

커드의 12규칙

- 규칙 0: 시스템은 "데이터베이스" 뿐만 아니라 "관리 시스템"으로써도 "관계형 모델" 자격을 얻어야 한다.

어떤 시스템이 관계형 데이터베이스 매니지먼트 시스템(RDBMS)으로서의 자격을 얻고자 한다면, 그 시스템은 그 시스템의 "관계형" 수단(만)을 "데이터베이스"를 "관리"하는 데 사용하여야 한다.

- 규칙 1: "정보 규칙":

데이터베이스 내의 모든 정보는 한 가지 방법으로만 표현되어야 한다. 즉 테이블 내 행 안의 열 안에 들어간 값으로 표현되어야 한다.

- 규칙 2: "보장된 접근 규칙":

모든 데이터는 모호함이 없이 접근되어야 한다. 이 규칙은 "기본 키"(primary keys)가 요구된다는 말을 다르게 표현한 것이다. 데이터베이스 내의 모든 개별적인 스칼라 값은, 그것을 포함하는 테이블의 이름, 열의 이름 그리고 행의 기본 키를 지정하면, 논리적으로 어드레스로 불러낼 수 있어야(addressable) 한다고 언급하고 있다.

- 규칙 3: "널 값의 체계적인 처리"

DBMS는 각 필드에 대해 "널(null)" 값을 갖게 할 수 있어야 한다. (다른 말로 비어 있게 할 수 있어야 한다는 말이다.) 특히, DBMS는 "정보가 없거나 활용할 수 없는 정보임"이라는 것에 대한 표현을 지원하되, 체계적이고, 다른 모든 통상적인 값들과는 구별되게 할 수 있어야 한다. (예를 들어, 수에 대해 "0이나 그 이외의 수들과는 구별됨"이라는 표현) "정보가 없거나 활용할 수 없는 정보임"이라는 것에 대한 표현은 데이터 타입도 독립적이어야 한다. 이 규칙은 DBMS 내에서는 그러한 표현들이 체계적인 방법으로 다루어져야 한다는 것을 암시한다.

- 규칙 4: "관계형 모델에 기반한 액티브 온라인 데이터베이스 카탈로그":

시스템은 온라인, 인라인 관계형 카탈로그를 지원해야 한다. 이 관계형 카탈로그는 접근 허용된 사용자들에게 통상의 쿼리 언어를 통해 접근 가능하여야 한다. 다시 말해, 사용자들은 데이터베이스의 데이터를 접근하는 때 쓰는 것과 동일한 쿼리 언어를 이용하여 데이터베이스의 구조(카탈로그)를 접근할 수 있어야 한다.

- 규칙 5: "종합적인 데이터 보조언어 규칙"

시스템은 적어도 하나의 관계형 언어를 지원하되, 그 관계형 언어는 다음의 특징을 가져야 한다.

1. 선형 문법을 가진다.
2. 대화형(interactively)으로 이용될 수도 있고, 애플리케이션 프로그램 안에서 이용될 수 있다.
3. 데이터 정의 동작(data definition operations)을 지원해야 한다. (뷰(view)에 대한 정의도 할 수 있어야 한다.) 데이터 조작 동작(data manipulation operations)을 지원해야 한다. 보안과 무결성 제약조건 그리고 트랜잭션 관리 동작(비긴, 커밋, 롤백)을 지원해야 한다.

- 규칙 6: "뷰 갱신 규칙"

이론적으로 갱신 가능한 모든 뷰(views)는 시스템에 의해서 갱신 가능하여야 한다.

- 규칙 7: "고급 삽입, 갱신, 제거":

시스템은 집합에 대해 한 번에(set-at-a-time) "삽입", "갱신", "제거" 동작을 지원해야 한다. 데이터베이스에서 여러 개의 행 또는 여러 개의 테이블에서 나온 데이터로 생성한 집합 내에서 다시 데이터를 획득할 수 있어야 한다는 말이다. 다시 말해, 이 규칙은 한 개의 테이블이나 한 개의 행에서 "삽입", "갱신", "제거" 동작이 행해지는 것이 아니라 여러 개의 테이블이나 여러 개의 행에서 그러한 동작이 행해져야 한다는 말이다.

- 규칙 8: "물리적 데이터 독립성":

물리적 레벨에서(데이터가 어떻게 저장되든지, 배열에 저장되든지 링크드 리스트에 저장되든지) 변경이 일어났더라도, 그 구조에 기반한 애플리케이션은 변경되어선 안 된다.

- 규칙 9: "논리적 데이터 독립성":

논리적 레벨에서(테이블, 열, 행,...) 변경이 일어났더라도, 그 구조에 기반한 애플리케이션은 변경되어선 안 된다. 논리적 데이터 독립성은 물리적 데이터 독립성보다 달성하기 어렵다.

- 규칙 10: "무결성 독립성":

무결성 제약조건들은 데이터베이스 카탈로그에 저장되어야 하며 애플리케이션 프로그램들과는 별도로 규정되어야 한다. 기존의 애플리케이션에 영향을 주지 않고 무결성 제약조건들을 변경하는 것이 가능해야 한다.

- 규칙 11: "분산 독립성":

여러 장소에 대한 데이터베이스의 분산 문제는 데이터베이스의 사용자에게 영향을 주지 않아야 한다. 다음과 같은 상황 하에서도, 기존의 동작하던 애플리케이션들은 계속 그대로 동작해야 한다:

1. 분산을 지원하는 DBMS가 처음 도입될 때, 그리고
2. 기존의 분산된 데이터가 시스템에 다시 재배치되어 분산될 때.

- 규칙 12: "무전복 규칙"

만약 시스템이 "저급"(record-at-a-time) 인터페이스를 제공한다면, 그 인터페이스는 시스템을 파괴하는 데 쓰여서는 안 된다. 예를 들어 관계형 "보안"/"무결성" 제약 등을 파괴하는 데 쓰여서는 안 된다.

릴레이션

속성 ↓	속성 ↓	속성 ↓	속성 ↓	
학번	이름	성별	학과	릴레이션 스키마
190001	홍길동	남	컴퓨터	튜플
190002	최민수	남	건축	튜플
190003	김미나	여	전기	튜플
190004	이영희	여	컴퓨터	튜플

성별의 도메인

릴레이션 인스턴스

예시) '학생' 릴레이션의 릴레이션 스키마, 릴레이션 인스턴스, 속성, 튜플

릴레이션 스키마

- 관계 모델에서 스키마는 데이터베이스 구조의 제약조건에 관한 전반적인 명세를 의미
- 개체와 개체의 특성을 나타내는 속성, 개체 간의 관계, 이들의 제약조건을 기술 한것
- 릴레이션 스키마 명세
 - 릴레이션 이름

- 필드 또는 열, 또는 속성의 이름
- 각 속성의 도메인 이름

릴레이션 인스턴스

- 레코드라고도 불리는 튜플의 집합
- 튜플
 - 릴레이션 스키마에서 정의된 각 속성(필드)로 구성되며, 하나의 데이터 묶음을 나타냄
 - 릴레이션 인스턴스에서 각 튜플을 하나의 행으로 구성
- 릴레이션 인스턴스의 튜플 수 : 카디널리티
- 릴레이션 인스턴스의 속성 수 : 차수
- 인스턴스의 튜플 순서는 중요하지 않음

도메인

- 하나의 속성이 가질 수 있는 동일한 유형의 원자값들의 집합
- 릴레이션 스키마는 릴레이션 인스턴스의 각 필드의 도메인을 명세(도메인 제약조건)
- 프로그래밍 언어 관점에서 필드의 데이터 타입을 의미 ``

관계 데이터베이스

- 관계 데이터베이스 : 서로 다른 릴레이션 이름을 가진 릴레이션들의 모임
- 관계 데이터베이스 스키마 : 데이터베이스에 속한 릴레이션의 스키마들의 집합

TIP!

- mysql은 전부다 멀티 바이트
- char은 고정바이트, varchar은 가변바이트
- varchar은 느리지만 저장공간 낭비가 없고
- char은 빠르지만 저장공간 낭비가 있다