프론트 엔드 개발자가 알아야 하는 컴퓨터 공학 지식
데이터베이스

데이터베이스 | 프론드 엔드 개발자가 알아야 하는 CS 지식

강사 나동빈

프론트 엔드 개발자가 알아야 하는 컴퓨터 공학 지식

데이터베이스

테이블(Table)

- **하나의 데이터베이스**는 여러 개의 테이블을 가진다.
- 하나의 테이블은 엑셀 시트(Sheet)와 유사한 형태를 보인다.
- 테이블 내 각 데이터는 행(row)과 열(column)이 만나는 지점에 데이터가 들어간다.
- 행(row) = 레코드(record) = 튜플(tuple)
- 열(column) = 필드(field) = 속성(attribute)

학생 번호	학생 이름	학생 성적
1	홍길동	75
2	나동빈	83
3	이순신	95
4	임꺽정	88

테이블 및 컬럼 컨벤션(conventions)

- 테이블(table)의 이름은 단수형을 사용한다.
- 이름을 붙일 때는 snake case를 사용한다: student, retired_employee
- 테이블의 key는 {테이블 이름의 단수형}_id 형태를 쓴다: student_id (O) 혹은 단순히 id (O)

[student 테이블]

- 속성 1: student_id
- 속성 2: name
- 속성 3: age
- 속성 4: grade

테이블(Table) 생성

- 테이블을 정의할 때는 CREATE 명령어를 사용한다.

```
CREATE TABLE `테이블 이름`
(
  {컬럼명 1} {자료형 1},
  {컬럼명 2} {자료형 2},
  {컬럼명 3} {자료형 3},
  ...
);
```

테이블(Table) 생성

컴퓨터
공학 지식
데이터베이스

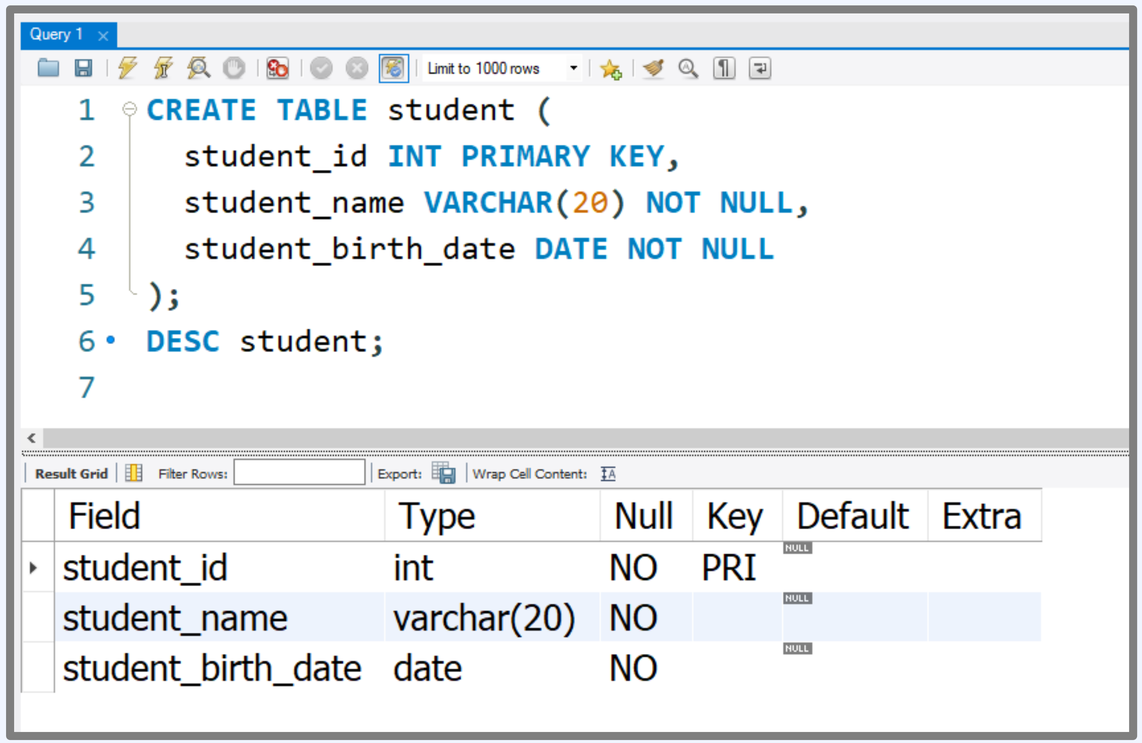
- 한 명의 학생에 대한 테이블(table)을 다음과 같이 생성할 수 있다.

```
CREATE TABLE student (
  student_id INT PRIMARY KEY,
  student_name VARCHAR(20) NOT NULL,
  student_birth_date DATE NOT NULL
);
```

테이블(Table) 생성

컴퓨터
공학 지식
데이터베이스

- 한 명의 학생에 대한 테이블(table)을 다음과 같이 생성할 수 있다.



테이블 제약 조건(Constraint) - 기본

- **NOT NULL**: NULL 값 비허용, 중복 허용
- **UNIQUE**: NULL 값 허용, 중복 비허용
- **PRIMARY KEY**: NULL 비허용, 중복 비허용, 테이블당 하나씩
- **DEFAULT**: 해당 컬럼의 기본 값을 설정

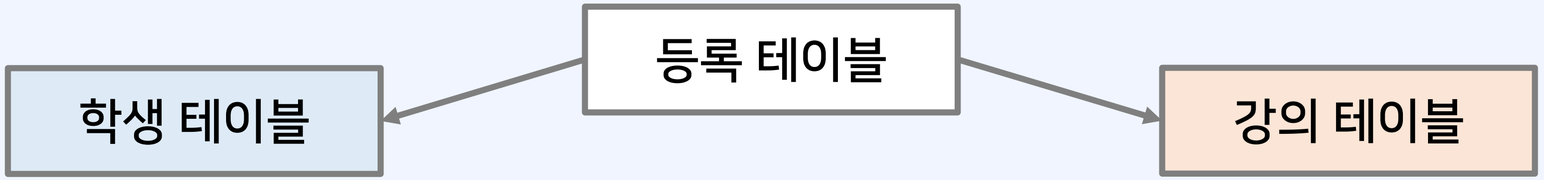
```
CREATE TABLE student
(
  student_id INT PRIMARY KEY,
  student_phone VARCHAR(20) UNIQUE,
  student_name VARCHAR(20) NOT NULL,
  student_address VARCHAR(20) DEFAULT 'seoul'
);
```

테이블 제약 조건(Constraint) - 외래키

- 특정한 학생(student)이 다른 강의(lecture)를 등록하는 상황을 고려하자.
- 이때 등록(registration) 테이블은 "학생 번호"와 "강의 번호"를 참조해야 한다.
- 이러한 기능을 위해 **외래키(foreign key)**가 제공된다.

테이블 제약 조건(Constraint) – 외래키

- 등록(registration) 테이블은 "학생 번호"와 "강의 번호"를 참조해야 한다.
- 만약 **[학생 테이블]**에 없는 학생을 추가하려는 경우 오류가 발생한다.



존재하지 않는 학생인 경우

- 학생 번호: 338283
- 강의 번호: 5
- 등록 날짜: 2012-05-05

오류 발생!

테이블 제약 조건(Constraint) – 외래키

- **외래키(foreign key)**는 하나의 테이블이 다른 테이블을 참조할 때 사용한다.
- 학생(student) ← 등록(registration) → 강의(lecture)

```
CREATE TABLE registration
(
  student_id INT,
  lecture_id INT,
  date DATETIME,
  FOREIGN KEY (student_id) REFERENCES student(student_id),
  FOREIGN KEY (lecture_id) REFERENCES lecture(lecture_id)
);
```