질의어

수식 : 항으로 숫자가 나온다. (변화 시킨다)

관계대수 : 항으로 Relation이 나오는 것 (관계대수식)

항에 주어진 Relation에 무엇이 적용이 되어 어떠한 Relation이 나올 것이다.

* 관계대수 연산자
* 언어에 따르는 연산을 수행하는 연산자
* 관계대수의 확장
* 사용자가 원하는 데이터를 마음대로 찾을 수 있도록 언어를 만든다. (질의어)

관계대수와 SQL

관계해석 : 원하는 데이터만 명시하고 질의를 어떻게 수행할 것인가는 명시하지 않는

{선언적인 언어}

뭐할 거라고만 얘기하고 어떻게 할건데? 🡺 선언적인 언어

(어떻게 할 것인지는 DBMS가 알아서 해라~~)

(관계해석은 쉬운 언어이다.)

관계대수 : SQL의 기반(SQL의 이론적인 기초)

(관계대수는 선언적인 언어, 관계해석에 비해서 절차적인 언어이다.) => 문제로 나올 수 있음(OX문제)

* 관계대수
* 이항연산자 : 두가지 - 단항연산자 : 한가지

관계대수에서도 이것들이 존재한다.

* 필수적인 연산자
* 셀렉션 : 튜플들 중에서 내가 원하는 튜플들만 잘라내는 것!

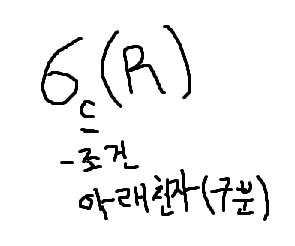
(입력 출력 릴레이션 사이의 스키마에 변화가 있는가? 없다.)

* 프로젝션 : 각각의 애트리뷰트 여러 개 중에서 몇 개의 애트리뷰트를 뽑아서 새로운 튜플튜 만드는 것. (입력 출력 릴레이션 사이의 스키마에 변화가 있는가? 있다.)
* 합집합 : 여러 튜플들을 합친 것
* 교집합(차집합) : 겹치는 튜플들
* 곱집합 (카티션 곱)
* 편의를 위해 유도된 연산자 (필수적인 연산자만 사용해도 됨)

Ex) 곱하기 (더하기를 여러 번하는 것)

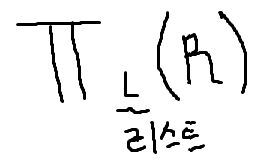
* 디비전 (나누기) : 좀 애매하다..?
* 셀렉션

표기형식



C 조건에 맞는 튜플들을 가지고 온다.

* 프로젝션



L리스트에 표기된 애트리뷰트만 추출함

연습문제를 한번 풀어보자. 많이 헷갈리기 때문에…

셀렉션과 프로젝션을 같이 쓰다 보면 교환법칙이 성립하는 경우와 성립하지 않는 경우로 나뉘게 된다. 그러므로 잘 생각하고 식을 써야 한다.

(셀렉션을 먼저 하는 것이 성능상 좋다!) 만약에 일억개 있으면 그거 먼저 인덱스 접근으로 찾아서 쪼개는게 더 빠르군!