

해당 창작물의 허가되지 않은 재배포 , 무단 복사 및 무단 게시를 엄격히 금지합니다.  
반드시 저작자에게 문의한 후 허가를 받아 사용해주세요.

( 자료 중 빨간글씨로 작성된 숫자는 [ SQL 자격검정 실전문제 , 한국데이터산업진흥원 ] 책의 문항을 의미하며 해당 개념으로 관련된 문항이라고 해석을 하시면 됩니다. )

# SQLD 특강

06 ~ 11 데이터 모델과 성능

강사 강태우

# SQLD 특강

06. 성능 데이터 모델링의 개요

07. 정규화와 성능

08. 반정규화와 성능

09. 대량 데이터에 대한 성능 (이부분은 DATA ON AIR에서 한번 읽어보세요)

10. 데이터베이스 구조와 성능

11. 분산 데이터베이스와 성능

## 06. 성능 데이터 모델링의 개요

# 성능 데이터 모델링이란?

데이터베이스 설계할 때부터 성능을 고려해서 성능이 좋은 프로그램을 만드는 것

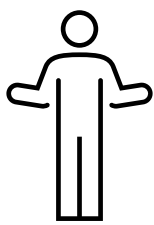
직원ID	패스워드	이름	성별	나이	입사일시	주민등록번호	연봉	연락처	주소
hello1234	12345	애호박	남	24	2022/12/05	912312-1523452	3500	062-123-1234	남구 <u>진월동</u> 아파트
hello1234	12345	애호박	남	24	2022/12/05	912312-1523452	3500	062-123-1234	서울 중구 12로
hello1234	12345	애호박	남	24	2022/12/05	912312-1523452	3500	010-1231-1231	남구 <u>진월동</u> 아파트
hello1234	12345	애호박	남	24	2022/12/05	912312-1523452	3500	010-1231-1231	서울 중구 12로

이 상태에서 연락처 010 -2222-4444 를 추가하는 방법은??

# 성능 데이터 모델링이란?

데이터베이스 설계할 때부터 성능을 고려해서 성능이 좋은 프로그램을 만드는 것

BAD..



직원

직원ID
패스워드
이름
성별
나이
입사일시
주민등록번호
연봉
연락처
주소

부서

부서ID
부서명
근무지

데이터 중복을 고려안하고 설계한 A씨

직원

직원ID
패스워드
이름
성별
나이
입사일시
주민등록번호
연봉
부서ID(FK)

부서

부서ID
부서명
근무지

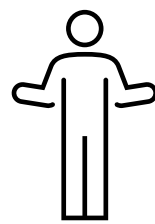
직원연락처

직원ID (FK)
연락처

직원주소

직원ID (FK)
주소

GOOD !



데이터 중복을 고려해 설계 중에 정규화한 B씨

# 성능 데이터 모델링이란?

데이터베이스 설계할 때부터 성능을 고려해서 성능이 좋은 프로그램을 만드는 것

직원

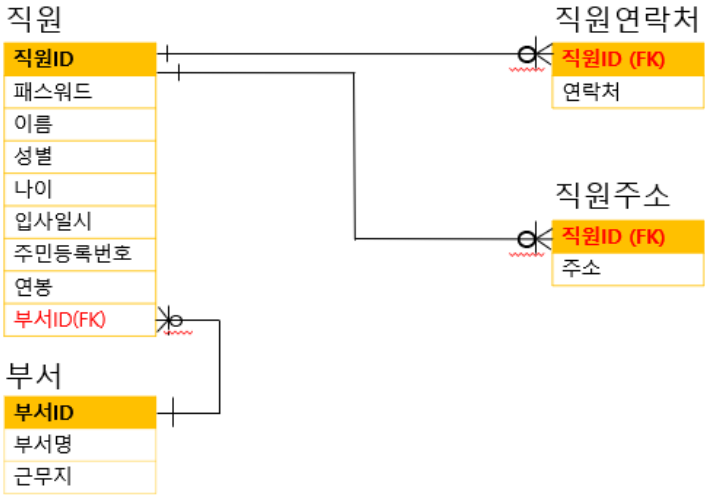
직원ID	패스워드	이름	성별	나이	입사일시	주민등록번호	연봉
hello1234	12345	애호박	남	24	2022/12/05	912312-1523452	3500

직원연락처

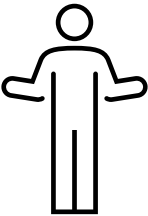
직원ID	연락처
hello1234	062-123-1234
hello1234	010-1231-1231

직원주소

직원ID	주소
hello1234	남구 진월동 아파트
hello1234	서울 중구 12로



GOOD !



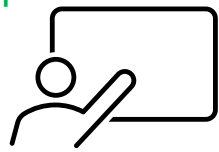
데이터 중복을 고려해 설계 중에 정규화한 B씨

# 성능 데이터 모델링 고려사항 및 순서

32,33

- ① 애초에 모델링할 때 정규화를 정확히 수행합니다. \*정규화는 뒤에서 자세하게 다룹니다
- ② 데이터베이스 용량을 체크(산정)해봅니다.
- ③ 데이터베이스에서 실행되는 트랜잭션 유형을 파악합니다.
- ④ 용량과 트랜잭션의 유형에 따라 반정규화를 수행합니다 \*반정규화는 뒤에서 자세하게 다룹니다
- ⑤ 이력모델의 조정, PK/FK조정, 슈퍼타입/서브타입 조정 등을 수행합니다.
- ⑥ 성능관점에서 데이터 모델을 검증한다.

성능을 고려한 데이터모델링은  
빨리 할수록 처리비용이  
감소합니다.



# 성능 데이터 모델링 고려사항 및 순서

① 애초에 모델링할 때 정규화를 정확히 수행합니다.      \*정규화는 뒤에서 자세히 다룹니다

이렇게 하면 데이터 중복을 방지해 데이터 입력,수정,삭제 시 무조건 성능이 증가합니다. (조회는 조건에 따라 다름)

직원ID	패스워드	이름	성별	나이	입사일시	주민등록번호	연봉	연락처	주소
hello1234	12345	애호박	남	24	2022/12/05	912312-1523452	3500	062-123-1234	남구 진월동 아파트
hello1234	12345	애호박	남	24	2022/12/05	912312-1523452	3500	062-123-1234	서울 중구 12로
hello1234	12345	애호박	남	24	2022/12/05	912312-1523452	3500	010-1231-1231	남구 진월동 아파트
hello1234	12345	애호박	남	24	2022/12/05	912312-1523452	3500	010-1231-1231	서울 중구 12로



직원

직원ID	패스워드	이름	성별	나이	입사일시	주민등록번호	연봉
hello1234	12345	애호박	남	24	2022/12/05	912312-1523452	3500

직원연락처

직원ID	연락처
hello1234	062-123-1234
hello1234	010-1231-1231

직원주소

직원ID	주소
hello1234	남구 진월동 아파트
hello1234	서울 중구 12로



# 성능 데이터 모델링 고려사항 및 순서

② 데이터베이스 용량을 체크(산정)해봅니다.

각각의 엔터티(or 테이블) 중에 어떤 곳에 데이터가 집중이 되는지 확인을 해보는 과정입니다.  
즉 어떤 엔터티(or 테이블) 이 대용량 데이터인지 확인을 하는 것입니다.

ex) 1억건의 데이터가 있는 테이블에서 1건 뽑기

직원

직원ID	패스워드	이름	성별	나이	입사일시	주민등록번호	연봉
hello1234	12345	애호박	남	24	2022/12/05	912312-1523452	3500

직원연락처

직원ID	연락처
hello1234	062-123-1234
hello1234	010-1231-1231

직원주소

직원ID	주소
hello1234	남구 <u>진월동</u> 아파트
hello1234	서울 중구 12로

# 성능 데이터 모델링 고려사항 및 순서

③ 데이터베이스에서 실행되는 **트랜잭션 유형을 파악**합니다. \* 트랜잭션은 업무 수행을 위한 개별적인 SQL 을 의미

예를 들어 “직원” 테이블에 대해 조회(R) 가 많다면 조회 중심으로 성능을 모델링할 수 있습니다.

직원

직원ID	패스워드	이름	성별	나이	입사일시	주민등록번호	연봉
hello1234	12345	애호박	남	24	2022/12/05	912312-1523452	3500

직원연락처

직원ID	연락처
hello1234	062-123-1234
hello1234	010-1231-1231

직원주소

직원ID	주소
hello1234	남구 진월동 아파트
hello1234	서울 중구 12로

CRUD 매트릭스

업무내용	직원	직원연락처	직원주소
직원을 등록한다	C		
직원 정보를 조회한다	R	R	R
직원의 입사일시를 기준으로 연봉을 조회한다	R		
직원 연락처를 등록한다.		C	
직원의 주민등록번호를 조회한다.	R		

# 성능 데이터 모델링 고려사항 및 순서

④ 용량과 트랜잭션의 유형에 따라 반정규화를 수행합니다

반정규화를 수행하려는 목적은 당연히 조회 성능 향상 입니다. ( 정규화 vs 반정규화 ??? )

직원

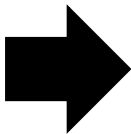
직원ID	패스워드	이름	성별	나이	입사일시	주민등록번호	연봉
hello1234	12345	애호박	남	24	2022/12/05	912312-1523452	3500

직원연락처

직원ID	연락처
hello1234	062-123-1234
hello1234	010-1231-1231

직원주소

직원ID	주소
hello1234	남구 <u>진월동</u> 아파트
hello1234	서울 중구 12로



직원ID	패스워드	이름	성별	나이	입사일시	주민등록번호	연봉	연락처	주소
hello1234	12345	애호박	남	24	2022/12/05	912312-1523452	3500	062-123-1234	남구 <u>진월동</u> 아파트
hello1234	12345	애호박	남	24	2022/12/05	912312-1523452	3500	062-123-1234	서울 중구 12로
hello1234	12345	애호박	남	24	2022/12/05	912312-1523452	3500	010-1231-1231	남구 <u>진월동</u> 아파트
hello1234	12345	애호박	남	24	2022/12/05	912312-1523452	3500	010-1231-1231	서울 중구 12로

반정규화는 데이터를 다시 중복, 통합 하는 것입니다.

## 07. 정규화와 성능

# 정규화란?

조회/입력/수정/삭제 등

데이터에 대한 중복을 제거하고 데이터가 관심사별로 처리되도록 엔터티를 쪼개 성능을 향상시키는 활동

단, 정규화가 대체적으로 좋은 것이며, 경우에 따라서는 반정규화가 더 이득일 수도 있음



정규화시 입력/삭제/수정은 성능 향상! 조회는 성능 향상 혹은 저하 될 수 있음

# 함수적 종속이란? $F(X) = Y$

정규화를 하기 위해 알아야 하는 개념으로

어떤 기준 값(컬럼)에 의해 데이터가 종속되는 현상을 의미

아이디	이름	나이	연봉	부서ID	부서명
Emp001	강태우	35	5000	D001	인사부
Emp002	김형준	32	4800	D001	인사부
Emp003	송송이	27	3400	D002	급여부
emp004	강태우	34	3700	D002	급여부

예) 결정자(아이디) -> 종속자( 이름,나이,연봉,부서ID,부서명) : 아이디가 나머지 모든 컬럼을 함수종속한다

예) 결정자(부서ID) -> 종속자( 부서명 ) : 부서ID 가 부서명 컬럼을 함수종속한다

# 정규화 과정 상세히 알아보기

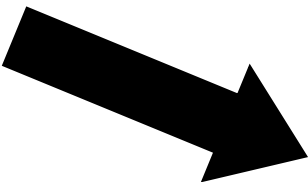
\* 실제로 처리하는 범위는 3NF 정도까지임



# 정규화 ( 1차 정규화 )

엔터티에서 **다중 속성**이 있거나 하나의 속성에 **여러 속성값**이 있으면 **원자값으로 쪼개준다.**

직원ID	이름	연봉	연락처	부서ID	부서명
A0001	강홍정	5000	010-1231-1234 , 062-123-1231	D001	인사부
A0002	정치호	6000	062-222-2222	D002	급여부
A0003	강태우	7000	010-3333-3333	D001	인사부



1차 정규화  
(도메인원자성)

직원ID	이름	연봉	연락처	부서ID	부서명
A0001	강홍정	5000	010-1231-1234	D001	인사부
A0001	강홍정	5000	062-123-1231	D001	인사부
A0002	정치호	6000	062-222-2222	D002	급여부
A0002	정치호	6000	062-222-2222	D002	급여부
A0003	강태우	7000	010-3333-3333	D001	인사부

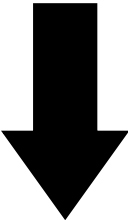


# 정규화 ( 1차 정규화 )

엔터티에서 **다중 속성**이 있거나 하나의 속성에 **여러 속성값**이 있으면 **원자값으로 쪼개준다.**

직원ID	이름	연봉	연락처	부서ID	주소1	주소2	부서명
A0001	강홍정	5000	010-1231-1234	D001	중구 택지로 1번지	서울 중양 보훈기관 123호	인사부

1차 정규화  
(도메인원자성)



직원ID	이름	연봉	연락처	부서ID	주소	부서명
A0001	강홍정	5000	010-1231-1234	D001	중구 택지로 1번지	인사부
A0001	강홍정	5000	010-1231-1234	D001	서울 중양 보훈기관 123호	인사부


# 정규화 ( 2차 정규화 )

2차 정규화는 [1정규형 -> 2정규형(NF)] 형태로 엔터티를 변경하는 활동이다.

식별자에 부분 종속 하는 대상을 쪼개어 준다.

학번	과목	성적	담당교수	소속학과
C0001	국어	A	김철민	교육융합과
C0001	수학	A	김철민	교육융합과
C0001	영어	B	김철민	교육융합과
C0002	국어	B	신두희	독일언어문학과
C0002	수학	C	신두희	독일언어문학과

2차 정규화  
(부분종속제거)



학번	담당교수	소속학과
C0001	김철민	교육융합과
C0002	신두희	독일언어문학과

학번	과목	성적
C0001	국어	A
C0001	수학	A
C0001	영어	B
C0002	국어	B
C0002	수학	C

이 테이블의 결정자는 학번과 과목이다.  
\*결정자(학번,과목) -> 종속자 (성적 , 담당교수 , 소속학과)

그런데 [학번] 만으로도 [담당교수 , 소속학과] 을 유일하게 구분할 수 있다.

즉, 결정자 전부에 나머지 컬럼이 완전 종속하지 않고, 결정자 부분적으로도 종속한다면 쪼개야 한다

# 정규화 ( 2차 정규화 )

\* 2차 정규화 푸는 팁 \*

학번 과목
성적 담임교수 소속학과

결정자 ( 학번 , 과목 ) -> 종속자 ( 성적 , 담임교수 , 소속학과 )

결정자 ( 학번 ) -> 종속자 ( 담임교수 , 소속학과 )

결정자 ( 학번 , 과목 ) -> 종속자 ( 성적 )

1. 위에서 **부분적 종속**한다고 하는 것들을 따로 뜯어냅니다 (결정자는 주식별자로)

학번
담임교수 소속학과

# 정규화 ( 2차 정규화 )

\* 2차 정규화 푸는 팁 \*

학번 과목
성적 담임교수 소속학과

결정자 ( 학번 , 과목 ) -> 종속자 ( 성적 , 담임교수 , 소속학과 )

결정자 ( 학번 ) -> 종속자 ( 담임교수 , 소속학과 )

결정자 ( 학번 , 과목 ) -> 종속자 ( 성적 )

2. 기존의 엔터티에서 학번이 가져간 일반속성을 제외하고 입력합니다.

학번
담임교수 소속학과

학번 과목
성적

# 정규화 ( 2차 정규화 ) 35

\* 2차 정규화 푸는 팁 \*

학번 과목
성적 담임교수 소속학과

결정자 ( 학번 , 과목 ) -> 종속자 ( 성적 , 담임교수 , 소속학과 )

결정자 ( 학번 ) -> 종속자 ( 담임교수 , 소속학과 )

결정자 ( 학번 , 과목 ) -> 종속자 ( 성적 )

3. 기존의 데이터를 파악하여 관계를 설정합니다.



# 정규화 ( 3차 정규화 )

엔터티에서 일반속성끼리 함수종속이 발생하면 쪼개줍니다.

직원ID	이름	연봉	연락처	주소	부서ID	부서명
A0001	강홍정	5000	010-1231-1234	중구 택지로 1번지	D001	인사부
A0002	정치호	6000	062-222-2222	광주 광산구 첨단로	D002	급여부
A0003	강태우	7000	010-3333-3333	서울 강남 택지로	D001	인사부

결정자( 직원ID ) -> 종속자( 나머지 컬럼들 ) // 완전함수종속! 2NF 만족

그런데 나머지 일반 컬럼중에서 결정자(부서ID) -> 종속자(부서명) 이 발생했음

이걸 쪼개줘야 함. 방법은 2차 정규화에서 했던 방식처럼 동일

# 정규화 ( 3차 정규화 )

엔터티에서 일반속성끼리 함수종속이 발생하면 쪼개줍니다.

직원ID	이름	연봉	연락처	주소	부서ID	부서명
A0001	강홍정	5000	010-1231-1234	중구 택지로 1번지	D001	인사부
A0002	정치호	6000	062-222-2222	광주 광산구 첨단로	D002	급여부
A0003	강태우	7000	010-3333-3333	서울 강남 택지로	D001	인사부

결정자( 직원ID ) -> 종속자( 나머지 컬럼들 ) // 완전함수종속! 2NF 만족

직원ID	이름	연봉	연락처	주소	부서ID	부서ID	부서명
A0001	강홍정	5000	010-1231-1234	중구 택지로 1번지	D001	D001	인사부
A0002	정치호	6000	062-222-2222	광주 광산구 첨단로	D002	D002	급여부
A0003	강태우	7000	010-3333-3333	서울 강남 택지로	D001		

# 정규화 ( 3차 정규화 )

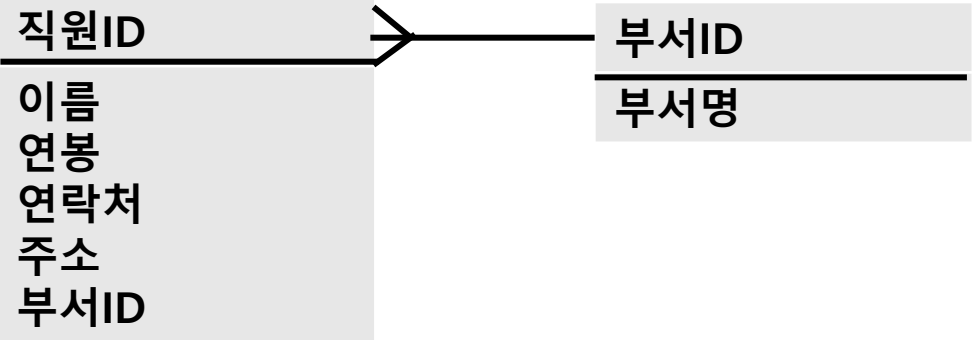
엔터티에서 일반속성끼리 함수종속이 발생하면 쪼개줍니다.

직원ID
이름
연봉
연락처
주소
부서ID
부서명

결정자 ( 직원ID ) -> 종속자 (이름,연봉,연락처,주소,부서ID,부서명)

결정자 (부서ID) -> 종속자 (부서명)

3차 정규화  
(이행함수종속제거)





## 08. 반정규화와 성능

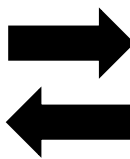
# 반정규화(=역정규화)

정규화한 데이터를 다시 합쳐서 중복되게 하거나 통합 , 분리 등을 수행하는 모델링 방식

- ① 테이블 반정규화 ( 테이블병합 , 테이블분할 , 테이블추가 등)
- ② 컬럼 반정규화 ( 중복컬럼 , 파생컬럼 , 이력테이블 컬럼 추가 등)
- ③ 관계 반정규화 (중복관계 추가)

# 정규화 vs 반정규화 ?

직원ID	패스워드	이름	성별	나이	입사일시	주민등록번호	연봉	연락처	주소
hello1234	12345	에호박	남	24	2022/12/05	912312-1523452	3500	062-123-1234	남구 <u>진월동</u> 아파트
hello1234	12345	에호박	남	24	2022/12/05	912312-1523452	3500	062-123-1234	서울 중구 12로
hello1234	12345	에호박	남	24	2022/12/05	912312-1523452	3500	010-1231-1231	남구 <u>진월동</u> 아파트
hello1234	12345	에호박	남	24	2022/12/05	912312-1523452	3500	010-1231-1231	서울 중구 12로



직원

직원ID	패스워드	이름	성별	나이	입사일시	주민등록번호	연봉
hello1234	12345	에호박	남	24	2022/12/05	912312-1523452	3500

직원연락처

직원ID	연락처
hello1234	062-123-1234
hello1234	010-1231-1231

직원주소

직원ID	주소
hello1234	남구 <u>진월동</u> 아파트
hello1234	서울 중구 12로

▶ Hello1234 직원의 이름, 연락처 , 주소를 가져오는 SQL을 작성해주세요.

[ 반정규화 상태에서 조회하기 ]

```
SELECT 이름
      , 연락처
      , 주소
FROM 직원
WHERE A.직원ID = 'hello1234' ;
```

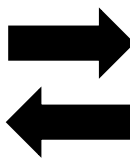


[ 정규화 상태에서 조회하기 ]

```
SELECT A.이름
      , B.연락처
      , C.주소
FROM 직원 A
      , 직원연락처 B
      , 직원주소 C
WHERE A.직원ID = B.직원ID
      AND B.직원ID = C.직원ID
      AND A.직원ID = 'hello1234' ;
```

# 정규화 vs 반정규화 ?

직원ID	패스워드	이름	성별	나이	입사일시	주민등록번호	연봉	연락처	주소
hello1234	12345	에호박	남	24	2022/12/05	912312-1523452	3500	062-123-1234	남구 <u>진월동</u> 아파트
hello1234	12345	에호박	남	24	2022/12/05	912312-1523452	3500	062-123-1234	서울 중구 12로
hello1234	12345	에호박	남	24	2022/12/05	912312-1523452	3500	010-1231-1231	남구 <u>진월동</u> 아파트
hello1234	12345	에호박	남	24	2022/12/05	912312-1523452	3500	010-1231-1231	서울 중구 12로



직원

직원ID	패스워드	이름	성별	나이	입사일시	주민등록번호	연봉
hello1234	12345	에호박	남	24	2022/12/05	912312-1523452	3500

직원연락처

직원ID	연락처
hello1234	062-123-1234
hello1234	010-1231-1231

직원주소

직원ID	주소
hello1234	남구 <u>진월동</u> 아파트
hello1234	서울 중구 12로

▶ Hello1234 직원의 이름을 가져오는 SQL을 작성해주세요.

[ 반정규화 상태에서 조회하기 ]

```
SELECT 이름
FROM 직원
WHERE A.직원ID = 'hello1234' ;
```

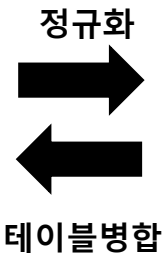
[ 정규화 상태에서 조회하기 ]

```
SELECT 이름
FROM 직원
WHERE A.직원ID = 'hello1234' ;
```



# 정규화 vs 반정규화 ?

직원ID	패스워드	이름	성별	나이	입사일시	주민등록번호	연봉	연락처	주소
hello1234	12345	에호박	남	24	2022/12/05	912312-1523452	3500	062-123-1234	남구 <u>진월동</u> 아파트
hello1234	12345	에호박	남	24	2022/12/05	912312-1523452	3500	062-123-1234	서울 중구 12로
hello1234	12345	에호박	남	24	2022/12/05	912312-1523452	3500	010-1231-1231	남구 <u>진월동</u> 아파트
hello1234	12345	에호박	남	24	2022/12/05	912312-1523452	3500	010-1231-1231	서울 중구 12로



직원								
직원ID	패스워드	이름	성별	나이	입사일시	주민등록번호	연봉	
hello1234	12345	에호박	남	24	2022/12/05	912312-1523452	3500	

직원연락처	
직원ID	연락처
hello1234	062-123-1234
hello1234	010-1231-1231

직원주소	
직원ID	주소
hello1234	남구 <u>진월동</u> 아파트
hello1234	서울 중구 12로

▶ Hello1234 직원의 휴대폰 '010-5555-5555' 를 추가해주세요.

[ 반정규화 상태에서 입력하기 ]

기존 내용이 중복되어 들어감

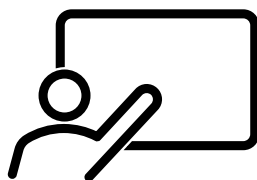
[ 정규화 상태에서 입력하기 ]

직원 연락처 엔터티에만 입력하면 됨



# 반정규화 대상 조사 방법

- 1. 반정규화가 필요한 대상을 찾아봅니다.
- 2. 다른 방법이 없는지 검토해봅니다.
- 3. 달리 방법이 없다면 반정규화를 적용합니다.



반정규화는 데이터의 무결성을 깨질 위험이 있습니다.  
그래서 가급적이면 다른 방법을 생각해보고 안되겠다  
싶으면 반정규화를 합니다.



- ① 테이블 반정규화 (테이블병합, 테이블분할, 테이블추가 등)
- ② 컬럼 반정규화 (중복컬럼, 파생컬럼, 이력테이블 컬럼, pk에 의한 컬럼 추가 등)
- ③ 관계 반정규화 (중복관계 추가)

# 테이블 반정규화

\* 종류가 많으므로 대표적인 예시만 몇 개 소개합니다.

(1) 수직 분할 과 수평 분할

- 테이블을 세로로 자릅니다. 자주 사용하는 컬럼만 떼어낼 때 사용할 수 있습니다.

직원ID	이름	연봉	연락처	주소	부서ID
A0001	강홍정	5000	010-1231-1234	종구 택지로 1번지	D001
A0002	정치호	6000	062-222-2222	광주 광산구 첨단로	D002
A0003	강태우	7000	010-3333-3333	서울 강남 택지로	D001

직원의 이름만 연봉만 자주 조회를 한다면 아래처럼 자주쓰는 컬럼만 따로 떼어준다.

직원ID	이름	연봉
A0001	강홍정	5000
A0002	정치호	6000
A0003	강태우	7000

직원ID	연락처	주소	부서ID
A0001	010-1231-1234	종구 택지로 1번지	D001
A0002	062-222-2222	광주 광산구 첨단로	D002
A0003	010-3333-3333	서울 강남 택지로	D001

# 테이블 반정규화

\* 종류가 많으므로 대표적인 예시만 몇 개 소개합니다.

(1) 수직 분할 과 수평 분할

- 테이블을 가로로 자릅니다. 자주 사용하는 튜플(행)만 떼어낼 수 있습니다.

[ 직원ID 가 'A0001' ~ 'A0100' 만 주로 쓴다면 수평으로 분할 ]

직원 9999명

직원ID	연락처
A0001	010-1231-1234
A0002	062-222-2222
A0003	010-3333-3333
A0004	010-4444-4444
...	...
A9999	010-9999-9999

자주 조회되는  
직원 100명

직원ID	연락처
A0001	010-1231-1234
A0002	062-222-2222
A0003	010-3333-3333
...	...
A0100	010-0100-0100

그외 9899명

직원ID	연락처
A0101	010-0101-0101
A0102	062-2022-2222
A0103	010-0003-3333
...	...
A9999	010-9999-9999



# 컬럼 반정규화

\* 종류가 많으므로 대표적인 예시만 몇 개 소개합니다.

## (1) 중복컬럼 추가

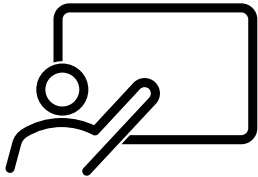
- 테이블에 중복되는 컬럼을 추가해 조인을 감소시킵니다.

### 직원

직원ID	패스워드	이름	성별	나이	입사일시	주민등록번호	연봉	연락처
hello1234	12345	애호박	남	24	2022/12/05	912312-1523452	3500	062-123-1234
hello1234	12345	애호박	남	24	2022/12/05	912312-1523452	3500	010-1231-1231

### 직원연락처

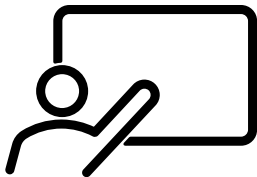
직원ID	연락처
hello1234	062-123-1234
hello1234	010-1231-1231



직원 엔터티에 연락처 컬럼을 추가하면 직원연락처 엔터티를 따로 조인하지 않아도 됩니다. 대신 직원연락처가 변경되면 똑같이 반영을 해줘야 합니다.

# 이력 컬럼 추가

직원ID	순번	연락처
A0001	1	010-1111-1111
A0001	2	010-1111-1112
A0001	3	010-1111-1113
A0001	4	010-1111-1114
A0002	1	010-2222-2221
A0002	2	010-2222-2222
A0002	3	010-2222-2223
A0002	4	010-2222-2224
A0002	5	010-2222-2225
A0002	6	010-2222-2226
A0003	1	010-3333-3331
A0003	2	010-3333-3332
A0003	3	010-3333-3333
...		

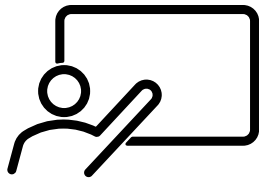


직원별로 가장 최근에 변경된 연락처 이력은?

```
SELECT *
FROM 직원연락처이력 A
WHERE 순번 = (
    SELECT MAX(순번)
    FROM 직원연락처이력
    WHERE 직원ID = A.직원ID);
```

# 이력 컬럼 추가 – 컬럼추가ver

직원ID	순번	연락처	최근여부
A0001	1	010-1111-1111	(null)
A0001	2	010-1111-1112	(null)
A0001	3	010-1111-1113	(null)
A0001	4	010-1111-1114	Y
A0002	1	010-2222-2221	(null)
A0002	2	010-2222-2222	(null)
A0002	3	010-2222-2223	(null)
A0002	4	010-2222-2224	(null)
A0002	5	010-2222-2225	(null)
A0002	6	010-2222-2226	Y
A0003	1	010-3333-3331	(null)
A0003	2	010-3333-3332	(null)
A0003	3	010-3333-3333	Y
...			



직원별로 가장 최근에 변경된 연락처 이력은?

```
SELECT * FROM 직원연락처이력 WHERE 최근여부 = 'Y' ;
```

# 관계 - 중복 관계 추가??

주문번호
P001
P002
P003
P004

주문번호	제품번호
P001	M001
P001	M002
P002	M003
P002	M001
P002	M002
P003	M001
P003	M002
P003	M003

제품번호	가격	제작소
M001	100	서울
M002	200	서울
M003	300	광주
M004	400	광주

제작소	주소
서울	중앙로
광주	충장로

아래 데이터모델에 대해 결과는 무엇인가?

주문번호
P001
P002
P003
P004

주문번호	제품번호
P001	M001
P001	M002
P002	M003
P002	M001
P002	M002
P003	M001
P003	M002
P003	M003

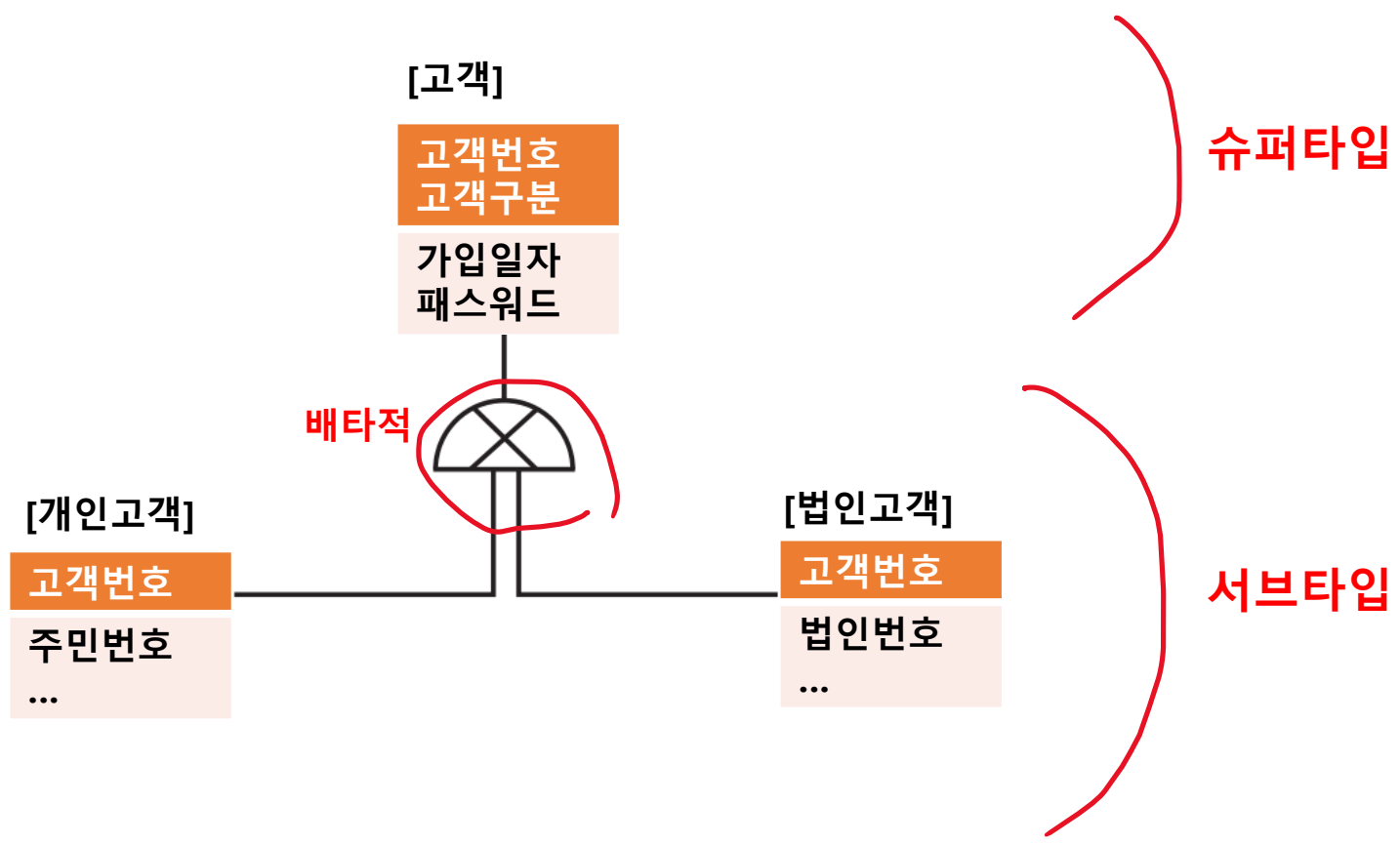
제품번호	단가
M001	100
M002	200
M003	300
M004	400

```
SELECT A.주문번호 , SUM(C.가격)
FROM 주문 A , 주문내역 B , 제품 C
WHERE A.주문번호 = B.주문번호
AND B.제품번호 = C.제품번호
GROUP BY A.주문번호 ;
```

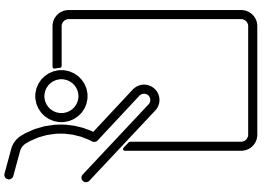
## 10. 데이터베이스 구조와 성능

# 슈퍼/서브 타입 데이터 모델

공통적인 특징을 가지는 것은 슈퍼모델로 , 각각의 차이는 서브모델로 표현하는 방식



이를 PLUS TYPE 이라고 합니다.  
- 조인이 발생합니다.



# 슈퍼/서브 타입 데이터 모델

공통적인 특징을 가지는 것은 슈퍼모델로 , 각각의 차이는 서브모델로 표현하는 방식

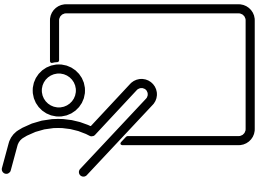
[고객]

고객번호
고객구분
가입일자
패스워드
주민번호
법인번호
...

하나로 모아서 관리

이를 SINGLE TYPE 이라고 합니다.

- 조인할 것들이 줄어 듭니다.
- 관리하기 편합니다.
- 개인고객을 찾고 싶어도 법인고객 까지 스캔하는 비효율이 있습니다.





# 슈퍼/서브 타입 데이터 모델

공통적인 특징을 가지는 것은 슈퍼모델로 , 각각의 차이는 서브모델로 표현하는 방식

[개인고객]

고객번호

주민번호  
가입일자  
패스워드  
...

[고객]

고객번호  
고객구분

가입일자  
패스워드

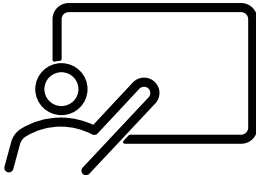
[법인가객]

고객번호

법인번호  
가입일자  
패스워드  
...

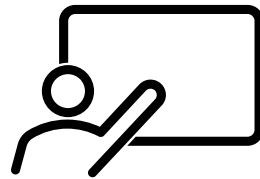
각각 개별로 생성

이를 ONE TO ONE TYPE  
이라고 합니다.  
  
- 조인이 많이 발생합니다.



# 11. 분산 데이터베이스와 성능

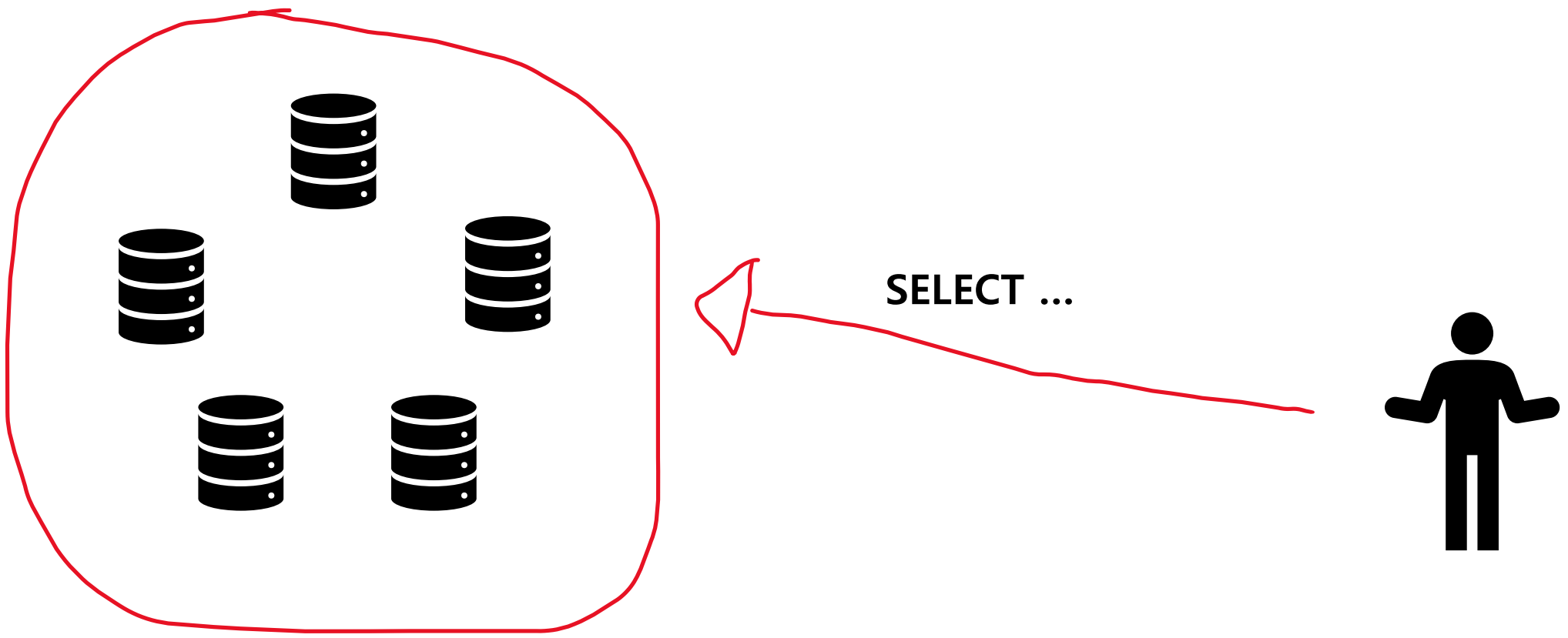
52



중요도가 낮습니다.  
특징과 장단점만 간단히 정리합니다.

# 분산 데이터베이스란?

물리적으로는 떨어져 있는 데이터베이스를 마치 하나의 시스템처럼 사용할 수 있는 논리적 개념의 데이터베이스를 의미



# 분산 데이터베이스의 투명성

분할 투명성 (단편화)

위치 투명성

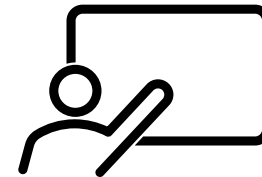
지역사상 투명성

중복 투명성

장애 투명성

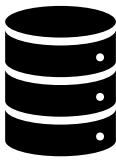
병행 투명성

투명성의 종류를 기억해두세요.

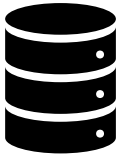


# 분산 데이터베이스의 투명성

## 분할 투명성 (단편화) & 중복투명성



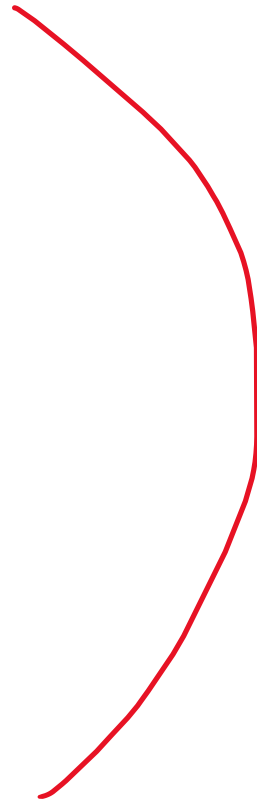
직원\_PART1 (A0001~A1000)  
직원\_PART2 (A1001 ~ A2000)



직원\_PART2 (A1001 ~ A2000)  
직원\_PART3 (A2001 ~ A3000)



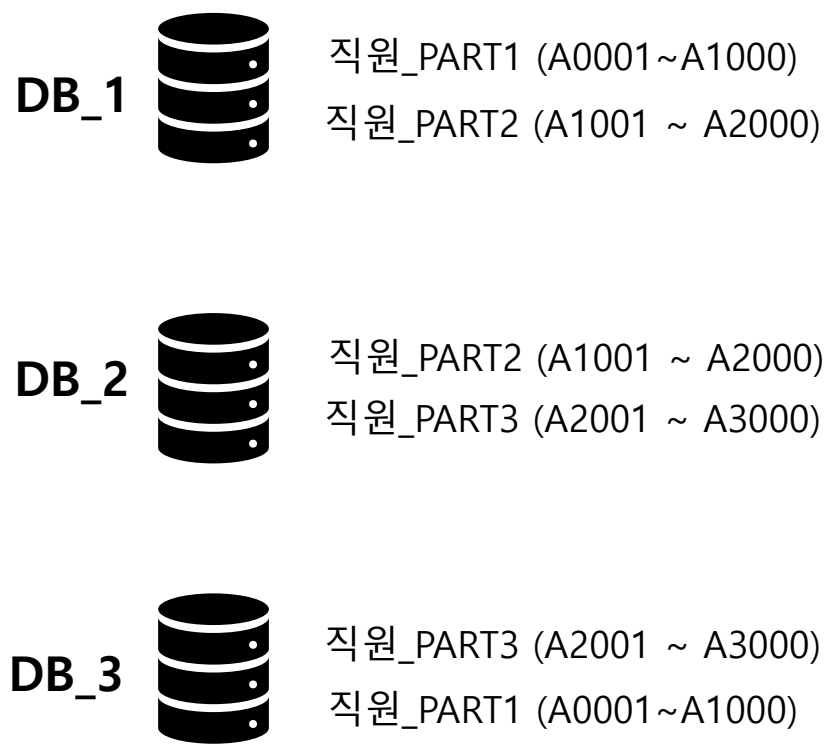
직원\_PART3 (A2001 ~ A3000)  
직원\_PART1 (A0001~A1000)



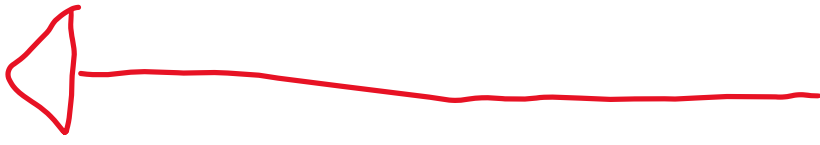
하나의 테이블을 여러 단편으로 분할 및 사본을  
여러 곳에 저장 &  
  
어디에 중복 되었는지는 우리는 알 필요가 없음

# 분산 데이터베이스의 투명성

## 위치 투명성 & 지역사상투명성



DB로 접속 및  
**SELECT \* FROM 직원**  
**WHERE 직원ID = 'A2005' ;**

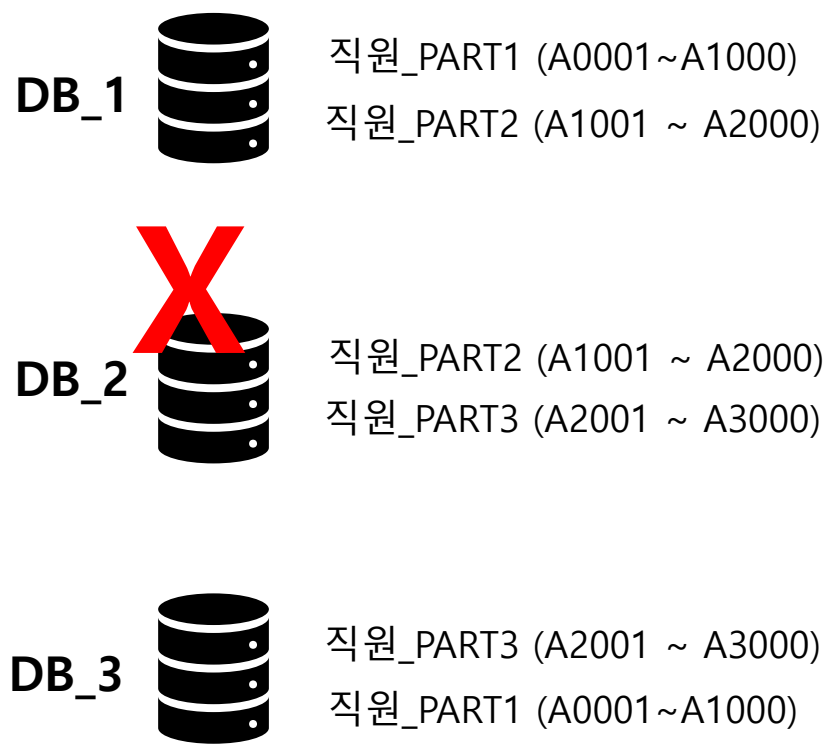


무슨 DB에 어떤 DB에 데이터가 있  
는지는 나는 잘 모르겠고,  
이거 뽑아주세요

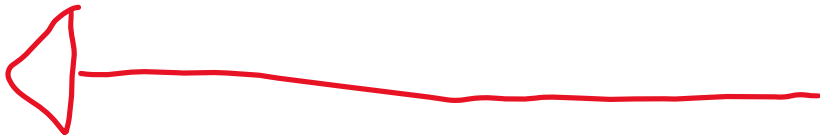
A black silhouette of a person with arms outstretched, representing a user who is asking for data without knowing its location.

# 분산 데이터베이스의 투명성

## 장애 투명성



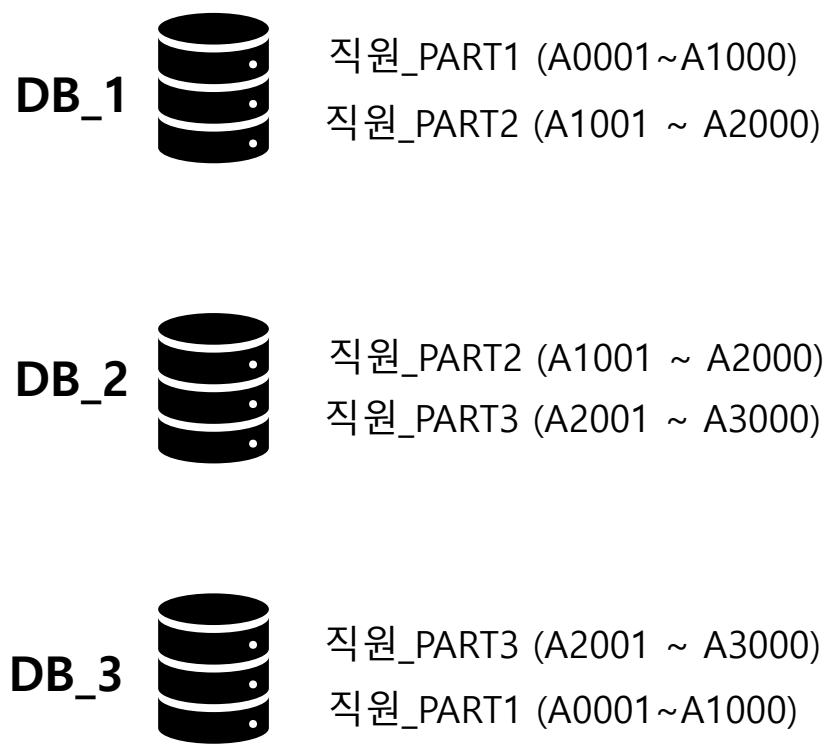
DB로 접속 및  
**SELECT \* FROM 직원**  
**WHERE 직원ID = 'A2005' ;**



DB\_2 가 장애 났었다구요?  
데이터 잘 출력되던데~

# 분산 데이터베이스의 투명성

## 병행 투명성



DB로 접속 및  
**SELECT \* FROM 직원**  
**WHERE 직원ID = 'A2005' ;**





# 분산 데이터베이스의 장단점

장점	단점
<ul style="list-style-type: none"><li>- 지역 자치성 , 용량 확장이 쉽다.<ul style="list-style-type: none"><li>- 신뢰성과 가용성</li><li>- 효율성과 융통성</li></ul></li><li>- 빠른 응답 속도와 통신비용 절감</li><li>- 각 지역 사용자의 요구 수용 증대</li></ul>	<ul style="list-style-type: none"><li>소프트웨어 개발 비용 증가</li><li>잠재적인 오류가 증가</li><li>처리할 비용이 증가</li><li>설계 및 관리가 어려움</li><li>분산이 되어 통제가 어려움</li><li>데이터가 중복되어 무결성 위험이 있음</li></ul>

해당 창작물의 허가되지 않은 재배포 , 무단 복사 및 무단 게시를 엄격히 금지합니다.  
반드시 저작자에게 문의한 후 허가를 받아 사용해주세요.

( 자료 중 빨간글씨로 작성된 숫자는 [ SQL 자격검정 실전문제 , 한국데이터산업진흥원 ] 책의 문항을 의미하며 해당 개념으로 관련된 문항이라고 해석을 하시면 됩니다. )

# SQLD 특강

06 ~ 11 데이터 모델과 성능 종료

강사 강태우

