

## 데이터베이스\_논리적설계LAB

### 1. 정규 엔티티 타입

EMPLOYEE(Empno, Name, Address, Title)

### 2. 2진 1:1 관계 타입

\*\*\*

일반적으로는 관계 타입의 별도의 릴레이션을 만드는 것은 좋지 않다. 하지만 릴레이션 R과 S 사이에 관계 집합의 크기가 작을때, 즉 NULL값이 많아지는 경우에는 관계 릴레이션을 따로 만들어서 실질적으로 관계 있는 것만을 표현하는 것이 낫다.

\*\*\*

(1)

EMPLOYEE(Emp-id, Name, Salary)

MANAGES(Emp-id, Dept-no, Start)

DEPARTMENT(Dept-no, Name, Location)

(2)

EMPLOYEE(Emp-id, Name, Salary)

DEPARTMENT(Dept-no, Emp-id, Name, Location, Start)

(위에서 emp-id는 FK, Start는 관계 애트리뷰트)

(1:N 이면 1쪽에서 기본키를 가져와야함, 그것이 N쪽에서는 Foreign Key가 됨)

### 3. M:N 관계 타입

STUDENT(Sid, Sname, Dept)

COURSE(Cid, Cname, Credits)

ENROLLED(Sid, Cid, Grade)

### 4. 약한 엔티티 타입, 관계

\*\*\* R2의 소유자 엔티티가 무엇인지 생각해볼것

R1(A, C)

R2(A, B, D)

R3(D, E, F)

R4(G, A, B)

Has(B, G)

(위에서 R2의 D는 FK, R4의 A,B는 FK)

### 5. 복합 애트리뷰트

STUDENT(Sid, Name, Email, Phone, Address, Street, City, State, Zip)

\*\*\* Address는 나타내지 않음

### 6.

STUDENT(Ssn, Name, Email, Deptname, Joiningdate)

DEPARTMENT(DeptName, Address, Phone)

7.

EMPLOYEE(Eid, Name, Dept, Designation)

Assignment(Eid, Pid, Date)

PROJECT(Pid, Duration, Location, Name)

8.

PARTICIPANT(Participartid, Name, Address)

COURT(Capacity, Courttype, Courtid)

TOURNAMENT(Notimespryear, location, Name, Tournamentid)

Match(Participartid, Courtid, Tournamentid, Time, Date)

9.

PATIENT(Patientid, Name, Address)

PET(Birthdate, Petname, Patientid)

10.

EMPLOYEE(Empno, Emphname, Address)

MVA(Empno, Skill)