

## 사원

사번
이름
주소
전화
정규직
연봉
임시직
월급
수당

### 1. 슈퍼타입 단위

전체를 하나의 테이블로 관리하는 것

테이블 명 : 사원

사번	이름	주소	전화	사원구분	연봉	월급	수당
PK							

---

테이블 명 : 사원

사번	이름	주소	전화	사원구분	연봉	월급	수당
PK							
1	홍길동	지리산	10	0	7000	NULL	NULL
2	임꺽정	구월산	11	0	8000	NULL	NULL
3	장길산	묘향산	12	1	NULL	200	10
4	김연아	서울	13	0	3400	NULL	NULL

1-1) 전체 사원 검색

SELECT \* FROM 사원;

1-2) 정규직 사원 검색

SELECT 사번,이름,주소,전화,연봉 FROM 사원 WHERE 사원구분=0

1-3) 임시직 사원 검색

SELECT 사번,이름,주소,전화,월급,수당 FROM 사원 WHERE 사원구분=1

장점 : 전체 사원 조회가 많을 때 유리한 설계 방식

단점 : NULL 이 많이 존재할 수 밖에 없다

고려할 조건

1) 전체 사원 조회가 많을 때

2) 사원구분에 따른 "고유 속성"이 많을 때

2. 서브타입 단위

서브타입 개수만큼 테이블을 설계하는 방식

테이블 명 : 정규직 사원

사번	이름	주소	전화	연봉
PK				

테이블 명 : 임시직 사원

사번	이름	주소	전화	월급	수당
PK					

테이블 명 : 정규직 사원

사번	이름	주소	전화	연봉
PK				
1	홍길동	지리산	10	7000
2	임꺽정	구월산	11	8000
4	김연아	서울	13	3400

테이블 명 : 임시직 사원

사번	이름	주소	전화	월급	수당
----	----	----	----	----	----

PK					
3	장길산	묘향산	12	200	10

장점 : 전체 조회보다 정규직/임시직을 구별해서 검색하는 경우가 많을 때 편리

단점 : 전체 조회할 때 Union연산으로 합쳐야 한다

관리가 어렵다(테이블간 데이터 이동/교집합 발생하지 않게 관리)

고려할 조건

- 1) 정규직/임시직을 구별해서 업무처리가 많을 때
- 2) 서브타입간에 "고유 속성"이 많을 경우

### 3. 중간 타입

테이블 명 : 사원

사번	이름	주소	전화	사원구분
PK				

테이블 명 : 정규직 사원

사번	연봉
----	----

PK, FK	

테이블 명 : 임시직 사원

사번	월급	수당
PK, FK		

급여 검색시 Join이 필요하다

전체 사원의 급여를 제외한 일반 검색시 유리

재귀적 관계, 순환적 관계

1. 각 사원에 대한 관리자(사수)에 대한 정보를 관리

테이블 명 : 사원

사번	이름	주소	입사일	급여	관리자사번
PK					FK



테이블 명 : 사원

사번	이름	주소	입사일	급여	관리자사번
PK					FK
1	홍길동	지리산	20100214	9000	NULL
2	임꺽정	구월산	20090121	8000	1
3	장길산	묘향산	20151013	6000	2
4	김연아	서울	20201124	4500	1

검색 시 Self Join을 해서 검색한다

테이블 명 : 부서

---

부서번호	부서명	위치	상위부서
PK			FK

테이블 명 : 부서

부서번호	부서명	위치	상위부서
PK			FK
1	본부1	서울	NULL
2	본부2	광주	NULL
11	부서11	서울	1
12	부서12	부산	1
21	부서21	서울	2
22	부서22	광주	2

Self Join으로 검색하면 된다

<최상위 부서만 검색>

```
SELECT * FROM 부서 WHERE 상위부서 IS NULL;
```

