[ 게임 데이터베이스 프로그래밍 ]

**관계형 및 비 관계형 데이터베이스에 대해 정리**

인하대학교 미래인재개발원 문화콘텐츠프로그래밍

김예슬

**1) 관계형 데이터베이스**

- Relational Database

- 키(Key)와 값(Value)들의 간단한 관계를 테이블 화 시킨 매우 간단한 원칙의 전산 정보

데이터베이스

- 1970년 에드거 F.커드가 제안한 데이터 관계형 모델에 기초하는 디지털 데이터베이스

- 데이터를 컬럼(Column)과 로우(Row / 레코드, 튜플)를 이루는 하나 이상의 테이블

(또는 관계)로 정리하며, 고유키가 각 로우를 식별

- 일반적으로 각 테이블/관계는 하나의 엔티티 타입을 대표하고, 로우는 그 엔티티 종류

의 인스턴스를 대표하며 컬럼은 그 인스턴스의 속성이 되는 값들을 대표

- Key

테이블의 각 로우에는 저만의 고유 키가 있음

한 테이블 안의 로우는 다른 테이블들의 로우로 연결이 가능

이는 연결된 로우의 고유 키를 위한 컬럼을 추가함으로써 이루어짐(외래 키)

- 관계

관계는 테이블 간에 둘 다 존재

이 관계들은 일대일, 일대다, 다대다 세 가지 형태로 이루어짐

대부분의 관계형 데이터베이스들은 각 로우의 각 컬럼이 하나의 값만을 보유할 수 있도록 설계

- 트랜잭션

데이터베이스 관리 시스템이 효율적이고 정확하게 운용되기 위해서는 ACID 트랜잭션을 갖추고 있어야 함

- 용어

|  |  |  |
| --- | --- | --- |
| **SQL 용어** | **관계형 데이터베이스 용어** | **설명** |
| [**로우**](https://ko.wikipedia.org/wiki/%EB%A1%9C%EC%9A%B0_(%EB%8D%B0%EC%9D%B4%ED%84%B0%EB%B2%A0%EC%9D%B4%EC%8A%A4)) | [**튜플**](https://ko.wikipedia.org/wiki/%ED%8A%9C%ED%94%8C) 또는 [**레코드**](https://ko.wikipedia.org/wiki/%EB%A1%9C%EC%9A%B0_(%EB%8D%B0%EC%9D%B4%ED%84%B0%EB%B2%A0%EC%9D%B4%EC%8A%A4)) | 하나의 항목을 대표하는 데이터 |
| [**컬럼**](https://ko.wikipedia.org/wiki/%EC%BB%AC%EB%9F%BC_(%EB%8D%B0%EC%9D%B4%ED%84%B0%EB%B2%A0%EC%9D%B4%EC%8A%A4)) | **속성**(어트리뷰트, attribute) 또는 **필드** | 튜플의 이름 요소 (예: "주소", "태어난 날짜") |
| [**테이블**](https://ko.wikipedia.org/wiki/%ED%85%8C%EC%9D%B4%EB%B8%94_(%EB%8D%B0%EC%9D%B4%ED%84%B0%EB%B2%A0%EC%9D%B4%EC%8A%A4)) | [**관계**](https://ko.wikipedia.org/wiki/%EA%B4%80%EA%B3%84_(%EB%8D%B0%EC%9D%B4%ED%84%B0%EB%B2%A0%EC%9D%B4%EC%8A%A4)) 또는 **기초 관계변수**(base relvar) | 같은 속성을 공유하는 튜플의 모임. (컬럼이나 로우의 모임.) |
| [**뷰**](https://ko.wikipedia.org/wiki/VIEW_(SQL)) 또는 [**결과 집합**](https://ko.wikipedia.org/w/index.php?title=%EA%B2%B0%EA%B3%BC_%EC%A7%91%ED%95%A9&action=edit&redlink=1) | **파생 관계변수**(derived relvar) | 튜플들의 모임. ([질의어](https://ko.wikipedia.org/wiki/%EC%A7%88%EC%9D%98%EC%96%B4)에 응답하는 RDBMS의 데이터 보고서) |

- 관계 또는 테이블

관계 : 속성을 지닌 튜플의 모임으로 정의

튜플 : 오브젝트와 그 오브젝트의 정보 대표

오브젝트 : 일반적으로 물리적인 오브젝트나 개념을 가리킴

관계 : 일반적으로 여러 열이나 컬럼으로 조직된 테이블로 기술하기도 함

- 정규화

관계형 모델의 필수적인 부분

단순하지 않은 도메인(비원자성 값)과 데이터 중복을 제거하기 위해 설계된 프로시저들의 모임을 아우르며, 데이터 조작의 변칙과 데이터 무결성의 손실을 예방

- 장점

명확하게 정의된 스키마, 데이터 무결성 보장

관계를 통해 각 데이터를 중복없이 한 번만 저장할 수 있음

- 단점

상대적으로 덜 유연하며, 데이터 스키마는 미리 알고 계획해야 함

JOIN문이 많은 매우 복잡한 쿼리가 만들어 질 수 있음

수평 확장이 어렵고, 보통 수직 확장만 가능

**2) 비 관계형 데이터베이스**

- NoSQL 데이터 베이스

- 스키마, 관계가 존재하지 않음

- 장점

스키마가 없기 때문에 유연성이 높음

데이터는 애플리케이션에 필요한 형식으로 저장

수직 및 수평 확장이 가능하므로 데이터베이스가 애플리케이션에서 발생시키는 모든 읽기 / 쓰기 요청 처리 가능

- 단점

유연성 때문에 데이터 구조 결정이 늦어질 수 있음

복사된 데이터가 변경되면 여러 콜렉션과 문서를 수정해야 함