREIGN KEY
  - 참조 무결성 제약
  - 일반적으로 REFERENCE 테이블의 PK를 참조
  - REFERENCE 테이블에 없는 값은 삽입 불가
  - REFERENCE 테이블의 레코드 삭제 시 동작
    - **ON DELETE CASCADE**: 해당하는 FK를 가진 참조행도 삭제
    - **ON DELETE SET NULL**: 해당하는 FK를 NULL로 바꿈

```
CREATE TABLE book (
        …
        author_id NUMBER(10),
        CONSTRAINT c_book_fk FOREIGN KEY (author_id)
        REFERENCE author(id)
        ON DELETE SET NULL
);
```

ADD / DROP CONSTAINTS

- 제약조건 추가
  - **ALTER TABLE** *테이블이름* **ADD CONSTRAINT** …
  - NOT NULL은 추가 못함

```
ALTER TABLE emp ADD CONSTRAINT emp_mgr_fk
   FOREIGN KEY (mgr)REFERENCES emp(empno);
```

- 제약조건 삭제
  - **ALTER TABLE** *테이블이름* **DROP CONSTRAINT** *제약조건이름*
  - PRIMARY KEY의 경우 FK 조건이 걸린 경우에는 CASCADE로 삭제해야함

```
ALTER TABLE book DROP CONSTRAINT c_emp_u;

ALTER TABLE dept DROP PRIMARY KEY CASCADE;
```

ENABLE/DISABLE CONSTRAINTS

- 제약조건 비활성화
  - 제약조건 검사를 중지함
  - CASCADE를 사용하여서 의존되어 있는 다른 조건을 함께 중지시킬 수 있음
  - 대규모 데이터 변경 등의 속도를 빠르게 함

```
ALTER TABLE emp DISABLE CONSTRAINT c_emp_pk CASCADE;
```

- 제약조건 활성화
  - 중지되어 있던 제약조건 검사를 활성화함
  - UNIQUE, PRIMARY KEY의 경우 인덱스 자동 생성

```
ALTER TABLE emp ENABLE CONSTRAINT c_emp_pk CASCADE;
```

CASCADE CONSTRAINT

- 제약조건이 걸려있는 테이블이나 컬럼은 삭제시 에러 발생
- 컬럼이나 테이블 DROP할 때 관련 제약조건도 함께 삭제 할 때

```
ALTER TABLE emp DROP (deptno) CASCADE CONSTRAINT;
```

```
DROP TABLE emp CASCADE CONSTRAINT;
```

Data Dictionary

- Oracle이 관리하는 모든 정보를 저장하는 카탈로그
- 내용
  - 모든 스키마 객체 정보, 스키마 객체의 공간 정보, 컬럼의 기본값, 제약 조건 정보, 오라클 사용자 정보, 권한 및 롤 정보, 기타 데이터베이스 정보 …
- Base-Table과 View로 구성됨
  - VIEW의 Prefix
    - USER: 로그인한 사용자 레벨
    - ALL: 모든 사용자 정보
    - DBA: 관리자
- SYS scheme에 속함
- 주의
  - DICTIONARY의 테이블이나 컬럼 이름은 모두 대문자 사용!

Dictionary 예

- 모든 Dictionary 정보

```
SELECT   * FROM      DICTIONARY
```

- 사용자 스키마 객체 확인 (테이블)

```
SELECT object_name
FROM user_objects
WHERE object_type = 'TABLE';
```

- 제약조건 확인 (EMP 테이블의)

```
SELECT constraint_name, constraint_type,search_condition
FROM user_constraints
WHERE table_name = 'EMP';
```

- 제약조건 컬럼 확인

```
SELECT constraint_name, column_name
FROM user_cons_columns
WHERE table_name = 'EMP';
```

Dictionary 예(2/2)

- 모든 사용자 확인

```
SELECT username,
       default_tablespace,
       temporary_tablespace
FROM DBA_USERS;
```

MYONGJI UNIVERSITY

DML

INSERT, DELETE, UPDATE

Data Manipulation Language

- 종류
  - Add new row(s)
    - **INSERT INTO** ***테이블이름*** **[(*컬럼리스트*)] VALUES (*값리스트*) ;**
  - Modify existing rows
    - **UPDATE *테이블이름* SET *변경내용* [WHERE *조건*];**
  - Remove existing rows
    - **DELETE FROM *테이블이름* [WHERE *조건*];**
- 트랜잭션의 대상
  - 트랜잭션은 DML의 집합으로 이루어짐.

INSERT

- 묵시적 방법: 컬럼 이름. 순서 지정하지 않음.
  테이블 생성시 정의한 순서에 따라 값 지정

```
INSERT INTO dept VALUES (777, 'MARKETING', NULL);
```

- 명시적 방법: 컬럼 이름 명시적 사용. 지정되지 않은 컬럼 NULL 자동 입력

```
INSERT INTO dept(dname, deptno) VALUES ('MARKETING', 777);
```

- Subquery 이용: 타 테이블로부터 데이터 복사 (테이블은 이미 존재하여야 함)

```
INSERT INTO deptusa
       SELECT deptno, dname FROM dept WHERE country = 'USA';
```

  - 참고: CREATE TABLE AS SELECT는 없는 테이블을 생성 & 데이터 복사

UPDATE

- 조건을 만족하는 레코드를 변경
  - 10번 부서원의 월급 100인상 & 수수료 0으로 변경

```
UPDATE emp SET sal = sal + 100, comm = 0
WHERE deptno = 10;
```

- WHERE 절이 생략되면 모든 레코드에 적용
  - 모든 직원의 월급 10%인상

```
UPDATE emp SET sal = sal * 1.1
```

- Subquery를 이용한 변경
  - 담당업무가 'SCOTT'과 같은 사람들의 월급을 부서 최고액으로 변경

```
UPDATE emp SET sal = (SELECT MAX(sal) FROM emp)
WHERE job = SELECT job FROM emp WHERE ename='SCOTT');
```

DELETE

- 조건을 만족하는 레코드 삭제

```
DELETE FROM emp WHERE ename = 'SCOTT';
```

- 조건이 없으면 모든 레코드 삭제 (주의!)
  - 모든 직원 정보 삭제

```
DELETE FROM emp;
```

- Subquery를 이용한 DELETE
  - 'SALES'부서의 직원 모두 삭제

```
DELETE FROM emp WHERE deptno =
            (SELECT deptno FROM dept WHERE dname = 'SALES');
```

참고

- 데이터 입력, 수정시 자주 사용되는 Pseudo 컬럼
  - USER : Current user name.
  - SYSDATE : Current date and time.
  - ROWID : Location information of rows

```
INSERT INTO emp(eno, hiredate) VALUES (200, SYSDATE);
```

- DEFAULT: default값이 정의된 컬럼에 기본값을 입력할 경우 사용할 수 있음

```
INSERT INTO book VALUES (200,'Gems', DEFAULT);
```

- DELETE 와 TRUNCATE의 차이점
  - Delete는 Rollback 가능 but 대량의 log 등을 유발하므로 Truncate보다 느림

- 모든 DML문은 Integrity Constraint를 어길 경우 에러 발생

학습마무리

- DDL: 데이터 정의
  - CREATE TABLE: 테이블 생성
  - ALTER TABLE: 테이블 관련 변경
  - DROP TABLE: 테이블 삭제
  - RENAME: 이름 변경
  - TRUNCATE: 테이블의 모든 데이터 삭제
  - COMMENT: 테이블에 설명 추가
- DML
  - INSERT INTO … VALUES …
  - UPDATE … SET …
  - DELETE … FROM