

2-1. SQL 정의와 종류, 테이블 생성

홍형경

chariehong@gmail.com

2020.06

1. SQL이란

(1) SQL의 정의

- SQL: Structured Query Language의 약자, 구조화된(구조적인) 질의 언어
- RDBMS와 데이터 처리를 위해 소통하는 언어
- 데이터베이스에 있는 데이터를 관리하는데 사용하는 언어
- 집합적 언어 (C, JAVA, PYTHON같은 다른 프로그래밍 언어는 절차적 언어)
- SQL 표준이 존재

1. SQL이란

(2) SQL의 특징

- 영어 기반으로 배우고 사용하기 쉽다
- 데이터 처리(입력, 삭제, 수정) 뿐만 아니라 데이터베이스 객체(테이블, 뷰 ...) 생성, 수정시 사용
- 특정 조건에 맞는 데이터를 한 번에 처리 ➔ 집합적 언어
- 탁월한 데이터 가공 처리

1. SQL이란

(3) SQL 표준

- ANSI(미국 국가표준 협회), ISO(국제 표준화 기구)의 SQL 표준이 존재
- SQL 표준이 존재 → DBMS 종류(오라클, MySQL, MSSQL 등)에 상관없이 사용 가능
- 1986년 최초로 ANSI의 SQL-87 제정
- 상용 RDBMS 발전에 비해 표준 제정이 뒤쳐짐
- 각 RDBMS 제조사들은 새 버전 출시 때 SQL 표준에 맞추려 노력하고 있음

1. SQL이란

(3) SQL 표준

- SQL 표준이 존재 → DBMS 종류(오라클, MySQL, MSSQL 등)에 상관없이 사용 가능
- 과연 이 말이 맞을까?
 - ➔ 일부는 맞고 일부는 틀림
 - ➔ 기본적인 문법은 차이가 없으나, 세부적으로 들어가면 DBMS 종류마다 문법이 상이

1. SQL이란

(3) SQL 표준

- 2 X 3의 결과를 구하는 경우....

DBMS	SQL 구문	결과
오라클	Select 2 * 3 from dual;	6
MSSQL	Select 2 * 3;	6
MySQL (MariaDB)	Select 2 * 3;	6

2. SQL 종류

(1) SQL의 종류

- DDL : 데이터 정의어
- DML : 데이터 조작어
- TCL : 트랜잭션 제어어
- DCL : 데이터 제어어

2. SQL 종류

(2) DDL (Data Definition Language, 데이터 정의어)

- 데이터베이스 객체(테이블, 뷰, 인덱스 등)를 생성, 수정, 삭제
- CREATE : 객체를 생성
- DROP : 생성된 객체를 삭제
- ALTER : 생성된 객체를 수정
- TRUNCATE TABLE : 테이블 데이터를 삭제
- RENAME: 객체의 이름을 변경

2. SQL 종류

(2) DML (Data Manipulation Language, 데이터 조작어)

- 데이터를 조회, 입력, 수정, 삭제
- **SELECT** : 테이블에 저장된 데이터를 조회
- **INSERT** : 테이블에 신규 데이터 입력/저장
- **UPDATE** : 기존 데이터를 수정
- **DELETE** : 데이터를 삭제
- **MERGE**: 조건에 따라 입력과 수정을 동시에 처리

2. SQL 종류

(3) TCL (Transaction Control Language, 트랜잭션 제어어)

- 트랜잭션 처리 ➔ 데이터 변경 작업 발생 시
- COMMIT : DML로 변경된(INSERT, UPDATE, DELETE, MERGE) 데이터를 데이터베이스에 저장/적용
- ROLLBACK : DML로 변경된 데이터를 변경 이전 상태로 되돌림

2. SQL 종류

(4) DCL (Data Control Language, 데이터 제어어)

- 권한을 할당하거나 회수
- GRANT : 사용자에게 권한을 할당
- REVOKE : 할당된 권한을 사용자로부터 회수

2. SQL 종류

(5) TRUNCATE TABLE과 DELETE

- 테이블에 있는 데이터를 삭제
- TRUNCATE TABLE
 - DDL로 데이터 삭제 후 ROLLBACK으로 원복 할 수 없음
 - 해당 테이블 데이터 전체를 삭제
- DELETE
 - DML로 ROLLBACK으로 삭제 이전 상태로 원복 가능
 - 특정 조건에 맞는 데이터만 삭제 가능 (WHERE 절)

2. SQL 종류

(6) 기타

- SQL 문장과 데이터베이스 객체명은 대소문자 구분을 안함
 - 테이블에 있는 데이터 값은 대소문자 구분을 함
 - 데이터 디렉터리에서 저장된 객체명은 모두 대문자로 들어가 있음
- 주석 (Comments) : 주석 처리된 부분은 무시됨
 - 한 줄 주석 : --
 - 여러 줄 주석 : /* */

3. 테이블 생성, 삭제, 수정

(1) 테이블 생성

- CREATE TABLE 문을 사용
- CREATE TABLE 테이블명 (
 컬럼명1 데이터형 NULL | NOT NULL,
 컬럼명2 데이터형 NULL | NOT NULL,
 ...
);

3. 테이블 생성, 삭제, 수정

(1) 테이블 생성

· 사원 (EMP) 테이블 생성

컬럼명	설명	데이터형	NULL 허용여부	PK 여부
EMP_NO	사원번호	VARCHAR2(30)	N	Y
EMP_NAME	사원이름	VARCHAR2(80)	N	
SALARY	월급	NUMBER	Y	
HIRE_DATE	입사일자	DATE	Y	

3. 테이블 생성, 삭제, 수정

(1) 테이블 생성

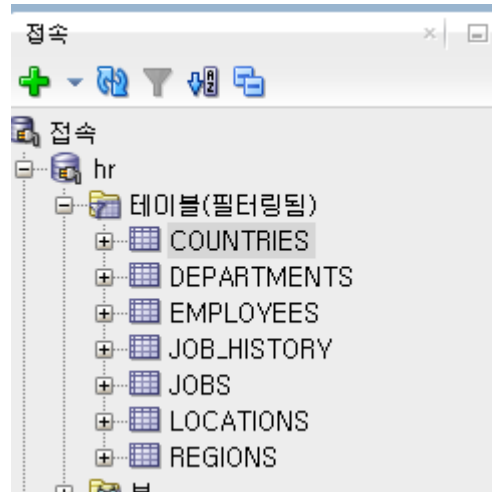
- CREATE TABLE EMP (
 EMP_NO VARCHAR2(30) NOT NULL,
 EMP_NAME VARCHAR2(80) NOT NULL,
 SALARY NUMBER NULL,
 HIRE_DATE DATE NULL
);
- NULL 허용 컬럼일 경우, NULL을 명시하지 않아도 됨

3. 테이블 생성, 삭제, 수정

(1) 테이블 생성

· 생성된 테이블 Layout

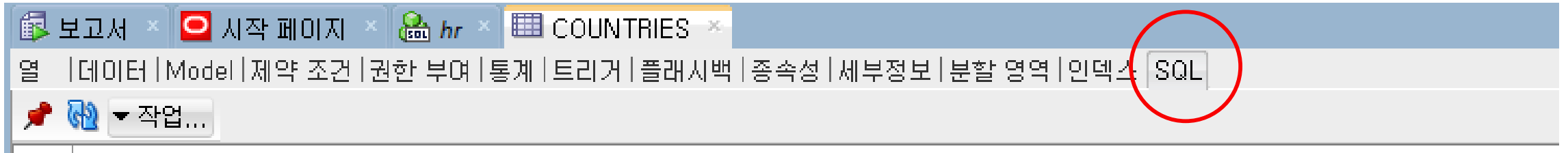
- DESC 테이블명;
- SELECT * FROM EMP;



COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 COUNTRY ID	CHAR(2 BYTE)	No	(null)	1	Primary key of countries table.
2 COUNTRY NAME	VARCHAR2(40 BYTE)	Yes	(null)	2	Country name
3 REGION ID	NUMBER	Yes	(null)	3	Region ID for the country. Foreign key to region id column in the departments tab

3. 테이블 생성, 삭제, 수정

(1) 테이블 생성



```
CREATE TABLE "HR"."COUNTRIES"
(
  "COUNTRY_ID" CHAR(2 BYTE) CONSTRAINT "COUNTRY_ID_NN" NOT NULL ENABLE,
  "COUNTRY_NAME" VARCHAR2(40 BYTE),
  "REGION_ID" NUMBER,
  CONSTRAINT "COUNTRY_C_ID_PK" PRIMARY KEY ("COUNTRY_ID") ENABLE,
  CONSTRAINT "COUNTR_REG_FK" FOREIGN KEY ("REGION_ID")
    REFERENCES "HR"."REGIONS" ("REGION_ID") ENABLE
) SEGMENT CREATION IMMEDIATE
ORGANIZATION INDEX NOCOMPRESS PCTFREE 10 INITRANS 2 MAXTRANS 255 LOGGING
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
TABLESPACE "SYSAUX"
PCTTHRESHOLD 50;

COMMENT ON TABLE "HR"."COUNTRIES" IS 'country table. Contains 25 rows. References with locations table.';
COMMENT ON COLUMN "HR"."COUNTRIES"."COUNTRY_ID" IS 'Primary key of countries table.';
COMMENT ON COLUMN "HR"."COUNTRIES"."COUNTRY_NAME" IS 'Country name';
COMMENT ON COLUMN "HR"."COUNTRIES"."REGION_ID" IS 'Region ID for the country. Foreign key to region_id column in the departments table.';
```

3. 테이블 생성, 삭제, 수정

(2) 테이블 수정

- ALTER TABLE 문 사용
- 새 컬럼 추가, 기존 컬럼 삭제, 기존 컬럼 수정
- 사원명 컬럼 길이를 100 BYTE로 수정
`ALTER TABLE EMP
MODIFY EMP_NAME VARCHAR2(100);`
- `DESC EMP;`

스크립트 출력 x		
작업이 완료되었습니다.(0.601초)		
이름	널?	유형

EMP_NO	NOT NULL	VARCHAR2 (30)
EMP_NAME	NOT NULL	VARCHAR2 (100)
SALARY		NUMBER
HIRE_DATE		DATE

3. 테이블 생성, 삭제, 수정

(2) 테이블 수정

- EMP_NAME2 컬럼 추가 → VARCHAR2(80)

```
ALTER TABLE EMP
```

```
ADD EMP_NAME2 VARCHAR2(80);
```

- DESC EMP;

스크립트 출력 x		
작업이 완료되었습니다.(0.045초)		
이름	널 ?	유형

EMP_NO	NOT NULL	VARCHAR2 (30)
EMP_NAME	NOT NULL	VARCHAR2 (100)
SALARY		NUMBER
HIRE_DATE		DATE
EMP_NAME2		VARCHAR2 (80)

3. 테이블 생성, 삭제, 수정

(2) 테이블 수정

- EMP_NAME2 컬럼명을 EMP_NAME3로 변경

- ALTER TABLE EMP

```
RENAME COLUMN EMP_NAME2  
TO EMP_NAME3;
```

- DESC EMP;

스크립트 출력 x		
작업이 완료되었습니다.(0.146초)		
이름	널 ?	유형

EMP_NO	NOT NULL	VARCHAR2 (30)
EMP_NAME	NOT NULL	VARCHAR2 (100)
SALARY		NUMBER
HIRE_DATE		DATE
EMP_NAME3		VARCHAR2 (80)

3. 테이블 생성, 삭제, 수정

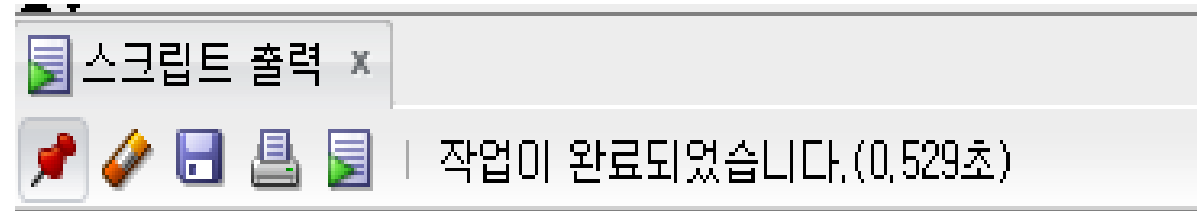
(2) 테이블 수정

- EMP_NAME3 컬럼 삭제

```
ALTER TABLE EMP
```

```
DROP COLUMN EMP_NAME3;
```

- DESC EMP;



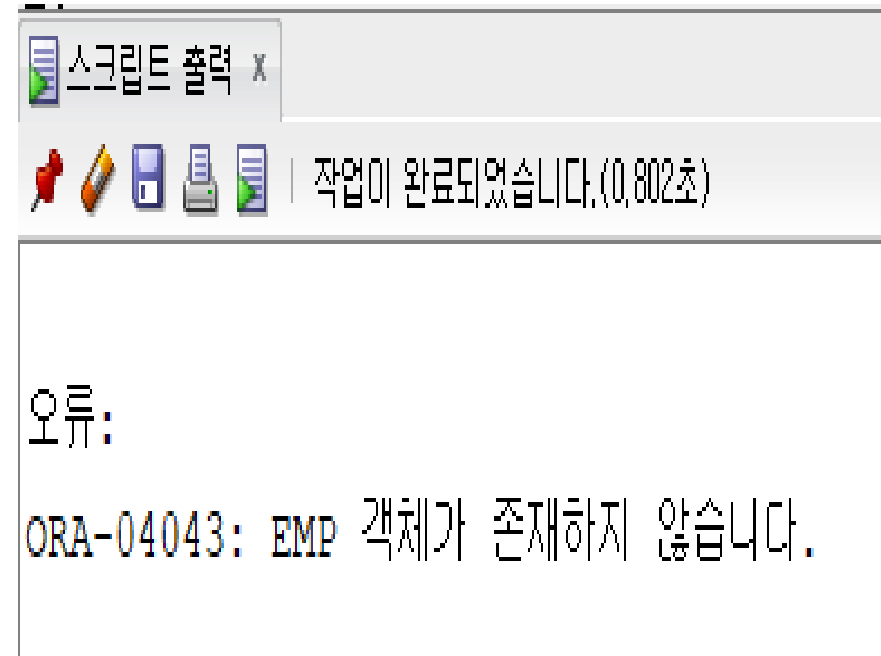
이름	널?	유형

EMP_NO	NOT NULL	VARCHAR2 (30)
EMP_NAME	NOT NULL	VARCHAR2 (100)
SALARY		NUMBER
HIRE_DATE		DATE

3. 테이블 생성, 삭제, 수정

(3) 테이블 삭제

- DROP TABLE 문 사용 → 저장된 데이터도 삭제
- DROP TABLE 테이블명;
- DROP TABLE EMP;
- DESC EMP;



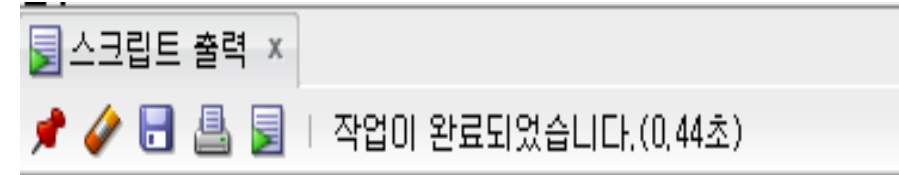
3. 테이블 생성, 삭제, 수정

(4) 기본 키 생성

· 테이블 생성 시 생성1

```
CREATE TABLE EMP (  
    emp_no    VARCHAR2(30) PRIMARY KEY,  
    emp_name  VARCHAR2(80) NOT NULL,  
    salary    NUMBER      NULL,  
    hire_date DATE        NULL  
);
```

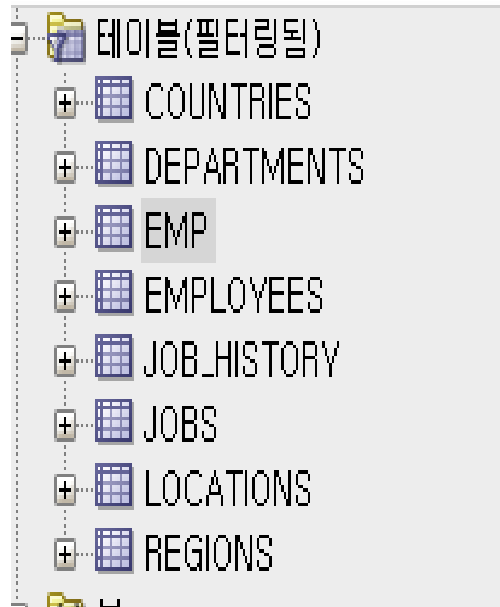
· DESC EMP;



이름	널?	유형
-----	-----	-----
EMP_NO	NOT NULL	VARCHAR2 (30)
EMP_NAME	NOT NULL	VARCHAR2 (80)
SALARY		NUMBER
HIRE_DATE		DATE

3. 테이블 생성, 삭제, 수정

(4) 기본 키 생성



열	데이터	Model	제약 조건	권한 부여	통계	트리거	플래시백	종속성	세부정보	분할 영역	인덱스
				작업...							
	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION								
1	SYS_C007415	Check	"EMP_NAME" IS NOT NULL								
2	SYS_C007416	Primary_Key	(null)								

열	데이터	Model	제약 조건	권한 부여	통계	트리거	플래시백	종속성	세부정보	분할 영역	인덱스
				새로고침: 0							
	COLUMN_NAME	COLUMN_POSITION									
1	EMP_NO	1									

3. 테이블 생성, 삭제, 수정

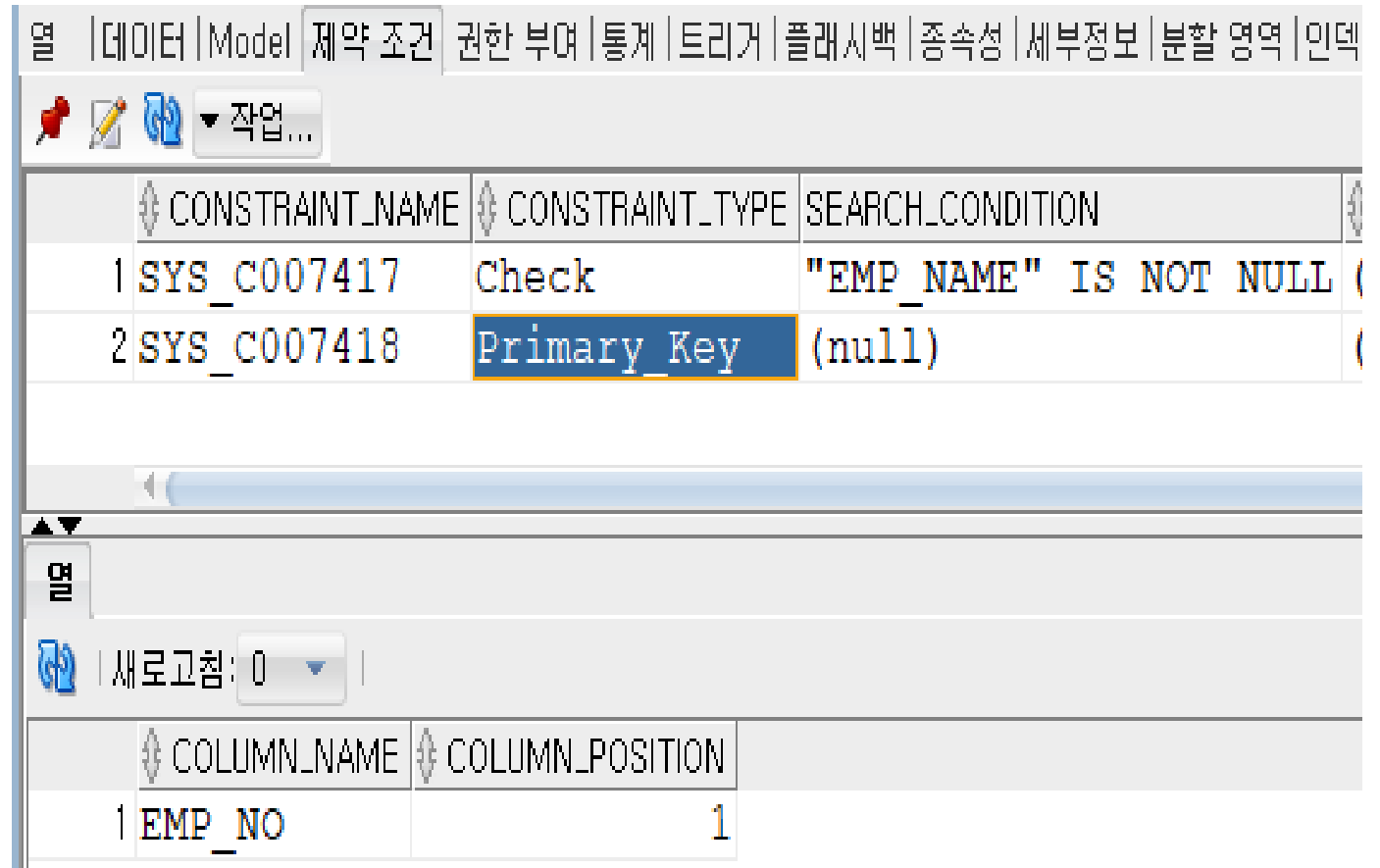
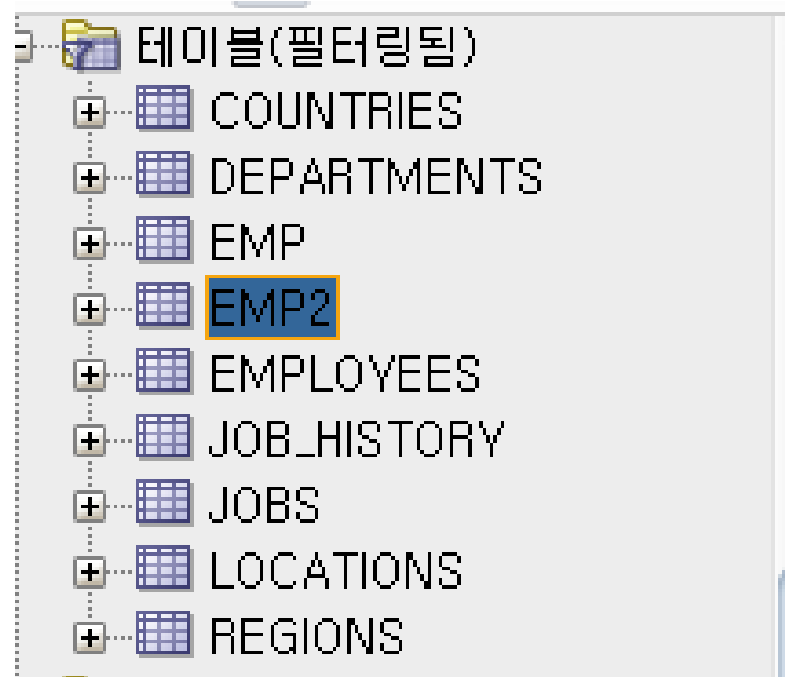
(4) 기본 키 생성

- 테이블 생성 시 생성2

```
CREATE TABLE EMP2 (  
    emp_no      VARCHAR2(30) ,  
    emp_name    VARCHAR2(80) NOT NULL,  
    salary      NUMBER      NULL,  
    hire_date   DATE        NULL,  
    PRIMARY KEY ( emp_no)  
);
```

3. 테이블 생성, 삭제, 수정

(4) 기본 키 생성



3. 테이블 생성, 삭제, 수정

(4) 기본 키 생성

- ALTER TABLE 구문 사용

```
CREATE TABLE EMP3 (  
    emp_no      VARCHAR2(30) ,  
    emp_name    VARCHAR2(80) NOT NULL,  
    salary      NUMBER      NULL,  
    hire_date   DATE        NULL);
```

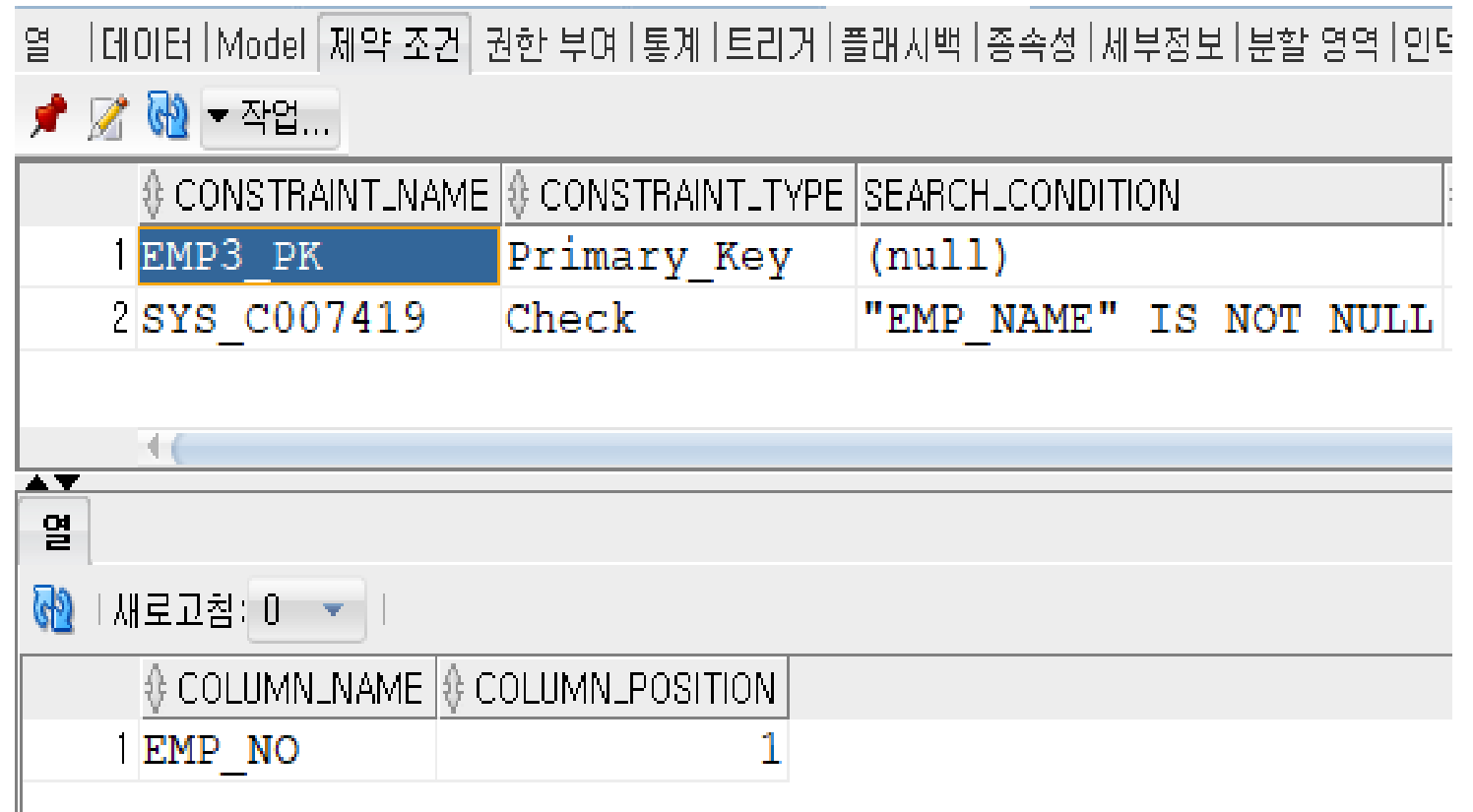
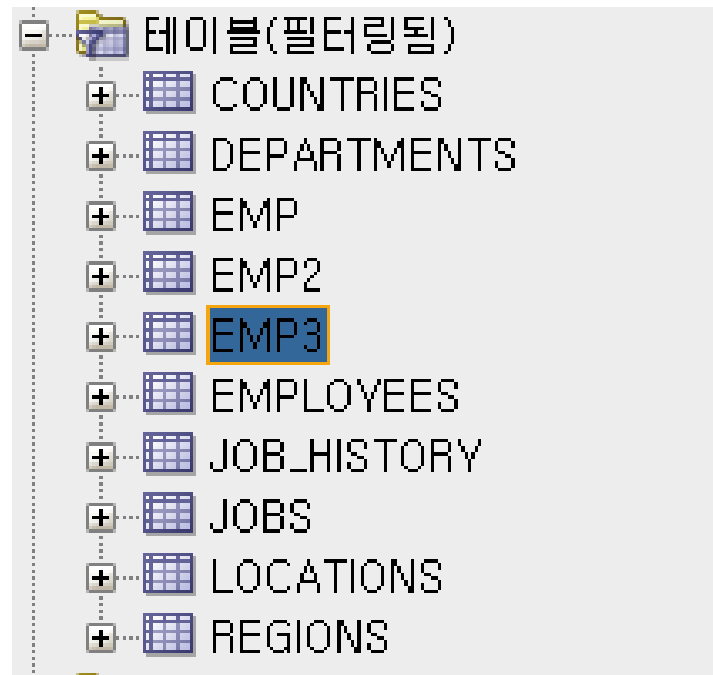
```
ALTER TABLE emp3
```

```
ADD CONSTRAINTS EMP3_PK PRIMARY KEY ( EMP_NO );
```

- ALTER TABLE 구문을 사용해 기본 키 생성 방법 권고

3. 테이블 생성, 삭제, 수정

(4) 기본 키 생성



4. 테이블 생성, 삭제, 수정 실습

· 부서 (DEPT_TEST) 테이블 생성

컬럼명	설명	데이터형	NULL 허용여부	PK 여부
DEPT_NO	부서번호	NUMBER	N	Y
DEPT_NAME	부서명	VARCHAR2(50)	N	
DEPT_DESC	부서설명	VARCHAR2(100)	Y	
CREATE_DATE	생성일자	DATE	Y	

· DEPT_DESC1 VARCHAR2(80) 컬럼 추가 → ALTER TABLE ADD

· DEPT_DESC1 컬럼 삭제 → ALTER TABLE ... DROP COLUMN

4. 테이블 생성, 삭제, 수정 실습

· 부서 (DEPT_TEST) 테이블 생성

컬럼명	설명	데이터형	NULL 허용여부	PK 여부
DEPT_NO	부서번호	NUMBER	N	Y
DEPT_NAME	부서명	VARCHAR2(50)	N	
DEPT_DESC	부서설명	VARCHAR2(100)	Y	
CREATE_DATE	생성일자	DATE	Y	

· DEPT_TEST_PK 란 이름으로 PK 생성

· DEPT_TEST 테이블 삭제

학습정리

- SQL의 종류는 DDL, DML, TCL, DCL 이 있다.
- DDL은 데이터베이스 객체들을 생성, 수정, 삭제 하며, DML은 실제로 데이터를 조회,입력, 수정 삭제하는 역할을 한다.
- 데이터를 담아두는 테이블은 CREATE TABLE 문을 사용해 생성할 수 있으며 테이블 생성 시 테이블을 구성하는 컬럼명과 데이터 형, 각종 제약조건을 정의해 생성한다
- 생성된 테이블의 수정은 ALTER TABLE, 테이블 삭제는 DROP TABLE 문을 사용한다.

Quiz

1. SQL의 종류로는 DDL, DML, TCL, DCL이 있는데 이 중 실제로 데이터를 가공하고 처리하는 역할을 수행하는 문장은 DML 문이다.

Quiz

2. 데이터를 삭제하는 방법에는 2 가지가 있다. 하나는 TRUNCATE TABLE 문이고 다른 하나는 DELETE 문이다. 이 중 데이터를 삭제한 뒤 복구할 수 없는 문장은 TRUNCATE TABLE 문이다.

Quiz

3. CREATE TABLE 문을 사용해 테이블을 생성한 후 테이블에 있는 컬럼의 데이터 형을 수정하려면 ALTER COLUMN 문을 사용해야 한다.