

제 3 장

관계형 데이터베이스



SQL Server

ORACLE®

목 차

3.1 데이터 모델

3.2 기존 데이터 모델

3.3 관계형 데이터 모델(Relational data model)

3.4 관계 데이터베이스 언어

3.1 데이터 모델

- 데이터 모델

- 데이터를 컴퓨터에 저장하려면, 현실세계와 그 데이터가 갖는 의미를 충분히 표현
- 데이터베이스의 구조뿐만 아니라 이런 구조에서 허용되는 연산 그리고 이런 구조와 연산에 대한 제약 조건을 포함하는 개념

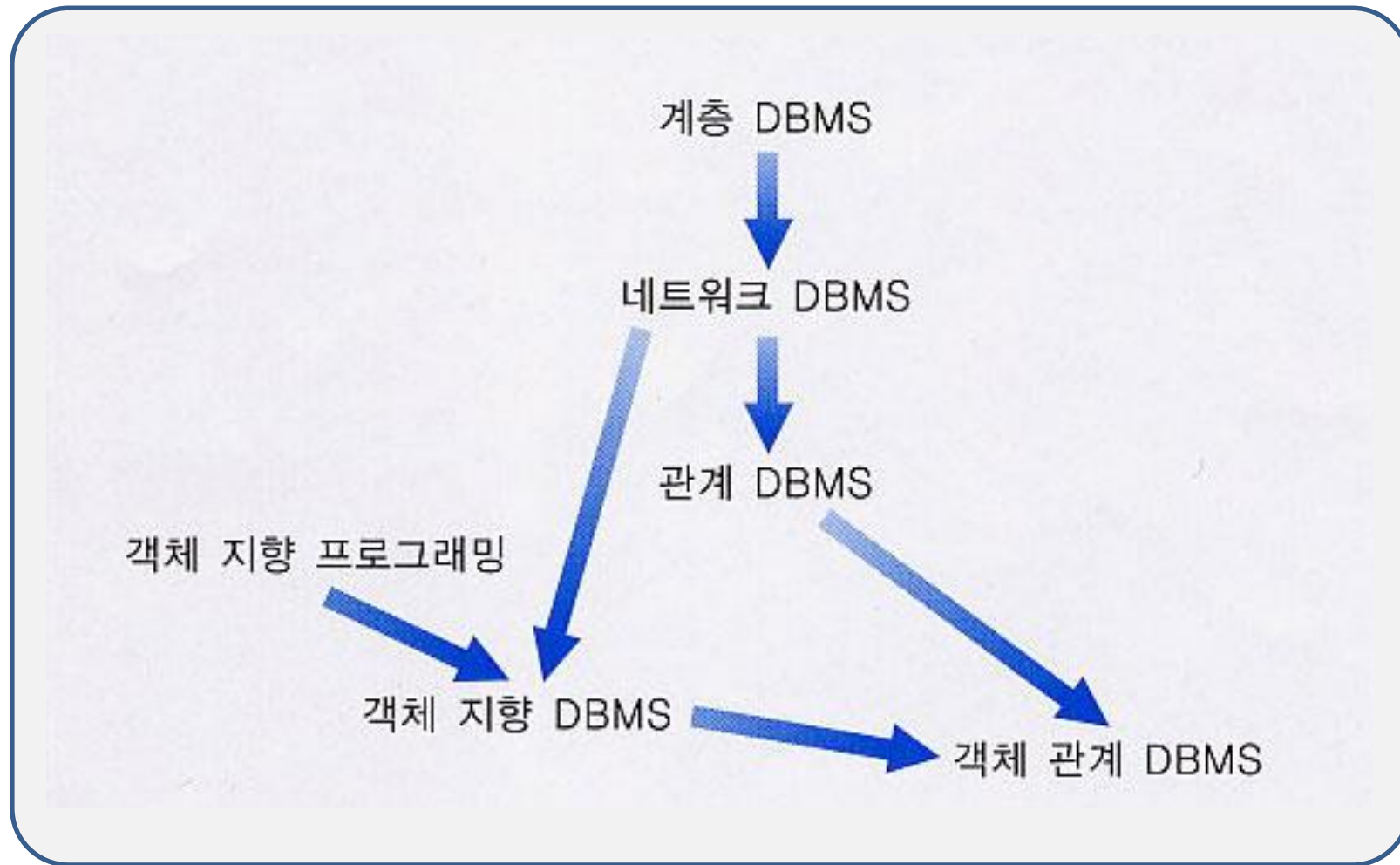
- 데이터 모델링

- 이러한 데이터 모델을 이용하여 현실 세계의 정보구조를 표현하려는 작업

- 데이터 모델

- 계층형 데이터 모델
- 네트워크형 데이터 모델
- 관계형 데이터 모델
- 객체관계형 데이터 모델

3.1 데이터 모델



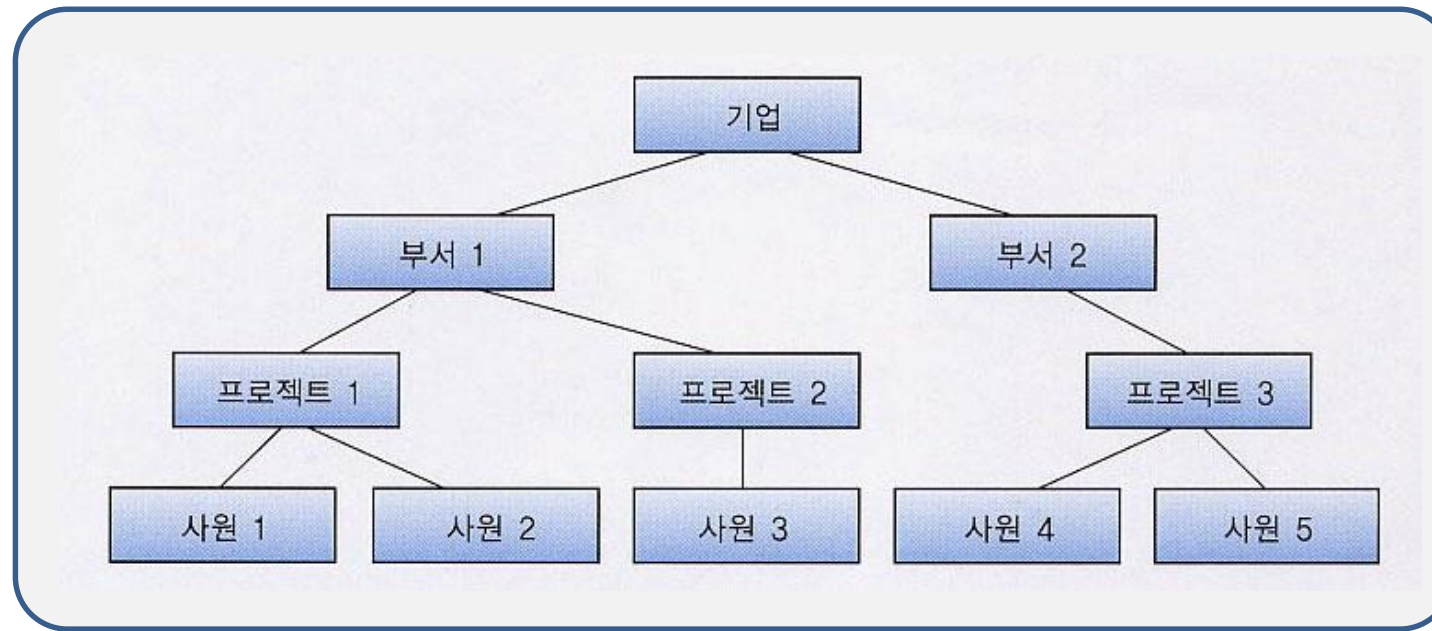
3.2 기존 데이터 모델

● 계층형 DBMS(Hierarchical DBMS)

- 1960년대 후반에 최초의 계층 DBMS가 등장(IBM사의 IMS)
- 트리 구조를 기반으로 하는 계층 데이터 모델을 사용한 DBMS
- 동일한 형태의 데이터들의 집합체를 나타내는데 세그먼트(segment)들 간의 부모/자식 관계
- 계층 데이터 모델은 네트워크 데이터 모델의 특별한 사례
- 장점
 - 어떤 유형의 응용에 대해서는 빠른 속도와 높은 효율성을 제공
- 단점
 - 어떻게 데이터를 접근하는가를 미리 응용 프로그램에 정의해야 함
 - 데이터베이스가 생성될 때 각각의 관계를 명시적으로 정의해야 함
 - 레코드들이 링크로 연결되어 있으므로 레코드 구조를 변경하기 어려움

3.2 기존 데이터 모델

- 계층형 DBMS(Hierarchical DBMS)



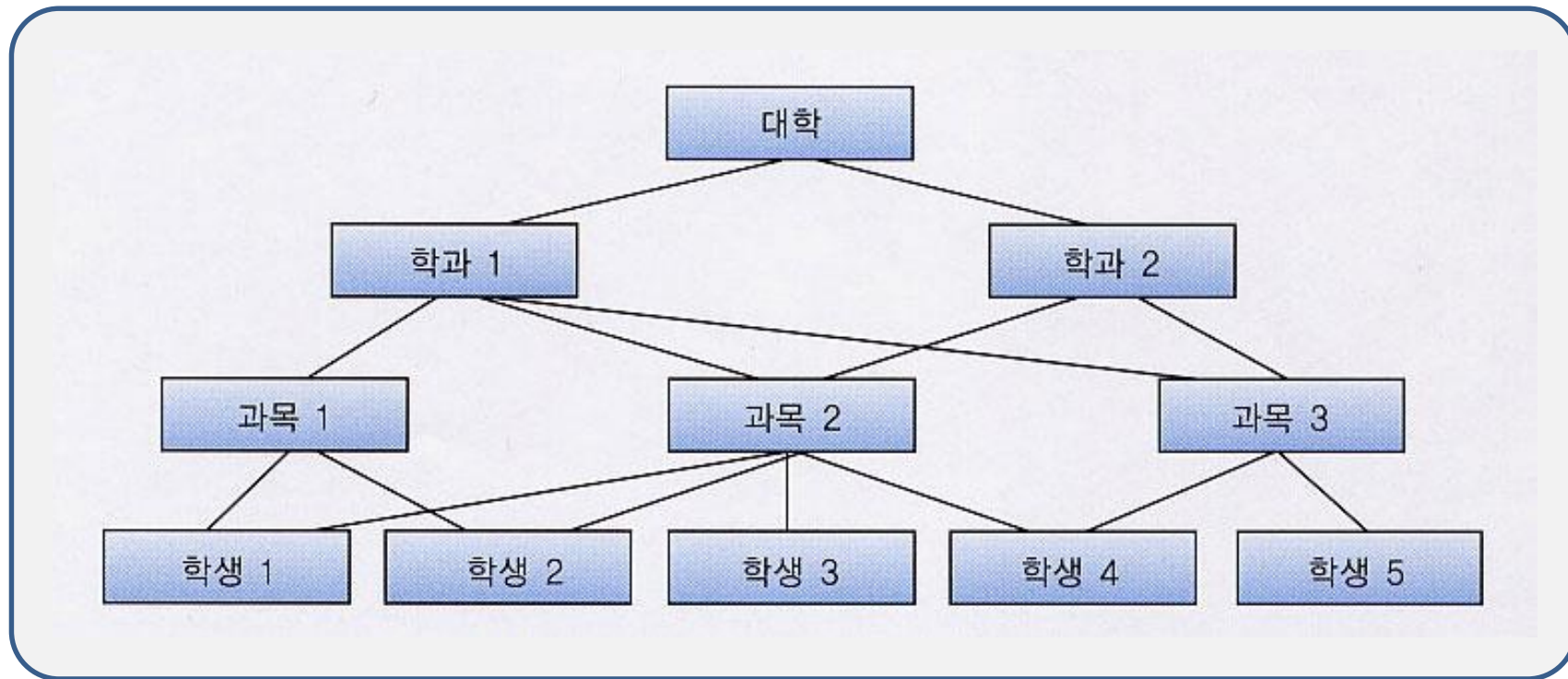
3.2 기존 데이터 모델

● 네트워크형 DBMS(Network DBMS)

- CODASYL 모델, DBTG모델
- 1960년대 초에 Charles Bachman이 하니웰(Honeywell)사에서 최초의 네트워크 DBMS인 IDS를 개발
- 레코드들이 노드로, 레코드들 사이의 관계가 간선으로 표현되는 **그래프**를 기반으로 하는 네트워크 데이터 모델을 사용
- 노드들 관계를 **오너(owner)-멤버(member)**의 관계를 표현
- 네트워크 DBMS에서도 레코드들이 링크로 연결되어 있으므로 레코드 구조를 변경하기 어려움

3.2 기존 데이터 모델

- 네트워크형 DBMS(Network DBMS)



※ 데이터 모델

● 객체 지향 DBMS

- 1980년대 후반 들어 새로운 데이터 모델인 객체 지향 데이터 모델이 등장
- 객체 지향 프로그래밍 패러다임을 기반으로 하는 데이터 모델
- 장점
 - 데이터와 프로그램을 그룹화하고, 복잡한 객체들을 이해하기 쉬우며, 유지와 변경이 용이함
- 예: ONTOS, OpenODB, GemStone, ObjectStore, Versant, O2 등

● 객체 관계 DBMS

- 1990년대 후반에 관계 DBMS에 객체 지향 개념을 통합한 객체 관계 데이터 모델이 제안됨
- 예: 오라클, Informix Universal Server 등

※ 데이터 모델

DBMS 분류

기준	종류
데이터 모델에 따른 분류	<ul style="list-style-type: none">• 계층 DBMS• 네트워크 DBMS• 관계 DBMS• 객체 지향 DBMS• 객체 관계 DBMS
사용자의 수에 따른 분류	<ul style="list-style-type: none">• 단일 사용자 DBMS (주로 PC용)• 다수 사용자 DBMS
사이트의 수에 따른 분류	<ul style="list-style-type: none">• 중앙 집중식 DBMS• 분산 DBMS
접근 방법에 따른 분류	<ul style="list-style-type: none">• 범용 DBMS• 특별한 DBMS (예: 공간 DBMS)

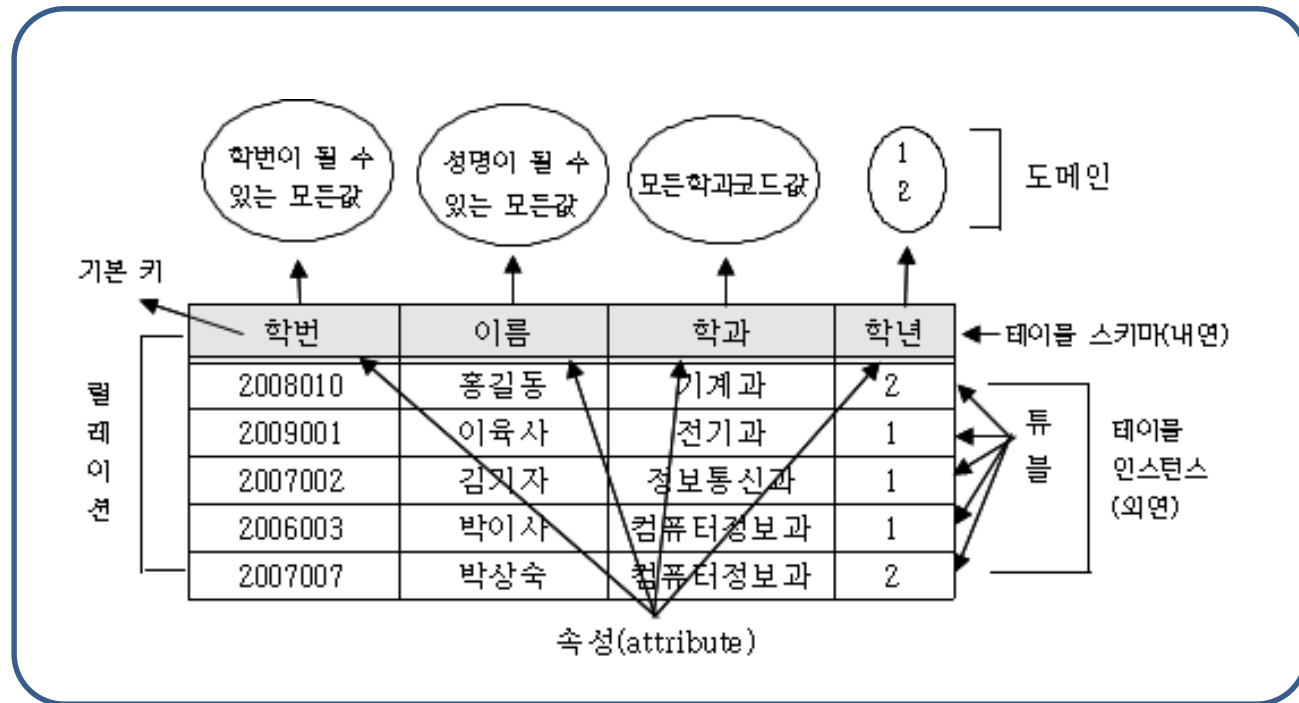
3.3 관계형 데이터 모델(relational data model)

- 관계형 DBMS

- 미국 IBM 연구소에서 진행된 “System R”
- 캘리포니아 버클리대에서 진행된 Ingres 프로젝트
- 1970년에 E.F. Codd가 IBM 연구소에서 관계 데이터 모델을 제안
- 장점
 - 모델이 간단하여 이해하기 쉬움
 - 사용자는 자신이 원하는 것만 명시하고, 데이터가 어디에 있는지, 어떻게 접근해야 하는지는 DBMS가 처리함.
- 예: 오라클, MS SQL Server, Sybase, DB2, Informix 등

3.3 관계형 데이터 모델(relational data model)

- 데이터 베이스는 테이블의 집합이며, 테이블은 행과 열로 구성됨



※학생 한 사람에 대한 데이터는 튜플 또는 엔티티이고, 학번, 성명, 학과 코드, 학년은 속성이다.

3.3 관계형 데이터 모델(relational data model)

(1) 릴레이션(relation)

정보 저장의 기본 형태가 2차원 구조의 테이블로 구성됨.

(2) 튜플(tuple)

테이블의 한 행(row), n 개의 속성으로 구성된 튜플이라면 n -튜플이라함.

(3) 속성(attribute)

테이블의 각 열(column), 속성(attribute)은 릴레이션이 갖는 성질(property)을 의미하며 관계데이터 모델에서 데이터의 가장 작은 논리적인 단위

(4) 도메인(domain)

각 속성이 취할 수 있는 값의 집합

- ① 단순 도메인(simple domain)
- ② 복합 도메인(composite domain)
- ③ 다치 속성(multivalued attribute)

3.3 관계형 데이터 모델(relational data model)

(5) 릴레이션 스키마(schema)

데이터베이스에 저장될 자료들의 논리적 구조 및 관계를 의미하고 릴레이션 이름과 속성들 그리고 스키마의 무결성 제약 조건으로 구성되며 시간에 무관한 정적 성질을 갖는 릴레이션의 영구 부분이다.

엔티티 유형(type), 릴레이션 유형, 릴레이션 내포(intension)

(6) 릴레이션 인스턴스(instance)

어느 한 시점에서 릴레이션이 포함하고 있는 전체 튜플을 의미하며 시간에 따라 변화하는 동적 성질을 갖고 있다.

엔티티 집합, 릴레이션 상태(state), 릴레이션 어커런스(occurrence), 릴레이션 외연(extension), 릴레이션(relation)

(7) 릴레이션의 차수(degree)

한 릴레이션을 구성하는 속성 수를 의미한다. 학생 릴레이션의 차수는 4이다.

(8) 카디널리티(cardinality)

특정 테이블의 튜플 개수를 의미한다. 학생 릴레이션의 카디널리티는 5이다.

3.3 관계형 데이터 모델(relational data model)

형식적 언어	유사한 용어	의 미
릴레이션(relation)	테이블, 파일(file)	이차원 테이블
속성(attribute)	열(column), 필드(field) 데이터 항목(data item)	릴레이션의 열
튜플(tuple)	행(row), 레코드(record)	릴레이션의 행
도메인(domain)		속성이 취할 수 있는 값의 범위
카디널리티(cardinality)		행의 개수
릴레이션 차수(degree)	내연(intension)	열의개수
릴레이션 스키마(schema)	인스턴스, 릴레이션 상태,	스키마 이름+속성들+계약조건
릴레이션 어커런스	외연(extension)	릴레이션의 튜플 집합
키(key)또는논리적 키 (logical key)		릴레이션의 튜플을 유일하게 식별하는 하나이상 의 속성으로 구성
물리적 키 (physical key) 또는 인덱스 (index)		빠른 검색이나 신속한 순차 접근을 위해 하나 이 상의 속성으로 구성

3.3 관계형 데이터 모델(relational data model)

- 모든 릴레이션은 다음과 같은 특성을 갖는다.

(1) 튜플의 유일성

릴레이션의 인스턴스는 튜플들의 집합이고 집합은 중복된 원소를 포함하지 않으므로 릴레이션에는 중복된 튜플이 존재하지 않는다.

(2) 튜플의 무순서(위에서 아래로)

릴레이션에 있는 튜플들의 순서는 의미가 없다.

(3) 속성의 무순서(왼쪽에서 오른쪽으로)

릴레이션의 속성 사이의 순서는 의미가 없다. 즉, 하나의 튜플은 (\langle 속성이름 \rangle , \langle 값 \rangle)쌍들의 집합으로 생각할 수 있으므로 이 쌍들 사이의 순서는 의미가 없다.

(4) 속성의 원자 값

모든 속성의 값은 원자 값이다. 즉, 논리적으로 더 이상 분해 할 수 없는 값으로 처리상의 단위 값을 의미한다.

3.3 관계형 데이터 모델(relational data model)

- 릴레이션의 키

- 각 튜플을 **고유하게 식별**할 수 있는 하나 이상의 어트리뷰트들의 모임
- 슈퍼 키(super key), 후보 키(candidate key), 기본 키(primary key), 대체 키(alternate key), **외래 키(foreign key)**

- 슈퍼 키(super key)

- 한 릴레이션 내의 특정 튜플을 고유하게 식별하는 하나의 어트리뷰트 또는 어트리뷰트들의 집합
- 예: 학생 테이블에서 (학번, 이름) 또는 (학번, 학과)
- 튜플들을 고유하게 식별하는데 꼭 필요하지 않은 어트리뷰트들을 포함할 수 있음

3.3 관계형 데이터 모델(relational data model)

- 후보 키(candidate key)

- 각 튜플을 고유하게 식별하는 **최소한**의 어트리뷰트들의 모임
- 두 개 이상의 어트리뷰트로 이루어진 경우에 **복합 키**(composite key)라고 함
- 예: (학번), (주민등록번호)

학생

학번	주민등록번호	이름	학과	학년
2012010	880101-1XXXXXX	홍길동	기계과	2
2011001	910505-1XXXXXX	이육사	전기과	1
2015002	900717-2XXXXXX	김기자	정보통신과	1
2015003	891225-1XXXXXX	박이사	컴퓨터정보과	1
2014007	880408-2XXXXXX	박상숙	컴퓨터시스템과	2

3.3 관계형 데이터 모델(relational data model)

- 기본 키(primary key)

- 한 릴레이션에 후보 키가 두 개 이상 존재할 경우 이들 중에서 하나를 기본 키로 선정함
- 후보 키중 되도록 짧은 것을 기본 키로 정함
- 후보 키가 없을 경우 인위적인 키 어트리뷰트를 릴레이션에 추가할 수 있음
- 예: (학번)

학생(학번, 주민등록번호, 이름, 학과, 학년)

학생(*학번, 주민등록번호, 이름, 학과, 학년)

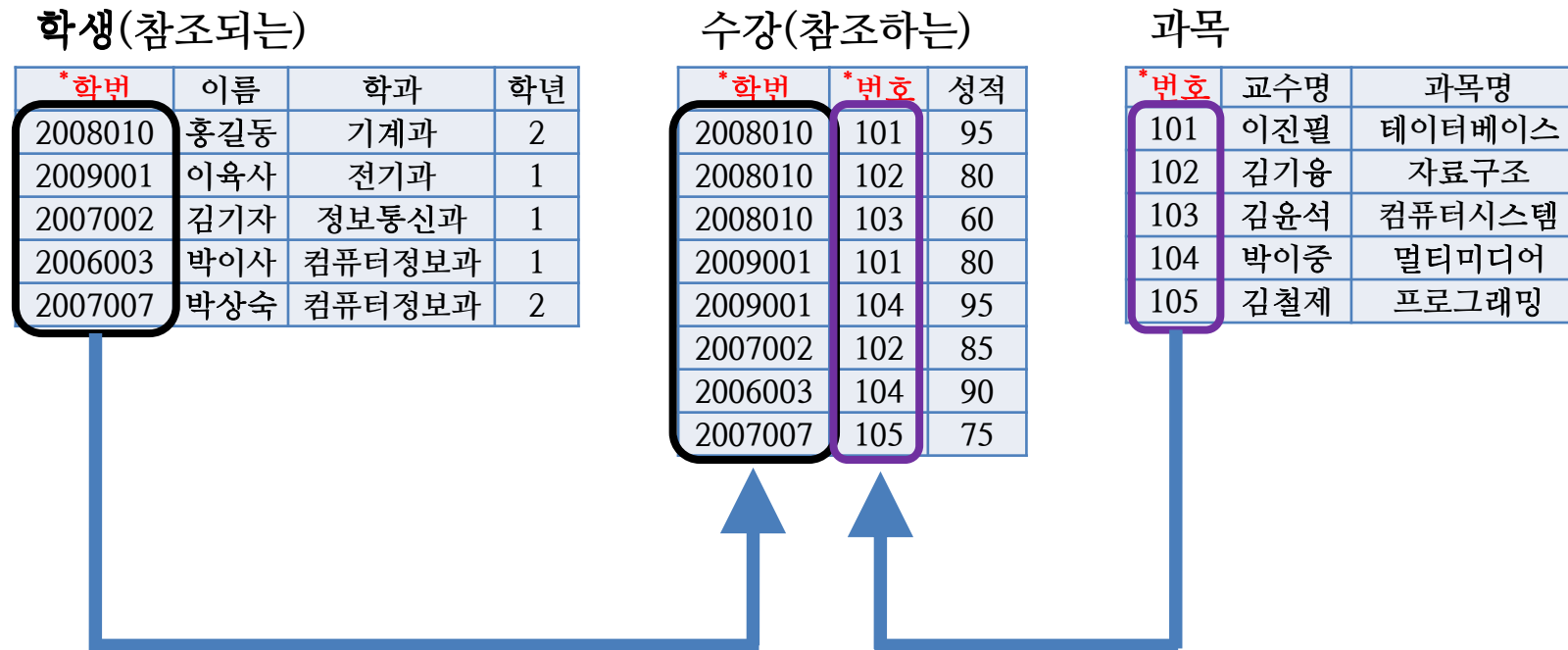
- 대체 키(alternate key)

- 기본 키가 아닌 후보 키
- 예: (주민등록번호)

3.3 관계형 데이터 모델(relational data model)

- 외래 키(foreign key)

- 어떤 릴레이션의 기본 키를 참조하는 어트리뷰트
- 외래 키 어트리뷰트는 참조되는 릴레이션의 기본 키와 동일한 도메인을 가져야 함
- 예 : 수강(학번), 수강(번호)



3.3 관계형 데이터 모델(relational data model)

● 제약 조건

- 관계형 데이터 베이스 상태(인스턴스)가 항상 만족해야 할 기본 규칙
- 데이터 베이스 인스턴스가 삽입, 삭제, 수정 등의 연산을 통해서 변하더라도 지속적으로 만족해야 할 조건

❖ 도메인 제약조건(domain constraint)

- 각 어트리뷰트 값은 반드시 원자값이어야 함.

❖ 키 제약조건(key constraint)

- 릴레이션의 모든 키는 유일한 값이어야 함

❖ 엔티티 무결성 제약조건(entity integrity constraint)

- 릴레이션의 기본키는 어떠한 경우에도 널값(null value)을 가질 수 없음

❖ 참조 무결성 제약조건(referential integrity constraint)

- 한 테이블에 있는 튜플이 다른 테이블에 있는 튜플을 참조하려면 반드시 참조되는 튜플이 그 테이블 내에 존재해야함.

3.4 관계형 데이터베이스 언어

- 관계 데이터 모델을 지원하는 두 가지 정형적인 언어

- ❖ 관계 해석(relational calculus)

- ✓ 원하는 데이터만 명시하고 질의를 어떻게 수행할 것인가는 명시하지 않는 선언적인 언어

- ❖ 관계 대수(relational algebra)

- ✓ 어떻게 질의를 수행할 것인가를 명시하는 절차적 언어

- ✓ 관계 대수는 상용 관계 DBMS들에서 널리 사용되는 SQL의 이론적인 기초

- ✓ 관계 대수는 SQL을 구현하고 최적화하기 위해 DBMS의 내부 언어로서도 사용됨

- ✓ 제한연산(RESTRICT), 프로젝트(PROJECT), 조인(JOIN), 디비전(DIVISION)

- ✓ 합집합(UNION), 교집합(INTERSECT), 차집합(DIFFEREN), 카티션프로덕트(CARTESIAN PRODUCT)

3.4 관계형 데이터베이스 언어

● 제한연산(RESTRICIT; σ)

- 특정 조건에 맞는 튜플만으로 결과 테이블을 작성함
- 수평적 부분집합
- 예 : 학생 테이블에서 컴퓨터정보과 학생들의 정보를 구하라.

학생

*학번	이름	학과	학년
2008010	홍길동	기계과	2
2009001	이육사	전기과	1
2007002	김기자	정보통신과	1
2006003	박이사	컴퓨터정보과	1
2007007	박상숙	컴퓨터정보과	2



결과

*학번	이름	학과	학년
2006003	박이사	컴퓨터정보과	1
2007007	박상숙	컴퓨터정보과	2

3.4 관계 데이터베이스 언어

- 프로젝트 연산(PROJECT; π)

- 특정 조건에 맞는 어트리뷰트만으로 결과 테이블을 작성함(수직적 부분집합)
- 예 : 학생 테이블에서 학생들의 학번과 이름을 구하라.

학생				결과	
*학번	이름	학과	학년	*학번	이름
2008010	홍길동	기계과	2	2008010	홍길동
2009001	이육사	전기과	1	2009001	이육사
2007002	김기자	정보통신과	1	2007002	김기자
2006003	박이사	컴퓨터정보과	1	2006003	박이사
2007007	박상숙	컴퓨터정보과	2	2007007	박상숙

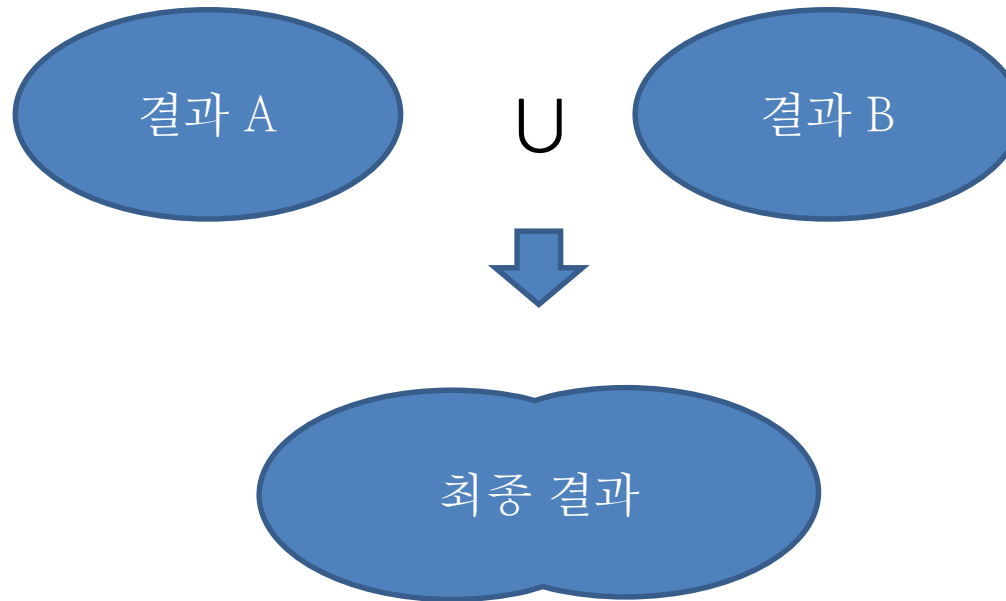
- 학생 테이블에서 컴퓨터정보과 학생들의 학번과 이름을 구하라.

학생				결과	
*학번	이름	학과	학년	*학번	이름
2008010	홍길동	기계과	2	2006003	박이사
2009001	이육사	전기과	1		
2007002	김기자	정보통신과	1		
2006003	박이사	컴퓨터정보과	1		
2007007	박상숙	컴퓨터정보과	2	2007007	박상숙

3.4 관계 데이터베이스 언어

- 합집합 연산(UNION; U)

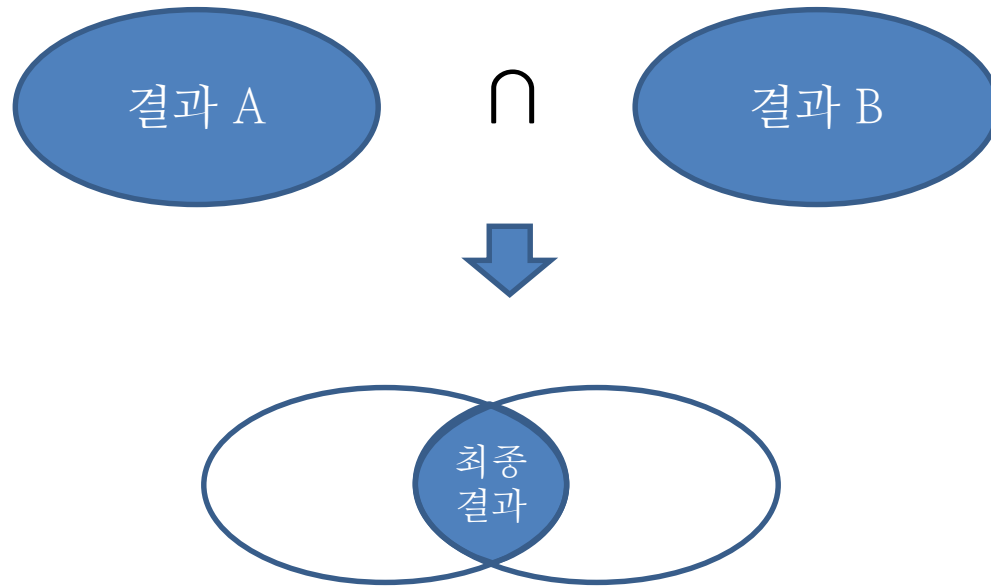
- 두 개 이상의 결과 테이블을 합쳐서 최종 결과를 작성함



3.4 관계 데이터베이스 언어

- 교집합 연산(INTERSECT; \cap)

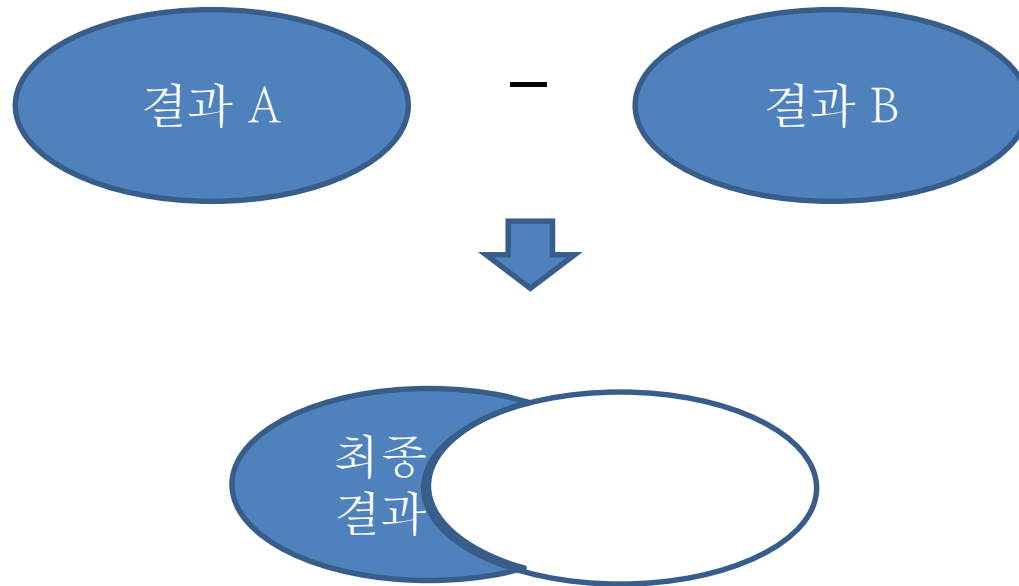
- 두 개 이상의 결과 테이블을 공통부분을 최종 결과를 작성함



3.4 관계 데이터베이스 언어

- 차집합 연산(DIFFERENCE;-)

- 결과 A에서 결과 B를 제외한 최종 결과를 작성함



3.4 관계 데이터베이스 언어

● 카티션 프로덕트 연산(CARTISIAN PRODUCT;X)

➤ 테이블 A와 테이블 B를 연결하는 모든 경우(튜플의 쌍) 나열하여 결과를 작성함

학생

*학번	이름	학과	학년
2008010	홍길동	기계과	2
2009001	이육사	전기과	1
2007002	김기자	정보통신과	1
2006003	박이사	컴퓨터정보과	1
2007007	박상숙	컴퓨터정보과	2

수강

*학번	*번호	성적
2008010	101	95
2008010	102	80
.....
2006003	104	90
2007007	105	75

X



학번	이름	학과	학년	학번	번호	성적
2008010	홍길동	기계과	2	2008010	101	95
2008010	홍길동	기계과	2	2008010	102	80
2008010	홍길동	기계과	2	2008010	103	60
2008010	홍길동	기계과	2	2009001	101	80
2008010	홍길동	기계과	2	2009001	104	95
2008010	홍길동	기계과	2	2001012	102	70
2008010	홍길동	기계과	2	2001012	103	80
2008010	홍길동	기계과	2	2007002	102	85
2008010	홍길동	기계과	2	2006003	104	90
2008010	홍길동	기계과	2	2007007	105	75
2009001	이육사	전기과	1	2008010	101	95
2009001	이육사	전기과	1	2008010	102	80
.....

3.4 관계 데이터베이스 언어

● 조인 연산(JOIN; ∞)

- 테이블 A와 테이블 B를 연결하여 **조건에 맞는** 결과를 작성함
- 예 : 홍길동 학생의 학번, 학과, 학년, 과목번호, 성적을 구하라.

학생

*학번	이름	학과	학년
2008010	홍길동	기계과	2
2009001	이육사	전기과	1
2007002	김기자	정보통신과	1
2006003	박이사	컴퓨터정보과	1
2007007	박상숙	컴퓨터정보과	2

수강

*학번	*번호	성적
2008010	101	95
2008010	102	80
.....
2006003	104	90
2007007	105	75

∞

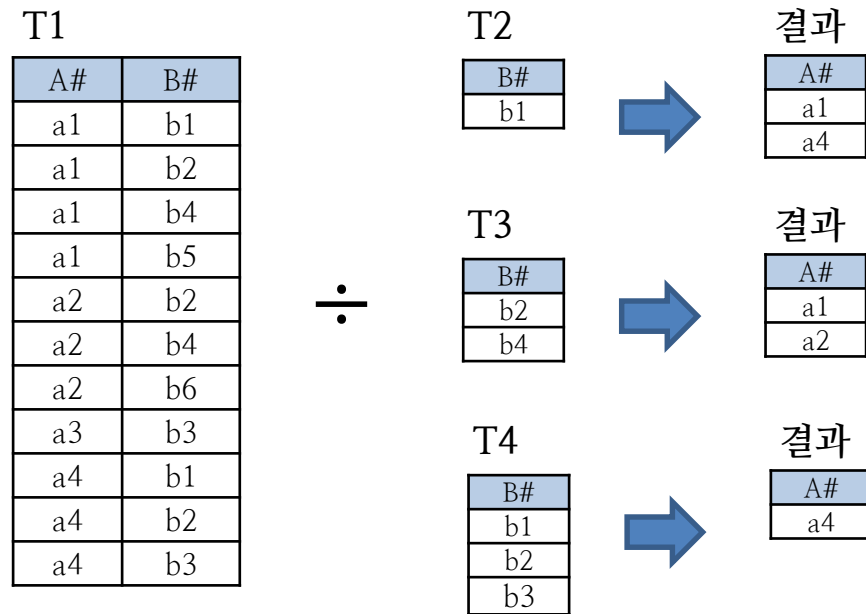


학번	이름	학과	학년	번호	성적
2008010	홍길동	기계과	2	101	95
2008010	홍길동	기계과	2	102	80
2008010	홍길동	기계과	2	103	60
2009001	이육사	전기과	1	101	80
2009001	이육사	전기과	1	104	95
2007002	김기자	정보통신과	1	102	85
2006003	박이사	컴퓨터정보과	1	104	90
2007007	박상숙	컴퓨터정보과	2	105	75

3.4 관계 데이터베이스 언어

● 디비전 연산(DIVISION;÷)

➤ 테이블 B의 모든 튜플을 가지는 릴레이션 A의 튜플을 추출하여 결과 테이블을 작성함



3장을 마치며.....

질의 & 응답

