



데이터베이스

목차

A table of contents

1

정의

2

특징

3

종류

4

스키마란

5

SQL과 NoSQL



정 의

1

1 데이터베이스 정의

데이터 베이스란

- 여러 사람이 공유하여 사용할 목적으로 체계화해 통합, 관리하는 데이터의 집합이다

특 징

2

1 데이터베이스의 특징

특징	설명
실시간 접근성	데이터 베이스는 실시간으로 서비스가 가능합니다.
계속적인 변화	데이터의 값은 시간에 따라 변화합니다 삽입,삭제,수정 등의 작업을 통해 데이터의 값을 저장합니다.
동시 공유	여러 사용자에게 동시에 서비스가 가능합니다.
내용에 따른 참조	데이터 베이스에 저장된 데이터는 데이터의 물리적 위치(서버의 위치)가 아니라 데이터 값에 따라 참조됩니다.

종 류

3

1 데이터베이스의 종류

종류	설명
계층형 데이터베이스	계층 구조로 데이터 관리, 최초의 현대적 데이터베이스
관계형 데이터베이스	2차원 형식으로 데이터관리, 가장 널리 사용됨 예) 오라클, IBM, DB2, PstgreSQL 등
객체지향형 데이터베이스	상용화가 되지 못함
NOSQL 데이터베이스	Not Only SQL 최근 각광 받는 중 관계형 데이터베이스의 기능 일부를 삭제

스키마란

4

1 스키마란?

스키마란?

- 데이터베이스에서 자료의 구조, 자료의 표현 방법, 자료 간의 관계를 형식 언어로 정의한 구조이다.
- 데이터베이스 관리 시스템(DBMS)이 주어진 설정에 따라 데이터베이스 스키마를 생성하며, 데이터베이스 사용자가 자료를 저장, 조회, 삭제, 변경할 때 DBMS는 자신이 생성한 데이터베이스 스키마를 참조하여 명령을 수행한다.

- 예시

고객 번호	이름	나이	주소
INT	CHAR(10)	INT	CHAR(20)

1 스키마의 계층

외부 스키마

사용자 뷰를 가리킵니다.
개인의 입장, "서브스키마"
라고도 한다
하나의 외부스키마를 여
럿이 공유가능하며 하나
의 DB시스템에 여러 개의
외부 스키마가 존재가 가
능합니다.

>>

개념 스키마

시스템 프로그래머나 설계
자의 관점에서 바라보는
스키마
**데이터베이스의 물리적 구
조**를 가리킵니다.(실제 저
장방법을 기술하는 물리적
인 저장장치와 관련됨)

>>

내부 스키마

전체적인 뷰를 가리킵니
다.(조직 전체의 입장)
개체간의 관계와 제약조건
을 나타내고
데이터베이스의 접근권한/
보안/무결성 규칙에 대한
명세를 정의합니다.
일반적으로 스키마는 내부
스키마를 뜻합니다.

SQL과 NoSQL

5

3 스키마의 사용유무

Lorem ipsum dolor sit amet, consectetur adipiscing elit

A

Customers			
Customer ID	First Name	Last Name	Phone
1	Yong	Kim	010-1111-1111
2	John	Park	010-2222-2222

Products		
Product ID	Name	Price
1	Pen	1.20
2	Pencil	0.80

Orders			
Order ID	Customer ID	Product ID	Quantity
1	2	2	12

스키마 사용

B

Customers		
{customer_id: 1, name: "Yong Kim", phone: "010-1111-1111"}		
{customer_id: 2, name: "John Park", phone: "010-2222-2222"}		

Products		
{product_id: 1, name: "Pen", price: 1.20}		
{product_id: 2, name: "Pencil", price: 0.80}		

중복된 데이터

Orders		
{order_id: 1, customer: {customer_id: 2, phone: "010..."}, product: {product_id: 2, price: 1.20}}		

스키마 미사용

SQL

장점

- 명확하게 정의된 스키마 데이터의 무결성을 보장.
- 관계는 각 데이터 중복 없이 한번만 저장.

단점

- 유연성이 NoSQL보다 떨어짐.
- 데이터 스키마를 사전에 계획하고 알려야함.
- 관계를 맺고 있어서 조인문이 많은 복잡한 복잡한 쿼리가 만들어질 수 있음.
- 대체로 수직적 확장만 가능함.

NoSQL

장점

- 스키마가 없어서 유연함
- 수직 수평 확장이 가능
- 상시 저장된 데이터를 조정하고 새로운 필드 확장 가능
- 데이터는 어플리케이션이 필요로 하는 형식으로 저장됨(데이터 읽어오는 속도 빨라짐)

단점

- 데이터 중복을 계속 업데이트가 필요
- 데이터가 여러 컬렉션에 중복이 되어있기에 수정 시 모든 컬렉션에서 수정을 해야함.
(SQL에서는 중복데이터가 없기에 한번에 가능)

VS.

SQL 데이터베이스 사용

VS.

NoSQL 데이터베이스 사용

1. 관계를 맺고 있는 데이터가 자주 변경되는 애플리케이션의 경우
(NoSQL에서는 여러 컬렉션을 모두 수정해야 하기 때문에 비효율적이기에 SQL 데이터베이스 사용)

2. 변경될 여지가 없고, 명확한 스키마가 사용자와 데이터에게 중요한 경우

1. 정확한 데이터 구조를 알 수 없거나 변경/확장 될 수 있는 경우

2. 읽기를 자주 하지만, 데이터 변경은 자주 없는 경우

3. 데이터베이스를 수평으로 확장해야 하는 경우
(막대한 양의 데이터를 다뤄야 하는 경우)



Thanks!