Chap 4 Algebraic Query Languages (대수 쿼리 언어)

◎ An Algebraic Query Languages (대수 쿼리 언어)

* 관계 대수 (Relational algebra)

- SQL의 원형
- 기존의 릴레이션에서 새로운 릴레이션을 만들어 낼수 있는 간단하고도 강력한 방법을 제공.

* 관계 대수의 장점

- C 나 Java 보다 찾기 편하다
- 컴파일러를 통해 최적화 됨

* 관계 대수의 연산

- 일반적인 집합연산 ($\overset{\cup}{\text{Union}}$, $\overset{\cap}{\text{Intersection}}$, $\overset{-}{\text{difference}}$)
- 관계의 일부를 제거하는 연산 ($\overset{\sigma}{\text{Selection}}$, $\overset{\pi}{\text{projection}}$)
- 두 관계의 튜플을 결합하는 연산 ($\overset{\times}{\text{Certesian product}}$, $\overset{\bowtie}{\text{Join}}$)
- 이름 변경 연산 ($\overset{\rho}{\text{renaming}}$)

중복제거 : $\delta(R)$
확장프로젝션 : $\pi_L(R)$
sum , AVG . MIN . MAX
그룹화 : $\gamma_L(R)$
$\bowtie$ $\bowtie_C$
정렬 : $\tau_L(R)$
외부조인 : $\overset{\circ}{\bowtie}_L$ $\overset{\circ}{\bowtie}_R$
$\overset{\circ}{\bowtie}$ $\overset{\circ}{\bowtie}_C$ 중

- 의존연산자 ($\overset{\cap}{\text{intersection}}$, $\overset{중}{\text{Theta - Join}}$ $\bowtie_C$, $\overset{\bowtie}{\text{Joins}}$)
- 독립연산자 ($\overset{\sigma}{\text{selection}}$, $\overset{\pi}{\text{Projection}}$, $\overset{\cup}{\text{Union}}$, $\overset{-}{\text{difference}}$, $\overset{중 \ \times}{\text{Certesian product}}$, $\overset{\rho}{\text{renaming}}$)
- 기본연산자 ($\overset{\sigma}{\text{Selection}}$. $\overset{\pi}{\text{Projection}}$, $\overset{\cup}{\text{Union}}$, $\overset{-}{\text{Difference}}$. $\overset{\times}{\text{Certesian product}}$)

* 셋연산자 : Union (∪) , 9ntersection (∩) , Difference (−)

- R과 S의 스키마 , 즉 동일한 애트리뷰트 집합으로 동일하게 정렬 되어야 함.

* 투영 (Projection , π) ⇒ $\pi_{A1, A2, \cdots An}(R)$

- R의 A1 , A2 , ⋯ A_n 속성만 가진 관계를 생성

(ex)

Movies

title	year	length	genre
...	...	...	...

$\pi_{\text{title. year}}$ (Movies)

title	year
...	...

* 선택 (Selection , σ) ⇒ $\sigma_C(R)$

— C 조건을 만족하는 튜플들의 집합

(예) Movies

title	year	length
StarWars	1977	124
GQ	1999	104

$\sigma_{length > 120}$ (Movies)

title	year	length
Starwars	1977	124

* 곱 (Cartesian Product , ×) ⇒ R × S

— R과 S의 튜플 쌍의 집합

(예)

A	B
1	2

R

B	C	D
2	5	6
4	7	8

S

A	R.B	S.B	C	D
1	2	2	5	6
1	2	4	7	8

R × S

* 자연조인 (Natural Joins , ⋈) ⇒ R ⋈ S

— R과 S의 공통속성에 대해 겹치는 튜플 쌍만의 집합, 중복된 열은 제거

(예)

A	B	C
1	2	5
3	4	7
3	4	8

R

B	C	D
2	5	6
4	7	8
9	10	11

S

A	B	C	D
1	2	5	6
3	4	7	8

R ⋈ S

* 세타조인 (Theta Joins . $\bowtie_C$) ⇒ σ_C (R×S) = $R \bowtie_C S$

— C 조건을 만족하는 튜플들만 선택

(예)

A	B	C
1	2	3
6	7	8
9	7	8

U

B	C	D
2	3	4
2	3	5
7	8	10

V

A	U.B	U.C	S.B	S.C	D
1	2	3	2	3	4
1	2	3	2	3	5
1	2	3	7	8	10
6	7	8	7	8	10
9	7	8	7	8	10

$U \bowtie_{A<D} V$

* 재정의 (Renaming , ρ) ⇒ $\rho_{S(A1, A2, \cdots An)}(R)$

— R의 이름을 S로 바꾸고, 애트리뷰트이름을 A1, A2 ... An 으로 재정의 한다

(예)

A	B
1	2
3	4

R

B	C	D
2	5	6
4	7	8

S

A	B	X	C	D
1	2	2	5	6
3	4	2	5	6
1	2	4	7	8
3	4	4	7	8

$R \times \rho_{S(X, C, D)}(S)$

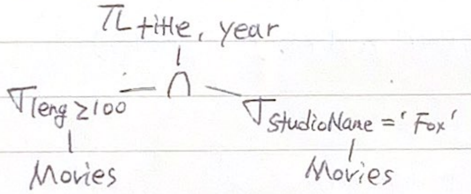
* 질의 작성을 위한 연산의 조합.

(예) 상영시간이 100분 이상인 Fox 사에서 제작된 영화의 제목과 연도를 찾아라

— 괄호를 이용한 관계 대수 표현

$$\overset{④}{\boxed{\pi_{title, year}}}\left(\overset{②}{\boxed{\sigma_{length \geq 100}}}\left(\overset{①}{\boxed{Movies}}\right) \overset{③}{\boxed{\cap}} \overset{②}{\boxed{\sigma_{studioNane = 'Fox'}}}\left(\overset{①}{\boxed{Movies}}\right)\right)$$

— 트리를 이용한 관계 대수 표현



◎ 릴레이션에 대한 제약

* 관계 무결성 제약 조건

(예) Movie 릴레이션에서 각 영화의 제작자는 반드시 MovieExec 릴레이션에 나타나야한다

- $\pi_{producerC\#}(Movies) \subseteq \pi_{certificate\#}(MovieExec)$
- $\pi_{producerC\#}(Movies) - \pi_{certificate\#}(MovieExec) = \emptyset$

Movies의 prod 도 MovieExec의 cert 에서 선택 ([redacted]게 관계무결성)

* 둘 이상의 애트리뷰트가 참여하는 참조무결성 제약

(예) Star In 릴레이션에 나타나는 영화는 Movie 릴레이션에도 반드시 나타나야 한다.

$\pi_{movieTitle,\ movieYear}(StarIn) \subseteq \pi_{title,\ year}(Movies)$

* 키제약 (함수적 종속성)

(ex) Name 이 key 값인 것을 표시하시오. (Moviestar의 name. address관계)

$\sigma_{\boxed{MS1.name = MS2.name}\ \boxed{AND}\ \boxed{MS1.address \neq MS2.address}}(MS1 \times MS2) = \emptyset$

단. $MS1 = \rho_{MS1(name,\ address)}(Moviestar)$

$MS2 = \rho_{MS2(name.\ address)}(Moviestar)$

* 도메인 제약

(ex) Moviestar 릴레이션에서 gender 애트리뷰트 값으로 'F'와 'M'만 허용.

$\sigma_{gender \neq 'F'\ AND\ gender \neq 'M'}(MovieStar) = \emptyset$

◎ 관계 대수의 확장연산자. (Extended Operators of Relational Algebra)

* 중복제거 (Duplicate elimination , δ) ⇒ $\delta(R)$

(ex)

A	B
1	2
3	4
1	2

R

A	B
1	2
3	4

$\delta(R)$

(=bag 에서 Set 으로 전환과정)

* 프로젝션 연산자의 확장 (π_L) ⇒ $\pi_L(R)$

(ex)

A	B	C
0	1	2
3	4	5

R

A	X
0	3
3	9

$\pi_{A,\ B+C \rightarrow X}(R)$

X	Y
1	1
1	1

$\pi_{B-A \rightarrow X,\ C-B \rightarrow Y}(R)$

* 집단값 연산자 (Aggregation operators, SUM, AVG. Min, MAX, COUNT) ⇒ SUM(B)

(예)

A	B
1	2
3	4
1	2

SUM(A) = 1+3+1 = 5

AVG(B) = (2+4+2)/3 = 2.666

COUNT (A) = 3

* 그룹화 연산자 (Grouping operator, γ_L) ⇒ $\gamma_L(R)$

- L을 기준으로 그룹화를 진행한다.

(ex) Movies

StudioName	length
Disney	97
Disney	120
MGM	150
MGM	95

$\gamma_{StudioName,\ SUM(length) \rightarrow lengthsum}(Movies)$

Movies

StudioName	length Sum
Disney	217
MGM	245

* 정렬 연산자 (Sorting operator, τ_L) ⇒ $\tau_L(R)$

- L에 따라서 튜플들을 정렬

* 외부조인

- 허상 튜플 (조인이 되지 못한 튜플) 도 조인결과에 포함

* 자연 외부조인 ($\overset{\circ}{\bowtie}$) ⇒ $R \overset{\circ}{\bowtie} S$

* 좌측 외부조인 ($\overset{\circ}{\bowtie}_L$) ⇒ $R \overset{\circ}{\bowtie}_L S$

* 우측 외부조인 ($\overset{\circ}{\bowtie}_R$) ⇒ $R \overset{\circ}{\bowtie}_R S$ (⊥ : NULL Value)

* 세타 외부조인 ($\overset{\circ}{\underset{C}{\bowtie}}$) ⇒ $R \overset{\circ}{\underset{C}{\bowtie}} S$

- 조건 C를 만족하는 외부조인

(ex)

A	B	C
1	2	3
4	5	6
7	8	9

U

B	C	D
2	3	10
2	3	11
6	7	12

V

A	B	C	D
1	2	3	10
1	2	3	11
4	5	6	⊥
7	8	9	⊥
⊥	6	7	12

$U \overset{\circ}{\bowtie} V$

A	B	C	D
1	2	3	10
1	2	3	11
4	5	6	⊥
7	8	9	⊥

$U \overset{\circ}{\bowtie}_L V$

A	B	C	D
1	2	3	10
1	2	3	11
⊥	6	7	12

$U \overset{\circ}{\bowtie}_R V$

A	U.B	U.C	V.B	V.C	D
4	5	6	2	3	10
4	5	6	2	3	11
7	8	9	2	3	10
7	8	9	2	3	11
1	2	3	⊥	⊥	⊥
⊥	⊥	⊥	6	7	12

$U \overset{\circ}{\underset{A > V.C}{\bowtie}} V$

● Relational Operations On Bags (백에 대한 관계연산)

* 관계 DB 에서 Bag

— Set 과 달리 값의 중복이 허용된다.

* 합집합, 교집합, 차집합 (Union, intersection, difference of bags, U. n. -)

(ex) $R = \{2.2.3\}$ $R \cup S = \{2.2.2.3.3.3\}$

$S = \{2.3.3\}$ $R \cap S = \{2.3\}$

$R - S = \{2\}$

A	B
1	2
3	4
1	2

R

A	B
1	2
3	4
3	4

S

A	B
1	2
1	2
1	2
3	4
3	4
3	4

RUS

A	B
1	2
3	4

R∩S

A	B
1	2

R-S

A	B
3	4

S-R

* 투영 (Projection of bags, π) ⇒ $\pi_{A.B}$

(예)

A	B	C
1	2	5
3	4	6
1	2	7
1	2	7

R

A	B
1	2
3	4
1	2
1	2

$\pi_{A,B}(R)$

* 선택 (Selection on bags, σ_L) ⇒ $\sigma_{C \geq 6}(R)$

(예)

A	B	C
1	2	5
3	4	6
1	2	7

R

A	B	C
3	4	6
1	2	7

$\sigma_{C \geq 6}(R)$

* 곱 (Product of bags , X) ⇒ R×S

(예)

A	B
1	2
1	2

R

B	C
2	3
4	5

S

A	R.B	S.B	C
1	2	2	3
1	2	4	5
1	2	2	3
1	2	4	5

R×S

* 조인 (Join of bags ; ⋈) ⇒ $R \bowtie_{R.B<S.B} S$

(예)

A	B
1	2
1	2

R

B	C
2	3
4	5
4	5

S

A	B	C
1	2	3
1	2	3

R ⋈ S

A	R.B	S.B	C
1	2	4	5
1	2	4	5
1	2	4	5
1	2	4	5

$R \bowtie_{R.B<S.B} S$

cf.

DB : DBMS를 이용하여 관리되어지는 데이터 수집.

DBMS : 대용량의 데이터를 효율적으로 생성 및 관리하고 오랜시간동안 안전하게 관리하기위해 사용하는 전문 S/W

데이터모델 : 데이터의 논리적인 구조를 지정하며 데이터와 데이터간의 관계를 논리적으로 설명하는 개념적도구.

E/R모델의 시작 : DB가 필요하다고 생각한 ideas 때문에 시작

E/R모델의 목적 : (개념설계) High-level-design

DB 디자인 : 데이터의 논리적인 구조와 제약조건을 지정

DB디자인을 위한 관계모델 : 여러 개념이 아닌 하나의 개념을 사용하여 단순하고 효율적이게 표현하는 것이 관계 모델의 장점

E/R 모델 : 데이터의 논리적구조를 엔티티, 애트리뷰트, 관계성을 가지고 표현하는 모델

E/R 다이어그램 : E/R모델을 시각적으로 표현하기 위한 표기법.

스키마 : 데이터의 논리적인 구조

릴레이션모델 : 데이터의 논리적인 구조를 관계라는 개념을 사용해 표현한 모델
수학의 집합론에 근거하면서 릴레이션의 연산을 논리적으로 표현할수 있어야 한다.
관계형 DB에 직접구현 할수 있도록 DB구조를 정의해야한다.

스키마 : 릴레이션의 이름과 릴레이션의 애트리뷰트들의 집합.