

Computer Science 정리

6. 데이터 베이스

1. 데이터 베이스의 기본 개념

- 데이터 베이스의 정의 및 특징

- 특정 조직의 여러 사용자가 공유하여 사용할 수 있도록 통합해서 저장한 운영 데이터의 집합.

실시간 접근	사용자의 데이터 요구에 실시간으로 응답
계속 변화	데이터의 계속적인 삽입,삭제,수정 등을 통해 현재의 정확한 데이터를 유지
동시 공유	서로 다른 데이터의 동시 공유 뿐만 아니라 같은 데이터의 동시 사용도 지원
내용 기반 참조	데이터가 저장된 주소나 위치가 아닌 내용으로 참조 ex) 재고량이 1000 이상인 제품의 이름을 검색

- 데이터 베이스 모델

- 데이터 모델의 개념

데이터 모델링 (data modeling)	<ul style="list-style-type: none">- 현실 세계에 존재하는 데이터를 컴퓨터 세계의 데이터 베이스로 옮기는 변환 과정- 데이터 베이스 설계의 핵심 과정
2단계 데이터 모델링	<ul style="list-style-type: none">- 개념적 데이터 모델링 : 현실 세계의 중요 데이터를 추출하여 개념 세계로 옮기는 작업- 논리적 데이터 모델링 : 개념 세계의 데이터를 데이터베이스에 저장하는 구조로 표현하는 작업.

- 데이터 모델링의 결과물을 표현하는 도구

- * 개념적 데이터 모델 : 사람의 머리로 이해할 수 있도록 현실 세계를 개념적 모델링하여 데이터 베이스의 개념적 구조로 표현하는 도구
ex) 개체 - 관계 모델
- * 논리적 데이터 모델 : 개념적 구조를 논리적 모델링 하여 데이터 베이스의 논리적 구조로 표현하는 도구
ex) 관계 데이터 모델

2.SQL

-sql의 분류

데이터 정의 언어 (DDL)	- 테이블을 생성하고 변경, 제거 하는 기능을 제공
데이터 조작 언어 (DML)	- 테이블에 새 데이터를 삽입하거나 , 테이블에 저장된 데이터를 수정, 삭제 , 검색 하는 기능을 제공
데이터 제어 언어 (DCL)	- 보안을 위해 데이터에 대한 접근 및 사용 권한을 사용자별로 부여하거나 취소하는 기능을 제공

- SQL의 데이터 정의 기능 : 테이블 생성, 변경, 삭제

- 테이블 생성 : CREATE TABLE
- 테이블 변경 : ALTER TABLE
- 테이블 삭제 : DROP TABLE
- 테이블 초기화 : TRUNCATE TABLE

- SQL의 데이터 조작 기능 : 데이터 검색, 데이터 삽입, 데이터 수정 / 삭제

- 데이터 검색 : SELECT
- 데이터 삽입 : INSERT
- 데이터 삭제 : DELETE
- 데이터 수정 : UPDATE

- SQL 데이터 제어 기능 : 권한 부여 / 회수

- 권한 부여 : GRANT
- 권한 회수 : REVOKE

- 뷰

- 기본 테이블을 들여다 볼 수 있는 창의 역할을 담당

활용	장점
- SELECT, INSERT, UPDATE, DELETE	<ul style="list-style-type: none"> - 질의문을 좀 더 쉽게 작성할 수 있음 - 데이터의 보안 유지에 도움이 됨 - 데이터를 좀 더 편리하게 관리할 수 있음

3. 데이터 정규화

- 이상 (anomaly) 현상

- 불필요한 데이터 중복으로 인해 릴레이션에 대한 데이터 삽입, 수정, 삭제 연산을 수행할 때 발생할 수 있는 부작용

삽입 이상 (insertion anomaly)	- 릴레이션에 새 데이터를 삽입하려면 불필요한 데이터도 함께 삽입하여야 하는 문제
갱신 이상 (update anomaly)	- 릴레이션의 중복된 튜플들 중 일부만 수정하여 데이터가 불일치 하게 되는 모순이 발생하는 문제
삭제 이상 (deletion anomaly)	- 릴레이션에서 튜플을 삭제하면 필수 데이터까지 손실되는 연쇄 삭제현상이 발생하는 문제

- 정규화

- 중복 데이터 없앴 -> 데이터 무결성 유지, 저장 용량 확보
- 이상현상을 제거하면서 데이터베이스를 올바르게 설계해나가는 과정
- 이상현상이 발생하지 않도록, 릴레이션을 관계 있는 속성들만으로 구성하기 위해 릴레이션을 분해(decomposition) 하는 과정
- 함수적 종속성을 판단하여 정규화를 수행함

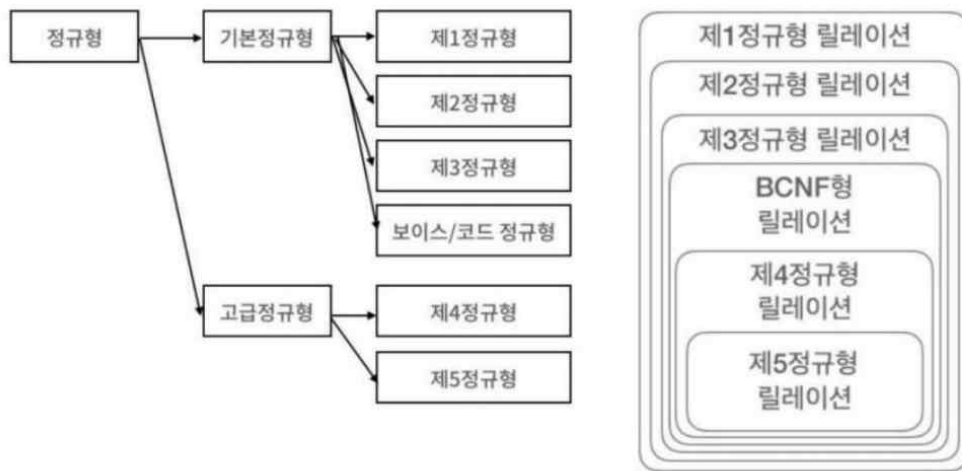
- 함수적 종속성 (FD : Functional Dependency)

- 함수 종속성을 이용하여, 릴레이션을 연관성이 있는 속성들만으로 구성되도록 분해하여 이상 현상 발생하지 않는 릴레이션으로 만드는 과정

- 정규형 (NF : Normal Form)

- 릴레이션이 정규화된 정도로 각 정규형마다 제약조건이 존재함
- 릴레이션의 특성을 고려하여 적합한 정규형을 선택

- 정규형의 종류



- 정규화 과정

제 1 정규형 (1NF; 1st Normal Form)	<ul style="list-style-type: none"> - 릴레이션의 모든 속성이 더는 분해되지 않는 원자값(atomic value) 만 가지면 만족함. - 제 1 정규형을 만족해야 관계 데이터베이스의 릴레이션이 될 자격이 있음.
제 2 정규형 (2NF; 2nd Normal Form)	<ul style="list-style-type: none"> - 릴레이션이 1 정규형을 만족하고 , 기본키가 아닌 모든 속성이 기본키에 완전 함수 종속되면 만족함.
제 3 정규형 (3NF; 3rd Normal Form)	<ul style="list-style-type: none"> - 릴레이션이 2 정규형을 만족하고 , 기본키가 아닌 모든 속성이 기본키에 이행적 함수 종속되지 않으면 만족
보이스/코드 정규형 (BCNF; Boyce/Codd NF)	<ul style="list-style-type: none"> - 하나의 릴레이션에 여러개의 후보키가 존재하는 경우, 제 3 정규형까지 모두 만족해도 이상현상 발생할 수 있음
제 4 정규형	<ul style="list-style-type: none"> - 릴레이션이 BCNF 만족하고 함수 종속이 아닌 다치 종속을 제거하면 만족 (MVD)
제 5 정규형	<ul style="list-style-type: none"> - 릴레이션이 제4 정규형 만족, 후보키를 통하지 않는 조인 종속을 제거하면 만족 (JD)

- 정규화시 주의사항

- 모든 릴레이션이 제5 정규형에 속해야 하는 것은 아님!!
- 일반적으로 제3 정규형이나 BCNF 에 속하도록 릴레이션을 분해하여 데이터 중복을 줄이고 이상 현상을 해결하는 경우가 많음

4. 트랜잭션

- 트랜잭션(transaction) 의 개념
 - 하나의 작업을 수행하기 위한 데이터베이스의 연산들을 모아놓은 것
 - 작업 수행에 필요한 SQL 문들의 모임
 - 논리적인 작업의 단위
 - 장애 발생 시 복구 작업이나 병행 제어 작업을 위한 단위로 사용
 - 데이터베이스 무결성이나 일관성을 보장하기 위해 작업 수행에 필요한 연산들을 하나의 트랜잭션으로 제대로 정의하고 관리해야 함
 - * 데이터 베이스의 상태를 변화시키는 작업의 단위! (여러개의 sql문)
- 트랜잭션의 특징

원자성 (atomicity)	- 트랜잭션의 연산들이 모두 정상적으로 실행되거나 하나도 실행되지 않아야 하는 all-or-nothing을 의미
일관성 (consistency)	- 트랜잭션이 성공적으로 수행된 후에도 데이터 베이스가 일관성을 유지해야 함을 의미
격리성 (isolation)	- 수행 중인 트랜잭션이 완료될 때까지 다른 트랜잭션들이 중간 연산 결과에 접근 할 수 없어야 함을 의미
지속성 (durability)	- 트랜잭션이 성공적으로 완료된 후 데이터 베이스에 반영한 수행 결과는 영구적이어야 함을 의미 - 지속성의 보장을 위해서는 장애 발생시 회복 기능이 필요.



- 트랜잭션의 주요 연산

commit 연산	rollback 연산
<ul style="list-style-type: none"> - 트랜잭션의 수행이 성공적으로 완료되었음을 선언하는 연산 - commit 연산이 실행되면 트랜잭션의 수행 결과가 데이터 베이스에 반영되고 일관된 상태를 지속적으로 유지하게됨 	<ul style="list-style-type: none"> - 트랜잭션의 수행이 실패했음을 선언하는 연산 - rollback 연산이 실행되면 지금까지 트랜잭션이 실행한 연산의 결과가 취소되고 데이터 베이스가 트랜잭션 수행전의 일관된 상태로 되돌아감