



**ORACLE<sup>®</sup>**  
**DB Management**

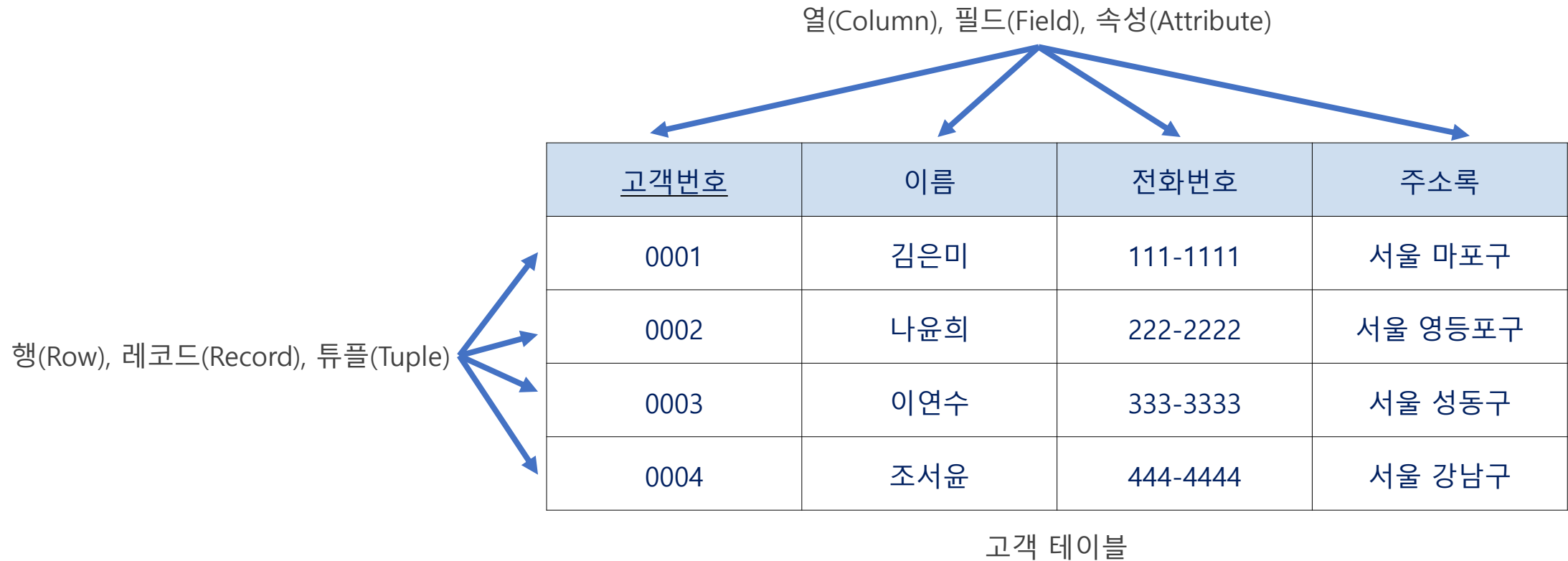
DDL 활용

# DDL

- DDL
  - ✓ Data Definition Language
  - ✓ 데이터 정의어
  - ✓ 데이터와 데이터 간의 관계를 정의하여 데이터베이스 구조를 설정하는 SQL문
  - ✓ 테이블(Table), 뷰(View) 등 데이터베이스 객체를 생성/수정/삭제하는 기능을 담당
  - ✓ 실행 후에는 작업을 취소할 수 없음
- DDL 종류
  - ① CREATE : 데이터베이스 객체 생성
  - ② ALTER : 데이터베이스 객체 수정
  - ③ DROP : 데이터베이스 객체 삭제
  - ④ TRUNCATE : 데이터베이스 객체 데이터 및 저장 공간 삭제

# 테이블(Table)

- 테이블(Table)
  - ✓ 관계형 데이터베이스에서는 정보를 테이블(릴레이션) 형태로 보관
  - ✓ 행(ROW)과 열(COLUMN)의 조합으로 구성



# 테이블 생성

- 테이블 생성
  - ✓ 테이블에 대한 구조를 정의하고, 데이터를 저장하기 위한 공간을 할당하는 과정
- 컬럼의 데이터 타입
  - ✓ NUMBER, VARCHAR2, DATE 등
- 무결성 제약조건 5가지
  - ① NOT NULL
  - ② UNIQUE
  - ③ CHECK
  - ④ PRIMARY KEY
  - ⑤ FOREIGN KEY
- 테이블 이름 규칙
  - ✓ 영문자, 숫자, 특수문자(underscore, \$, #)을 사용할 수 있다.
  - ✓ 영문자로 시작하고 30자 이내로 한다.
  - ✓ 대소문자를 구별하지 않는다.
  - ✓ 서로 다른 테이블에서 동일한 데이터를 저장하는 컬럼의 이름은 가능하면 같은 이름을 사용한다.
  - ✓ 완성된 설계도(테이블 정의서)에 의해 테이블을 생성하길 권장한다.

# 무결성 제약조건

1. NOT NULL ➤ 필수, NULL값을 허용하지 않음
2. UNIQUE ➤ 중복 값을 허용하지 않음
3. PRIMARY KEY ➤ 각 레코드를 구별하는 칼럼(NOT NULL + UNIQUE)
4. FOREIGN KEY ➤ 다른 테이블의 값을 참조할 때 사용하는 키
5. CHECK ➤ 값의 유효성을 검사

dept_no	dept_name	location	emp_no	name	depart
1	영업부	대구	1001	구창민	1
2	인사부	서울	1002	김민서	1
3	총무부	대구	1003	이은영	2
4	기획부	서울	1004	한성일	2

PRIMARY KEY

FOREIGN KEY

# 테이블 생성 방법(제약조건이름 X)

```
CREATE TABLE 테이블_이름(  
    칼럼_이름 데이터_타입 [ 제약조건 ],  
    칼럼_이름 데이터_타입 [ 제약조건 ],  
    칼럼_이름 데이터_타입 NOT NULL PRIMARY KEY,  
    칼럼_이름 데이터_타입 UNIQUE,  
    칼럼_이름 데이터_타입 CHECK(조건식),  
    칼럼_이름 데이터_타입 REFERENCES 부모테이블(칼럼_이름)  
);
```

```
CREATE TABLE 테이블_이름(  
    칼럼_이름 데이터_타입 [ 제약조건 ],  
    칼럼_이름 데이터_타입 [ 제약조건 ],  
    [ PRIMARY KEY(칼럼_이름) ],  
    [ UNIQUE(칼럼_이름) ],  
    [ CHECK(조건식) ],  
    [ FOREIGN KEY(칼럼_이름) REFERENCES 부모테이블(칼럼_이름) ]  
);
```

# 테이블 생성 방법(제약조건이름 O)

```
CREATE TABLE 테이블_이름(  
    칼럼_이름 데이터_타입 [ 제약조건 ],  
    칼럼_이름 데이터_타입 [ 제약조건 ],  
    칼럼_이름 데이터_타입 NOT NULL CONSTRAINT 제약조건_이름 PRIMARY KEY,  
    칼럼_이름 데이터_타입 CONSTRAINT 제약조건_이름 UNIQUE,  
    칼럼_이름 데이터_타입 CONSTRAINT 제약조건_이름 CHECK(조건식),  
    칼럼_이름 데이터_타입 CONSTRAINT 제약조건_이름 REFERENCES 부모테이블(칼럼_이름)  
);
```

```
CREATE TABLE 테이블_이름(  
    칼럼_이름 데이터_타입 [ 제약조건 ],  
    칼럼_이름 데이터_타입 [ 제약조건 ],  
    [ CONSTRAINT 제약조건_이름 PRIMARY KEY(칼럼_이름) ],  
    [ CONSTRAINT 제약조건_이름 UNIQUE(칼럼_이름) ],  
    [ CONSTRAINT 제약조건_이름 CHECK(조건식) ],  
    [ CONSTRAINT 제약조건_이름 FOREIGN KEY(칼럼_이름) REFERENCES 부모테이블(칼럼_이름) ]  
);
```

# 연습. BANK와 CUSTOMER 테이블

## BANK

칼럼명	데이터 타입	제약조건
BANK_CODE	VARCHAR2(20 BYTE)	기본키
BANK_NAME	VARCHAR2(30 BYTE)	

## CUSTOMER

칼럼명	데이터 타입	제약조건
NO	NUMBER	기본키
NAME	VARCHAR2(30 BYTE)	필수
PHONE	VARCHAR2(30 BYTE)	중복불가
AGE	NUMBER	0 ~ 100 사이만 가능
BANK_CODE	VARCHAR2(20 BYTE)	BANK 테이블의 bank_code 칼럼을 참조하는 외래키



# 테이블 구조 확인

- DESC[RIBE]
  - ✓ 테이블 생성 여부와 테이블의 구조를 확인하기 위한 명령

- ✓ 확인할 수 있는 사항

- ✓ 칼럼 이름
- ✓ NOT NULL 제약조건
- ✓ Data Type

- ADDRESS 테이블 구조 확인

```
DESC ADDRESS
```

```
SQL> DESC address
```

Name	Null?	Type
ID		NUMBER(3)
NAME		VARCHAR2(50)
ADDR		VARCHAR2(100)
PHONE		VARCHAR2(15)
EMAIL		VARCHAR2(100)

# Data Type

데이터 타입	의미
CHAR(size)	size 크기의 고정 길이 문자 타입 최대크기 : 2,000 바이트 최소크기 : 1 바이트
VARCHAR2(size)	size 크기의 가변 길이 문자 타입 최대크기 : 4,000 바이트 최소크기 : 1 바이트
NVARCHAR2(size)	국가별 문자 집합에 따른 size 크기의 문자 또는 바이트의 가변길이 문자 타입 최대크기 : 4,000 바이트 최소크기 : 1 바이트
NUMBER(p, s)	정밀도(p)와 스케일(s)로 표현되는 숫자 타입
DATE	날짜 타입

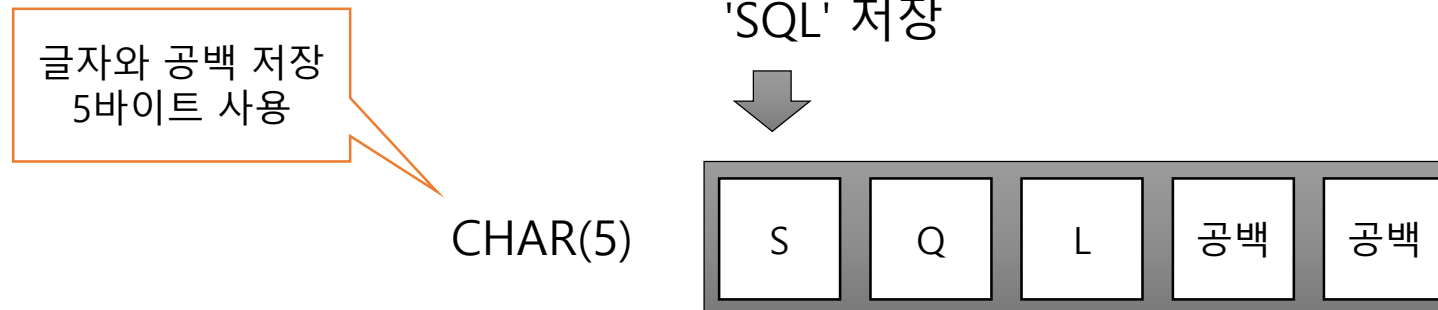
# Data Type

데이터 타입	의미
BLOB	대용량의 바이너리 데이터를 저장하는 타입 (그래픽, 동영상, 사운드 등) 최대크기 : 4GB
CLOB	대용량의 텍스트 데이터를 저장하는 타입 (e-Book.html 등) 최대크기 : 4GB
BFILE	대용량의 바이너리 데이터를 파일 형태로 저장하기 위한 데이터 타입 (읽기 전용) 최대크기 : 4GB
TIMESTAMP(n)	DATE 타입의 확장 타입 n은 millisecond 자리수로 최대 9자리까지 표현

# CHAR

- CHAR Type

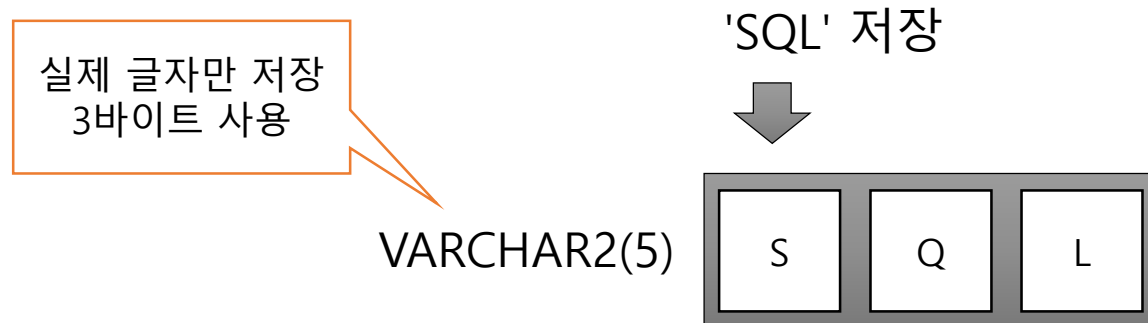
- ① 고정 길이의 문자열 저장 **최대 2,000바이트**
- ② 지정된 길이보다 짧은 데이터가 입력되면 빈 공간은 공백으로 채움
- ③ 데이터 입력 시 사용자가 데이터를 입력하지 않으면 NULL 저장
- ④ 지정된 길이보다 긴 데이터를 입력하면 오류 발생
- ⑤ 길이의 편차가 심한 데이터(예 : 주소)는 저장 공간의 낭비로 이어짐
- ⑥ 주민등록번호, 학번 등 길이가 일정하거나 비슷한 경우에 사용하는 것이 유리



# VARCHAR2

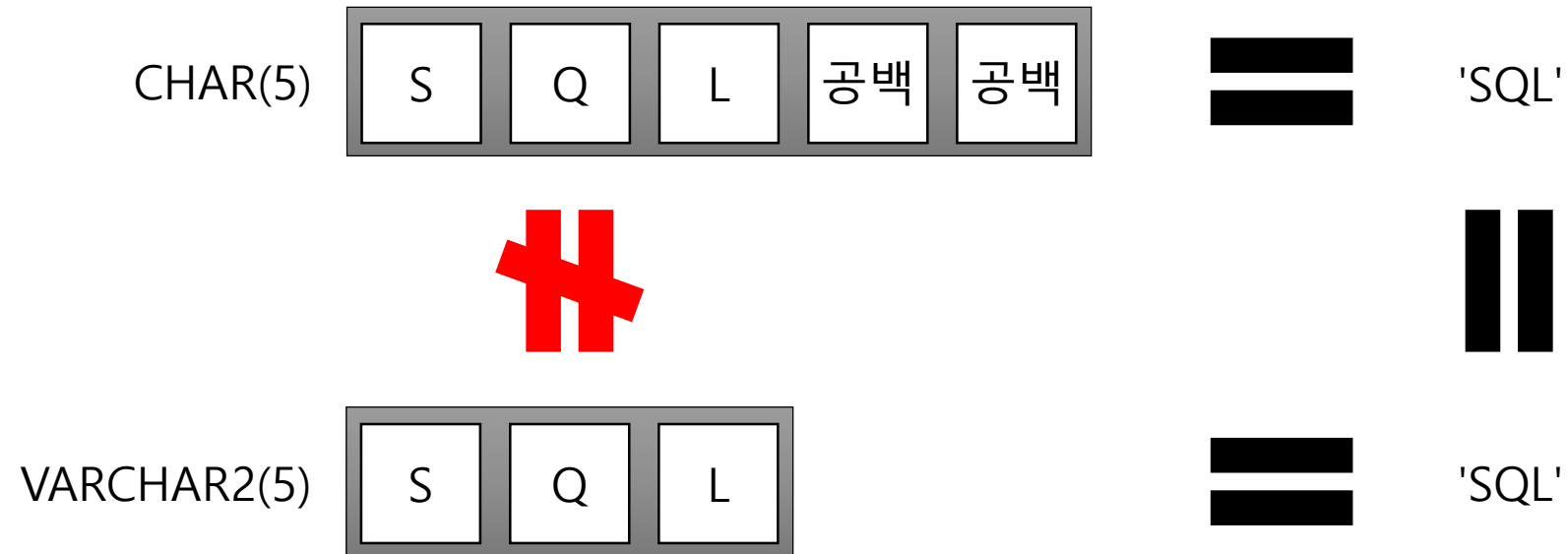
- VARCHAR2 Type

- ① 가변 길이의 문자열 저장 **최대 4,000바이트**
- ② 지정된 길이보다 짧은 데이터가 입력되면 빈 공간은 NULL로 처리(저장공간의 낭비 없음)
- ③ 데이터 입력 시 사용자가 데이터를 입력하지 않으면 NULL 저장
- ④ 지정된 길이보다 긴 데이터를 입력하면 오류 발생
- ⑤ 편차가 심한 데이터나 NULL이 많이 입력되는 경우에 사용하는 것이 유리
- ⑥ 실무에서는 CHAR 타입보다 **VARCHAR2 타입**이 많이 사용



# CHAR vs VARCHAR2

- Type에 따라(CHAR 또는 VARCHAR2) 동일한 텍스트도 다른 텍스트로 인식될 수 있음
- 동일한 의미의 칼럼들은 같은 타입과 크기를 가지는 것이 유리



# NUMBER

- NUMBER Type
  - ✓ 숫자(정수, 실수) 저장
  - ✓ 22BYTE 가변 길이 데이터 타입
  - ✓ 최대 38자리까지 지원
- NUMBER 표현 범위

$1.0 \times 10^{-130} \sim 96.9... \times 10^{125}$
- 소수점 처리 방식
  - ① NUMBER(precision, scale) 형식 이용
  - ② 지정된 자리 이하에서 반올림되어 저장된다.
  - ③ precision : 정수부 + 소수부 모두를 포함하는 전체 유효 숫자(0은 유효 숫자에서 제외)
  - ④ scale : 소수부(소수점 이하) 자리 수
  - ⑤ precision을 지정하지 않고 숫자를 입력하면 입력되는 숫자 값의 크기만큼 저장공간이 할당
  - ⑥ scale을 지정하지 않으면 정수로 인식(소수점을 입력하면 정수로 반올림되어 저장)

# DATE

- DATE Type
  - ✓ 날짜/시간(세기, 년도, 월, 일, 시, 분, 초) 저장
  - ✓ 7BIT 고정 길이 데이터 타입
  - ✓ 시간이 입력되지 않으면 12:00:00로 인식
  - ✓ 날짜/시간 연산 가능
  - ✓ 날짜 데이터를 문자 데이터로 처리하는 경우도 많음(DATE Type의 성능이 문자 Type에 비해 떨어짐)
- DATE 관련 함수
  - ① TO\_DATE
    - 문자 형식으로 저장된 날짜 데이터를 날짜 형식으로 변환
  - ② SYSDATE
    - 시스템의 현재 날짜와 시간을 반환



# TIMESTAMP

- TIMESTAMP Type
  - ✓ DATE Type의 확장 타입
  - ✓  $\mu s$ (마이크로초, 백만분의 1초) 표현 가능
  - ✓ ms(밀리초, 천분의 1초)의 기본 값은 6자리, 최대 9자리까지 사용 가능
- TIMESTAMP 종류
  - ① TIMESTAMP WITH TIME ZONE
    - TIMESTAMP Type에 지역 시간대(TIME ZONE)를 함께 저장
    - 지역 시간대(TIME ZONE)는 세계 표준 시간대(UTC)를 기준으로 현 지역 시간대를 환산
  - ② TIMESTAMP WITH LOCAL TIME ZONE
    - 사용자 데이터베이스의 지역 시간대(TIME ZONE)를 따름
    - 지역 시간대(TIME ZONE)를 다른 곳으로 옮겨도 따로 변경할 것이 없음

# 테이블 정의 변경

## 1. 칼럼 추가하기

```
ALTER TABLE 테이블_이름 ADD 칼럼_이름 데이터_타입 [제약조건];
```

## 2. 칼럼 수정하기

```
ALTER TABLE 테이블_이름 MODIFY 칼럼_이름 데이터_타입 [제약조건];
```

## 3. 칼럼 삭제하기

```
ALTER TABLE 테이블_이름 DROP COLUMN 칼럼_이름;
```

## 4. 칼럼 이름 바꾸기

```
ALTER TABLE 테이블_이름 RENAME COLUMN 칼럼_이름 TO 새_칼럼_이름;
```

## 5. 테이블 이름 바꾸기

```
ALTER TABLE 테이블_이름 RENAME TO 새_테이블_이름;
```

# 테이블 삭제

## 1. 테이블 삭제하기

```
DROP TABLE 테이블_이름;
```

## 2. 제약조건 테이블 삭제하기

```
DROP TABLE 테이블_이름 CASCADE CONSTRAINTS;
```

## 주의 !!

DROP TABLE 명령어는 테이블을 완전히 삭제한다.

테이블에 데이터가 있을 경우, 모든 데이터가 지워지고

이를 복구할 수 없으므로 주의해서 사용해야 한다.

# 테이블 데이터 삭제

## 1. 테이블 데이터 삭제하기

```
TRUNCATE TABLE 테이블_이름;
```

### 주의 !!

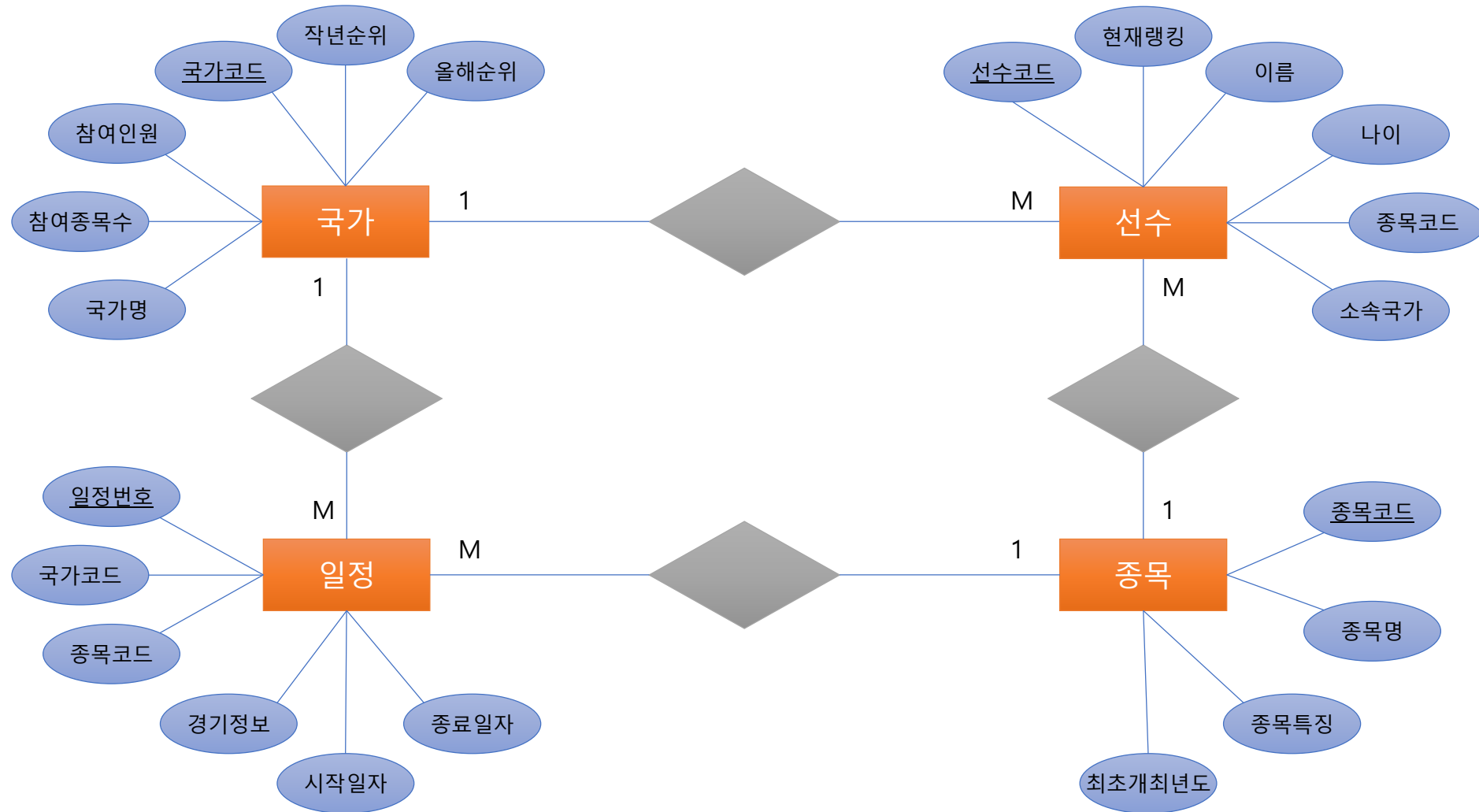
TRUNCATE TABLE 명령어는 테이블의 구조는 남기고,

모든 행(ROW) 만 삭제하는 명령이다.

DELETE 문과 비교하면 각 행(ROW)의 삭제 로그가 남지 않기 때문에

삭제 자체는 빠르지만 삭제된 데이터를 복구할 수 없다.

# Olympics 데이터베이스



# Olympics 데이터베이스

국가(NATION) 테이블

칼럼 한글명	칼럼 영문명	데이터 타입	길이	NN	UQ	PK	FK	DEFAULT
국가코드	N_CODE	NUMBER	3	Y		Y		
국가명	N_NAME	VARCHAR2	30	Y				
참여인원	N_PARTI_PERSON	NUMBER						0
참여종목개수	N_PARTI_EVENT	NUMBER						
작년순위	N_PREV_RANK	NUMBER						
현재순위	N_CURR_RANK	NUMBER						

종목(EVENT) 테이블

칼럼 한글명	칼럼 영문명	데이터 타입	길이	NN	UQ	PK	FK	DEFAULT
종목코드	E_CODE	NUMBER		Y		Y		
종목명	E_NAME	VARCHAR2	30	Y				
최초개최년도	E_FIRST_YEAR	NUMBER	4					
종목특징	E_INFO	VARCHAR2	100					

# Olympics 데이터베이스

선수(PLAYER) 테이블

칼럼 한글명	칼럼 영문명	데이터 타입	길이	NN	UQ	PK	FK	DEFAULT
선수코드	P_CODE	NUMBER	3	Y		Y		
선수명	P_NAME	VARCHAR2	30	Y				
소속국가코드	N_CODE	NUMBER	3	Y			Y	
종목코드	E_CODE	NUMBER		Y			Y	
현재순위	P_RANK	NUMBER						
나이	P_AGE	NUMBER	3					

일정(SCHEDULE) 테이블

칼럼 한글명	칼럼 영문명	데이터 타입	길이	NN	UQ	PK	FK	DEFAULT
일정번호	S_NO	NUMBER	3	Y		Y		
국가코드	N_CODE	NUMBER	3				Y	
종목코드	E_CODE	NUMBER					Y	
시작일자	S_START_DATE	DATE						
종료일자	S_END_DATE	DATE						
경기정보	S_INFO	VARCHAR2	100					