0404

DDL (DATE DEFINITION LANGUAGE)

-데이터 정의 언어이다
-객체를 CREATE / ALTER / DROP 하는 구문을 DDL이라 한다

ALTER (바꾸다, 변조하다)

- 컬럼(추가 , 수정 , 삭제)
- 제약조건(추가 , 삭제) [수정은 안됨]
- 이름 변경 (테이블 , 컬럼 , 제약 조건)

[작성법]
테이블을 수정하는 경우

ALTER TABLE 테이블명 ADD / MODIFY / DROP 수정할 내용 ;

- ADD : 추가
- MODIFY : 수정
- DROP : 삭제

제약 조건을 추가

- ALTER TABLE 테이블명 ADD [CONSTRAINT 제약 조건명] 제약조건 (컬럼명) [REFERENCES 테이블명 [(컬럼명)]]
- 제약조건의 설명

제약조건을 삭제

- ALTER TABLE 테이블명 DROP CONSTRAINT 제약조건명 ;

작성법 중 []대괄호는 생략할 수도 안할 수도 있다.

```
ALTER TABLE DEPT_COPY ADD CONSTRAINT DEPT_TITLE_U UNIQUE(DEPT_TITLE);
```

- DEPT_COPY 테이블 DEPT_TITLE컬럼에 UNIQUE 제약조건을 추가 했다
- DEPT_TITLE_U라는 제약조건명을 추가하고

```
ALTER TABLE DEPT_COPY
ADD CONSTRAINT LOCATION_ID_CHK CHECK (LOCATION_ID IN('L1','L2','L3','L4','L5'));
```

- DEPT_COPY 테이블의 LOCATION_ID 컬럼에 CHECK제약조건을 설정 했다
- 제약조건의 별칭은 LOCATION_ID_CHK 으로 추가했다.

```
ALTER TABLE DEPT_COPY
ADD PRIMARY KEY (DEPT_ID);
```

- DEPT_COPY 테이블에 있는 DEPT_ID 컬럼에 PRIMARY KEY 제약조건을 설정했다
- 별칭은 생략했다

```
ALTER TABLE DEPT_COPY
MODIFY DEPT_TITLE NOT NULL;  --NOT NULL은 MODIFY
```

- DEPT_TITLE 컬럼의 제약 조건을 NOT NULL을 추가 했다
- 이 때 ADD가 아닌 MODIFY를 이용해야 한다. (컬럼의 성질을 추가(ADD)하는 것이 아닌 변경(MODIFY)하는 것으로 인식한다.)

```
--DEPT_COPY에 추가한 제약 조건중 PK빼고 모두 삭제
ALTER TABLE DEPT_COPY DROP CONSTRAINT DEPT_TITLE_U;

ALTER TABLE DEPT_COPY DROP CONSTRAINT LOCATION_ID_CHK;
```

- DEPT_COPY 테이블에 있는 제약조건들의 별칭을 지정해 삭제 했다

```
--NOT NULL만 제거시 MODIFY 사용
ALTER TABLE DEPT_COPY MODIFY DEPT_TITLE CONSTRAINT SYS_C008448 NULL ;
```

- NOT NULL은 MODIFY 사용 생각하기.

컬럼의 ADD/MODIFY/DROP

컬럼 추가 ALTER TABLE 테이블명 ADD (컬럼명 데이터 타입 [DEFAULT '값'])
컬럼 수정 ALTER TABLE 테이블명 MODIFY 컬럼명 데이터 타입; (데이터의 타입 변경)
컬럼명 DEFAULT '값' (기본값 변경)
컬럼 삭제 ALTER TABLE 테이블 명 DROP COLUMN (삭제할 컬럼명)
DROP 삭제할 컬럼명 (이게 더 편하다)

```
ALTER TABLE DEPT_COPY ADD (CNAME VARCHAR2(20));
```

- DEPT_COPY 테이블에 VARCHAR2(20)자료형 CNAME을 추가

```
ALTER TABLE DEPT_COPY
```

```
ADD (LNAME VARCHAR2(30) DEFAULT '한국')
```

- DEPT_COPY에 LNAME을 추가 후 DEFAULT 값을 한국으로 생성

```
ALTER TABLE DEPT_COPY
MODIFY DEPT_ID VARCHAR2(3);
```

- DEPT_ID 컬럼의 자료형을 VARCHAR2(3)으로 변경
- 이때 이미 변경할 값보다 큰 값을 사용 할 경우 변경이 제한된다

```
ALTER TABLE DEPT_COPY
MODIFY LNAME DEFAULT '대한민국';
```

- LNAME이 DEFAULT 값을 대한민국으로 변경한다
- 이때 이미 세팅되어 있는 DEFAULT 값은 변경되지 않는데
- 변경하고 싶을 경우 UPDATE - SET을 사용해야한다.

```
UPDATE DEPT_COPY SET LNAME = '대한민국';
UPDATE DEPT_COPY SET LNAME = DEFAULT;
```

```
ALTER TABLE DEPT_COPY DROP (CNAME) ;
```

- DEPT_COPY 테이블에 CNAME의 컬럼을 삭제
- 이 때 컬럼이 하나더 없는 경우는 테이블이 성립될 수 없기 때문에 허용되지않는다

테이블의 삭제

DROP TABLE 테이블명 [CASCADE CONSTRAINTS]

- 관계가 있는 경우 CASCADE를 이용해 정리해야한다

```
DROP TABLE DEPT_COPY;
```

- DEPT_COPY 테이블을 삭제한다

```
--**관계가 형성된 테이블중 부모 테이블 삭제**
DROP TABLE TB1; --ORA-02449: 외래 키에 의해 참조되는 고유/기본 키가 테이블에 있습니다

--해결 방법 1 : 자식 먼저 삭제하기 (참조하는 테이블이 없으면 삭제 가능)
DROP TABLE TB2;
```

```
DROP TABLE TB1;

--해결 방법 2 : CASCADE CONSTRAINT 옵션 사용
--(제약조건까지 모두 삭제하겠다)
-- == FK 제약조건으로 인해 불가능하지만 제약조건을 없애버려 FK 관계를해재
DROP TABLE TB1 CASCADE CONSTRAINTS;--TKRWP TJDRHD
DROP TABLE TB2;
```

- 삭제는 '관계'를 잘 생각해야한다.

```
--4.컬럼 , 제약조건 , 테이블 이름 변경 (RENAME)

--1) 컬럼명 변경   :  ALTER TABLE 테이블명 RENAME COLUMN 컬럼명 TO 변경명
ALTER TABLE DEPT_COPY RENAME COLUMN DEPT_TITLE TO DEPT_NAME;

--2) 제약 조건명 변경 :  ALTER TABLE 테이블명 RENAME CONSTRAINT 제약조건명 TO 변경명
ALTER TABLE DEPT_COPY RENAME CONSTRAINT PK_DCOPY TO DEPT_COPY_PK;

--3)테이블명 변경 : ALTER TABLE 테이블명 RENAME TO 변경명;
ALTER TABLE DEPT_COPY RENAME TO DCOPY;
```

- 각 형식에 맞게 RENAME을 잘 설정하면 된다

VIEW

VIEW : SELECT문의 실행 결과 RESULT SET을 저장하는 객체

사용목적

- 복잡한 SELECT문을 쉽게 재사용할 수 있다
- 테이블의 모든 정보(진짜 모습)을 감출 수 있어 보안상 유리하다

VIEW 사용지 주의사항

- 가상의 테이블이기 때문에 ALTER 사용 불가능
  - ALTER : DDL (데이터에 직접 접근함)
- VIEW를 통해 INSERT / UPDATE / DELETE를 사용가능하지만 **하지마라 그냥**

```
  [VIEW 생성 방법]
```

```
CREATE [OR REPLACE] [FORCE | NOFORCE] VIEW 뷰이름 [(alias[,alias]...]    AS subquery
[WITH CHECK OPTION]
[WITH READ ONLY];
```

- OR REPLACE : 기존에 동일한 뷰 이름이 존재하는 경우 덮어쓰고 , 없을 경우 새로 생성한다
- FORCE / NOFORCE : 서브쿼리에 사용된 테이블이 존재하지 않아도 뷰 생성 / 테이블이 있어야지만 뷰 생성
- WITH CHECK OPTION : 옵션을 설정한 컬럼의 값을 수정 불가능하게 함
- WITH READ ONLY : 뷰를 이용해 SELECT만 사용가능 (DML를 사용한 뻘짓을 예방 가능하다)

```
--사원의 사번 , 이름 , 부서명 직급명 , 조회 VIEW
CREATE VIEW V_EMP AS
SELECT EMP_ID,EMP_NAME,DEPT_TITLE,JOB_NAME
FROM EMPLOYEE
LEFT JOIN DEPARTMENT ON(DEPT_CODE = DEPT_ID)
JOIN JOB USING(JOB_CODE);
```

- EMPLOYEE를 기반으로 한 VIEW 생성한다
- 이 때 VIEW 생성은 관리자 계정한대 권한을 부여받아야 한다

```
CREATE OR REPLACE VIEW V_EMP AS
SELECT EMP_ID 사번,EMP_NAME 이름 ,DEPT_TITLE 부서명 ,JOB_NAME 직급명
FROM EMPLOYEE
LEFT JOIN DEPARTMENT ON(DEPT_CODE = DEPT_ID)
JOIN JOB USING(JOB_CODE);

--OR REPLACE 같은 뷰가 있어도 생성 가능

SELECT * FROM V_EMP
WHERE 직급명 = '대리';
```

- OR REPLACE를 이용해 이미 있는 V_EMP를 덮어 씌우면서 생성한다
- 이 때 생성한 별칭은 VIEW 컬럼명칭이 된다
- VIEW를 사용 할때 JOB_TITLE를 사용하지 못하고 별칭으로만 사용 가능하다

---

*WITH READ ONLY 옵션 *

```
CREATE OR REPLACE VIEW V_DCOPY2 AS
SELECT DEPT_ID,LOCATION_ID FROM DEPARTMENT
WITH READ ONLY ; --SELECT전용의 VIEW 생성
```

```
INSERT INTO V_DCOPY2 VALUES('D0','L3');
--SQL 오류: ORA-42399: 읽기 전용 뷰에서는 DML 작업을 수행할 수 없습니다.
```

- 뻘짓 못하게 막는다