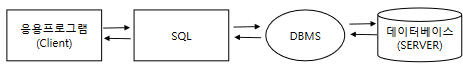
**<8주차 : SQL - DDL>**

**1. SQL 개요**

**# SQL 개념**

(1) SQL(Structured Query Language)은 IBM이 1970년대 초에 SEQUEL(Structured English Query Language)로 개발 -> SQL로 변경

(2) 1987년에 ISO(국제 표준화 기구)에서 SQL을 표준으로 제정



**## SQL 특징**

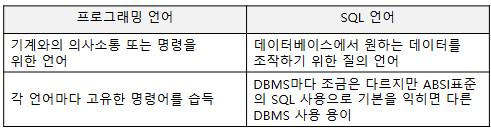
(1) 문법이 사용하기 쉽습니다.

(2) 실행 순서와 상관없이 처리 내용을 명령할 수 있습니다.

(3) 데이터의 검색, 조작, 정의가 용이합니다.

(4) 표준 언어로 구성되어 있습니다.

**### 프로그래밍 언어와 비교**

****

**#### SQL 명령어 종류**

(1) DDL(Data Definition Language)

: CREATE, ALTER, DROP, RENAME, TRUNCATE(테이블 정의와 구조 수정)

(2) DML(Data Manipilation Language)

: SELECT, INSERT, UPDATE, DELETE(검색, 삽입, 수정, 삭제)

(3) DCL(Data Control Language)

: GRANT, REVOKE(접근 권한 관리)

(4) Cf) TCL((Transaction Control Language)

: COMMIT, ROLLBACK, SAVEPOINT(DML에서 실행한 사항을 관리)

**##### SQL문 작성 규칙**

(1) SQL 문은 대·소문자 구분하지 않음(SELECT와 select는 동일)

(2) 한 줄 또는 여러 줄로 작성할 수 있음

(3) 문장 마지막은 세미콜론(;)으로 끝남

(4) 명령어, 객체명, 변수명은 대/소문자 구분이 없음 -> 명령어는 대문자로, 나머지는 소문자로 작성하는 게 좋음

- 데이터 값은 대/소문자를 구분함

(5) 날짜와 문자열에는 작은 따옴표를 사용

(6) 단어와 단어 사이는 공백 또는 줄바꿈으로 구분

(7) 주석문

• -- 이것은 주석입니다.

• /\* 여기부터 여기까지 주석입니다. \*/

**2. DDL**

**# 테이블 생성 규칙**

(1) 테이블명

▪ 객체를 의미할 수 있는 이름으로, 단수형을 권고함

▪ 다른 테이블의 이름과 중복되지 않아야 함

(2) 칼럼명

▪ 한 테이블 내에서는 칼럼명이 중복되지 않아야 함

▪ 테이블 생성시 각 칼럼들은 괄호 내에서 콤마로 구분됨

▪ 칼럼 뒤에 데이터 유형이 반드시 지정되어야 함

(3) 테이블명 & 칼럼명

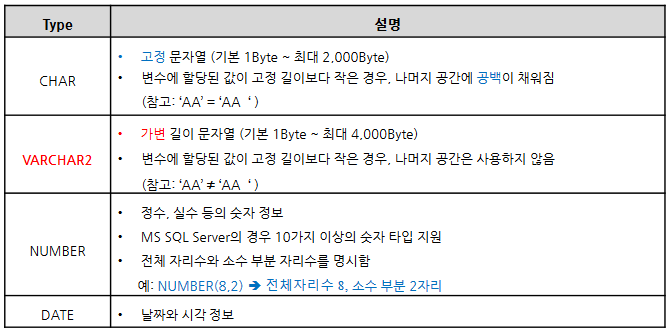
▪ 사전에 정의된 예약어(Reserved word)는 사용 불가

▪ 테이블명과 칼럼명에는 문자, 숫자, 일부 기호(\_, $, #)만 허용됨

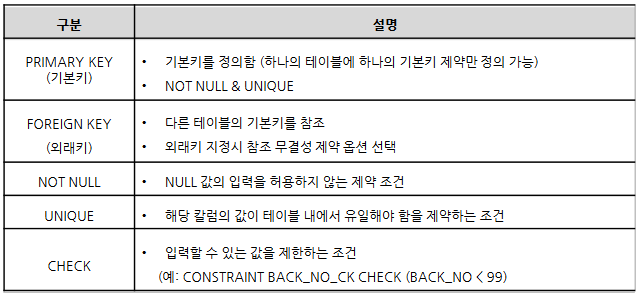
▪ 테이블명과 칼럼명은 반드시 문자로 시작해야 함 (숫자, 기호 불가)

(4) 제약조건명: 다른 제약조건의 이름과 중복되지 않아야 함

**## Oracle의 주요 데이터 타입**



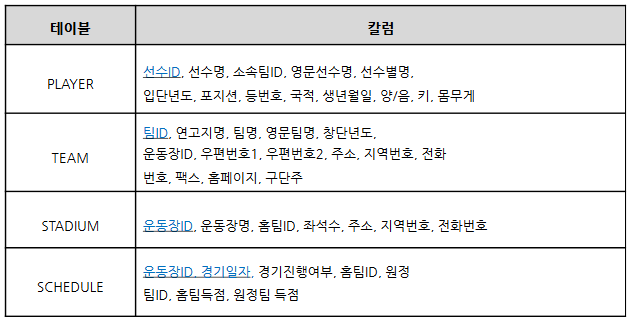
**## Oracle의 주요 데이터 타입**

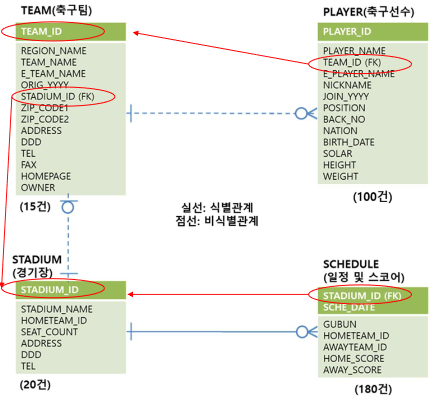


- 제약조건명을 명시적으로 부여할 수도 있고,

묵시적으로 제약조건명 없이 제약조건을 설정할 수도 있음.

**## 예제 데이터 (K-리그)**





**### 테이블 생성 SQL문**

(1) STADIUM

CREATE TABLE STADIUM (

STADIUM\_ID CHAR(3) NOT NULL,

STADIUM\_NAME VARCHAR2(40) NOT NULL,

HOMETEAM\_ID CHAR(3),

SEAT\_COUNT NUMBER,

ADDRESS VARCHAR2(60),

DDD VARCHAR2(3),

TEL VARCHAR2(10),

CONSTRAINT STADIUM\_PK PRIMARY KEY (STADIUM\_ID)

);

(2) TEAM

CREATE TABLE TEAM (

TEAM\_ID CHAR(3) NOT NULL,

REGION\_NAME VARCHAR2(8) NOT NULL,

TEAM\_NAME VARCHAR2(40) NOT NULL,

E\_TEAM\_NAME VARCHAR2(50),

ORIG\_YYYY CHAR(4),

STADIUM\_ID CHAR(3) NOT NULL,

ZIP\_CODE1 CHAR(3),

ZIP\_CODE2 CHAR(3),

ADDRESS VARCHAR2(80),

DDD VARCHAR2(3),

TEL VARCHAR2(10),

FAX VARCHAR2(10),

HOMEPAGE VARCHAR2(50),

OWNER VARCHAR2(10),

CONSTRAINT TEAM\_PK PRIMARY KEY (TEAM\_ID),

CONSTRAINT TEAM\_FK FOREIGN KEY (STADIUM\_ID) REFERENCES STADIUM(STADIUM\_ID)

);

마지막 제약조건(TEAM\_FK)은 STADIUM 테이블이 먼저 생성되어 있어야 함.

(3) SCHEDULE

CREATE TABLE SCHEDULE (

STADIUM\_ID CHAR(3) NOT NULL,

SCHE\_DATE CHAR(8) NOT NULL,

GUBUN CHAR(1) NOT NULL,

HOMETEAM\_ID CHAR(3) NOT NULL,

AWAYTEAM\_ID CHAR(3) NOT NULL,

HOME\_SCORE NUMBER(2),

AWAY\_SCORE NUMBER(2),

CONSTRAINT SCHEDULE\_PK PRIMARY KEY (STADIUM\_ID, SCHE\_DATE),

CONSTRAINT SCHEDULE\_FK FOREIGN KEY (STADIUM\_ID)

REFERENCES STADIUM(STADIUM\_ID)

조건이름 외래키 STADIUM 테이블의 STADIUM\_ID를 참조함.

);

마지막 제약조건(SCHEDULE\_FK)은 STADIUM 테이블이 먼저 생성되어 있어야 함.

(4) PLAYER

CREATE TABLE PLAYER (

PLAYER\_ID CHAR(7) NOT NULL,

PLAYER\_NAME VARCHAR2(20) NOT NULL,

TEAM\_ID CHAR(3) NOT NULL,

E\_PLAYER\_NAME VARCHAR2(40),

NICKNAME VARCHAR2(30),

JOIN\_YYYY CHAR(4),

POSITION VARCHAR2(10),

BACK\_NO NUMBER(2),

NATION VARCHAR2(20),

BIRTH\_DATE DATE,

SOLAR CHAR(1),

HEIGHT NUMBER(3),

WEIGHT NUMBER(3),

CONSTRAINT PLAYER\_PK PRIMARY KEY (PLAYER\_ID),

CONSTRAINT PLAYER\_FK FOREIGN KEY (TEAM\_ID) REFERENCES

TEAM(TEAM\_ID)

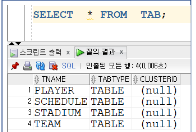
);

마지막 제약조건(PLAYER\_PK)은 TEAM 테이블이 먼저 생성되어 있어야 함.

**#### 테이블 목록 및 구조 확인**

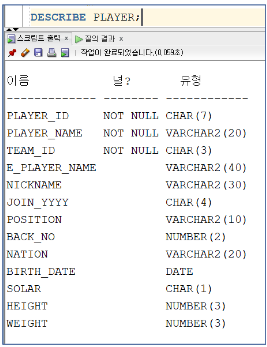
(1) 테이블 목록 확인

SELECT \* FROM TAB;



(2) 테이블 구조 확인

DESCRIBE PLAYER;



15페이지

**3. 평가**

**# 데이터 모델링 (data modeling)**

(1) 현실세계의 복잡한 정보와 관계를 컴퓨터 세계에서 처리할 수 있도록 변환하는 과정

(2) 추상화 (↔ 구체화)