6. 관계 데이터 연산

관계 데이터 연산의 개념

- 데이터 모델은 데이터 구조, 연산, 제약조건으로 구성된다
  - 데이터 구조, 제약조건은 5장 참조

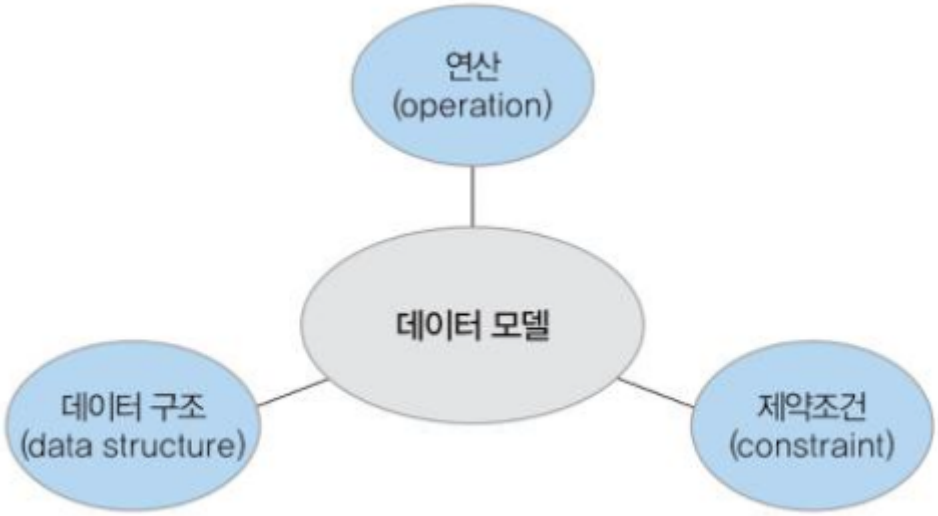


그림 6-1 데이터 모델의 구성

- 관계 데이터 모델에서 연산은 원하는 데이터를 얻기 위해 릴레이션에 필요한 처리 요구를 수행하는 것으로, 데이터베이스 시스템의 구성 요소 중 데이터 언어의 역할을 한다
- 관계 데이터 모델의 연산을 간단히 관계 데이터 연산(relationship data operation)이라고도 한다
- 대표적인 관계 데이터 연산으로 관계 대수와 관계 해석이 있다

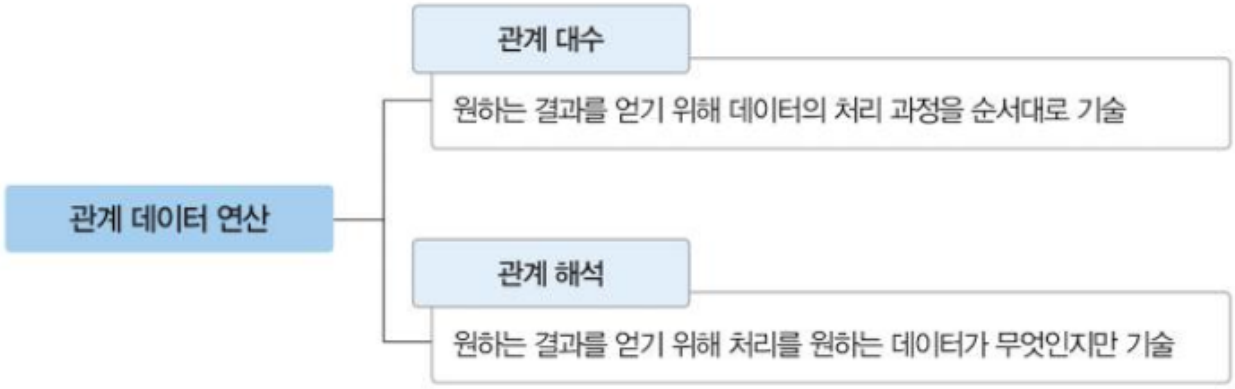


그림 6-2 관계 데이터 연산의 종류

- 관계 대수와 관계 해석은 원하는 데이터를 얻기 위한 처리 절차를 얼마나 자세히 기술하느냐에서 큰 차이를 보인다
  - 관계 대수(relational algebra)는 원하는 결과를 얻기 위해 데이터의 처리 과정을 순서대로 기술하는 절차 언어다
  - 관계 해석(relational calculus)은 원하는 결과를 얻기위해 처리를 원하는 데이터가 무엇인지만 기술하는 비절차 언어다
- 사용자 입장에서는 처리 과정을 자세히 기술하는 것보다 처리를 원하는 데이터가 무엇인지만 기술하는 비절차 언어가 더 편리하게 느껴질 수 있다
- 하지만 데이터를 처리하는 기능과 처리를 요구하는 표현력에서 관계 대수와 관계 해석은 능력이 동등하다
  - 관계 대수로 기술된 데이터 처리 요구는 관계 해석으로도 기술할 수 있고, 관계 해석으로 기술된 데이터 처리 요구를 관계 대수로도 기술할 수 있다
- 데이터 처리에 대한 요구를 일반적으로 질의(query)라 한다
- 데이터 언어가 관계 대수나 관계 해석으로 기술할 수 있는 모든 질의를 새로 제안된 데이터 언어로 기술할 수 있으면 관계적으로 완전(relationally complete)하다고 하고, 해당 언어가 어느정도 검증되었다고 판단한다

관계 대수

- 관계 대수의 개념과 연산자
  - 관계 대수는 원하는 결과를 얻기 위해 릴레이션을 처리하는 과정을 순서대로 기술하는 언어다
  - 연산자들의 집합의로도 정의할 수 있다
  - 일반적으로 연산자와 함께 연산의 대상이 되는 피연산자가 존재하기 마련인데 관계 대수에서는 피연산자가 릴레이션이다
    - 즉, 관계 대수는 릴레이션을 연산한다
    - 피연산자인 릴레이션에 연산자를 적용해 얻은 결과도 릴레이션이다
    - 이러한 관계 대수의 특성을 폐쇄 특성(closure property)이라 하는데, 이는 '콩 심은데 콩 나고 팥 심은데 팥 난다'는 속담을 떠올리게 한다
  - 관계 대수에 속하는 대표적인 연산자 8개는 특성에 따라 일반 집합 연산자(set operation)와 순수 관계 연산자(relational operation)로 분류할 수 있다

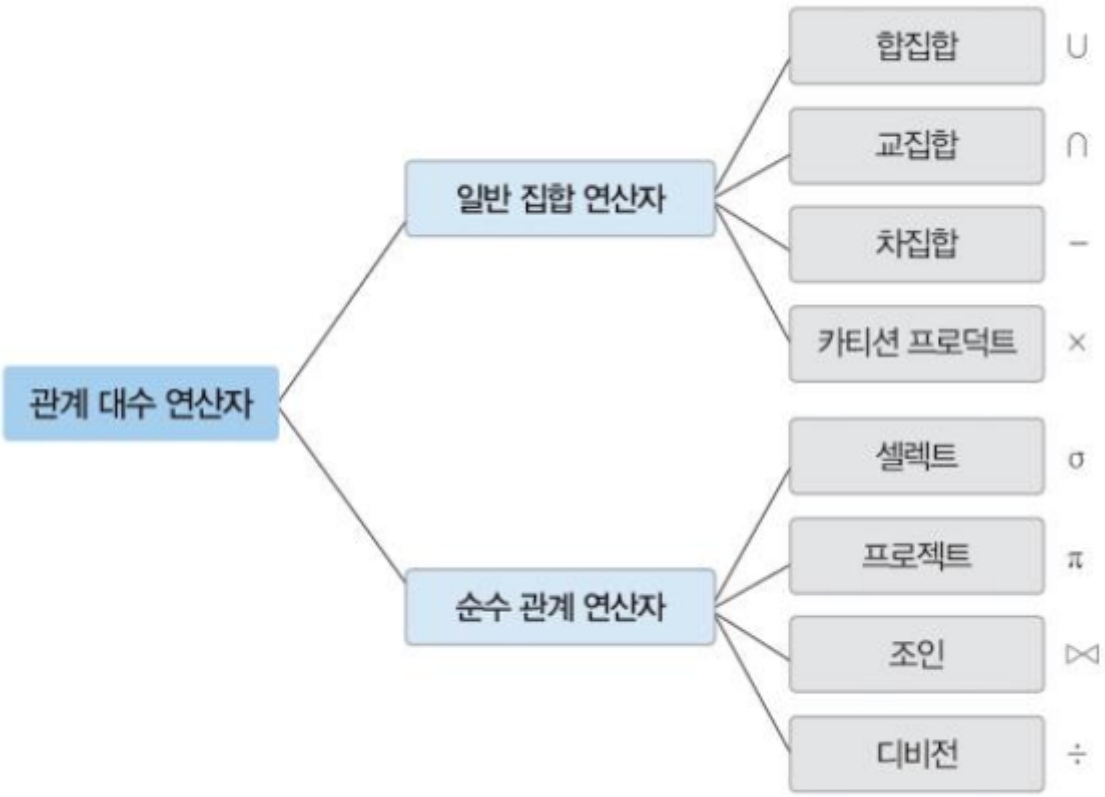


그림 6-3 관계 대수 연산자의 종류

- 일반 집합 연산자는 릴레이션이 튜플의 집합이라는 개념을 이용하는데 이는 수학의 집합 관련 연산자를 차용한 것이다
- 일반 집합 연산자의 종류와 기능은 다음과 같다

연산자	기호	표현	의미
합집합	∪	R∪S	릴레이션 R과 S의 합집합을 반환
교집합	∩	R∩S	릴레이션 R과 S의 교집합을 반환
차집합	−	R−S	릴레이션 R과 S의 차집합을 반환
카티션 프로덕트	×	R×S	릴레이션 R의 각 투플과 릴레이션 S의 각 투플을 모두 연결하여 만든 새로운 투플을 반환

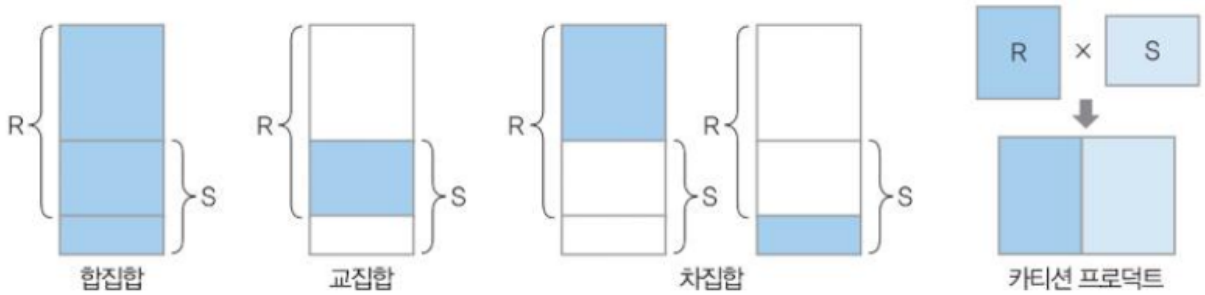


그림 6-4 일반 집합 연산자의 종류와 기능

- 순수 관계 연산자는 릴레이션의 구조와 특성을 이용하는 것으로 관계 데이터 모델에서 새로 제시된 연산자다
- 순수 관계 연산자에 속하는 연산자의 종류와 기능은 다음과 같다

연산자	기호	표현	의미
셀렉트	σ	$\sigma_{조건}(R)$	릴레이션 R에서 조건을 만족하는 투플들을 반환
프로젝트	π	$\pi_{속성리스트}(R)$	릴레이션 R에서 주어진 속성들의 값으로만 구성된 투플들을 반환
조인	⋈	R ⋈ S	공통 속성을 이용해 릴레이션 R과 S의 투플들을 연결하여 만든 새로운 투플들을 반환
디비전	÷	R ÷ S	릴레이션 S의 모든 투플과 관련이 있는 릴레이션 R의 투플들을 반환

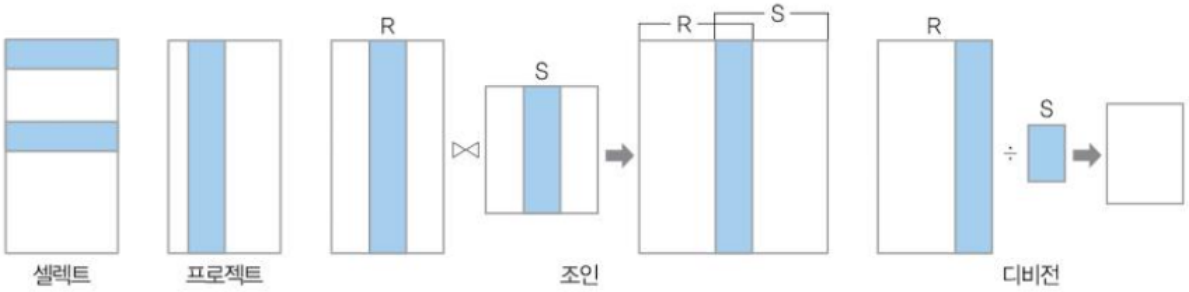


그림 6-5 순수 관계 연산자의 종류와 기능

- 일반 집합 연산자
  - 일반 집합 연산자의 제약조건
    - 일반 집합 연산자는 연산을 위해 피연산자가 2개 필요하다
    - 합집합, 교집합, 차집합은 피연산자인 2개의 릴레이션이 합병 가능(union-compatible)해야 한다
      - 다음 두 조건을 만족하면 합병 가능하다
        - 두 릴레이션의 차수가 같다. 즉, 두 릴레이션은 속성 개수가 같다
        - 2개의 릴레이션에서 서로 대응되는 속성의 도메인이 같다. 단, 도메인이 같으면 속성의 이름은 달라도 된다

고객 릴레이션

고객번호	고객이름	나이
INT	CHAR(20)	INT
100	정소화	20
200	김선우	35
300	고명석	24

직원 릴레이션

직원번호	직원이름	직위
INT	CHAR(20)	CHAR(20)
10	김용욱	부장
20	채광주	과장
30	김수진	대리

그림 6-6 합병이 불가능한 예

- 위 예는 직위와 나이 속성의 도메인이 달라서 합병 불가능 하다

고객 릴레이션

고객번호	고객이름	나이
INT	CHAR(20)	INT
100	정소화	20
200	김선우	35
300	고명석	24

직원 릴레이션

직원번호	직원이름	나이
INT	CHAR(20)	INT
10	김용욱	40
20	채광주	32
30	김수진	28

그림 6-7 합병이 가능한 예

- 위 예는 둘 다 차수가 3으로 같고 모든 속성의 도메인이 같으므로 합병 가능하다

- 합집합
  - 합병이 가능한 두 릴레이션 R과 S의 합집합(union)은 R ∪ S로 표현한다
  - R ∪ S는 릴레이션 R에 속하거나 릴레이션 S에 속하는 모든 튜플로 결과 릴레이션을 구성한다

R

번호	이름
100	정소화
200	김선우
300	고명석

S

번호	이름
100	정소화
101	채광주
102	김수진

합집합 연산

R∪S

번호	이름
100	정소화
200	김선우
300	고명석
101	채광주
102	김수진

그림 6-8 합집합 연산의 예

  - 합집합 연산 결과 중복되는 튜플은 한 번만 나타난다
  - 합집합 연산 결과 릴레이션은 피연산자인 R과 S의 차수와 같다
  - 합집합 연산 결과 카디널리티(튜플의 전체 개수)는 R과 S의 튜플 개수를 합친 것보다 작거나 같다
  - 교환법칙, 결합법칙이 성립한다

- 교집합
  - 합병이 가능한 두 릴레이션 R과 S의 교집합(intersection)은 RnS 로 표현한다
  - 교집합 연산은 릴레이션 R과 S에 공통으로 속하는 튜플로 결과 릴레이션을 구성한다

그림 6-9 교집합 연산의 예

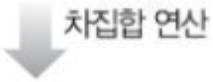
  - 교집합 연산 결과 릴레이션의 차수는 릴레이션 R과 S의 차수와 같다
  - 교집합 연산 결과 카디널리티는 릴레이션 R과 S의 각각의 카디널리티보다 작거나 같다
  - 교환법칙, 결합법칙이 성립한다
- 차집합
  - 합병이 가능한 두 릴레이션 R과 S의 차집합(difference)은 R-S로 표현한다
  - 차집합 연산은 릴레이션 R에는 존재하지만 릴레이션 S에는 존재하지 않는 튜플들로 결과 릴레이션을 구성한다

R

번호	이름
100	정소화
200	김선우
300	고명석

S

번호	이름
100	정소화
101	채광주
102	김수진



R–S

번호	이름
200	김선우
300	고명석

S–R

번호	이름
101	채광주
102	김수진

그림 6-10 차집합 연산의 예

- 차집합 연산의 결과 릴레이션 차수는 피연산자의 차수와 같다
- 차집합 연산의 결과 카디널리티는 피연산자 각각의 카디널리티보다 작거나 같다
- 교환법칙, 결합법칙이 성립하지 않는다

- 카티션 프로덕트
  - 두 릴레이션 R과 S의 카티션 프로덕트(cartesian product)는 R X S로 표현한다
  - 카티션 프로덕트 연산은 릴레이션 R에 속한 각 튜플과 릴레이션 S에 속한 각 튜플을 모두 연결하여 만들어진 새로운 튜플을 결과 릴레이션을 구성한다

R

번호	이름
INT	CHAR(20)
100	정소화
200	김선우
300	고명석

S

번호	나이
INT	INT
100	40
101	30
102	25

카티션 프로덕트 연산

R×S

R.번호	R.이름	S.번호	S.나이
INT	CHAR(20)	INT	INT
100	정소화	100	40
100	정소화	101	30
100	정소화	102	25
200	김선우	100	40
200	김선우	101	30
200	김선우	102	25
300	고명석	100	40
300	고명석	101	30
300	고명석	102	25

그림 6-11 카티션 프로덕트 연산의 예

- 카티션 프로덕트 연산에 참여하는 릴레이션 R,S의 차수는 같지만 도메인은 다르다
  - 카티션 프로덕트 연산은 합병이 불가능한 경우에도 가능하다
- 결과 릴레이션의 속성을 '릴레이션이름.속성이름' 형식으로 표기한것은 속성이 원래 어느 릴레이션 소속인지를 나타내기 위한 표기법이다
- 카티션 프로덕트 연산의 결과 릴레이션의 차수는 피연산자의 차수를 더한것과 같다
- 그리고 결과 카디널리티는 릴레이션 R,S의 카디널리티를 곱한것과 같다
- 교환법칙, 결합법칙이 성립한다

- 순수 관계 연산자
  - 순수 관계 연산자는 릴레이션의 구조와 특성을 이용하는 연산자다
    - 대표적으로 셀렉트, 프로젝트, 조인, 디비전이 있다
  - 순수 관계 연산자는 관계 데이터 모델에서 새로 제시된 것이다

- 셀렉트
  - 셀렉트(select)연산은 릴레이션에서 주어진 조건을만족하는 튜플만 선택하여 결과 릴레이션을 구성한다
  - 따라서 결과 릴레이션은 주어진 릴레이션을 수평으로 절단한 모양이 된다
  - 즉, 해당 릴레이션에서 수평적 부분집합(horizontal subset)을 생성한 것과 같다

$$\sigma_{조건식}(릴레이션)$$

  - 셀렉트 연산의 표기법
  - 셀렉트 연산은 하나의 릴레이션을 대상으로 수행한다
  - 조건식은 비교 연산자를 이용해 구성하는데, 조건식을 비교식 또는 프레디킷(predicate)라고 한다

$$(>,\ \geq,\ <,\ \leq,\ =,\ \neq)$$

  - 조건식을 속성과 상수의 비교나 다른 속성들 간의 비교로 표현할 수 있다
  - 속성과 상수를 비교할 때는 상수의 데이터 타입이 속성의 도메인과 일치해야 한다
  - 다른 속성들 간의 비교할 때는 속성들의 도메인이 같아야 비교가 가능하다
  - 비교연산자와 함께 논리 연산자를 사용할 수도 있다

$$\wedge,\ \vee,\ \neg$$

  - 왼쪽부터 and, or, not 이다
  - 셀렉트 연산은 where문을 이용해 일반적인 데이터 언어의 형식으로 표현할 수 있다

```
릴레이션 where 조건식
```

고객아이디	고객이름	나이	등급	직업	적립금
apple	김현준	20	gold	학생	1000
banana	정소화	25	vip	간호사	2500
carrot	원유선	28	gold	교사	4500
orange	정지영	22	silver	학생	0

그림 6-12 셀렉트 연산을 적용할 릴레이션 예 : 고객 릴레이션

- 위의 예시에서 셀렉트를 적용한 예제

예제 6-1

고객 릴레이션에서 등급이 gold인 투플을 검색하시오.

▶▶ $\sigma_{등급='gold'}(고객)$ 또는 `고객 where 등급 = 'gold'`

결과 릴레이션

고객아이디	고객이름	나이	등급	직업	적립금
apple	김현준	20	gold	학생	1000
carrot	원유선	28	gold	교사	4500

예제 6-2

고객 릴레이션에서 등급이 gold이고, 적립금이 2000 이상인 투플을 검색하시오.

▶▶ $\sigma_{등급='gold' \wedge 적립금 \geq 2000}(고객)$ 또는 `고객 where 등급 = 'gold' and 적립금 ≥ 2000`

결과 릴레이션

고객아이디	고객이름	나이	등급	직업	적립금
carrot	원유선	28	gold	교사	4500

- 셀렉트는 릴레이션의 모든 튜플 중 조건식을 참으로 만드는 일부분의 튜플만 선택한다
  - 튜플은 릴레이션의 행에 해당되므로 셀렉트는 연산 대상 릴레이션의 수평적 부분집합으로 결과를 구성하는 수평적 연산자이다
- 교환법칙이 성립한다

$$\sigma_{적립금 \geq 2000}(\sigma_{등급='gold'}(고객)) = \sigma_{등급='gold'}(\sigma_{적립금 \geq 2000}(고객)) = \sigma_{등급='gold' \wedge 적립금 \geq 2000}(고객)$$

- 프로젝트
  - 프로젝트(project)연산은 릴레이션에서 선택한 속성에 해당하는 값으로 결과 릴레이션을 구성한다
  - 결과 릴레이션이 주어진 릴레이션의 일부 열로만 구성되어 해당 릴레이션에서 수직접 부분집합(vertical subset)을 생성하는 것과 같다
  - 프로젝트 연산은 파이 기호를 사용해서 나타낸다

    $\pi_{속성리스트}(릴레이션)$

  - 일반 데이터 언어의 형식으로 표현하면 다음과 같다

    릴레이션[속성리스트]

고객아이디	고객이름	나이	등급	직업	적립금
apple	김현준	20	gold	학생	1000
banana	정소화	25	vip	간호사	2500
carrot	원유선	28	gold	교사	4500
orange	정지영	22	silver	학생	0

그림 6-14 프로젝트 연산을 적용할 릴레이션 예 : 고객 릴레이션

- 위의 예제에서 프로젝트 연산을 적용한 예시는 다음과 같다

예제 6-3

고객 릴레이션에서 고객이름, 등급, 적립금을 검색하시오.

▶▶ $\pi_{\text{고객이름,등급,적립금}}(\text{고객})$ 또는 고객[고객이름,등급,적립금]

결과 릴레이션

고객이름	등급	적립금
정소화	gold	1000
김선우	vip	2500
고명석	gold	4500
김용욱	silver	0

예제 6-4

고객 릴레이션에서 등급을 검색하시오.

▶▶ $\pi_{\text{등급}}(\text{고객})$ 또는 고객[등급]

결과 릴레이션

등급
gold
vip
silver

- 프로젝트 연산을 한 결과 릴레이션에도 동일한 튜플이 중복되지 않고 한 번만 나타난다
  - 프로젝트 연산의 결과도 릴레이션이기 때문에 중복되는 튜플이 존재할 수 없다는 릴레이션의 기본 특성을 유지하는 것
- 프로젝트는 릴레이션의 모든 속성 중 일부분만 선택하는 연산이고 속성은 릴레이션에서 열에 해당되므로 프로젝트는 대상 릴레이션의 수직접 부분집합으로 결과 릴레이션을 구성하는 수직적 연산자

이다

고객아이디	고객이름	나이	등급	직업	적립금
apple	김현준	20	gold	학생	1000
banana	정소화	25	vip	간호사	2500
carrot	원유선	28	gold	교사	4500
orange	정지영	22	silver	학생	0

고객이름	등급	적립금
김현준	gold	1000
정소화	vip	2500
원유선	gold	4500
정지영	silver	0

그림 6-15 프로젝트 연산의 수행 과정 예 : 고객 릴레이션

- 조인
  - 릴레이션 하나로 원하는 데이터를 얻을 수 없어 관계가 있는 여러 릴레이션을 함께 사용해야 하는 경우 조인(join)연산을 이용한다
  - 조인 연산은 조인 속성(join attribute)을 이용해 두 릴레이션을 조합하여 하나의 결과 릴레이션을 구성한다
    - 조인 속성은 두 릴레이션이 공통으로 가지고 있는 속성으로, 두 릴레이션이 관계가 있음을 나타낸다
  - 조인 연산한 결과 릴레이션은 피연산자 릴레이션에서 조인 속성의 값이 같은 튜플만 연결하여 만든 새로운 튜플을 포함한다
    - 즉, 조인 연산의 결과는 두 릴레이션에 대해 카티션 프로덕트 연산을 수행한 후 조인 속성의 값이 같은 조건을 만족하는 튜플을 반환하는 셀렉트 연산을 수행한 것과 같다
  - 조인 연산의 기호는 다음과 같다

    릴레이션1 ⋈ 릴레이션2

- 조인 연산을 적용할 릴레이션들의 관계가 다음과 같다면

그림 6-16 조인 연산을 적용할 예제 릴레이션들의 관계

- 주문 릴레이션의 외래키인 고객아이디 속성은 고객 릴레이션의 기본키를 참조한 것이며 두 릴레이션의 관계를 표현한다
  - 그러므로 주문 릴레이션의 고객아이디 속성과 고객 릴레이션의 고객아이디 속성은 두 릴레이션에 공통으로 존재하는 조인 속성이다
- 두 릴레이션을 조인할 때는 조인 속성의 값이 같은 튜플만 연결하여 결과 릴레이션을 구성해야 한다

$$고객 \bowtie 주문$$

- 위의 표현식으로 조인 연산을 실행하면

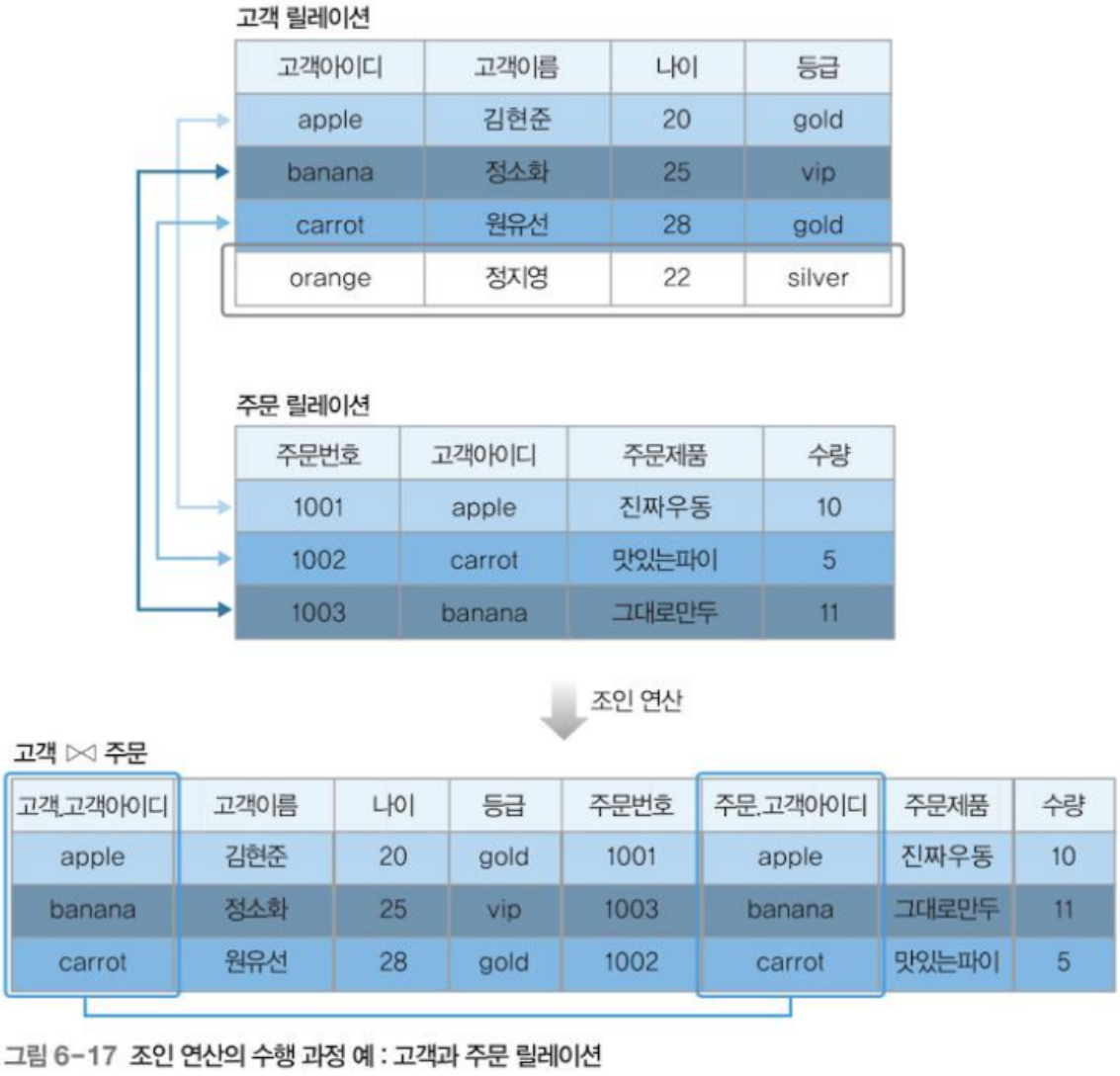


그림 6-17 조인 연산의 수행 과정 예 : 고객과 주문 릴레이션

- 이런 결과가 나온다
- 고객 릴레이션의 조인 속성인 orange 고객아이디 속성은 주문 릴레이션의 조인 속성인 고객아이디 속성에 존재하지 않는 값이기 때문에 조인 연산의 결과에서 제외된다
- 조인 연산을 하는 이유는
  - 예를 들어 정소화 고객이 주문한 제품을 검색하려면 고객 데이터와 고객과 관련된 주문 데이터가 모두 필요하므로, 고객 릴레이션과 주문 릴레이션을 조인하여 결과 릴레이션을 구성한 후 조건에 맞는 튜플을 찾기 위해서다
- 지금까지의 조인연산은 정확히 분류하면 동등 조인(equi-join)이다
  - 질의에서 동등 조인의 개념이 많이 사용되기 때문에 조인이라고 하면 보통 동등 조인을 의미하고 동등 조인의 기호는 다음과 같다

    ⋈
- 동등 조인에 비해 좀 더 일반화된 조인으로 세타 조인(theta-join)이 있다
- 세타 조인은 주어진 조인 조건을 만족하는 두 릴레이션의 모든 튜플을 연결한 새로운 튜플로 결과 릴레이션을 구성한다

- 세타 조인의 기호는 다음과 같다

$$릴레이션1 \bowtie_{A\,\theta\,B} 릴레이션2$$

- 동등 조인의 기호에서 조건이 추가된것
- A는 릴레이션1의 속성이고 B는 릴레이션2의 속성이다
- 세타는 비교연산자다
  - 속성 값에 대한 비교 연산이 가능하도록 A와 B의 도메인은 같아야 한다
- 세타 조인 연산을 한 후 얻게 되는 결과 릴레이션의 차수는 릴레이션1의 차수와 릴레이션2의 차수를 더한 것과 같다
- 세타 조인은 세타, 즉 비교 연산자를 이용해 다양한 조인 조건을 표현할 수 있는데, 특히 세타연산자가 '='인 세타 조인이 동등 조인인 것
- 위의 동등 조인 연산을 세타 조인으로 표현하면

$$고객 \bowtie_{고객.고객아이디=주문.고객아이디} 주문$$

- 동등 조인의 결과 릴레이션에는 두 릴레이션의 모든 속성이 포함되기 때문에 조인 속성이 중복되어 나타난다
  - 릴레이션.속성 의 표기법으로 둘을 구분했다
- 자연 조인(natural join)은 동등 조인의 결과 릴레이션에서 중복된 속성을 제거하여 조인 속성이 한 번만 나타나고, 기호는 다음과 같다

$$고객 \bowtie_{N} 주문$$

- 자연 조인의 결과는

고객 릴레이션

고객아이디	고객이름	나이	등급
apple	김현준	20	gold
banana	정소화	25	vip
carrot	원유선	28	gold
orange	정지영	22	silver

주문 릴레이션

주문번호	고객아이디	주문제품	수량
1001	apple	진짜우동	10
1002	carrot	맛있는파이	5
1003	banana	그대로만두	11

자연 조인 연산

고객 $\bowtie_N$ 주문

고객아이디	고객이름	나이	등급	주문번호	주문제품	수량
apple	김현준	20	gold	1001	진짜우동	10
banana	정소화	25	vip	1003	그대로만두	11
carrot	원유선	28	gold	1002	맛있는파이	5

그림 6-18 자연 조인 연산의 예 : 고객과 주문 릴레이션

- 조인 속성이 여러개의 속성으로 구성되어 있다면 결과는 다음과 같다

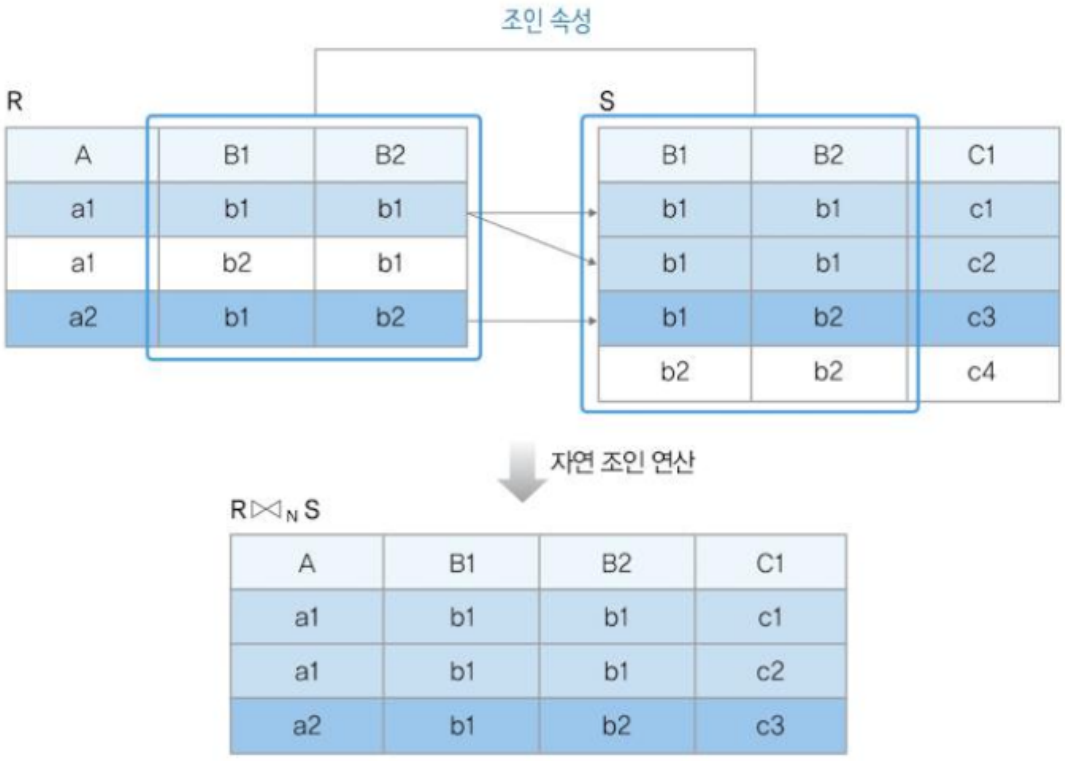


그림 6-19 2개의 속성으로 이루어진 조인 속성을 이용하는 자연 조인 연산의 예 : R과 S 릴레이션

- 디비전
  - 두 릴레이션 R과 S의 디비전(division) 연산은 R / S (나누기 기호)로 표현한다
  - 디비전 연산은 릴레이션S의 모든 튜플과 관련있는 릴레이션 R의 튜플로 결과 릴레이션을 구성한다
    - 단, 릴레이션 R이 릴레이션 S의 모든 속성을 포함하고 있어야 R / S 연산이 가능하다
    - 이는 릴레이션 S의 모든 속성과 도메인이 같은 속성을 릴레이션 R이 포함하고 있어야 한다는 의

미

고객 릴레이션

고객아이디	고객이름	나이	등급	직업	적립금
apple	김현준	20	gold	학생	1000
NULL	정소화	25	vip	간호사	2500
carrot	원유선	28	gold	교사	4500
NULL	정지영	22	silver	학생	0

우수등급 릴레이션

등급
gold

디비전 연산

고객 ÷ 우수등급

고객아이디	고객이름	나이	직업	적립금
apple	김현준	20	학생	1000
carrot	원유선	28	교사	4500

그림 6-20 디비전 연산의 예 1 : 고객과 우수등급 릴레이션

- 위는 간단한 디비전 연산의 예
  - 나누는 릴레이션인 우수 등급 릴레이션의 속성의 값인 gold와 같은 튜플이 선택되고 결과 릴레이션에서는 gold속성(등급속성)이 제외되었다

주문내역 릴레이션

주문고객	제품이름	제조업체
apple	진짜우동	한빛식품
carrot	맛있는파이	마포과자
banana	그대로만두	한빛식품
apple	그대로만두	한빛식품
carrot	그대로만두	한빛식품

제품1 릴레이션

제품이름
진짜우동
그대로만두

제품2 릴레이션

제품이름	제조업체
그대로만두	한빛식품

디비전 연산

주문내역 ÷ 제품1

주문고객	제조업체
apple	한빛식품

주문내역 ÷ 제품2

주문고객
banana
apple
carrot

그림 6-21 디비전 연산의 예 2 : 주문내역, 제품1, 제품2 릴레이션

- 다양한 디비전 연산의 예
  - 제품1 릴레이션으로 디비전 연산을 하면 '진짜우동', '그대로만두' 속성을 하나라도 포함하고 있는 모든 튜플을 선택하여 제품이름 속성을 제외한 결과 릴레이션을 구성
  - 제품2 릴레이션으로 디비전 연산을 하면 '그대로만두', '한빛식품' 속성을 모두 포함하고 있는 모든 튜플을 선택하여 제품이름, 제조업체 속성을 제외한 결과 릴레이션을 구성

- 관계 대수를 이용한 질의 표현

고객 릴레이션

고객아이디	고객이름	나이	등급
apple	김현준	20	gold
banana	정소화	25	vip
carrot	원유선	28	gold
orange	정지영	22	silver

주문 릴레이션

주문번호	주문고객	주문제품	수량
1001	apple	진짜우동	10
1002	carrot	맛있는파이	5
1003	banana	그대로만두	11

주문 릴레이션의 외래키

그림 6-22 질의 표현에 사용할 예제 릴레이션들 : 고객과 주문 릴레이션

- 다음과 같은 관계의 릴레이션들에서 몇 가지 질의 예제를 살펴보자

예제 6-5

등급이 gold인 고객의 이름과 나이를 검색하시오.

▶▶ $\pi_{\text{고객이름,나이}}(\sigma_{\text{등급='gold'}}(\text{고객}))$

결과 릴레이션

고객이름	나이
김현준	20
원유선	28

예제 6-6

고객이름이 원유선인 고객의 등급과, 원유선 고객이 주문한 주문제품, 수량을 검색하시오.

▶▶ $\pi_{\text{등급,주문제품,수량}}(\sigma_{\text{고객이름='원유선'}}(\text{고객} \bowtie \text{주문}))$

결과 릴레이션

등급	주문제품	수량
gold	맛있는파이	5

예제 6-7

주문수량이 10개 미만인 주문 내역을 제외하고 검색하시오.

▶▶ $\text{주문} - (\sigma_{\text{수량<10}}(\text{주문}))$

결과 릴레이션

주문번호	주문고객	주문제품	수량
1001	apple	진짜우동	10
1003	banana	그대로만두	11

- 확장된 관계 대수 연산자
  - 자연 조인 연산을 확장한 세미 조인과 외부 조인을 알아보자
  - 세미 조인
    - 세미 조인 연산은 릴레이션 S의 조인 속성으로만 구성한(프로젝트한) 릴레이션을 릴레이션 R에 자연 조인 하는것이다
    - 세미 조인의 기호

$$R \ltimes S$$

그림 6-25 고객과 주문 릴레이션의 세미 조인 연산

- 주문 릴레이션에서 조인 속성인 고객아이디 속성을 프로젝트 한 후 고객 릴레이션에 자연 조인하면 결과를 얻을 수 있다
- 결과적으로 세미 조인은 고객 릴레이션에서 자연 조인 연산에 참여할 수 있는 튜플만 선택하여 결과 릴레이션을 구성한다
- 주문을 한 적이 있는 고객의 데이터만 검색하고 싶을 때 세미 조인을 이용하면 검색에 불필요한 속성을 미리 제거하여 조인 연산의 비용을 줄일 수 있다
- 하지만 세미 조인은 교환법칙이 성립하지 않는다

- 외부 조인
  - 외부 조인(outer join)연산은 두 릴레이션에 자연 조인 연산을 수행할 때 조인 속성 값이 같은 튜플이 상대 릴레이션에 존재하지 않아 조인 연산에서 제외된 모든 튜플을 결과 릴레이션에 포함시킨다
  - 이때, 결과 릴레이션에 속성 값이 없는 경우는 널 값으로 처리한다
  - 외부 조인은 왼쪽 외부 조인 연산, 오른쪽 외부 조인 연산, 완전 외부 조인 연산으로 나뉜다

고객 릴레이션

고객아이디	고객이름	나이
apple	김현준	20
banana	정소화	25
carrot	원유선	28
orange	정지영	22

주문 릴레이션

주문번호	고객아이디	주문제품
1001	apple	진짜우동
1002	carrot	맛있는파이
1003	banana	그대로만두
1004	NULL	얼큰라면

- 위의 상황에서 외부 조인 연산을 모두 적용하면

(1) 왼쪽 외부 조인 연산 : 고객 ⟕ 주문

고객아이디	고객이름	나이	주문번호	주문제품
apple	김현준	20	1001	진짜우동
banana	정소화	25	1003	그대로만두
carrot	원유선	28	1002	맛있는파이
orange	정지영	22	NULL	NULL

(2) 오른쪽 외부 조인 연산 : 고객 ⟖ 주문

고객아이디	고객이름	나이	주문번호	주문제품
apple	김현준	20	1001	진짜우동
banana	정소화	25	1003	그대로만두
carrot	원유선	28	1002	맛있는파이
NULL	NULL	NULL	1004	얼큰라면

(3) 완전 외부 조인 연산 : 고객 ⟗ 주문

고객아이디	고객이름	나이	주문번호	주문제품
apple	김현준	20	1001	진짜우동
banana	정소화	25	1003	그대로만두
carrot	원유선	28	1002	맛있는파이
orange	정지영	22	NULL	NULL
NULL	NULL	NULL	1004	얼큰라면

그림 6-27 세 가지 외부 조인 결과 비교 : 고객과 주문 릴레이션

관계 해석

- 관계 해석은 처리를 원하는 데이터가 무엇인지만 기술하는 비절차 언어로, 관계 대수처럼 관계 데이터 연산의 한 종류다
- 데이터를 처리하는 기능과 처리를 요구하는 표현력에서 관계 대수와 관계 해석은 능력이 모두 동일하다

- 관계 해석은 관계 데이터 모델의 제안자인 코드가 수학의 프레디킷 해석(predicate calculus)에 기반을 두고 제안했다
- 튜플 관계해석(tuple relational calculus), 도메인 관계 해석(domain relational calculus)으로 분류된다