

## 1. 관계대수(Relational Algebra)

- 관계대수: 관계형 데이터베이스에서 원하는 정보와 그 정보를 어떻게 유도하는가를 기술하는 절차적 언어
- 관계대수: 하나 또는 두 개의 릴레이션을 입력으로 받아 들어 결과로서 새로운 릴레이션을 생성해내는 연산들의 집합이다.
- 관계대수 연산: 1) 기본연산인 SELECT, PROJECT, JOIN, DIVISION 등  
2) 집합 연산인 UNION, DIFFERENCE, INTERSECTION, CARTESIAN PRODUCT 등이 있다.

## 2. 순수 관계 연산자

### 1) SELECT

- SELECT 연산은 릴레이션에 존재하는 튜플들 중에서 특정 조건을 만족하는 튜플들의 부분집합을 구하여 새로운 릴레이션을 만든다.
- 릴레이션의 행에 해당하는 튜플을 구하는 것이므로 수평 연산이라고도 한다.
- SELECT 연산의 기호는 그리스 문자 시그마 ( $\sigma$ ) 이다.
- 표기 형식

$\sigma_{\langle \text{조건} \rangle}(R)$

- 단, R은 릴레이션이다.
- 조건에서는  $=, \neq, <, \leq, >, \geq$  등의 기호를 사용한 비교 연산이 허용되며, AND( $\wedge$ ), OR( $\vee$ ), NOT( $\neg$ )등의 논리 연산자를 사용하여 여러 개의 조건들을 하나의 조건으로 결합시킬 수도 있다.

<성적>

이름	국어	영어	수학	과학
고길동	70	90	78	88
이순신	90	80	88	98
둘리	80	68	98	78
아무개	79	79	70	57
임꺽정	67	98	80	84

<학적부>

이름	주소	나이
고길동	모진동	25
이순신	화양동	20
둘리	자양동	24
아무개	화양동	19
임꺽정	구의동	18

**예제 1**  $\sigma_{\text{국어} \geq 80}(\text{성적})$  : <성적> 릴레이션에서 국어 점수가 80점 이상인 튜플들을 추출한다.

**예제 2**  $\sigma_{\text{국어} \geq 80 \vee \text{영어} \geq 80}(\text{성적})$  : <성적> 릴레이션에서 국어 점수 또는 영어 점수가 80점 이상인 튜플들을 추출한다.

### 2) PROJECT

- PROJECT 연산은 주어진 릴레이션에서 속성 리스트에 제시된 속성 값만을 추출하여 새로운 릴레이션을 만든다, 단, 연산 결과에 중복이 발생하면 중복이 제거된다,
- 릴레이션에서 열에 해당하는 속성을 추출하는 것이므로 수직 연산이라고도 한다.
- PROJECT 연산의 기호는 그리스 문자 파이( $\pi$ )를 사용한다.
- 표기형식

$\pi_{\langle \text{속성리스트} \rangle}(R)$

- 단, R은 릴레이션이다.

**예제**  $\pi_{\text{이름}, \text{국어}}(\text{성적})$  : <성적> 릴레이션에서 이름과 국어 속성을 추출한다,

3) JOIN

- JOIN 연산은 공통 속성을 중심으로 2개의 릴레이션을 하나로 합쳐서 새로운 릴레이션을 만든다.
- JOIN 연산의 결과로 만들어진 릴레이션의 차수는 조인된 두 릴레이션의 차수를 합한 것과 같다.
- JOIN 연산의 결과는 CARTESIAN PRODUCT 연산을 수행한 다음 SELECT 연산을 수행한 것과 같다.
- JOIN 연산의 기호는 ⋈ 이다.
- 표기 형식

$R \bowtie_{(JOIN\ 조건)} S$

- 단, R 과 S는 릴레이션이다.

**예제** 성적 ⋈<sub>이름=이름</sub> 학적부 : <성적> 릴레이션과 <학적부> 릴레이션으로부터 이름 속성이 같은 튜플들을 결합하여 새로운 릴레이션을 생성한다.

<성적>

<학적부>

이름	국어	영어	수학	과학
고길동	70	90	78	88
이순신	90	80	88	98
둘리	80	68	98	78
아무개	79	79	70	57
임꺽정	67	98	80	84

이름	주소	나이
고길동	모진동	25
이순신	화양동	20
둘리	자양동	24
아무개	화양동	19
임꺽정	구의동	18

성적,이름	국어	영어	수학	과학	학적부,이름	주소	나이
고길동	70	90	78	88	고길동	모진동	25
이순신	90	80	88	98	이순신	화양동	20
둘리	80	68	98	78	둘리	자양동	24
아무개	79	79	70	57	아무개	화양동	19
임꺽정	67	98	80	84	임꺽정	구의동	18

**해설** 위 그림에서와 같이 JOIN 조건이 '=' 일 때 동일한 속성이 2번 나타나게 된다.

- =, ≠, <, ≤, >, ≥ 등의 비교 연산자를  $\Theta$ 로 일반화 하여  $\Theta$ 로 표현될 수 있는 조인을 세타( $\Theta$ -join, Theta JOIN)이라고 한다. 위의 예제는  $\Theta$ 가 '='인 동일조인(Equi JOIN)이다.
- JOIN 조건이 '='일 때 동일한 속성이 2번 나타나게 되는데, 이 중 중복된 속성을 제거하여 같은 속성은 1번만 나타나게 하는 연산을 NATURAL JOIN 이라 한다.

4) DIVISION

- 두 릴레이션 R(X)와 S(Y)에 대해  $Y \subseteq X$ 이고  $X - Y = Z$ 라고 하면, R(X)와 R(X,Y)는 동일한 표현이다. 이때, 릴레이션 R(X,Y)에 대한 S(Y)의 DIVISION연산은 S(Y)의 모든 튜플에 연관되어 있는 R(Z)의 튜플을 선택하는 것이다.

- 표기형식

$R[속성r \div 속성s]S$

- 단, 속성 r은 릴레이션 R의 속성이고, 속성s는 릴레이션 S의 속성이며, 속성 r과 s는 동일 속성 값을 가지는 속성이어야 한다.

<구입자>

<생산물1>

<생산물2>

번호	이름	구입품코드
1	고길동	A
1	고길동	B
2	이순신	A
2	이순신	B
2	이순신	C
4	둘리	A

생산물코드
A
B

이름	생산물코드
이순신	A
이순신	B

**예제1** 구입자[구입품코드÷생산물코드]생산물1의 결과는 다음과 같다.

번호	이름
1	고길동
2	이순신

**해설** <구입자> 릴레이션에서 구입품 코드 속성을 제외하고 번호와 이름만 추출한다. 단 <생산품1> 릴레이션의 모든 튜플과 연관되어 있는 튜플만 추출한다.

**예제1** 구입자[이름, 구입품 코드 ÷ 이름, 생산품코드]생산품2의 결과는 다음과 같다.

번호
2

**해설** <구입자> 릴레이션에서 이름, 구입품 코드 속성을 제외하고 번호만 추출한다, 단 <생산품2> 릴레이션의 모든 튜플과 연관되어 있는 튜플만 추출한다.

### 3.일반 집합 연산자

- 집합 연산은 수학적 집합 이론에서 사용하는 연산자로서 릴레이션 연산에도 그대로 적용할 수 있다.
- 집합 연산 중 UNION, DIFFERENCE, INTERSECTION,을 처리하기 위해서는 병합조건을 만족해야 한다.
- 두 릴레이션 R과 S가 있을 때 각각 집합연산의 특징을 요약하면 다음과 같다.

연산자	기능 및 수학적 표현	카디널리티
합집합 UNION 연산기호 : $\cup$	<ul style="list-style-type: none"> <li>• 두 릴레이션에 존재하는 튜플의 합집합을 구하는 연산</li> <li>• 결과로 생성된 릴레이션에서 중복되는 튜플은 제거된다.</li> <li>• <math>R \cup S = \{t   t \in R \vee t \in S\}</math></li> <li>※ t는 릴레이션 R 또는 S에 존재하는 튜플이다.</li> </ul>	$ R \cup S  \leq  R  +  S $ 합집합의 카디널리티는 두 릴레이션 카디널리티의 합보다 크지 않다.
교집합 INTERSECTION 연산기호 : $\cap$	<ul style="list-style-type: none"> <li>• 두 릴레이션에 존재하는 튜플의 교집합을 구하는 연산</li> <li>• <math>R \cap S = \{t   t \in R \wedge t \in S\}</math></li> <li>※ t는 릴레이션 R 과 S에 동시에 존재하는 튜플이다.</li> </ul>	$ R \cap S  \leq \min\{ R ,  S \}$ 교집합의 카디널리티는 두 릴레이션중 카디널리티가 적은 릴레이션의 카디널리티보다 크지 않다.
차집합 DIFFERENCE 연산기호 : $-$	<ul style="list-style-type: none"> <li>• 두 릴레이션에서 존재하는 튜플의 차집합을 구하는 연산</li> <li>• <math>R - S = \{t   t \in R \wedge t \notin S\}</math></li> <li>※ t는 릴레이션 R에만 존재하고 S에는 존재하지 않는 튜플이다.</li> </ul>	$ R - S  \leq  R $ 차집합의 카디널리티는 릴레이션 R의 카디널리티 보다 크지 않다.

카티션 곱 CARTESIAN PRODUCT 연산기호 : $\times$	<ul style="list-style-type: none"> <li>• 두 릴레이션에 존재하는 튜플들의 결합된 정보를 구하는 연산</li> <li>• <math>R \times S = \{r \cdot s   r \in R \wedge s \in S\}</math></li> <li>※ <math>r \cdot s</math>는 R에 존재하는 튜플 r과 S에는 존재하는 튜플 s를 결합한 튜플이다.</li> </ul>	$ R \times S  =  R  \times  S $ 카디시언 프로덕트의 카디널리티는 두 릴레이션 카디널리티를 곱한 것과 같다.
--	--	--

#### <사원>

사번	이름
1	고길동
2	이순신

#### <직원>

사번	이름
2	이순신
3	둘리

**예제1**  $\pi_{\text{이름}}(\text{사원}) \cup \pi_{\text{이름}}(\text{직원})$  : <사원> 릴레이션과 <직원> 릴레이션에서 이름을 추출한 것의 합집합을 구한다.

**예제2**  $\pi_{\text{이름}}(\text{사원}) \cap \pi_{\text{이름}}(\text{직원})$  : <사원> 릴레이션과 <직원> 릴레이션에서 이름을 추출한 것의 교집합을 구한다.

**예제3**  $\pi_{\text{이름}}(\text{사원}) - \pi_{\text{이름}}(\text{직원})$  : <사원> 릴레이션과 <직원> 릴레이션에서 이름을 추출한 것의 차집합을 구한다.

**예제 4**  $\pi$ 이름(사원)  $\times$   $\pi$ 이름(직원) : <사원> 릴레이션과 <직원> 릴레이션에서 이름을 추출한 것의 카티션 프로덕트를 구한다.

기출문제

문제1 ) 괄호 안의 내용에 가장 적합한 번호를 답항 보기 에서 한 가지만 선택하시오.

관계대수는 관계형 데이터베이스에서 원하는 정보와 그 정보를 검색하기 위해서 어떻게 유도하는가를 기술하는 절차적인 언어로서 두 그룹의 연산이 있다.

첫 번째 그룹은 수학적 집합이론으로부터 나온 일반 집합의 연산이다. 릴레이션은 튜플의 집합이기 때문에 집합 연산은 기본적으로 릴레이션에 그대로 적용할 수 있다. 일반 집합 연산에는 합집합, 차집합, 교집합, 카티션 프로덕트가 있다.

두 번째 그룹은 관계 데이터베이스에 적용할 수 있도록 특별히 개발한 순수 관계 연산들이 있다.

순수 관계 연산자들 중 (① )연산은 릴레이션에서 주어진 조건을 만족하는 튜플 들을 선택하는 연산이다.

(① )연산의 결과 릴레이션은 주어진 릴레이션을 수평적으로 절단하여 그 일부를 가지고 구성한 것과 같기 때문에 수평연산이라고도 한다. 또한 (① )연산에서는 부울 연산자 AND, OR, NOT 등으로 결합시켜 여러 조건을 부여하여 이용할 수도 있다.

<학생>릴레이션

학번	이름	학년	학과
1001	나기태	4	컴퓨터
1002	송찬영	3	전기
1003	정연	1	컴퓨터
1004	박병화	4	건축
1005	김미나	1	컴퓨터

<학생> 릴레이션을 대상으로 <예1>과 같은 릴레이션을 생성하는 (① )연산의 표기형식은 (② )와 같이 나타낼 수 있다.

<예1>

학번	이름	학년	학과
1001	나기태	4	컴퓨터
1004	박병화	4	건축

순수 관계 연산들 중 (③ )연산은 릴레이션의 애트리뷰트 들을 대상으로 하는 연산으로, 연산에 명시된 애트리뷰트 값들만 선택한다, (③ )연산의 결과로 만들어진 릴레이션은 주어진 릴레이션의 몇몇 열들로 구성된 것으로 릴레이션의 수직적 부분 집합과 같으므로 수직연산이라고도 한다. (③ )연산의 결과 릴레이션은 중복된 튜플을 배제하고 생성된다. 예를들어, <학생>릴레이션을 대상으로 <예2>와 같은 결과 릴레이션을 생성하는 (③ )연산의 표기 형식은 (④ )(으)로 나타낼 수 있다.

<예2>

학년	학과
4	컴퓨터
3	전기
1	컴퓨터
4	건축

관계대수 연산을 이용하여 <예3>과 같이 <학생>릴레이션에 대해 학과가 “컴퓨터”이고, 학년이 1인 학생의 학번과 이름으로 구성된 릴레이션 표기형식은  $\pi_{\text{학번},\text{이름}}(\sigma_{\text{학과}=\text{“컴퓨터”} \wedge \text{학년}=1}(\text{학생}))$ 으로 나타낼 수 있다.

<예3>

학번	이름
1003	정연
1005	김미나

또한, 순수 관계 연산에는 공통 속성을 중심으로 2개의 릴레이션을 하나로 합쳐서 새로운 릴레이션을 만드는 (⑤ )연산 등이 있고, (⑤ )연산의 연산자 기호는 “ $\bowtie$ ”를 사용한다.

## 답항보기

1	JOIN	2	UNION	3	PROJECT	4	$\pi$ 학년,학과(학생)	5	DIFFERENCE
6	CARTESIAN	7	$\sigma$ 학년=4(학생)	8	SELECT	9	제약(Constraint)	10	DIVISION

문제2 ) 괄호 안의 내용에 가장 적합한 번호를 답항 보기 에서 한 가지만 선택하시오.

관계 대수는 질의에 대한 결과를 생성하기 위해 연산의 순서를 어떻게 해야 하는지 명시하는 절차 언어이다. 관계 대수의 연산은 하나의 릴레이션이나 여러 개의 릴레이션으로 부터 일정한 조건에 맞는 튜플들을 찾아 새로운 릴레이션을 만들거나 여러 개의 릴레이션들을 통합하여 새로운 하나의 릴레이션을 만드는 연산을 수행하기 때문에 연산의 피연산자가 모두 릴레이션이고 그 결과도 릴레이션이다.

관계 대수의 연산 중 조인 연산에 대하여 살펴 보자.

조인(Join)연산은 두 개의 릴레이션으로 부터 연관된 튜플(Tuple)들을 결합하는 연산이다. 결과만 비교 한다면 조인 연산은 (① ) 연산을 수행한 후, (② ) 연산을 적용한 것과 같다. (② ) 연산은 조건을 만족하는 릴레이션의 수평적 부분 집합으로 구성되며, 연산자의 기호는 그리스 문자 시그마를 사용한다.

조인 연산은 조인을 효율적으로 시행하는 여러 가지 알고리즘이 개발되었다.

조인을 비교 연산자 'θ'로 일반화하여, 'θ'로 표현될 수 있는 조인을 (③ ) 조인이라 한다. 두 릴레이션 R(A1,A2,...,An)과 S(B1,B2,...,Bm)의 (③ ) 조인의 결과는 차수(Degree)가 (n+m)이고, 애트리뷰트가 (A1,A2,...,An,B1,B2,...,Bm)이며, 조인 조건을 만족하는 튜플들로 이루어진 집합이다.

(③ )는(은) 비교 연산자 중의 하나이다. (③ ) 조인을 수행하기 위해서 양쪽 릴레이션의 애트리뷰트(일반적으로 기본 키와 외래키 관계)들이 (③ ) 조인 조건을 만족하는 튜플들만 골라낸다.

이것이 (① ) 연산과 조인 연산의 주요 차이점이다. (① ) 연산은 릴레이션 R에 존재하는 튜플과 릴레이션 S에 존재하는 튜플들을 모두 접속시킨 새로운 결과 릴레이션을 만드는 연산이다. 따라서, (① ) 연산의 결과로 만들어지는 릴레이션의 차수는 피연산자 R과 S 릴레이션의 차수의 합과 같고, 카디널리티는 R의 카디널리티와 S 카디널리티의 곱과 같다.

동등조인은 (③ ) 조인 중에서 비교연산자가 '='인 조인이다.

동등조인의 형식은 다음과 같이 표현한다.

$R(\text{④})_{R.attribute=S.attribute}$

조인의 기호로는 (④ )를(을) 사용한다.

다음과 같은 두 릴레이션이 있다고 가정하자.

EMPLOYEE

EMPNO	EMPNAME	DNO
1234	김원호	2
1235	박영주	1
1236	이수미	3
1237	조민수	2
1238	최종인	3

DEPARTMENT

DEPTNO	DEPTNAME
1	영업
2	기획
3	개발
4	총무

'EMPLOYEE(④ )  $DNO=DEPTNO$  DEPARTMENT' 로 동등 조인한 결과는 다음과 같다.

EMPNO	EMPNAME	DNO	DEPTNO	DEPTNAME
1234	김원호	2	2	영업
1235	박영주	1	1	기획
1236	이수미	3	3	개발
1237	조민수	2	2	총무
1238	최종인	3	3	개발

동등 조인에서는 두 릴레이션에서 조인 조건에 사용된 두 애트리뷰트가 결과 릴레이션에 포함된다. 그러나, 이 두 애트리뷰트는 이름이 다를 수 있어도 결과 릴레이션의 각 튜플에서 두 애트리뷰트 값이 같으므로 둘 중 하나의 애트리뷰트만 포함시켜도 무방하다. 즉 속성의 중복은 결과에 아무런 의미를 주지 못하기 때문에 제거되어 연산을 수행하는 것이 바람직하다, 동등조인의 결과 릴레이션에서 중복된 속성을 제거하여 수행하는 연산을 (⑤ )조인이라고 한다.

즉 동등조인에서 중복 속성 중 하나가 제거된 것이 (⑤ )조인이다.

(⑤ )조인의 핵심은 두 릴레이션의 공통된 속성을 매개체로 하여 두 릴레이션의 정보를 ‘관계’로 묶어 내는 것이다. 일반 테이블에서는 보통 외래키가 매개체 역할을 담당한다.

**답항보기**

1	JOIN	2	UNION	3	PROJECT	4	⋈	5	DIVISION
6	Cartesian Product	7	선택(Select)	8	세타(Theta)	9	관계연산자	10	자연(Natural)