

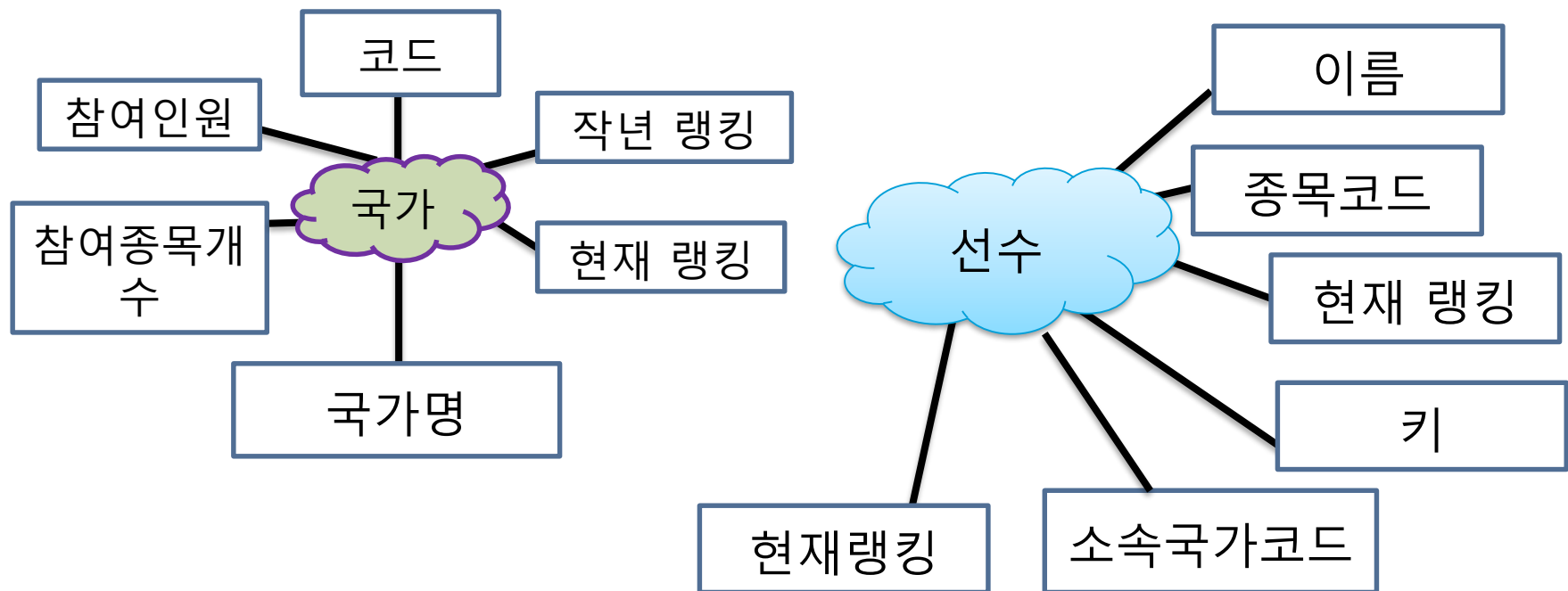
기본 SQL 작성하기_DDL 활용



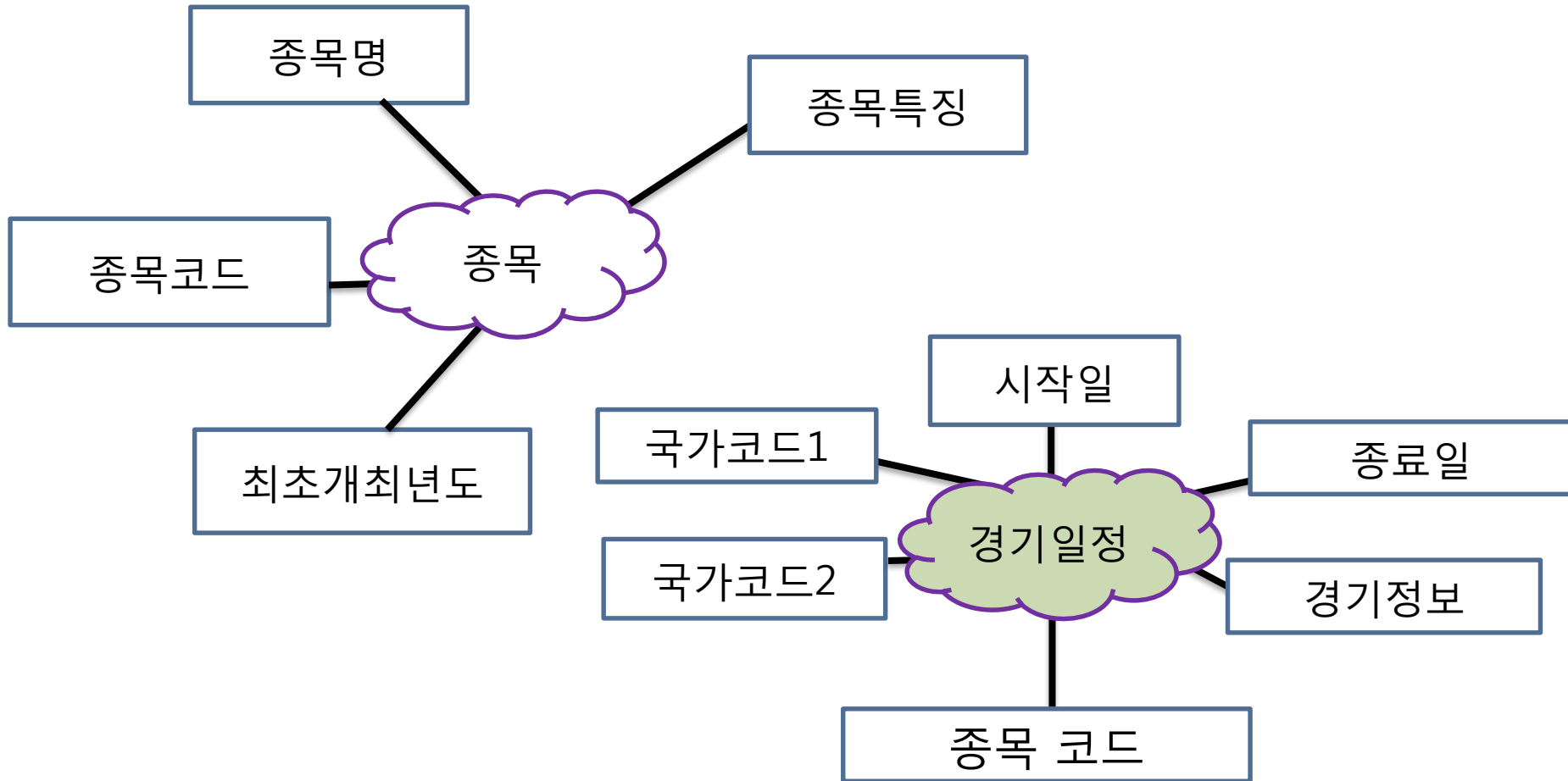
데이터베이스 설계

다음 데이터베이스에 저장해야 할 데이터를 생각해보자

동계올림픽이 우리나라에서 열린다. 조직위원회에서는 올림픽 기간 동안 사용할 프로그램을 개발하려 한다. 이 때 프로그램에서 사용될 데이터베이스를 설계하고 생성하려 한다. 필요한 데이터를 마인드맵으로 연결하여 그려보자. (데이터베이스명 : Olympics)



데이터베이스 설계



더 추가할 정보가 있을까?

데이터베이스 생성

```
CREATE DATABASE 데이터베이스명;
```

```
CREATE DATABASE testDB;
```

SHOW DATABASES; 데이터베이스 목록 확인

```
DROP DATABASE 데이터베이스명;
```

```
DROP DATABASE testDB;
```

주의!! 데이터베이스에 저장된 모든 정보가 삭제

테이블

- 관계형 데이터베이스에서는 정보를 테이블 형태로 보관

The diagram illustrates a database table structure with the following annotations:

- 고객 릴레이션** (Customer Relation): A red arrow points to the entire table structure.
- 속성(컬럼명)** (Attribute/Column Name): Three arrows point to the column headers: 고객번호, 이름, and 전화번호.
- 컬럼** (Column): A purple box highlights the entire table structure, with an arrow pointing to the header row.
- 행(튜플)** (Row/Tuple): Three arrows point to the data rows (0001, 0002, 0003).

<u>고객번호</u>	이름	전화번호	주소록
0001	김은미	001-145-1111	서울 마포구
0002	나윤희	001-111-1111	부산 동래구
0003	이연수	001-112-1111	경북 대구
0004	조서윤	001-124-1111	대전 동구

데이터정의어(DDL:DATA DEFINITION LANGUAGE)

테이블 정의 (설계된 데이터베이스를 실제 테이블로 정의)

```
CREATE TABLE 테이블_이름(  
    {속성_이름 데이터_타입 [NOT NULL] [AUTO_INCREMENT]}  
    [PRIMARY KEY(컬럼명),]  
    [UNIQUE(컬럼명),]  
    [DEFAULT 값]  
    [FOREIGN KEY(컬럼명) REFERENCES  
        참조테이블(컬럼명)]  
);
```

값을 꼭 입력해야 한다.

기본키

자동증가

속성의 모든 값이 유일한 값

값이 입력되지 않을 때 저장되는 값

외래키

제약조건

제약조건

- **NOT NULL** - NULL값을 허용하지 않음
- **AUTO_INCREMENT** - 자동으로 값 증가

(**AUTO_INCREMENT=1001**) 1001 부터 1씩 증가

- **UNIQUE** - 컬럼에 같은 값을 허용하지 않음
- **PRIMARY KEY** - 각 튜플을 구별하는 컬럼(**NOT NULL, UNIQUE**)
- **FORGIEN KEY** - 다른 테이블의 값을 참조할 때 사용하는 키
- **DEFAULT**- 값이 없을 경우 입력되는 값

department

dept_no	dept_name	location
1	영업부	대구
2	인사부	서울
3	총무부	대구
4	기획부	서울

employee

emp_no	name	department
1001	구창민	1
1002	김민서	1
1003	이은영	2
1004	한성일	2

FOREIGN KEY

테이블 정의

player(선수) 테이블

```
create table player(  
  p_code int AUTO_INCREMENT,  
  p_name varchar(30) NOT NULL,  
  n_code int,  
  e_code int,  
  p_rank int,  
  age int,  
  height int,  
  PRIMARY KEY(p_code),  
  FOREIGN KEY(n_code) REFERENCES nation(code));
```

Diagram illustrating the table definition for 'player' (선수) and its constraints. The table structure is defined by the SQL statement above. The constraints are highlighted in purple and connected by blue lines to a box on the right labeled '제약 조건' (Constraint).

→ 제약 조건

예) 고객 테이블 만들기

<u>고객번호</u>	이름	전화번호	주소록
0001	김은미	001-145-1111	서울 마포구
0002	나윤희	001-111-1111	부산 동래구
0003	이연수	001-112-1111	경북 대구
0004	조서윤	001-124-1111	대전 동구

- 각 컬럼의 데이터형을 생각해 보자
- 각 컬럼의 제약조건을 생각해 보자
 - 기본키가 되는 컬럼을 생각해 보자
 - null값 허용 여부
 - default 값 여부
 - AUTO_INCREMENT(1씩 자동 증가)

```
CREATE TABLE customer(  
    고객번호 int NOT NULL AUTO_INCREMENT,  
    이름 varchar(20) NOT NULL,  
    전화번호 varchar(30),  
    주소록 varchar(50),  
    PRIMARY KEY(고객번호),  
    UNIQUE(전화번호),  
);
```

테이블 만들기

데이터베이스 접속
(USE Olympics)

데이터베이스 내에 테이블 정의
CREATE TABLE 명령어 이용
4개 테이블 정의(nation, player, game, event)
- 테이블 구조 (컬럼의 데이터형)
- 필요 제약 조건

nation 테이블 정의

국가코드	국가명	참여인원	참여종목개수	현재 랭킹	지난대회 랭킹
int	varchar(30)	int	int	int	int
primary key	not null				

player 테이블 정의

선수코드	이름	소속국가코드	종목코드	현재랭킹	나이	키
int	varchar(30)	int	int	Int	Date	Float
primary key						

game 테이블 정의

시작일	종료일	경기정보	국가코드1	국가코드2	종목코드
date	date	text	Int	int	int
			primary key		

event 테이블 정의

종목코드	종목명	시작일	종료일
int	varchar(20)	date	date
primary key			

테이블 정의 변경하기

테이블에 구조 확인

```
DESC 테이블_이름;
```

테이블에 컬럼 추가

```
ALTER TABLE 테이블_이름 ADD 컬럼명 데이터형;
```

테이블에 컬럼 삭제

```
ALTER TABLE 테이블_이름 DROP [COLUMN] 컬럼명;
```

테이블에 컬럼 데이터형 변경

```
ALTER TABLE 테이블_이름 MODIFY COLUMN 컬럼명 데이터형;
```

DROP 테이블

정의된 테이블 삭제

```
DROP TABLE 테이블_이름;
```

drop table school;

school 테이블을 삭제한다.

주의 !! drop table 명령어는 테이블안에 데이터가 있을 경우, 모든 데이터를 지우고 실행되므로 주의해서 사용해야 한다.