02/01 db 면접 질문 정리

목적 DB의 중요한 내용과 다루지 않았던 내요을 전체적으로 정리

해당 내용은 실제 면접 질문과 중요 개념을 위주로 정리하였음.

정규화의 종류

정규화의 목적 ⇒ **중복된 데이터를 허용하지 않는 것**

Customer

Customer ID	First Name	Surname	Telephone Number
123	Pooja	Singh	555-861-2025
123	Pooja	Singh	192-122-1111
456	San	Zhang	182-929-2929
456	San	Zhang	(555) 403-1659 Ext. 53
789	John	Doe	555-808-9633

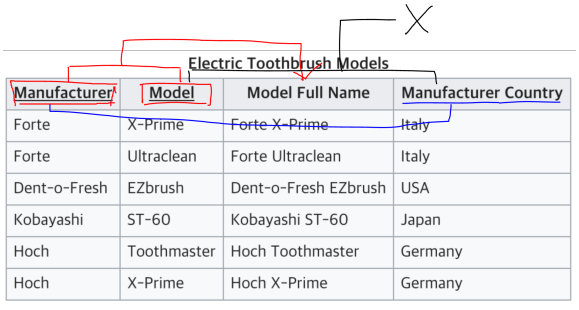
제1 정규화 불만족

Customer

Customer ID	First Name	Surname	Telephone Number
123	Pooja	Singh	555-861-2025
123	Pooja	Singh	192-122-1111
456	San	Zhang	182-929-2929
456	San	Zhang	(555) 403-1659 Ext. 53
789	John	Doe	555-808-9633

제1 정규화 만족

1정규화 ⇒ 원자값으로 분할



Electric Toothbrush Models

Manufacturer	Model	Model Full Name	Manufacturer Country
Forte	X-Prime	Forte X-Prime	Italy
Forte	Ultraclean	Forte Ultraclean	Italy
Dent-o-Fresh	EZbrush	Dent-o-Fresh EZbrush	USA
Kobayashi	ST-60	Kobayashi ST-60	Japan
Hoch	Toothmaster	Hoch Toothmaster	Germany
Hoch	X-Prime	Hoch X-Prime	Germany

제2 정규화 불만족

Electric Toothbrush Manufacturers

Manufacturer	Manufacturer Country
Forte	Italy
Dent-o-Fresh	USA
Kobayashi	Japan
Hoch	Germany

Electric Toothbrush Models

Manufacturer	Model	Model Full Name
Forte	X-Prime	Forte X-Prime
Forte	Ultraclean	Forte Ultraclean
Dent-o-Fresh	EZbrush	Dent-o-Fresh EZbrush
Kobayashi	ST-60	Kobayashi ST-60
Hoch	Toothmaster	Hoch Toothmaster
Hoch	X-Prime	Hoch X-Prime

제2 정규화 만족

2정규화 ⇒ 완전 함수적 종속을 만족

테이블에서 기본키가 복합키(키1, 키2)로 묶여있을 때, 두 키 중 하나의 키만으로 다른 컬 럼을 결정지을 수 있으면 안된다.

Tournament Winners

Tournament	Year	Winner	Winner Date of Birth
Indiana Invitational	1998	Al Fredrickson	21 July 1975
Cleveland Open	1999	Bob Albertson	28 September 1968
Des Moines Masters	1999	Al Fredrickson	21 July 1975
Indiana Invitational	1999	Chip Masterson	14 March 1977

제3 정규화 ⇒ 이행적 종속을 없애기 위해 테이블을 분리

내가 이해한 것 기본키가 아닌 것에 종속을 가진 경우을 없앤다.

(

모든 비기본 키 속성이 기본 키에만 직접적으로 종속되어야 하는 상태)

Tournament Winners

Tournament	Year	Winner
Indiana Invitational	1998	Al Fredrickson
Cleveland Open	1999	Bob Albertson
Des Moines Masters	1999	Al Fredrickson
Indiana Invitational	1999	Chip Masterson

Winner Dates of Birth

Winner	Date of Birth
Chip Masterson	14 March 1977
Al Fredrickson	21 July 1975
Bob Albertson	28 September 1968

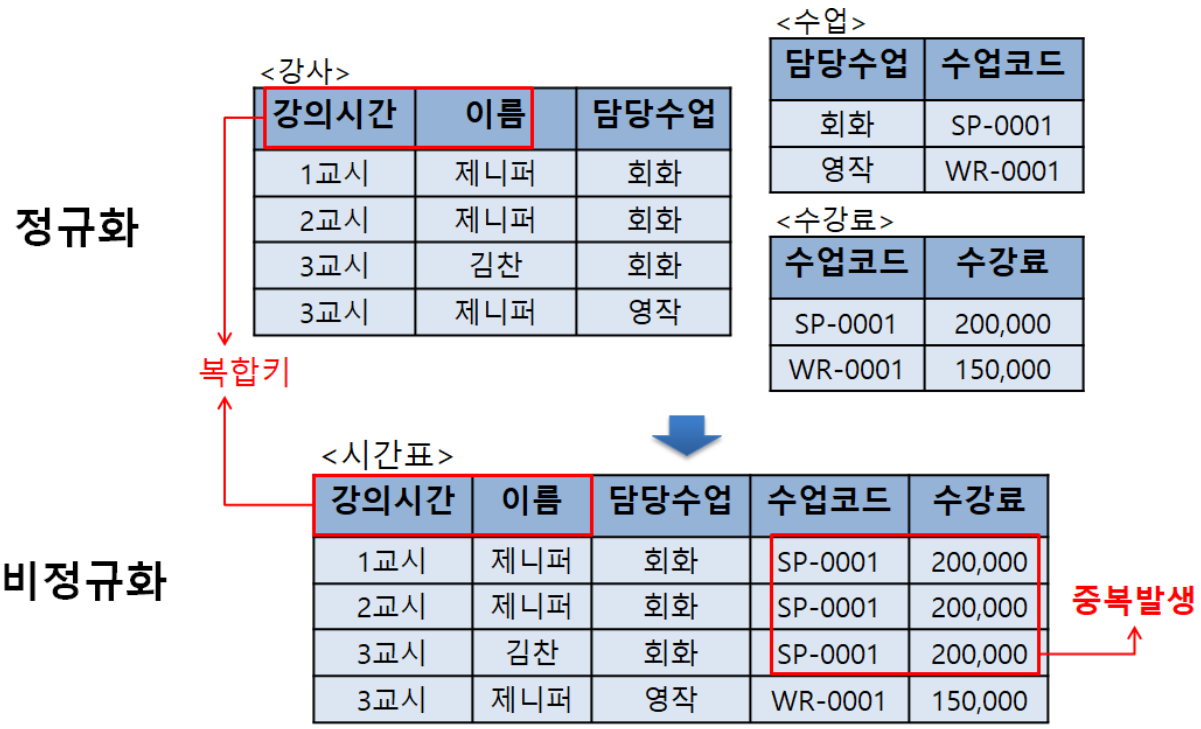
비정규화 vs 정규화 데이터베이스

정규화 데이터베이스 ⇒ 중복을 최소화하도록 설계된 데이터베이스

외래키를 통해서 다른 테이블 참조

비정규화 데이터베이스 ⇒ 읽는 시간을 최적화하도록 설계된 데이터베이스

데이터를 중복하여 설계



정규화 데이터베이스에서 **조인을 통한 연산이 많이 필요하다고 판단되는 경우** 비정규화를 통해 해당 정보를 **중복적으로 저장하여 설계하는 방식 ⇒ 비정규화 데이터 베이스**

Transaction 관리를 위한 DBMS의 전략

1. DBMS의 구조

> 크게 2가지 : Query Processor (질의 처리기), Storage System (저장 시스템)
>
> 입출력 단위 : 고정 길이의 page 단위로 disk에 읽거나 쓴다.
>
> 저장 공간 : 비휘발성 저장 장치인 disk에 저장, 일부분을 Main Memory에 저장

2. Page Buffer Manager or Buffer Manager

DBMS의 Storage System에 속하는 모듈 중 하나로, Main Memory에 유지하는 페이지를 관리하는 모듈

> Buffer 관리 정책에 따라, UNDO 복구와 REDO 복구가 요구되거나 그렇지 않게 되므로, transaction 관리에 매우 중요한 결정을 가져온다.

UNDO와 REDO 복구 메커니즘

1. **UNDO 복구**
   - 목적: 트랜잭션이 실패하거나 취소될 경우, 변경사항을 원래 상태로 되돌리는 것
   - 예시: 트랜잭션이 데이터를 변경하던 중 시스템이 실패하여 트랜잭션이 완료되지 못했다면, UNDO를 통해 변경 전 상태로 되돌린다.
2. **REDO 복구**
   - 목적: 이미 커밋된 트랜잭션의 변경사항이 디스크에 영구적으로 반영되도록 보장하는 것
   - 예시: 트랜잭션이 커밋된 후 시스템에 장애가 발생했다면, REDO를 통해 디스크에 커밋된 상태를 재확인하고 필요한 경우 다시 적용한다.

예시: 버퍼 관리와 복구 메커니즘

- 상황: 데이터베이스에 '계좌 이체' 트랜잭션이 수행된다고 할 때, 트랜잭션은 한 계좌에서 다른 계좌로 금액을 이체하는 과정

- UNDO 필요성:
  - 트랜잭션이 '금액 차감' 작업을 수행한 후, 시스템이 실패한다.
  - 이 경우, 버퍼 매니저는 아직 디스크에 반영되지 않은 변경사항을 메모리에서 추적해야 한다.
  - 시스템 복구 시, 버퍼 매니저는 이러한 변경사항을 원래 상태(금액 차감 전)로 되돌려야 한다.(UNDO).
- REDO 필요성:
  - 트랜잭션이 '금액 추가' 작업을 완료하고 커밋한 후, 시스템이 실패한다.
  - 이 경우, 버퍼 매니저는 커밋된 변경사항이 디스크에 올바르게 반영되었는지 확인해야 한다.
  - 시스템 복구 시, 버퍼 매니저는 이러한 변경사항을 다시 적용하여 디스크 상태를 최신 커밋 상태로 유지한다.(REDO).

  UNDO = 실패시 변경사항을 원래 상태로 되돌리는 것

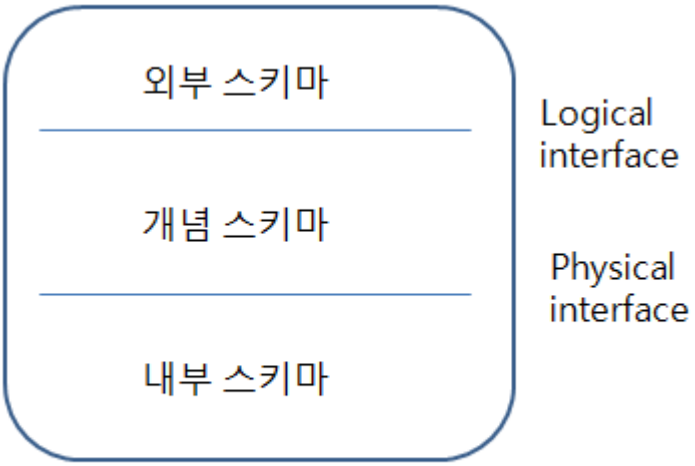
  REDO = (작업 완료 후)시스템 실패시 이미 커밋된 트랜잭션의 변경사항이 시스템 장애 후에도 영구적으로 반영되도록 보장하는 것

DB에서 View는 무엇인가? 가상 테이블이란?

> 허용된 데이터를 제한적으로 보여주기 위한 것
>
> 하나 이상의 테이블에서 유도된 가상 테이블이다.
>
> - 사용자가 view에 접근했을 때 해당하는 데이터를 원본에서 가져온다.
>
> view에 나타나지 않은 데이터를 간편히 보호할 수 있는 장점 존재

DBMS를 사용하는 목적

DBMS는 데이터 독립성(Data Independence)을 보장하기 위해 사용



3 – level Architecture

- **외부 스키마**
  - 사용자 입장에서 보는 논리적인 데이터베이스의 구조를 의미
- **개념 스키마**
  - 데이터베이스의 전체적인 논리구조, 즉 테이블을 의미
- **내부 스키마**
  - 데이터가 실제로 저장되는 물리적인 데이터베이스의 구조를 의미

3-level architecture의 장점은 **각 스키마 간의 데이터의 독립성이 보장**된다는 것

뷰를 사용하는 이유

뷰를 사용하는 이유 역시, 독립성을 보장하기 위해

1. **데이터 독립성 (Growth)**
   a. **뷰에 새로운 속성(Attribute) 추가 시 -** 뷰에 새로운 속성을 추가해도 기본 테이블(Base Table)에는 영향을 주지 않는다.
   b. **기본 테이블에 새로운 속성 추가 시 -** 기본 테이블에 새로운 속성이 추가되더라도, 기존에 정의된 뷰에는 자동으로 반영되지 않는다.
2. **구조 재편성 (Restructuring)**
   a. **뷰를 통한 데이터 선별 -** 뷰를 사용하면 기본 테이블의 일부 데이터만을 선택하여 보여줄 수 있다.

b. **연산 감소 및 보안성 향상 -** 매번 기본 테이블 전체에 접근하지 않고, 필요한 부분만을 포함하는 뷰에 접근함으로써 연산량을 줄일 수 있다.

데이터베이스를 설계할 때 가장 중요한 것이 무엇이라고 생각하는가?

> 무결성
>
> **무결성 보장 방법은?**
>
> 데이터를 조작하는 프로그램 내에서 데이터 생성, 수정, 삭제 시 무결성 조건을 검증한다.
>
> 트리거 이벤트 시 저장 SQL을 실행하고 무결성 조건을 실행한다.
>
> DB제약조건 기능을 선언한다.

데이터베이스 무결성이란?

> 테이블에 있는 모든 행들이 유일한 식별자를 가질 것을 요구함 (같은 값 X)
>
> 외래키 값은 NULL이거나 참조 테이블의 PK값이어야 함
>
> 한 컬럼에 대해 NULL 허용 여부와 자료형, 규칙으로 타당한 데이터 값 지정

트랜잭션 특징

- 원자성(Atomicity)

  > 트랜잭션이 DB에 모두 반영되거나, 혹은 전혀 반영되지 않아야 된다.

- 일관성(Consistency)

  > 트랜잭션의 작업 처리 결과는 항상 일관성 있어야 한다.

- 독립성(Isolation)

> 둘 이상의 트랜잭션이 동시에 병행 실행되고 있을 때, 어떤 트랜잭션도 다른 트랜잭션 연산에 끼어들 수 없다.

- 지속성(Durability)

> 트랜잭션이 성공적으로 완료되었으면, 결과는 영구적으로 반영되어야 한다.

Commit

하나의 트랜잭션이 성공적으로 끝났고, DB가 일관성있는 상태일 때 이를 알려주기 위해 사용하는 연산

Rollback

하나의 트랜잭션 처리가 비정상적으로 종료되어 트랜잭션 원자성이 깨진 경우

transaction이 정상적으로 종료되지 않았을 때, last consistent state (예) Transaction의 시작 상태) 로 roll back 할 수 있음.

이상현상

이상 현상(Anomalies)은 데이터베이스의 릴레이션(테이블) 설계가 적절하지 않을 때 발생하는 여러 문제

데이터의 중복으로 인해 삽입(Insert), 갱신(Update), 삭제(Delete) 작업 시 예상치 못한 문제 발생

주로 데이터의 정규화가 충분히 이루어지지 않았을 때 발생

insert, update, delete 문제 ⇒ **정규화를 통해 해결 가능.**

궁금한점 :

모든 비기본 키 속성이 기본 키에만 직접적으로 종속되어야 하는 상태 ⇒ 3NF

테이블에 있는 모든 행들이 유일한 식별자를 가질 것을 요구함 ⇒ 기본키

같은 목적이 아닌가?

참조 :

tech-interview-for-developer

코딩인터뷰 완전분석 - 게일 라크만 맥도웰