

## < 15일차 강의>

>>데이터베이스의 정의

>>파일 시스템 (옛날 방식)

1. 데이터 종속
2. 데이터 중복

>> 데이터 중복 시 문제점 4가지

>>데이터베이스의 특징

>>오라클 삭제방식

>>오라클 설치

>>내 계정 확인

Show user;

>>다른 유저 확인

Select \* from user\_users;

>>사용자 계정 만들기

Create user Scott identified by tiger account unlock;

>>스캇에게 권한 주기위해 시스템 계정으로 접속한다

Connect system/oracle12

>>스캇에게 권한 주기

Grant resource, connect  
To Scott;

>>계정 생성: create user 유저명 identified by 비번 account unlock;

>>권한 생성: grant resource, connect to 유저명;

>>테이블 만들기

>>테이블 골격확인  
Desc amp;

>>sql 설치후 실행\

>>데이터 모델

>>데이터 모델링의 이해

## < DB 구현 정리 >

>> 모델링의 이해  
추상화, 단순화, 명확화

>>데이터 모델링의 3단계 진행

- 개념적 데이터 모델링
- 논리적 데이터 모델링
- 물리적 데이터 모델링

>>프로젝트 라이프 사이클

>>스키마(Schema) : db계정 -> 스키마

- 개념 스키마
- 외부 스키마
- 내부 스키마

>> 데이터 모델링의 중요한 세가지 개념

엔티티 - 테이블

속성 - 컬럼

관계

>> IE / Crow's Foot 표기법

>> 엔티티 개념

>> 엔티티의 특징

>> 속성의 개념

>> 속성의 정의

>> 속성의 명명규칙

>> 도메인(Domain)

>> 관계형 데이터베이스

>> SQL (Structured Query Language) 의 정의

>> < 데이터의 활용 >

- 데이터 조작어 (DML : Data Manipulation Language)

Select

Insert

Update

delete

>> < 데이터의 구현>

- 데이터 정의어 (DDL : Data Definition Language)

Create

Alter

Drop

Rename

- 데이터 제어어 (DCL : Data Control Language)

Grant

Revoke

- 트랜잭션 제어어 (TCL : Transaction Control Language)

Commit

Rollback

< 17일차 강의>

>> select : 조회하다

>> 생성된 테이블 볼수있는 조작어

Select \* from tab;

>> 테이블 만들기

>> 데이터 유형 - 크기

- 글자 Varchar2( )

- 숫자 number( 전체 길이, 소수점 이하 자리수)
- 날짜 date (sys date- 시스템상 날짜)

>> 테이블 삭제

Drop table 테이블명;

>> commit 이란?

>> 트랜잭션 이란?

>> 트랜잭션의 특징

- 원자성 - all or nothing
- 일관성
- 고립성
- 지속성

@ 위의 일련의 과정을 DBMS가 해주고 있다  
우리는 명령어만 치면됨

>> 트랜잭션 요약

데이터 처리의 한단위

>> TCL

- commit
- rollback

< 컬럼 추가, 삭제, 속성 변경 >

>> 테이블 구조 변경하는 명령어 ALTER

Alter table 명령문

DDL

- 컬럼의 추가, 삭제, 컬럼의 타입이나 길이를 변경할때 사용
- 테이블 구조를 변경하면 기존 데이터에 영향을 주게 된다

>> 새로운 컬럼 추가

ALTER TABLE 테이블명  
ADD (컬럼명, 데이터 타입, ...)

>> ROLLBACK 안되는 경우

- COMMIT 한것에 대해선 ROLLBACK 되지않음
- DROP TABLE 한것에 대해선 ROLLBACK 되지않음

>> 기존 컬럼 속성 변경하기

- 테이블에 이미 존재하는 컬럼을 변경

ALTER TABLE 테이블명  
MODIFY (컬럼명, 데이터 타입, ...)

>> 컬럼을 변경한다는 것은

- 컬럼에 대한 데이터 타입이나 크기, 기본값을 변경한다는 의미

>> 기존 컬럼 삭제하기

Alter table 테이블명  
Drop column 컬럼명;

>> 기존 컬럼 속성 변경하기

Alter table 테이블명;  
Modify (컬럼명 데이터 타입, ...)

>>아래 과정 수행 중 오류뜸

기존 JOB 9

이후 JOB 30

그 다음 JOB 5 (오류 발생!)

cannot decrease column length because some value is too big  
값 데이터?

>> JOB에 PRESIDENT(9글자) 이걸 5자로 줄이려면 값에 손실이 오므로 오류 발생

!!!!꼭 기억!!<정리 - 수정은 마음대로 할수 없다>

>>데이터 길이를 변경하려는 경우

1) 해당 컬럼에 자료가 있는 경우

@ 크기를 늘릴 수는 있지만

현재 가지고 있는 데이터의 크기보다 작은 크기로 변경할 수 없다.

@컬럼의 데이터 타입을 변경할 수 없다

>> 꼭 5글자로 바뀌어야 한다면 어떻게?

데이터를 없애 버리면 가능

```
DELETE FROM EMP02;
```

```
ALTER TABLE EMP02
```

```
MODIFY (JOB VARCHAR2(5));
```

2) 해당 컬럼에 자료가 없는 경우

@ 컬럼의 크기를 변경할 수 있다(마음대로)

@ 컬럼의 데이터 타입을 변경 할 수 있다

단, 해당 컬럼이 NULL 값만 가지고 있으면 데이터 유형을 변경할 수 있다.

(미래에 데이터를 넣을거지만 어떤 것을 넣을지 모를때 NULL로 구성해둬 - 하지만 NULL은 최대한 없애야함 - 공간이 제대로 활용되지 않고 있다는 것이므로)

오류발생!

column to be modified must be empty to change datatype

>> 데이터 타입을 변경하려면  
존재하고 있는 데이터가 없어야 된다  
(숫자를 데이터로 바꿀수 없다)

```
DELETE FROM EMP02;  
ALTER TABLE EMP02  
MODIFY (SAL VARCHAR2(100));
```

>> 컬럼명 바꾸기  
Alter table 테이블명  
rename column 변경해야 할 컬럼명 to 새로운 컬럼명;

>> 테이블명 바꾸기  
Rename 변경 전 테이블명 to 변경 후 테이블명

>> 테이블 내용을 수정 UPDATE문  
업데이트 하나의 단위는 행이다

@전체 행의 내용 변경  
Update 테이블명  
Set 변경하고자 하는 데이터가 있는 컬럼명 = value1  
Column2 = value2

@테이블의 특정 행만 변경한다  
Where 컬럼명 = 값( value)

<WHERE문 -> 조건문>  
UPDATE 테이블명  
SET 변경하고자 하는 데이터가 있는 컬럼명 = VALUE1  
COLUMN2 = VALUE2  
WHERE 조건문;



예)

```
UPDATE DEPT01  
SET DNAME = 'ACCOUNT'  
WHERE DNAME = 'ACCOUNTING';  
(ACCOUNTING을 ACCOUNT 로 변경한 것)
```

>> 테이블의 불필요한 행을 삭제하는 DELETE  
Delete from 삭제를 원하는 정보가 들어있는 테이블명  
Where 조건절;

< COMMIT 명령어 ROLLBACK 명령어 >  
@데이터 무결성이 보장된다  
@영구적인 변경 전에 데이터의 변경사항을 확인할 수 있다  
@논리적으로 연관된 작업을 그룹화 할 수 있다

트랜잭션은  
ALL OR NOTHING

DML (SELECT, INSERT, DELETE, UPDATE)  
VS  
DDL (CREATE, ALTER, DROP, RENAME)

@DML에 관련된 작업은 어딘가 임시 저장소에 보관된다  
그 내용은 확인이 가능하다  
임시저장소에 있는 내용은 ROLLBACK 가능하다

@임시저장소가 아닌 실제 저장소에 보관된 데이터는 ROLLBACK 불가능하다

@DDL 은 내용을 수행 후 마지막에 COMMIT; 를 수행한다

## < 계정을 만드는 방법 >

>>DBA 권한을 가진 system 만이 계정을 만들고 권한을 줄수 있음

>>계정을 만들자

```
create user totoExam IDENTIFIED by toto;
```

>>권한을 주자

```
GRANT CONNECT TO totoExam;
```

>>테이블을 만드는 권한을 주자

```
GRANT RESOURcE TO totoExam;
```

>>계정을 삭제하자!

```
drop user totoexam;
```

>>계정을 언락하자

```
create user totoExam IDENTIFIED by toto account unlock;
```

## < 제약 조건 >

>>!!시험!!무조건 알아야

< 무결한 데이터의 5가지 제약 조건 >

(아예 애초에 무결한 데이터가 되도록 원천봉쇄하자는 것)

무결성 제약조건      역할

NOT NULL              NULL을 허용하지 않는다

UNIQUE              중복된 값을 허용하지 않는다. 항상 유일한 값을 갖  
도록 한다  
(각 행 구분)

PRIMARY KEY              NULL을 허용하지 않고 중복된 값을 허용하지  
않는다  
NOT NULL 조건과 UNIQUE 조건을 결합한 형태

FOREIGN KEY              참조되는 테이블의 컬럼의 값이 존재하면 허용  
한다

CHECK              설정한 값만을 허용한다

>> 제약조건 확인하기(CONSTRAINTS)

DESC USER\_CONSTRAINTS;

SELECT \*  
FROM USER\_CONSTRAINTS;

@<NOT NULL >

>>오류

cannot insert NULL into ("TOTO"."EMP02"."EMPNO")  
테이블 생성할때 컬럼에 뒤에 NOT NULL 을 붙였기 때문에

```
CREATE TABLE EMP02(  
    EMPNO  NUMBER(4) NOT NULL,  
    ENAME  VARCHAR2(10) NOT NULL,  
    JOB    VARCHAR2(9),  
    DEPTNO NUMBER(2)
```

@< UNIQUE >

유일한 값만 허용하는 유니크 제약조건

특정 컬럼에 자료가 중복되지 않게 하는 것  
즉, 지정된 컬럼에 유일한 값이 입력될 수 있도록 하는 것

```
CREATE TABLE EMP03(  
    EMPNO  NUMBER(4) UNIQUE,           --UNIQUE 제약조  
    건  
    ENAME  VARCHAR2(10) UNIQUE,  
    JOB    VARCHAR2(9),  
    DEPTNO NUMBER(2)  
);
```

```
INSERT INTO EMP03  
VALUES(7499, 'AAA', 'SALES', 30);
```

```
INSERT INTO EMP03  
VALUES(7499, 'BBB', 'MANAGER', 20);
```

!!오류!!

unique constraint (TOTO.SYS\_C007017) violated

>>제약조건명을 명시하여 작성

[형식]

컬럼명 타입 CONSTRAINT 제약조건명 제약조건

```
CREATE TABLE EMP04(  
    EMPNO  NUMBER(4) CONSTRAINT EMP04_EMPNO_UK  
    UNIQUE,  
    ENAME  VARCHAR2(10) CONSTRAINT EMP_ENAME_NN NOT  
    NULL,  
    JOB    VARCHAR2(9),  
    DEPTNO NUMBER(2)  
);
```

>>제약조건명 명명규칙

테이블명\_컬럼명\_제약조건명

```
CONSTRAINT 테이블명_컬럼명_UK(UNIQUE KEY)  
CONSTRAINT 테이블명_컬럼명_NN(NOT NULL)
```

@@!!중요!!

< PRIMARY KEY > - 기본키  
데이터 구분을 위한  
PRIMARY KEY 제약조건

이름이 같은 사람은 존재해도 주민번호가 같은 사람은 없다.  
그래서 주민번호는 사람을 구분하기 위해 사용할 수 있다  
테이블에 저장된 데이터를 구분하기 위해서는  
특정 컬럼에 PRIMARY KEY 제약조건을 주면 좋다.

!!어려움!!

< 참조 무결성을 위한 FOREIGN KEY 제약 조건 >  
참조 무결성이란?  
테이블 사이의 관계에서 발생하는 개념

마트

과자 테이블

제품번호   과자명   과자가격   제조날짜  
제품번호는 숫자 4자리   유일하고 널이 들어갈수 없다  
과자명은 글자50  
과자가격은 정수 4자리, 소수점 1  
제조날짜는 날짜

고객 테이블

사람이름   주민번호   구매물품  
사람이름은 문자20  
주민번호는 문자14 유일하고 널이 들어갈수 없다  
구매물품은 과자 테이블의 제품번호와 도메인이 동일하며  
참조하고 있다

과자 데이터 입력

1000   알아서   SYSDATE

1001   알아서 SYSDATE

1002   알아서 SYSDATE

고객 데이터는 알아서 4개 입력

>>CHECK와 DEFAULT의 제약 조건

DEFAULT 제약 조건 설정하기

BONUS

DEPT

EMP

SALGRADE

TEST

사원   부서

널이 허용되어있는지 없는지 각 컬럼의 여부

기본키 여부

외래키 여부

>>오라클 제약조건 추가wor

기본키 : 유니크 키 + NOT NULL + INDEX

유니크 키 + NOT NULL: