

정규화

문혜영

학습 목차

- 01 이상현상
- 02 정규화
- 03 반정규화
- 04 해싱

- 이상현상(Anomaly)

과목코드	과목명	학번	이름	연락처
C01	DB개론	1101	홍길동	010-1111-2222
C02	DB실무	1101	홍길동	010-1111-2222
A01	마이닝	1102	강감찬	010-2222-3333
B01	통계	1101	홍길동	010-1111-2222
C01	DB개론	1103	김유신	010-3333-4444

- 삭제 이상 (delete anomaly)
 - '1102' 학생이 'A01'을 수강한 기록을 삭제할 때
- 삽입 이상 (insert anomaly)
 - (1104, 유관순, 010-4444-5555)인 학생을 추가하고자 할 때
- 갱신 이상 (update anomaly)
 - '1101' 학생의 연락처가 변경된 경우

이상현상

- 기본키는 회원번호+수강과목

회원번호	수강과목	전화
01	D1	111-1111
01	S1	111-1111
02	D1	222-2222
03	D1	333-3333
03	S1	333-3333
04	B1	444-4444
04	D1	444-4444
04	S1	444-4444

1) 삽입이상

- (회원번호=5, 전화번호=999-9999) 삽입

2) 삭제이상

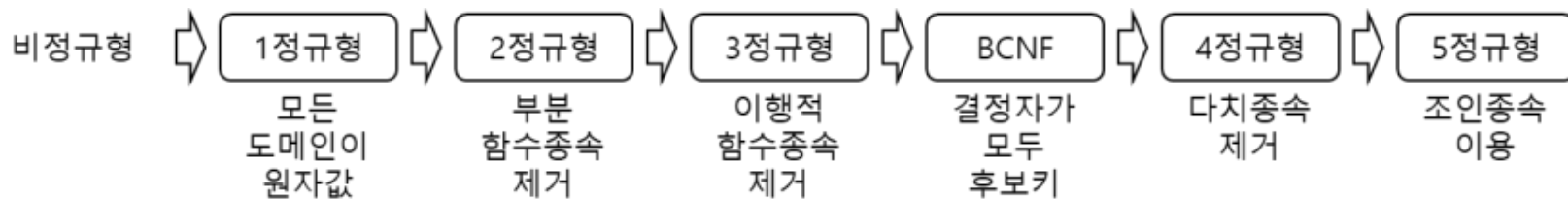
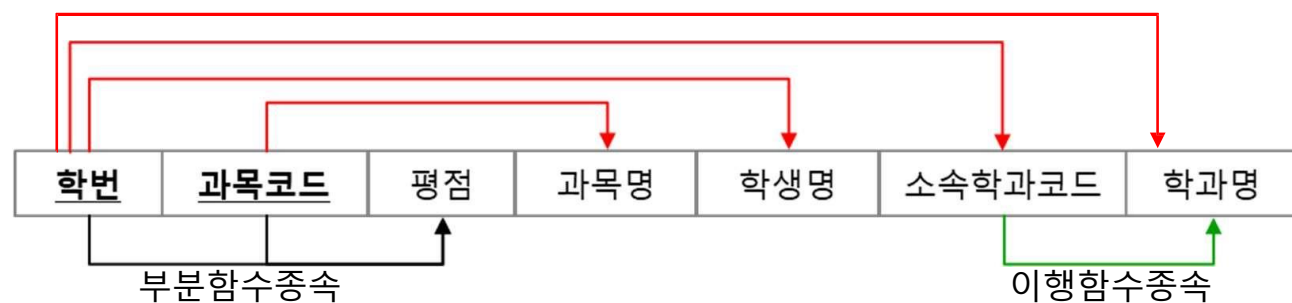
- (회원번호=2, 수강과목=D1) 삭제

3) 갱신이상

- (회원번호=4, 전화번호=000-0000)수정

● 정규화

- 목적: 삽입/삭제/갱신 이상현상 방지
- 함수적 종속성(FD, Functional Dependency)에 기반



➤ 종류

- 1NF: 모든 값이 원자값을 가짐
- 2NF: 부분함수종속 제거
- 3NF: 이행함수종속 제거

- ① 제1정규형

- 도메인들의 값이 원자값(atomic value)만으로 되어 있는 릴레이션이다.

이름	주문일	도서명
홍길동	08/15	가시나무
		누구나 생각
김하나	0516	세상 친절

정규화

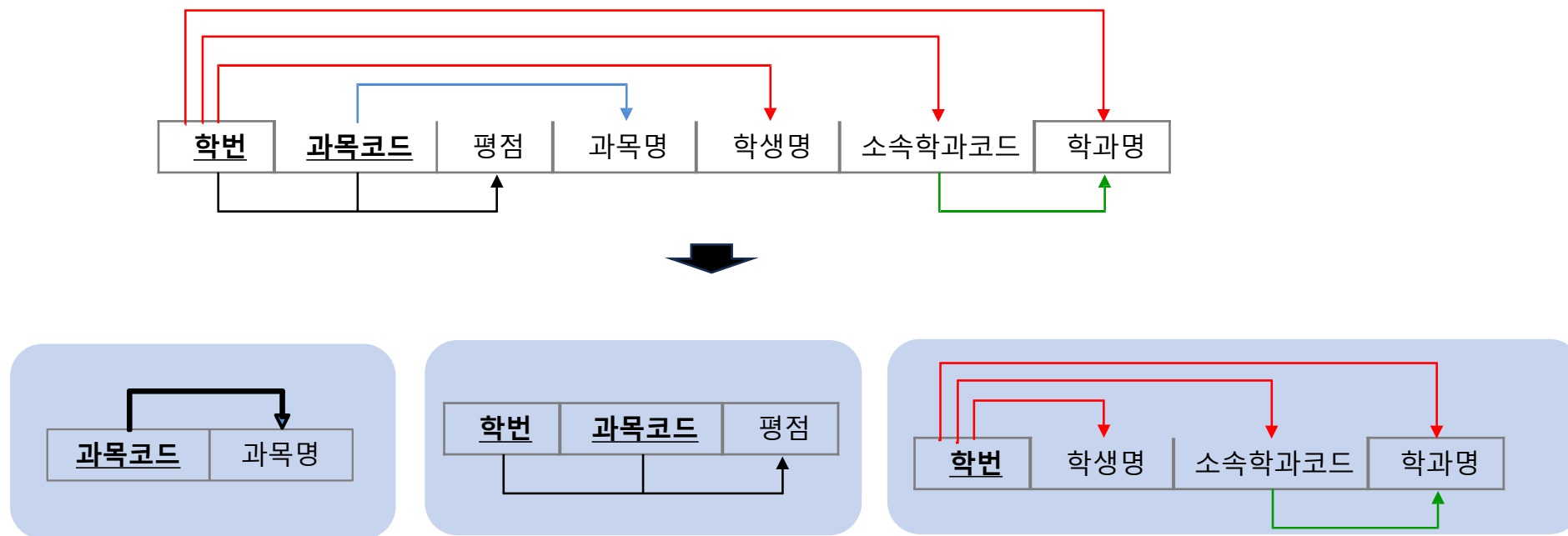


이름	주문일	도서명
홍길동	08/15	가시나무
홍길동	08/15	누구나
홍길동	08/15	생각
김하나	0516	세상
김하나	0516	친절

- 제2정규화

- 2차 정규화를 통해 2NF를 생성하는 예

- 부분함수종속 제거



• ② 제2정규형

- 부분적 함수 종속 제거 -> 모든 속성들이 기본키에 완전 함수적 종속
- 부분 함수 종속:
 - 기본키를 구성하는 부분 속성만으로도 결정되어지면 부분 함수 종속.
 - (학번, 과목) -> 교수 but (과목)-> 교수가 존재하면 부분 함수 종속.
- 완전 함수 종속:
 - 기본키 전체로만 속성이 결정될 때 완전 함수 종속.
 - (학번, 과목) -> 점수는 성립하지만 (학번)->점수이나, (과목)->(점수)은 성립하지 않으므로 완전 함수 종속.

학번	과목	교수	점수
100	정보통신	정민기	90
100	웹디자인	홍길동	80
103	정보통신	정민기	86
103	웹디자인	홍길동	95
104	사이버안보	정수현	88

부분 함수 종속



학번	과목	점수
100	정보통신	90
100	웹디자인	80
103	정보통신	86
103	웹디자인	95
104	사이버안보	88

완전 함수 종속

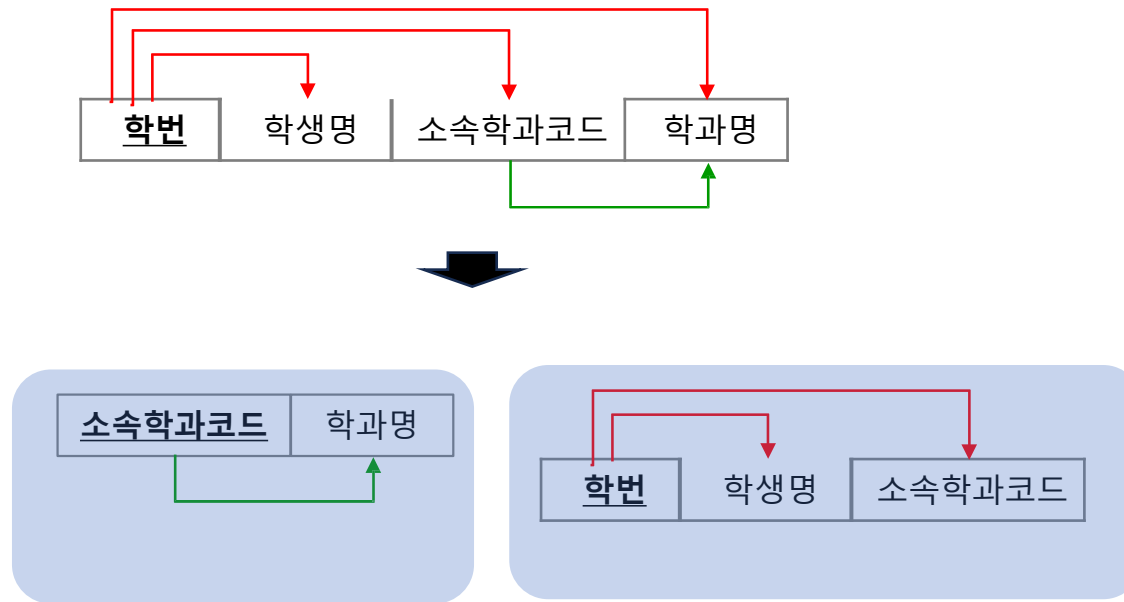
과목	교수
정보통신	정민기
웹디자인	홍길동
사이버안보	정수현

- ③ 제3정규형

- 이행함수종속제거

- 이행함수종속 : $A \rightarrow B$ 이고 $B \rightarrow C$ 이면 $A \rightarrow C$ 인 관계

- 기본키이외의 속성이 다른 속성을 결정하는 함수 종속을 제거



- ④ BCNF(Boyce-Codd Normalization, 보이스-코드 정규화)
 - 릴레이션의 모든 결정자가 후보키이다.
- ⑤ 제4정규형
 - 릴레이션 R에서 다치종속 $A \twoheadrightarrow B$ 가 성립하는 경우
R의 모든 속성이 A에 함수 종속인 릴레이션이다.
- ⑥ 제5정규형
 - 모든 조인 종속성이 후보키를 통해서만 성립되는 릴레이션이다.

정규화 예시1

- 제1정규화

[주문서]

주문번호	아이디	이름	주문일	도서명	출판사	정가	수량	금액
0235001	kbs01	홍길동	08/15	가시나무	대한출판	8,000	4	32,000
				누구나	기리나라	8,000	5	40,000
				생각	아름출판	7,000	3	21,000
0235002	mbc01	김하나	0516	세상	마고나	7,000	5	35,000
				친절	아름출	8,000	5	40,000
0235003	sbs01	이태원	0817	나무	알컷	7,000	3	21,000
				파도	알사모	8,000	5	40,000
				친절	아름출	8,000	3	24,000

- 제1정규화 :
각 필드의 값들이 단일값

주문번호	아이디	이름	주문일	도서명	출판사	정가	수량	금액
0235001	kbs01	홍길동	08/15	가시나무	대한출판	8,000	4	32,000
0235001	kbs01	홍길동	08/15	누구나	기리나라	8,000	5	40,000
0235001	kbs01	홍길동	08/15	생각	아름출판	7,000	3	21,000
0235002	mbc01	김하나	0516	세상	마고나	7,000	5	35,000
0235002	mbc01	김하나	0516	친절	아름출	8,000	5	40,000
0235003	sbs01	이태원	0817	나무	알컷	7,000	3	21,000
0235003	sbs01	이태원	0817	파도	알사모	8,000	5	40,000
0235003	sbs01	이태원	0817	친절	아름출	8,000	3	24,000

주문번호	번호	아이디	이름	주문일	도서명	출판사	정가	수량	금액
0235001	1	kbs01	홍길동	08/15	가시나무	대한출판	8,000	4	32,000
0235001	2	kbs01	홍길동	08/15	누구나	기리나라	8,000	5	40,000
0235001	3	kbs01	홍길동	08/15	생각	아름출판	7,000	3	21,000
0235002	1	mbc01	김하나	08/16	세상	마고나	7,000	5	35,000
0235002	2	mbc01	김하나	08/16	친절	아름출	8,000	5	40,000
0235003	1	sbs01	이태원	08/17	나무	알컷	7,000	3	21,000
0235003	2	sbs01	이태원	08/17	파도	알사모	8,000	5	40,000
0235003	3	sbs01	이태원	0817	친절	아름출	8,000	3	24,000

정규화 예시1

• 제2정규화

- 부분함수종속제거
- 기본키(주문번호, 번호)
- '아이디', '이름', '주문일' 필드값은 기본키 필드중에서 '주문번호' 필드에 종속적인 관계에 있다.
- 0235003주문번호는 하나의 아이디, 이름, 주문일로 처리할 수 있으므로 종속적이다.

주문번호	번호	아이디	이름	주문일	도서명	출판사	정가	수량	금액
0235001	1	kbs01	홍길동	08/15	가시나무	대한출판	8,000	4	32,000
0235001	2	kbs01	홍길동	08/15	누구나	기리나라	8,000	5	40,000
0235001	3	kbs01	홍길동	08/15	생각	아름출판	7,000	3	21,000
0235002	1	mbc01	김하나	08/16	세상	마고나	7,000	5	35,000
0235002	2	mbc01	김하나	08/16	친절	아름출	8,000	5	40,000
0235003	1	sbs01	이태원	08/17	나무	알컨	7,000	3	21,000
0235003	2	sbs01	이태원	08/17	파도	알사모	8,000	5	40,000
0235003	3	sbs01	이태원	0817	친절	아름출	8,000	3	24,000

[주문서 테이블]

주문번호	번호	도서명	출판사	정가	수량	금액
0235001	1	가시나무	대한출판	8,000	4	32,000
0235001	2	누구나	기리나라	8,000	5	40,000
0235001	3	생각	아름출판	7,000	3	21,000
0235002	1	세상	마고나	7,000	5	35,000
0235002	2	친절	아름출	8,000	5	40,000
0235003	1	나무	알컨	7,000	3	21,000
0235003	2	파도	알사모	8,000	5	40,000
0235003	3	친절	아름출	8,000	3	24,000

[주문정보 테이블]

주문번호	아이디	이름	주문일
0235001	kbs01	홍길동	08/15
0235002	mbc01	김하나	08/16
0235003	sbs01	이태원	08/17

정규화 예시1

- 3정규화
 - 이행적 함수 종속 제거

[주문서 테이블]

주문번호	번호	도서명	출판사	정가	수량	금액
0235001	1	가시나무	대한출판	8,000	4	32,000
0235001	2	누구나	기리나라	8,000	5	40,000
0235001	3	생각	아름출판	7,000	3	21,000
0235002	1	세상	마고나	7,000	5	35,000
0235002	2	친절	아름출	8,000	5	40,000
0235003	1	나무	알컴	7,000	3	21,000
0235003	2	파도	알사모	8,000	5	40,000
0235003	3	친절	아름출	8,000	3	24,000

[주문정보 테이블]

주문번호	아이디	이름	주문일
0235001	kbs01	홍길동	08/15
0235002	mbc01	김하나	08/16
0235003	sbs01	이태원	08/17

주문서 테이블

주문번호	번호	제품코드	수량	금액
0235001	1	1	4	32,000
0235001	2	2	5	40,000
0235001	3	3	3	21,000
0235002	1	4	5	35,000
0235002	2	5	5	40,000
0235003	1	6	3	21,000
0235003	2	7	5	40,000
0235003	3	5	3	24,000

도서 테이블

제품코드	도서명	출판사	정가
1	가시나무	대한출판	8,000
2	누구나	기리나라	8,000
3	생각	아름출판	7,000
4	세상	마고나	7,000
5	친절	아름출	8,000
6	나무	알컴	7,000
7	파도	알사모	8,000

[주문정보 테이블]

주문번호	아이디	이름	주문일
0235001	kbs01	홍길동	08/15
0235002	mbc01	김하나	08/16
0235003	sbs01	이태원	08/17

정규화 예시1

• 원본

주문번호	아이디	이름	주문일	도서명	출판사	정가	수량	금액
0235001	kbs01	홍길동	08/15	가시나무	대한출판	8,000	4	32,000
				누구나	기리나라	8,000	5	40,000
				생각	아름출판	7,000	3	21,000
0235002	mbc01	김하나	0516	세상	마고나	7,000	5	35,000
				친절	아름출	8,000	5	40,000
0235003	sbs01	이태원	0817	나무	알컨	7,000	3	21,000
				파도	알사모	8,000	5	40,000
				친절	아름출	8,000	3	24,000

• 제1정규형

주문번호	아이디	이름	주문일	도서명	출판사	정가	수량	금액
0235001	kbs01	홍길동	08/15	가시나무	대한출판	8,000	4	32,000
0235001	kbs01	홍길동	08/15	누구나	기리나라	8,000	5	40,000
0235001	kbs01	홍길동	08/15	생각	아름출판	7,000	3	21,000
0235002	mbc01	김하나	0516	세상	마고나	7,000	5	35,000
0235002	mbc01	김하나	0516	친절	아름출	8,000	5	40,000
0235003	sbs01	이태원	0817	나무	알컨	7,000	3	21,000
0235003	sbs01	이태원	0817	파도	알사모	8,000	5	40,000
0235003	sbs01	이태원	0817	친절	아름출	8,000	3	24,000

주문번호	번호	아이디	이름	주문일	도서명	출판사	정가	수량	금액
0235001	1	kbs01	홍길동	08/15	가시나무	대한출판	8,000	4	32,000
0235001	2	kbs01	홍길동	08/15	누구나	기리나라	8,000	5	40,000
0235001	3	kbs01	홍길동	08/15	생각	아름출판	7,000	3	21,000
0235002	1	mbc01	김하나	08/16	세상	마고나	7,000	5	35,000
0235002	2	mbc01	김하나	08/16	친절	아름출	8,000	5	40,000
0235003	1	sbs01	이태원	08/17	나무	알컨	7,000	3	21,000
0235003	2	sbs01	이태원	08/17	파도	알사모	8,000	5	40,000
0235003	3	sbs01	이태원	0817	친절	아름출	8,000	3	24,000

• 제2정규형

[주문서 테이블]

주문번호	번호	도서명	출판사	정가	수량	금액
0235001	1	가시나무	대한출판	8,000	4	32,000
0235001	2	누구나	기리나라	8,000	5	40,000
0235001	3	생각	아름출판	7,000	3	21,000
0235002	1	세상	마고나	7,000	5	35,000
0235002	2	친절	아름출	8,000	5	40,000
0235003	1	나무	알컨	7,000	3	21,000
0235003	2	파도	알사모	8,000	5	40,000
0235003	3	친절	아름출	8,000	3	24,000

[주문정보 테이블]

주문번호	아이디	이름	주문일
0235001	kbs01	홍길동	08/15
0235002	mbc01	김하나	08/16
0235003	sbs01	이태원	08/17

• 제3정규형

주문서 테이블

주문번호	번호	제품코드	수량	금액
0235001	1	1	4	32,000
0235001	2	2	5	40,000
0235001	3	3	3	21,000
0235002	1	4	5	35,000
0235002	2	5	5	40,000
0235003	1	6	3	21,000
0235003	2	7	5	40,000
0235003	3	5	3	24,000

도서 테이블

제품코드	도서명	출판사	정가
1	가시나무	대한출판	8,000
2	누구나	기리나라	8,000
3	생각	아름출판	7,000
4	세상	마고나	7,000
5	친절	아름출	8,000
6	나무	알컨	7,000
7	파도	알사모	8,000

[주문정보 테이블]

주문번호	아이디	이름	주문일
0235001	kbs01	홍길동	08/15
0235002	mbc01	김하나	08/16
0235003	sbs01	이태원	08/17

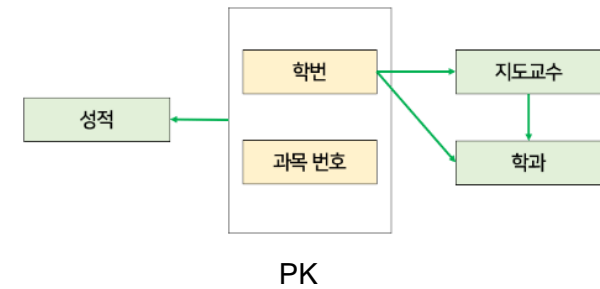
정규화(예시2)

- 원본

학번	과목번호	성적	지도교수	학과
100	C123,C234	A,B	K1	경영학
200	C123	B	K2	경영학
300	C400	A	K3	전자
400	C500	C	K4	영어

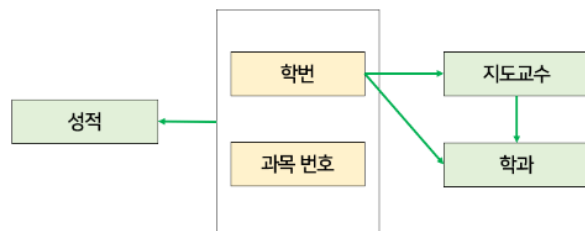
- 1정규화

학번	과목번호	성적	지도교수	학과
100	C123	A	K1	경영학
100	C234	B	K1	경영학
200	C123	B	K2	경영학
300	C400	A	K3	전자
400	C500	C	K4	영어

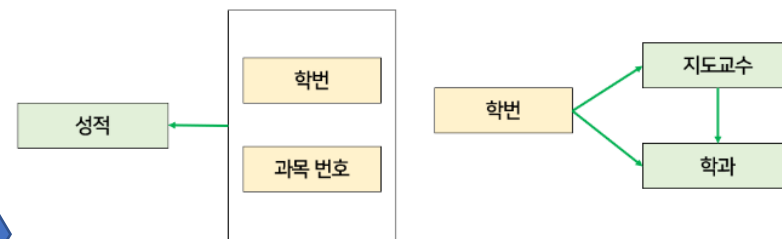


정규화(예시2)

•2정규화



학번	과목번호	성적	지도교수	학과
100	C123	A	K1	경영학
100	C234	B	K1	경영학
200	C123	B	K2	경영학
300	C400	A	K3	전자
400	C500	C	K4	영어

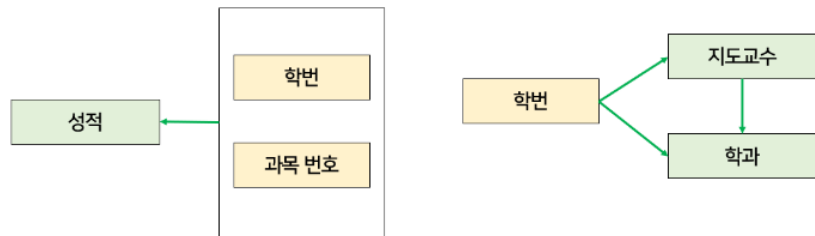


학번	과목번호	성적
100	C123	A
100	C234	B
200	C123	B
300	C400	A
400	C500	C

학번	지도교수	학과
100	K1	경영학
100	K1	경영학
200	K2	경영학
300	K3	전자
400	K4	영어

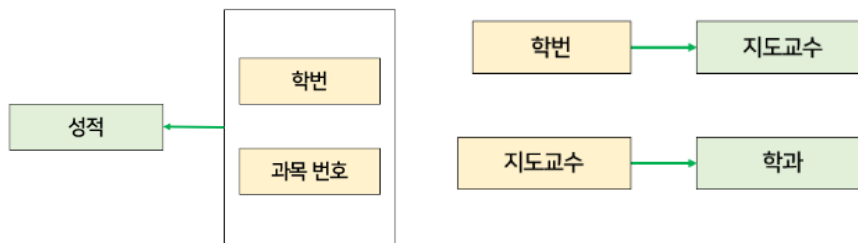
정규화(예시2)

• 3정규화



학번	과목코드	성적
100	C123	A
100	C234	B
200	C123	B
300	C400	A
400	C500	C

학번	지도교수	학과
100	K1	경영학
100	K1	경영학
200	K2	경영학
300	K3	전자
400	K4	영어



학번	과목코드	성적
100	C123	A
100	C234	B
200	C123	B
300	C400	A
400	C500	C

학번	지도교수
100	K1
100	K1
200	K2
300	K3
400	K4

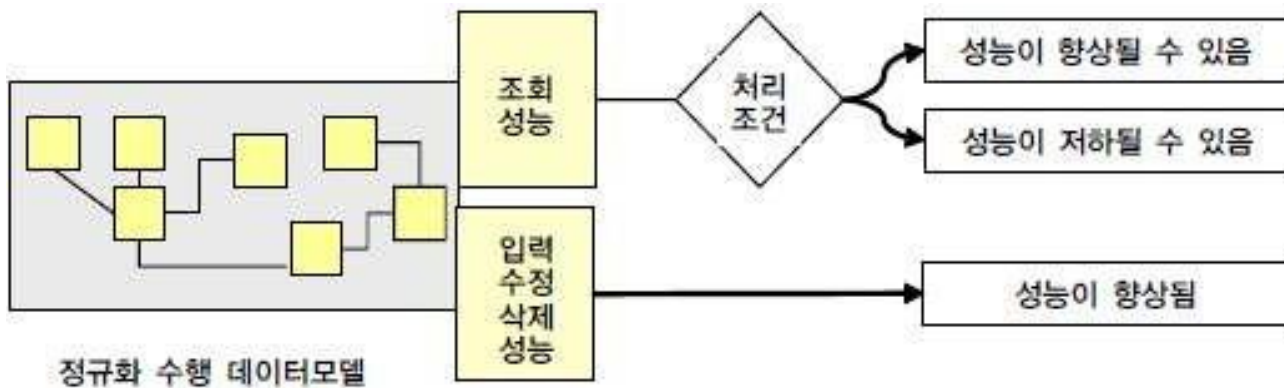
지도교수	학과
K1	경영학
K1	경영학
K2	경영학
K3	전자
K4	영어

- 정규화

- 효과

- 데이터 중복 감소 → 성능 향상
 - 데이터가 관심사별로 묶임 → 성능 향상
 - 조회 질의에서 조인이 많이 발생 → 성능 저하

- → 정규화를 통해 일반적으로 성능이 향상되나,
조회시 경우 처리 조건에 따라 성능이 향상되거나 저하되는 경우가 있을 수 있다.



- 정규화를 통한 성능 개선 예



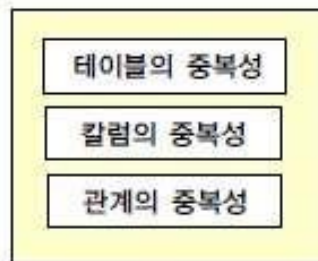
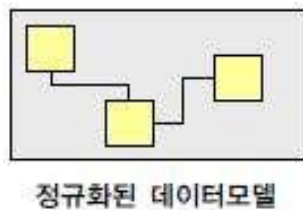
- 특정 학생의 학년 변경시
 - 정규화 전: 해당 학생이 수강 테이블에서 나타나는 수만큼 변경
 - 정규화 후: 해당 학생의 학년 한 건만 변경
- 3학년 학생의 학번, 이름 조회시
 - 정규화 전: 수강 이력이 누적된 수만큼 인스턴스 조회
 - 정규화 후: 학생 수만큼 인스턴스 조회 (성능 향상)
- 학번, 학생명, 과목명, 평점 조회시
 - 정규화 전: 하나의 테이블에서 모두 조회
 - 정규화 후: 조인 필요 (약간의 성능 저하)

- 반정규화의 정의

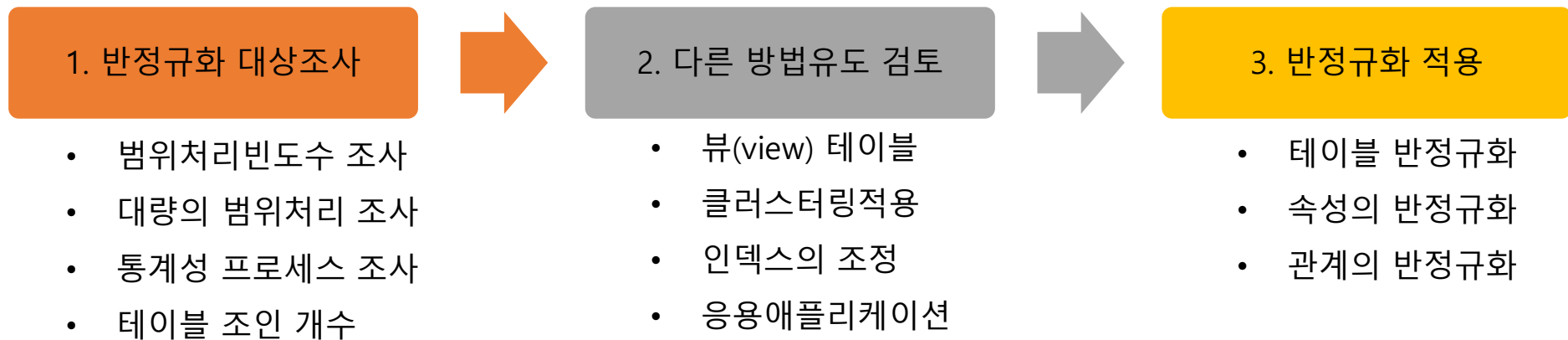
- 반정규화(=역정규화 = Denormalization)
- 정규화된 엔터티, 속성, 관계에 대해 성능 향상을 목적으로 **중복, 통합, 분리**를 수행하는 데이터 모델링 기법
 - cf) 비정규화: 정규화를 아예 수행하지 않음

- 특징

- 테이블, 칼럼, 관계의 반정규화를 종합적으로 고려해야 함
 - 일반적으로 속성(칼럼)의 중복을 시도함
- 과도한 반정규화 → 데이터 무결성을 침해하게 됨



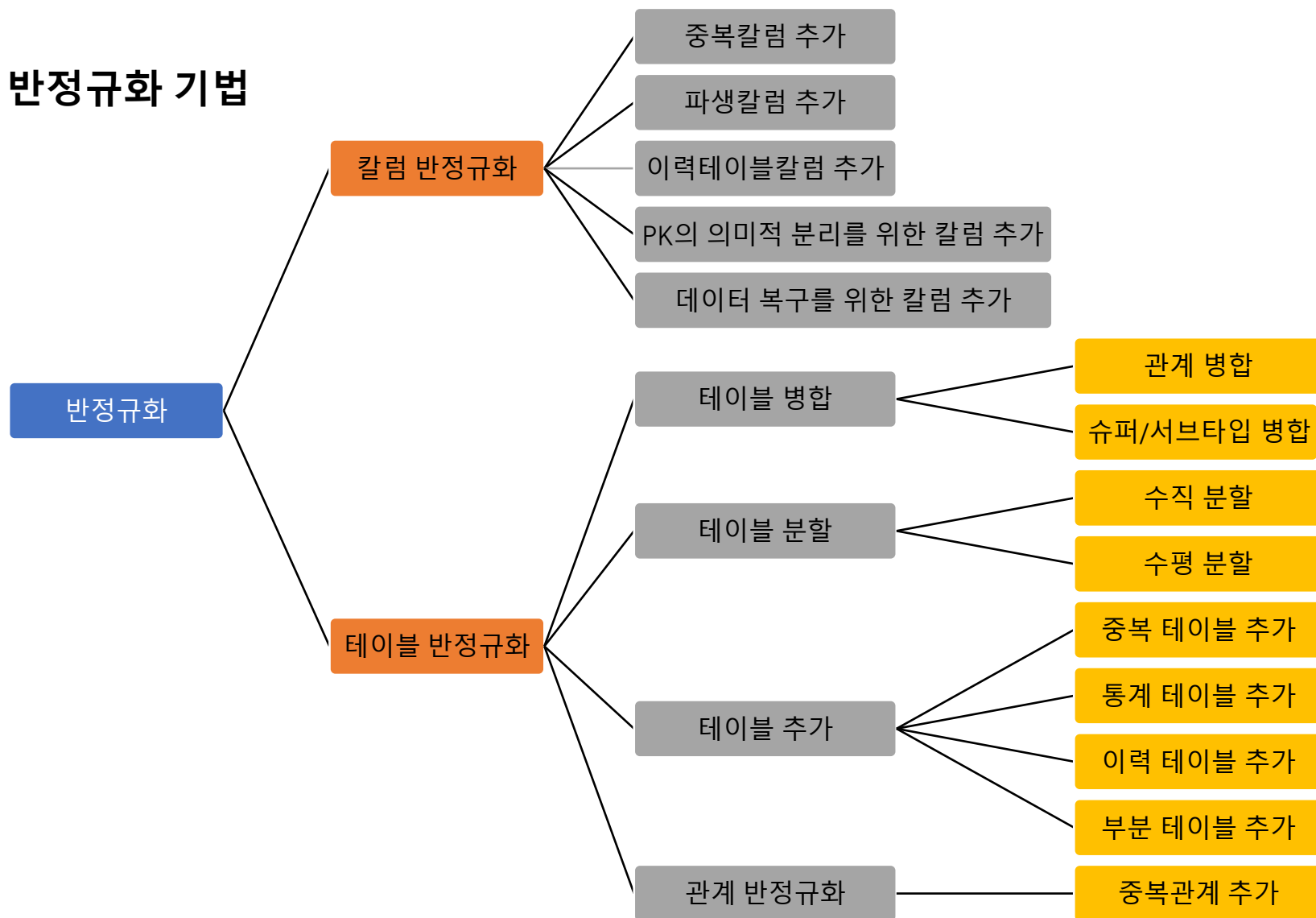
- 반정규화의 사전 절차



- 다른 방법 유도

- 뷰(View) 생성: 뷰 자체가 성능 향상을 가져오지는 않음, **신중하게 설계된 뷰**를 재사용할 때 성능 향상
- 클러스터링: 자주 사용되는 테이블의 데이터를 디스크의 같은 블록에 저장
- 인덱스의 조정: 인덱스 추가, 삭제 및 순서 조정
- 응용 애플리케이션: 데이터 처리를 위한 로직 변경

● 반정규화 기법



반정규화(칼럼)

- 반정규화 기법 - 칼럼 반정규화

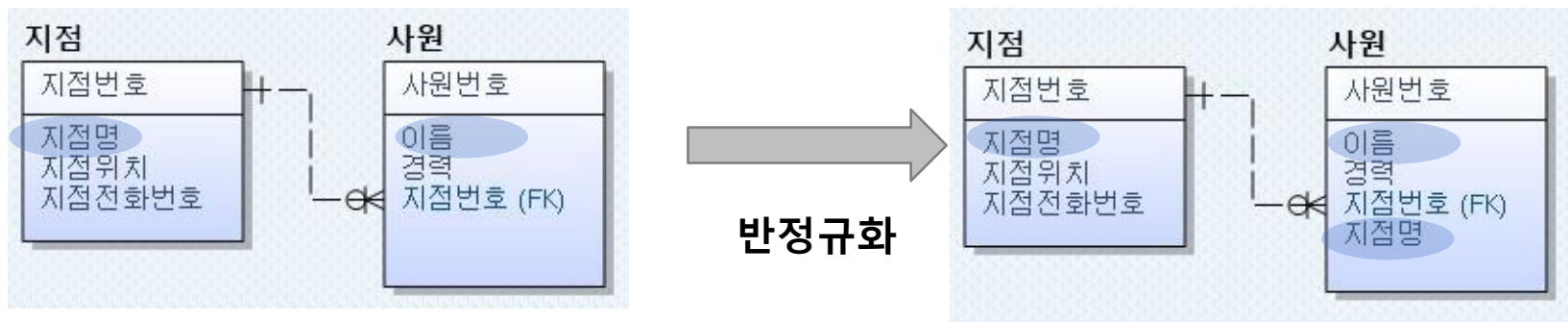
반정규화 기법	내용
중복칼럼 추가	조인 횟수를 감소시키기 위해, 다른 테이블의 칼럼을 중복으로 저장함
파생칼럼 추가	값의 계산으로 인한 시간 지연을 줄이기 위해, 예상되는 값을 미리 계산하여 중복으로 저장함 (Derived Attribute)
이력 테이블 칼럼 추가	대량 이력 데이터 처리의 성능 향상을 위해 종료 여부, 최근값 여부 등의 칼럼을 추가로 저장함
PK의 의미적 분리를 위한 칼럼 추가	PK가 복합 의미를 갖는 경우 구성 요소 값의 조회 성능 향상을 위해 일반 속성을 추가함 (예: 차량번호가 '지역'+ '일련번호'로 구성된 경우 '지역' 일반속성 추가)
데이터 복구를 위한 칼럼 추가	사용자의 실수 또는 응용프로그램 오류로 인해 데이터가 잘못 처리된 경우, 원래 값으로의 복구를 위해 이전 데이터를 임시적으로 중복 저장

반정규화(칼럼)

● 중복칼럼 추가

– 해당 테이블에서 자주 사용하는 칼럼을 중복시킴

- 예: 각 사원의 이름과 소속지점명을 조회하는 질의가 많은 경우



• 중복 속성을 추가하는 경우

- 조인이 자주 발생하는 속성인 경우
- 접근 경로가 복잡한 속성인 경우
- 액세스의 조건으로 자주 사용되는 속성인 경우
- 기본키의 형태가 적절하지 않거나 여러 개의 속성으로 구성된 경우

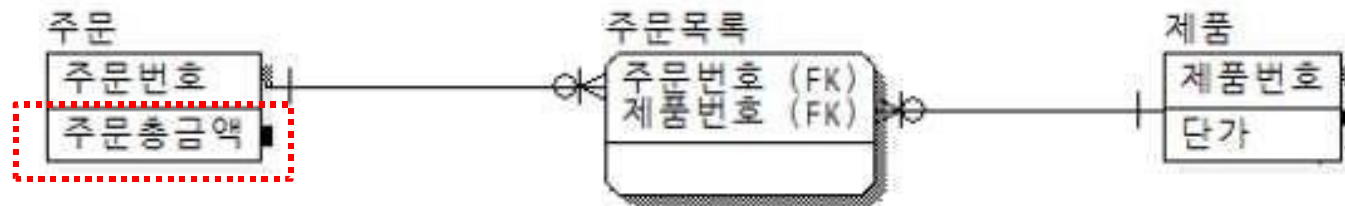
• 중복 속성 추가 시 고려 사항

- 테이블 중복과 속성의 중복을 고려한다.
- 데이터 일관성 및 무결성에 유의해야 한다.
- 저장 공간의 지나친 낭비를 고려한다.

반정규화(칼럼)

● 파생칼럼 추가

- 질의가 예상되는 값을 미리 계산하여 저장함
- 칼럼에 의한 파생



- 로우에 의한 파생
 - 특정 로우가 변경되면 모든 로우 값 다시 계산 → 가급적 사용하지 않아야 함

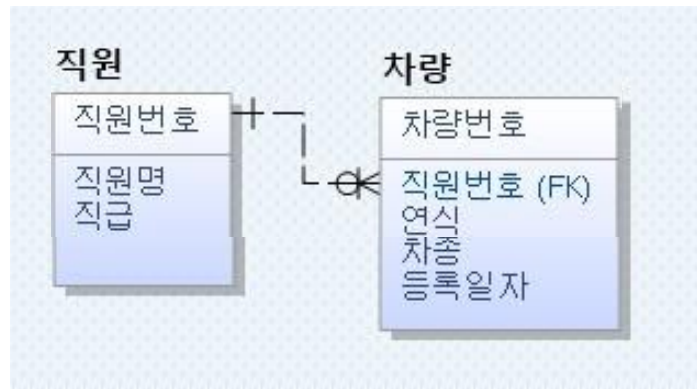
공사비	
공사번호	
공사일자	
공사비누적	

공사번호	공사일자	공사비	공사비누적
2001-001	2001년01월01일	7500000원	7500000원
2001-001	2001년03월05일	2000000원	9500000원
2001-001	2001년07월01일	9721239원	19221239원

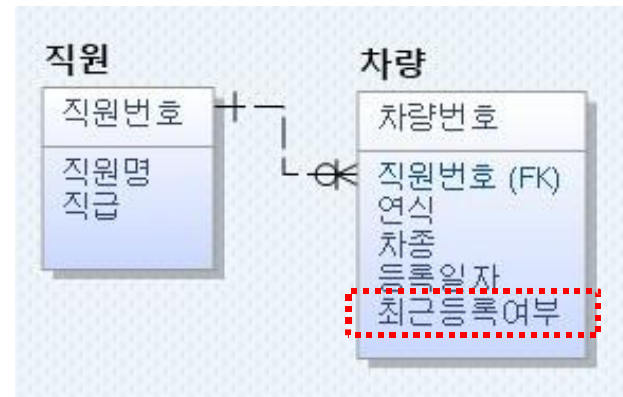
반정규화(칼럼)

- 이력테이블칼럼 추가

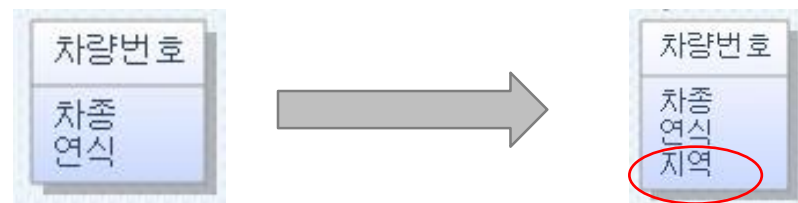
- 이력테이블에 최근값, 종료여부 등의 칼럼을 중복으로 추가
 - 예: 직원이 가장 최근 등록한 차량을 조회는 경우



반정규화

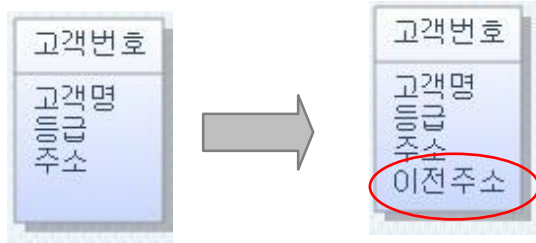


- PK의 의미적 분리를 위한 칼럼 추가
 - 차량번호 “서울 가1234”와 같이 ‘지역’ + ‘일련번호’로 구성된 경우
 - ‘서울’ 지역으로 등록된 차량만 조회하려면?
 - PK를 활용할 수 없음 → 일반 속성 추가



- 데이터 복구를 위한 칼럼 추가

- 사용자의 실수 또는 응용프로그램의 오류로 인한 데이터 손실 대비
- 고객 정보 중 이전 주소 추가



반정규화(테이블)

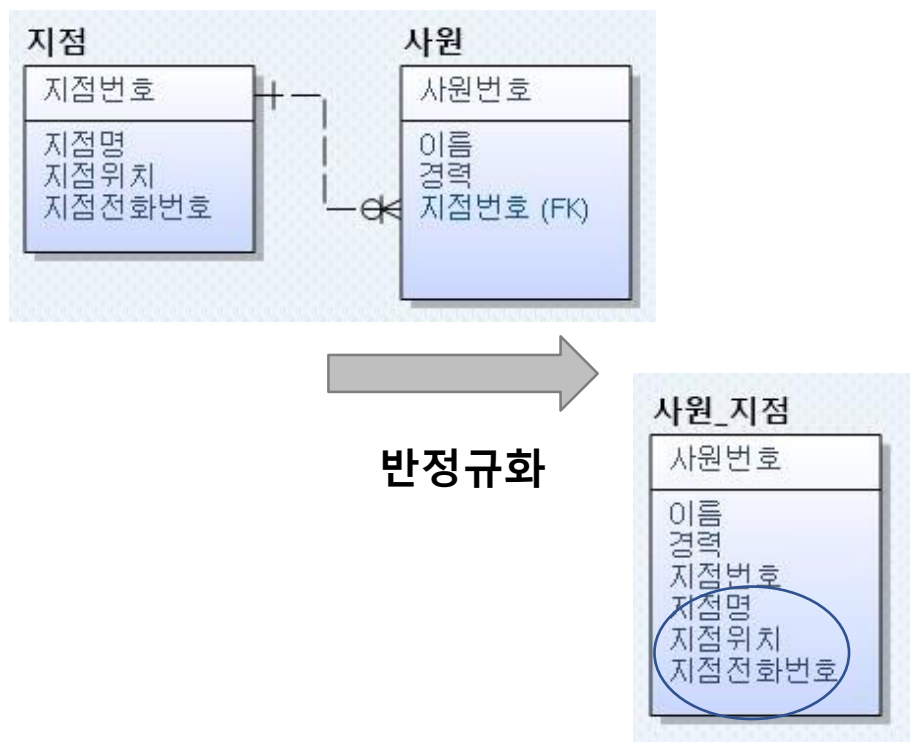
- 반정규화 기법 - 테이블 반정규화

분류	기법	내용
테이블병합	관계 병합	1:1 또는 1:M 관계를 병합함 (두 테이블의 동시 조회가 많은 경우)
	슈퍼/서브타입 병합	슈퍼/서브타입 관계를 병합함 (one to one type → single type / plus type)
테이블분할	수직 분할	디스크 I/O의 분산을 위해 테이블을 칼럼(Column) 단위로 분리함
	수평 분할	디스크 I/O의 분산을 위해 테이블을 로우(Row) 단위로 분리함
테이블추가	중복 테이블 추가	원격 조인(다른 업무 또는 다른 서버 간 조인)을 제거하기 위해 동일한 테이블 구조를 중 복사킴 (분산 DB 참고)
	통계 테이블 추가	SUM, AVG 등의 통계값을 미리 계산하여 저장 (분산 DB 참고)
	이력 테이블 추가	이력 테이블 중 일부 레코드를 마스터 테이블에서 중복 관리
	부분 테이블 추가	하나의 테이블에서 집중적으로 이용되는 칼럼들만을 추출하여 별도의 테이블 생성 (테이블 수직 분할과 유사하지만, 원본 테이블을 유지하면서 추가함)

반정규화(테이블)

- 관계 병합

- 두 테이블의 동시 조회가 많은 경우

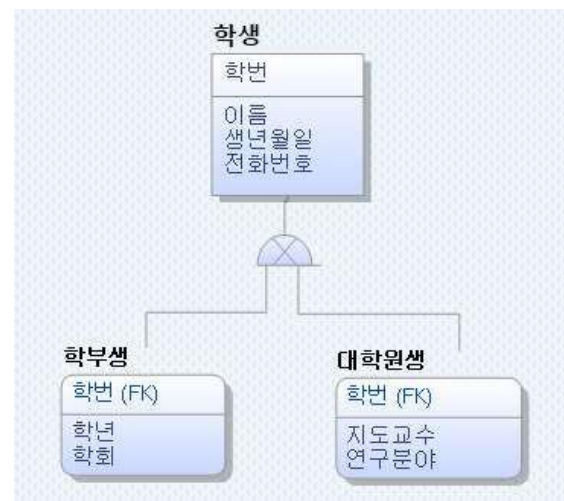


반정규화(테이블)

- 슈퍼/서브타입 모델의 변환

- Super Type / Sub Type

- 일반화(Generalization) 관계를 표현
 - 여러 엔터티의 공통 속성을 Super Type, 개별 속성을 Sub Type으로 구성



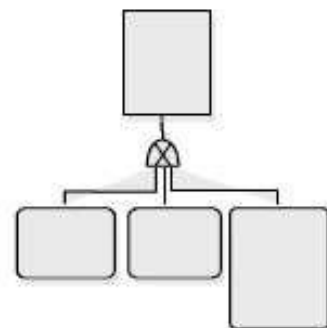
슈퍼타입 : 동물
아이디, 이름, 출생일, 성별, 무게

서브타입 : 강아지
아이디, 품종, 예방접종, 훈련수준

서브타입 ; 고양이
아이디, 품종, 무게, 털의 유형

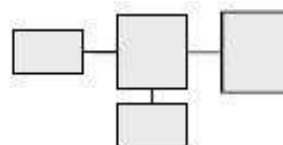
- 데이터의 양 및 트랜잭션의 유형에 따라 테이블 구조 결정

슈퍼/서브타입 모델



기준은?

One to One Type



1:1 타입

Plus Type



슈퍼+서브 타입

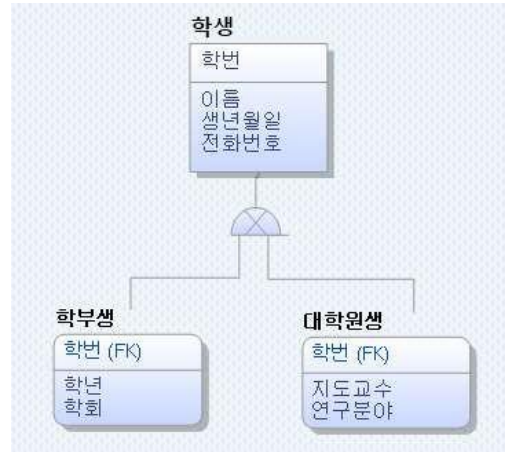
Single Type



All in One 타입

반정규화(테이블)

- 슈퍼/서브타입 모델의 변환
 - 개별 접근 트랜잭션이 많은 경우
 - One to One Type 으로 변환



파일(F) 편집(E) 보기(V) 즐겨찾기(A) 도구(T) 도움말(H)

x [변환] [선택]

☆

기본정보

학번 1234 이름 홍길동

생년월일 20001122 전화번호 010-1111-2222

상세정보조회

파일(F) 편집(E) 보기(V) 즐겨찾기(A) 도구(T) 도움말(H)

x [변환] [선택]

☆

상세정보

학번 1234

지도교수 강감찬

연구분야 Data Science

학번 (FK)

지도교수

연구분야

반정규화(테이블)

- 슈퍼/서브타입 모델의 변환

- 슈퍼타입+서브타입 접근 트랜잭션이 많은 경우

- → **Plus Type** 으로 변환

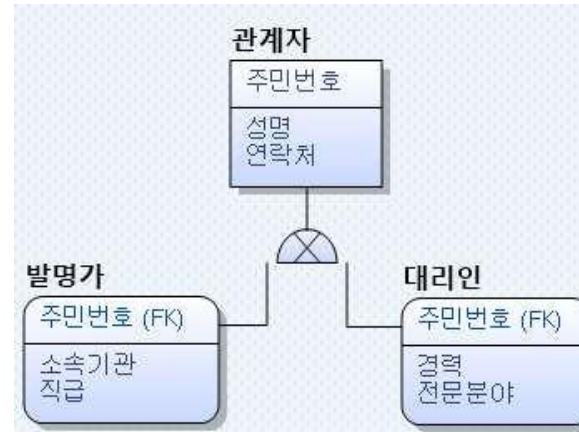
- 예: 대학원생 100명, 학부생 10만명에서 대학원생 정보 조회가 빈번한 경우

파일(F)	편집(E)	보기(V)	즐거찾기(A)	도구(T)	도움말(H)
X	변환	선택			
학번	1234	이름	홍길동		
생년월일	20001122	전화번호	010-1111-2222		
지도교수	강감찬	연구분야	Data Science		

반정규화(테이블)

● 슈퍼/서브타입 모델의 변환

- 여러 서브타입에 대한 동시 접근이 많을 때
 - All in One Type으로 변환



- 슈퍼/서브타입 모델 변환 타입의 비교

구분	One To One Type	Plus Type	Single Type
특징	개별 테이블 유지	슈퍼+서브타입 테이블	하나의 통합 테이블
확장성	우수함	보통	나쁨
조인 필요 수	많음	보통	적음
I/O 성능저하	양호	양호	나쁨
관리 용이성	나쁨	나쁨	좋음
적합 트랜잭션 유형	개별 테이블로 접근이 많은 경우	슈퍼+서브 형식 데이터 처리가 많은 경우	전체에 대한 일괄 처리가 많은 경우

반정규화(테이블)

● 테이블 분할 절차

- 데이터 모델링 수행
- 데이터베이스 용량 산정
- 대량 데이터가 처리되는 테이블에 대해 트랜잭션 처리 패턴 분석
- 트랜잭션이 칼럼 단위로 집중되는 경우 수직 분할, 로우 단위로 집중되는 경우 수평 분할 수행

<수직 분할>

사원번호	이름	전공
201001	홍길동	정보처리
a201002	유관순	네트워크
a201003	윤봉길	인공지능
a201004	홍길동	정보처리
a201005	이순신	정보처리
a201006	유관순	네트워크

사원번호	이름
a201001	홍길동
a201002	유관순
a201003	윤봉길
a201004	홍길동
a201005	이순신
a201006	유관순

사원번호	전공
a201001	정보처리
a201002	네트워크
a201003	인공지능
a201004	정보처리
a201005	정보처리
a201006	네트워크

<수평 분할>

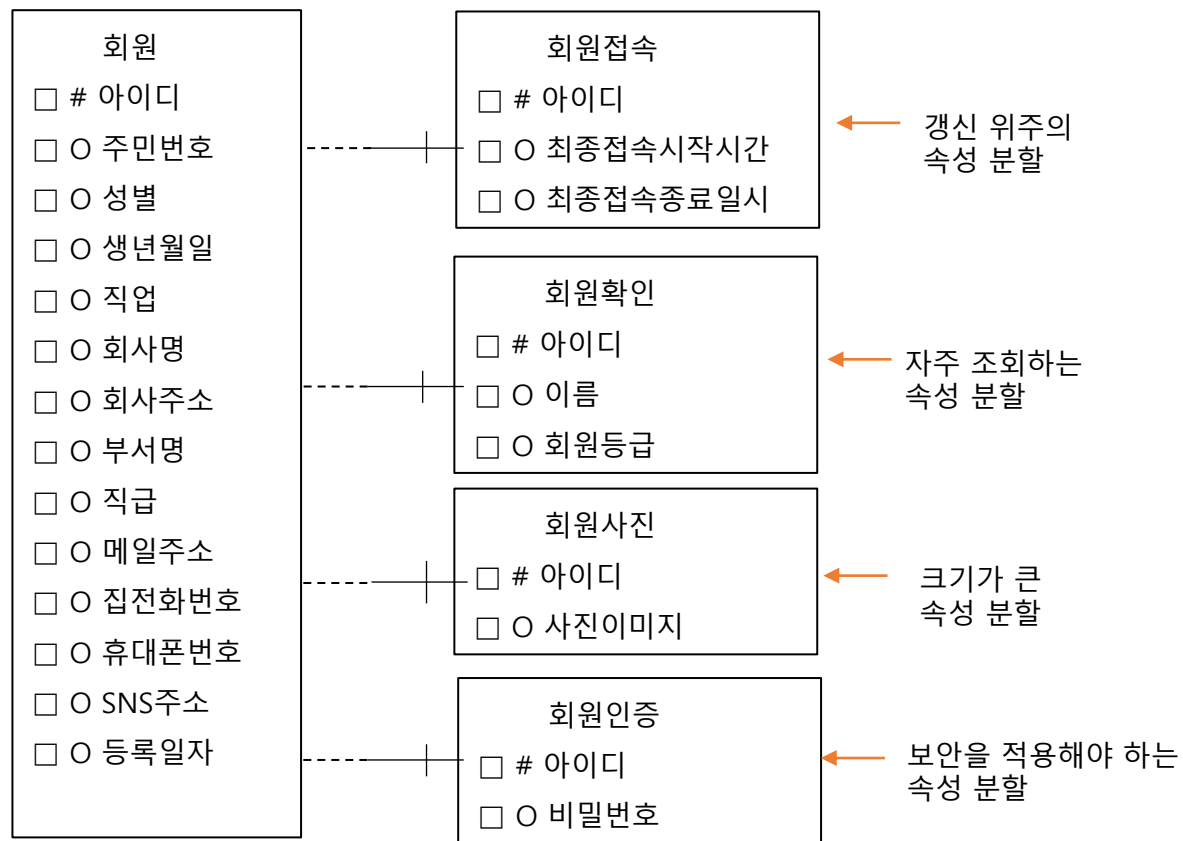
사원번호	이름	전공
a201001	홍길동	정보처리
a201002	유관순	네트워크
a201003	윤봉길	인공지능
a201004	홍길동	정보처리
a201005	이순신	정보처리
a201006	유관순	네트워크

사원번호	이름	전공
a201001	홍길동	정보처리
a201002	유관순	네트워크
a201003	윤봉길	인공지능

사원번호	이름	전공
a201004	홍길동	정보처리
a201005	이순신	정보처리
a201006	유관순	네트워크

반정규화(테이블)

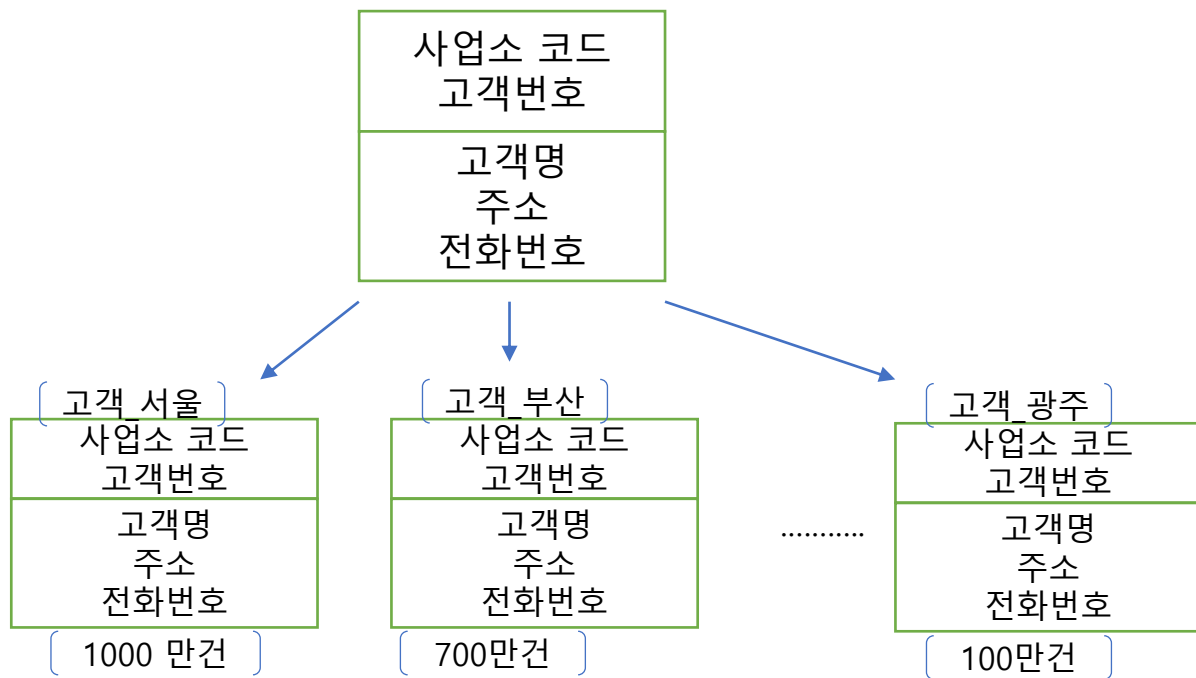
- 테이블 수직 분할(Vertical Partitioning)
 - 하나의 테이블에 속성이 너무 많을 경우 속성을 기준으로 테이블을 분할
 - 갱신 위주의 속성 분할 :
 - 갱신이 자주 일어나는 속성들을 수직 분할
 - 자주 조회되는 속성 분할 :
 - 테이블에서 자주 조회되는 속성이 극히 일부일 경우 자주 사용되는 속성들을 수직 분할
 - 크기가 큰 속성 분할 :
 - 이미지나 2GB 이상 저장될 수 있는 텍스트 형식 등으로 된 속성들을 수직 분할
 - 보안을 적용해야 하는 속성 분할 :
 - 테이블 내의 특정 속성에 대해 보안을 적용할 수 없으므로 보안을 적용해야 하는 속성들을 수직 분할



반정규화(테이블)

- 테이블 수평 분할

- 테이블이 많은 양의 데이터를 가질 것으로 예상되는 경우 → Partitioning
 - Range Partition: 범위로 분할 (고객번호: 1~1000, 1001~2001 등)
 - List Partition: 값으로 분할 - (지역: 서울, 대구, 부산 등)
 - Hash Partition: 해시 함수로 분할



- 해시 함수(Hash Function)
 - 임의의 길이의 데이터를 짧은 길이의 데이터로 매핑하는 함수

123456 → 16

203050 → 20

765321 → 71

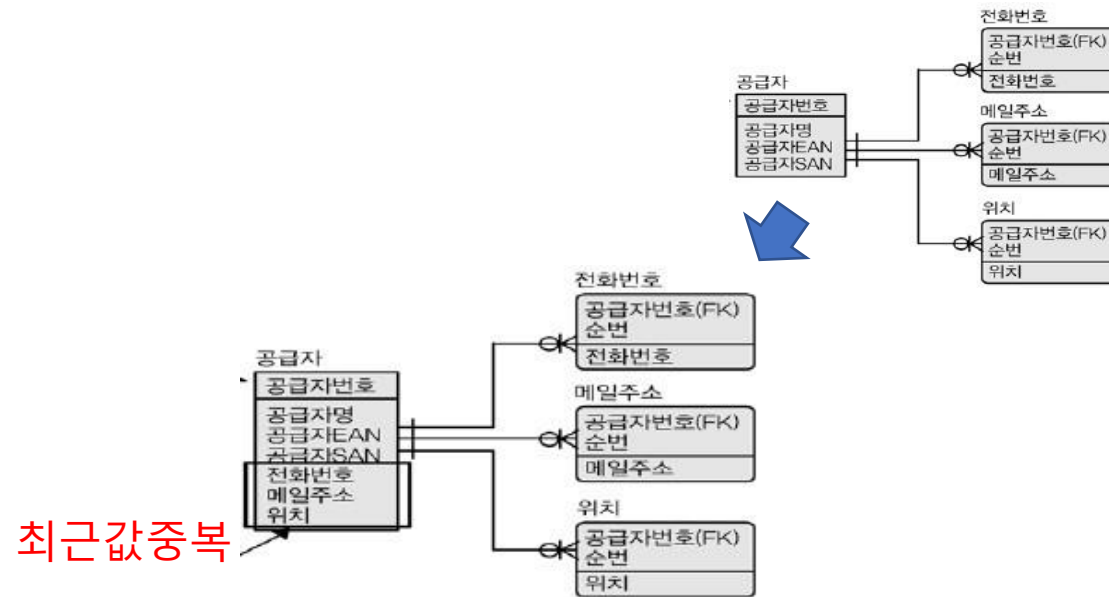
740721 → 71

반정규화(테이블)

- 이력테이블 **추가**

- 공급자번호 1001 ~ 1005의 최근 전화번호, 메일주소, 위치 조회?

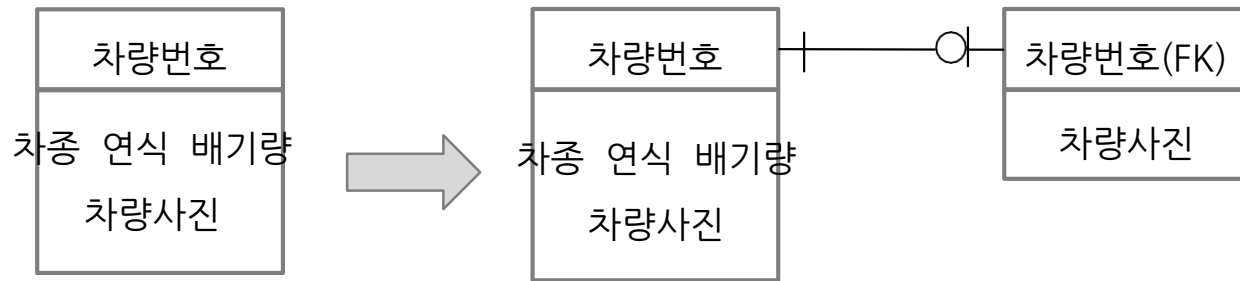
- 가장 최근값을 중복으로 기록한 테이블 생성



반정규화(테이블)

- 부분테이블 추가

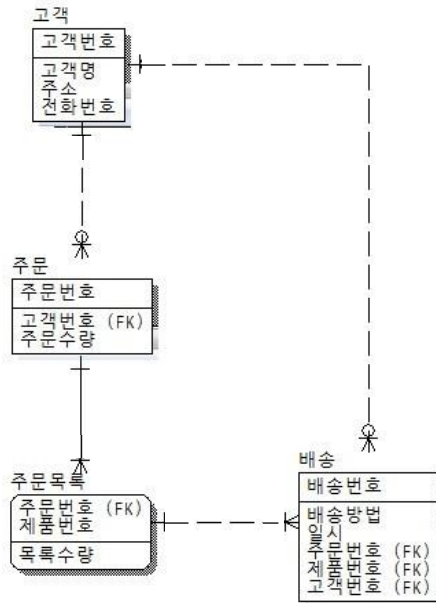
- 집중적으로 자주 사용되는 특정 속성들을 추출하여 별도 테이블 구성
 - 예) 차량번호에 대한 차량사진 조회 질의가 많은 경우
- 테이블 수직 분할과 유사하지만 원본 테이블이 그대로 유지됨



반정규화(테이블)

- 반정규화 기법 - 관계 반정규화

반정규화 기법	내용
<u>중복관계추가</u>	조인을 통해 정보 조회가 가능하지만, 조인 경로 단축을 위해 중복관계를 추가함



특정배송건의고객명을 알고자하는경우!

반정규화(테이블)

- **인덱스(Index)**

- Select 학번, 이름, 학년

- from 학생

- where 학번 > 3000 and 학번 < 5000;

→ 해당되는 학생은 누구인가?

학생

학번	이름	학년
1122	강감찬	2
2233	김유신	2
2211	홍길동	3
3321	을지문덕	3
2354	유관순	4
5434	이순신	1
2324	이성계	1
4321	단군	2

반정규화(테이블)

● 인덱스(Index)

Index

Index Key	Row ID
1122	
2211	
2233	
2324	
2354	
3321	
4321	
5434	

학생

학번	이름	학년
1122	강감찬	2
2233	김유신	2
2211	홍길동	3
3321	을지문덕	3
2354	유관순	4
5434	이순신	1
2324	이성계	1
4321	단군	2

- 검색 속도의 향상을 위한 기술 → 실제 테이블을 Full Scan하지 않고 **인덱스 테이블을 검색**
- 지나치게 많은 인덱스 생성시 시간 및 공간 낭비
- 인덱스된 필드의 업데이트시 시간 증가
- 자동 생성(PK 또는 Unique 조건) / 수동 생성(Create Index 구문)

1) 제산 방법(division)

- $H(k) = k \bmod m$ ($H(k)$: 홈 주소, k : 키 값, m : 버킷의 수에서 가장 가까운 소수)

2) 기수(radix) 변환법

- 주어진 키 값을 다른 진법의 수로 판단

3) 중간 제곱 함수(mid-square)

- 키 값을 제곱한 후 중간에 정해진 자리 수 만큼을 취함
- $H(k) = k^2$

4) 숫자 분석(digit analysis) 방법

- 필요한 자리만큼 선택하여 홈 주소를 결정하는 방식

5) 중첩(Folding)

- 키값을 버킷 주소 크기 만큼 분할한 후,
분할한 것을 더하거나 XOR 연산하여
버킷의 주소를 만드는 방법.(주소의 크기를 벗어나는 것은 버림)

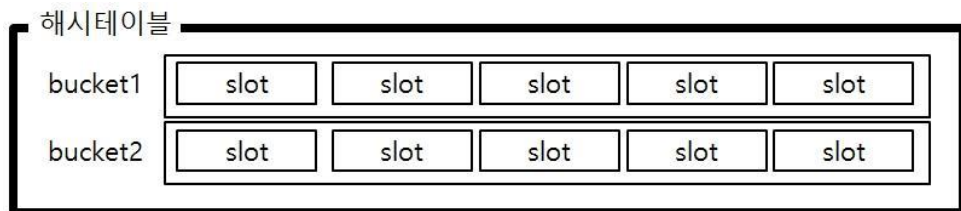
키	값	주	소
02 - 835 - 2316		2	3 6
02 - 836 - 3518		3	5 8
02 - 836 - 6435		6	4 5
02 - 835 - 2719		2	7 9
02 - 835 - 3113		3	1 3
02 - 835 - 1221		1	2 1

- 용어

- 버킷(Bucket) : 하나의 주소를 갖는 파일의 한 구역을 의미, 버킷의 크기는 같은 주소에 포함될 수 있는 레코드 수.
- 슬롯(Slot) : 1개의 레코드를 저장할 수 있는 공간으로, n개의 슬롯이 모여 하나의 버킷을 형성.
- 충돌(collision) : 서로 다른 두 개의 key가 같은 버킷 주소를 갖게 되어 충돌하는 현상.
- 동의어(synonym) : 같은 버킷 주소를 갖는 레코드들의 집합
- 오버플로우(overflow) : 더 이상 버킷에 보관될 수 없는 현상

- 해싱의 문제점

- 서로 다른 두 개 이상의 키 값들이 해시 함수에 의해 동일한 주소로 변환되는 경우 충돌이 발생.
- 충돌의 발생이 빈번하면 발생 시간이 길어지는 등 성능이 저하
-> 해시 함수의 수정이나 해시 테이블의 크기가 적절히 조절되어야 한다.
- 일반적으로 충돌이 발생할 경우 버킷이 여러 슬롯으로 구성되어 있다면 다른 슬롯에 저장
그러나 모든 슬롯이 채워지면 오버플로우가 발생



- ① 선형 탐색법(linear probing)
 - 충돌이 발생했을 경우 다음 버킷부터 차례로 빈 버킷을 찾음,
 - 간단방법, 검색 시간 증가, 집중 현상으로 인한 성능 저하.
 - 키 k_i 의 해시 테이블 주소가 $h(k_i)$ 라 할 때 충돌 발생시
다음 버킷 주소는 $h(k_i) + 1$ 번지, $h(k_i) + 1$ 의 주소
버킷이 비어 있지 않으면 $h(k_i) + 2$ 의 순서로 $h(k_i) + n$ 의 빈 버킷을 계속 찾아 삽입
- ② 2차 탐색법(quadratic probing)
 - 선형 방법에서 발생하는 집중 문제를 해결하기 위한 방법,
특정한 수만큼 떨어진 곳을 순환적으로 찾아서 빈 공간을 찾아 저장하는 방법이다.
 - $h(k) + 1, h(k) + 2^2, h(k) + 3^2, \dots, n^2$

평가(정규화)

- 다음 <조건>에 따라 입력 키 값을 해시(hash) 테이블에 저장하였을 때
- 해시 테이블의 내용으로 옳은 것은?
- 조건
 - 해시 테이블의 크기는 7이다.
 - 해시 함수는 $h(k) = k \bmod 7$ 이다.
(단, k 는 입력 키 값이고, \bmod 는 나머지를 구하는 연산자이다)
 - 충돌은 이차 조사법(quadratic probing)으로 처리한다.
 - 키 값의 입력 순서: 9, 16, 2, 6, 20

①

0	6
1	2
2	9
3	16
4	
5	
6	20

해시
테이블

②

0	6
1	20
2	9
3	16
4	
5	
6	2

해시
테이블

③

0	20
1	
2	9
3	16
4	2
5	
6	6

해시
테이블

④

0	20
1	2
2	9
3	
4	16
5	
6	6

해시
테이블

평가(정규화)

- 다음과 같이 왼쪽 릴레이션을 오른쪽 릴레이션으로 정규화를 하였을때 어떤 정규화 작업을 한것인가?

국가	도시
대한민국	서울, 부산
미국	워싱턴, 뉴욕
중국	베이징

국가	도시
대한민국	서울
대한민국	부산
미국	워싱턴
미국	뉴욕
중국	베이징

- ① 제1정규형
- ② 제2정규형
- ③ 제3정규형
- ④ 제4정규형

수고하셨습니다 🙌