



Seoul
Software
ACademy

데이터베이스 응용

- table relation -

SeSAC 도봉 1기
웹 풀스택 과정

JOIN

조인이란?

데이터베이스의 조인은 **두 개 이상의 테이블**에서 데이터를 결합하여 새로운 결과를 생성

- 테이블 간의 관계를 통해 의미 있는 정보를 얻어내야 할 때 조인을 사용하여 관련된 데이터를 결합하고 원하는 정보를 추출

조인 종류

- **INNER JOIN**: 공통된 값만을 가진 행 반환.
- **LEFT JOIN**: 왼쪽 테이블 모든 행 포함하며, 오른쪽 테이블과 일치하는 경우 함께 반환.
- **RIGHT JOIN**: 오른쪽 테이블 모든 행 포함하며, 왼쪽 테이블과 일치하는 경우 함께 반환.
- **FULL JOIN**: 양쪽 테이블 모든 행을 포함하며, 일치하는 경우 함께 반환.

조인

```
SELECT 조회할 필드 FROM 테이블명 INNER JOIN 조인할테이블명 ON 일치해야할 필드
```

예시

```
SELECT orders.order_id, customers.customer_name FROM orders
INNER JOIN customers ON orders.customer_id = customers.customer_id;
```

Relationship

Relationship?

- 테이블들은 서로 관계를 맺을 수 있으며, 이를 통해 복잡한 데이터 구조를 단순화하고 관리
 - 1:1
 - 1:다
 - 다:다
- 복습) 외래키: 다른 릴레이션(테이블)의 키를 참조하는 것을 외래키

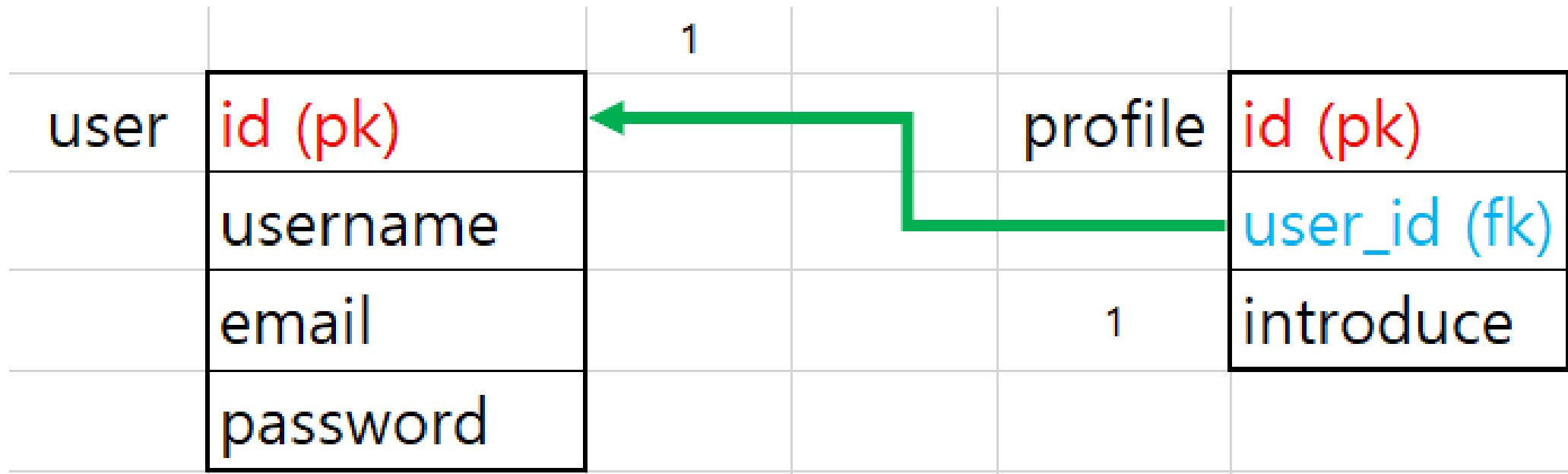
1:1 관계

- 1:1 관계는 하나의 레코드가 다른 하나의 레코드와 관련되어 있는 관계를 의미
- 이 관계에서 각 레코드는 서로 한 가지 정보만을 공유
- 예시) 사용자와 프로필, 직원과 연차, 주문과 송장

생성방법

```
CREATE TABLE user (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    username VARCHAR(50) NOT NULL,  
    email VARCHAR(100) NOT NULL,  
    password VARCHAR(100) NOT NULL  
);  
  
CREATE TABLE profile (  
    id INT PRIMARY KEY,  
    user_id INT,  
    introduce VARCHAR(100),  
    FOREIGN KEY (user_id) REFERENCES user(id)  
);
```

User 와 Profile 의 관계 (1:1)



생성방법

```
User.hasOne(Profile, { foreignKey: 'user_id', onDelete: 'CASCADE' });  
Profile.belongsTo(User, { foreignKey: 'user_id', onDelete: 'CASCADE' });
```

- hasOne: 한 모델이 다른 모델을 가리키는 1:1 관계를 설정하는 데 사용
- belongsTo: 다른 모델이 한 모델을 가리키는 1:1(1:다) 관계를 설정하는 데 사용

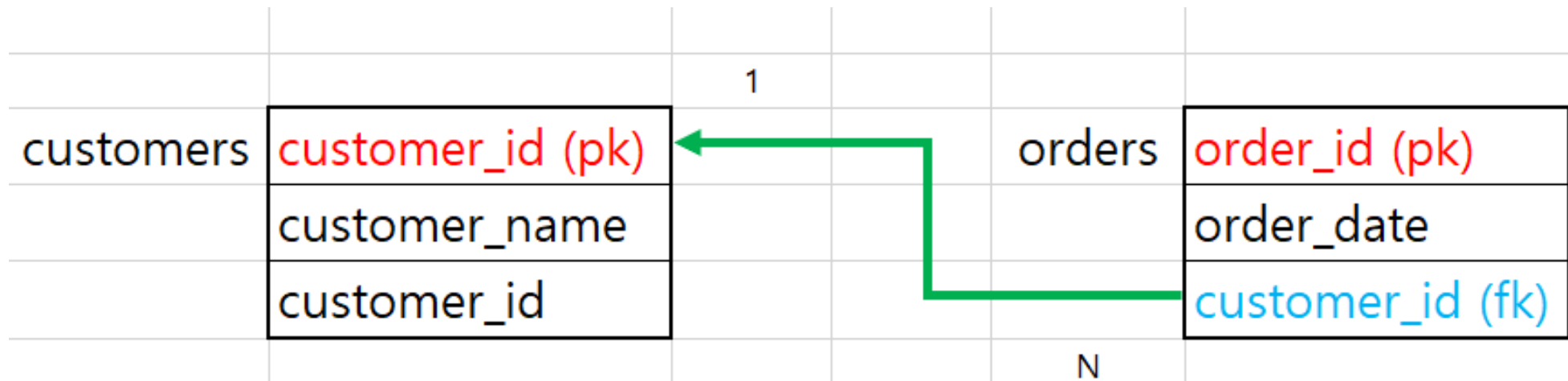
1:다 관계

- 1:다 관계는 한 쪽 레코드가 다른 쪽 레코드 여러 개와 관련되어 있는 관계
- 보통 부모 엔티티와 자식 엔티티 사이의 관계를 나타내는데 사용
- 예시) 부서와 직원, 고객과 주문, 학교와 학생

생성방법

```
CREATE TABLE customers (  
    customer_id INT PRIMARY KEY AUTO_INCREMENT,  
    customer_name VARCHAR(255),  
    customer_email VARCHAR(255)  
);  
  
CREATE TABLE orders (  
    order_id INT PRIMARY KEY,  
    order_date DATE,  
    customer_id INT ,  
    FOREIGN KEY (customer_id) REFERENCES customers(customer_id)  
);
```

Customers 와 Orders의 관계 (1:N)



고객(customers)은 여러 개의 주문을 할 수 있고,
하나의 주문/orders)은 고객 한 명에게 속해 있다.

생성방법

```
Customer.hasMany(Order, { foreignKey: 'customer_id' });  
Order.belongsTo(Customer, { foreignKey: 'customer_id' });
```

- hasMany: 한 개의 모델(테이블)이 다른 모델과 1:n 관계를 가질 때 사용
- belongsTo: 다른 모델이 한 모델을 가리키는 1:1(1:다) 관계를 설정하는 데 사용

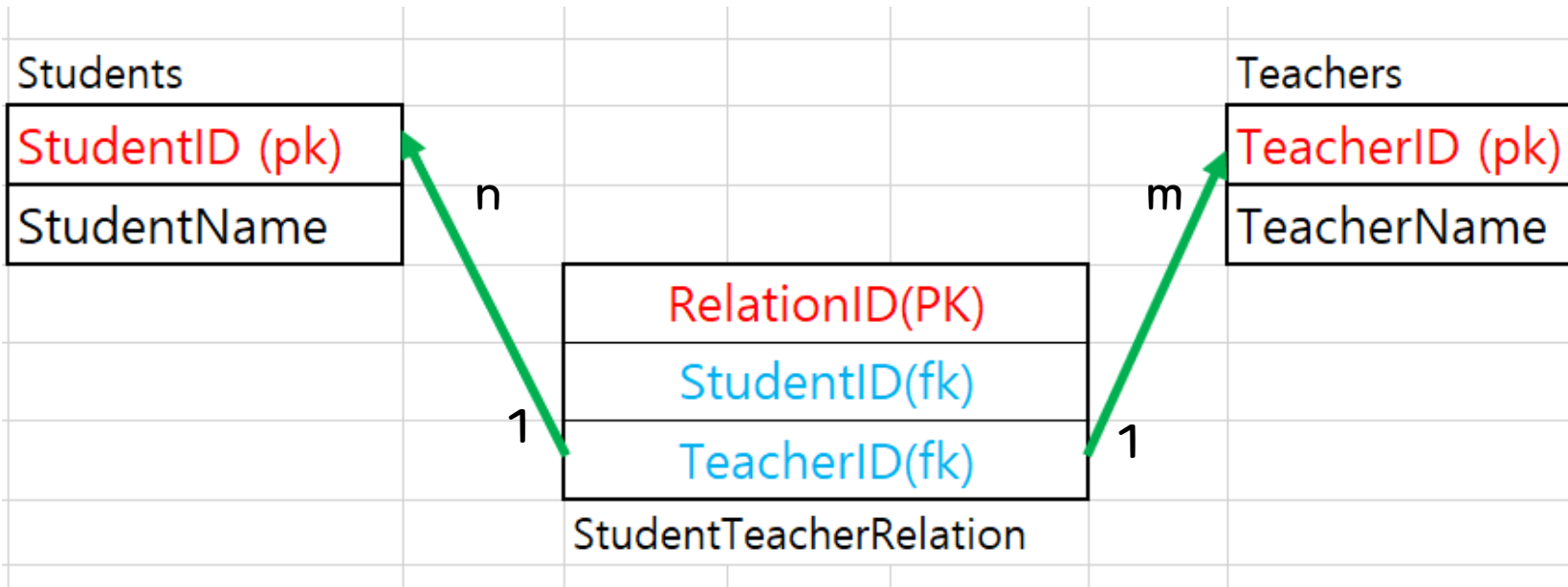
다:다 관계

- 다:다 관계는 한 쪽 레코드가 다른 쪽 레코드 여러 개와 관련되고, 그 반대쪽 레코드도 다른 쪽 레코드 여러 개와 관련되어 있는 관계를 의미
- 이 관계는 중간 테이블을 사용하여 구현
- 예시) 학생과 과목, 주문과 제품, 배우와 영화

생성방법

```
CREATE TABLE Students (  
    StudentID INT PRIMARY KEY,  
    StudentName VARCHAR(50)  
);  
  
CREATE TABLE Teachers (  
    TeacherID INT PRIMARY KEY,  
    TeacherName VARCHAR(50)  
);  
  
CREATE TABLE StudentTeacherRelation (  
    RelationID INT PRIMARY KEY,  
    StudentID INT,  
    TeacherID INT,  
    FOREIGN KEY (StudentID) REFERENCES Students(StudentID),  
    FOREIGN KEY (TeacherID) REFERENCES Teachers(TeacherID)  
);
```

Students와 Teacher의 관계 (N:M)



생성방법

```
Student.belongsToMany(Teacher, { through: StudentTeacherRelation });  
Teacher.belongsToMany(Student, { through: StudentTeacherRelation });
```

- belongsToMany: 다:다 관계를 설정할 때 사용되는 메서드