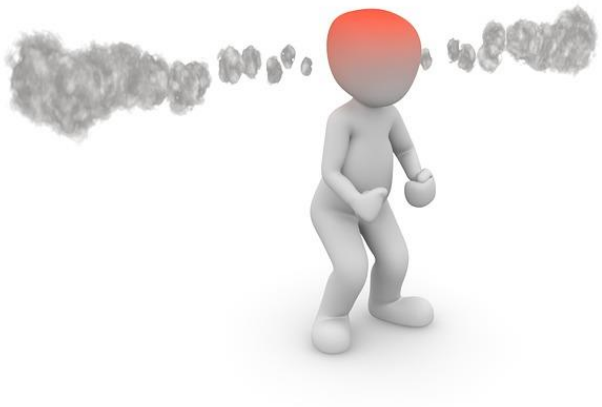
정보처리기사

제1과목 데이터베이스

01 시험안내

문혜영교수

- http://www.q-net.or.kr
- 큐넷
- 한국산업인력관리공단



정보처리기사

제1과목 데이터베이스

02 데이터베이스의 개념

문혜영교수

목차

1 정보 시스템

2 데이터와 정보

3 데이터베이스 개념

4 DBMS

정보시스템

- 자료와 정보
  - ✓자료(Data) : 현실 세계에서 단순한 관찰이나 측정을 통해 수집된 사실이나 값
  - ✓정보(Information) : 의사 결정에 도움을 줄 수 있는 유용한 형태로, 자료를 가공(처리)해서 얻을 수 있는 결과

- 정보 시스템의 정의
  - ✓정보 시스템 : 한 조직체에 필요한 Data를 수집, 저장해 두었다가 필요 시에 처리해서 의사결정에 유용한 정보를 생성하고 분배하는 수단

- 자료 처리 시스템
  - ✓일괄처리 시스템
    - ✓시스템 중심의 자료 처리 방법으로 시스템 성능이 높다.
    - ✓순차 접근 방법을 사용하는 업무에 적합하다.
    - ✓트랜잭션 당 처리 비용이 적다.
    - ✓급여계산, 회계 마감 업무, 세무 처리, 수도/전기요금 처리, 연말 결산 등의 업무에 사용된다.
  - ✓온라인 실시간 처리 시스템
  - ✓분산 처리 시스템

❖ 데이터웨어하우스(Data Warehouse) : 조직이나 기업체의 중심이 되는 주요 업무 시스템에서 추출되어 새로이 생성된 데이터베이스로서 의사결정지원 시스템을 지원하는 주체적, 통합적, 시간적 데이터의 집합

문제

- 정보(Information)의 의미로 거리가 먼 것은?
  ① 자료(Data)를 처리하여 얻은 결과
  ② 사용자가 목적하는 값
  ③ 현실세계에서 관찰을 통해 얻은 값
  ④ 의사결정을 위한 값

정답 3

데이터베이스의 개념

- 데이터베이스의 정의

✓통합된 데이터(Integrated Data)

✓저장된 데이터(Stored Data)

✓운영 데이터(Operational Data) : 조직의 고유한 업무를 수행하기 위해 존재 가치가 확실하고 반드시 필요한 자료(데이터)를 의미한다.

✓공용 데이터(Shared Data)

- 데이터베이스의 특징

✓실시간 접근성(Real-Time Accessibility) : 수시적이고 비전형적인 질의(조회)에 대하여 실시간 처리에 의한 응답이 가능해야 한다.

✓계속적인 변화(Continuous Evolution) : 데이터의 삽입, 삭제, 갱신으로 항상 최신의 데이터를 유지해야 한다.

✓동시 공용(Concurrent Sharing) : 같은 내용의 데이터를 다수의 사용자가 동시에 공용할 수 있다.

✓내용에 의한 참조(Content Reference) : 데이터의 참조는 저장되어 있는 데이터 레코드들의 주소나 위치에 의해서가 아니라, 사용자가 요구하는 데이터 내용으로 데이터를 찾는다.

문제

- 데이터베이스의 특성으로 옳지 않는 것은?
  ① 같은 내용의 데이터를 여러 사람이 동시에 공용할 수 있다.
  ② 데이터베이스는 데이터의 삽입, 삭제, 갱신으로 내용이 계속적으로 변한다.
  ③ 수시적이고 비전형적인 질의에 대하여 실시간 처리로 응답할 수 있어야 한다.
  ④ 데이터의 참조는 저장되어 있는 데이터 레코드들의 주소나 위치에 의해서 이루어진다.

정답 4

DBMS

- DBMS의 발전 배경

✓종래의 파일 처리 방식에서의 데이터 구성 : 종래의 파일 처리 방식에서는 중복성과 종속성을 초래하는 단점이 있다.

  ✓종속성으로 인한 문제점

  ✓중복성으로 인한 문제점

✓데이터베이스 도입에서의 데이터 구성

  ✓업무 처리에 필요한 자료들의 중복을 최소화하여 한 곳에 모아서 구성한다.

정보처리기사

제1과목 데이터베이스

03 데이터베이스의 개념

문혜영교수

목차

4 스키마의 3계층

3 스키마

2 DBMS장단점

1 DBMS

- DBMS의 필수 기능
  - ✓정의(조직)(Definition) 기능
    - ✓모든 응용 프로그램들이 요구하는 데이터 구조를 지원하기 위해 데이터베이스에 저장된 데이터의 형(Type)과 구조에 대한 정의, 이용 방식, 제약 조건 등을 명시하는 기능
  - ✓조작(Manipulation) 기능
    - ✓ 데이터 검색, 갱신, 삽입, 삭제 등을 체계적으로 처리하기 위해 사용자와 데이터베이스 사이의 인터페이스 수단을 제공하는 기능
  - ✓제어(Control) 기능
    - ✓데이터베이스를 접근하는 갱신, 삽입, 삭제 작업이 정확하게 수행되어 데이터의 무결성이 유지되도록 제어해야 한다.
    - ✓정당한 사용자가 허가된 데이터만 접근할 수 있도록 보안을 유지하고 권한을 검사할 수 있어야 한다.
    - ✓여러 사용자가 데이터베이스를 동시에 접근하여 데이터를 처리할 때 처리 결과가 항상 정확성을 유지하도록 병행 제어를 할 수 있어야 한다.

- DBMS의 장점
  - ✓데이터의 중복을 피할 수 있어 기억공간이 절약된다.
  - ✓저장된 자료를 공동으로 이용할 수 있다.
  - ✓데이터의 일관성을 유지할 수 있다.
  - ✓데이터의 무결성을 유지할 수 있다.
  - ✓보안을 유지할 수 있다.
  - ✓데이터를 표준화할 수 있다.

❖ 백업(Backup) : 장비 고장 등의 비상사태에도 데이터베이스가 보존되도록 복사하는 작업

문제

- 데이터베이스 관리 시스템(DBMS)의 주요 필수 기능과 거리가 먼 것은?
  ① 데이터베이스 구조를 정의할 수 있는 정의 기능
  ② 데이터 사용자의 통제 및 보안 기능.
  ③ 데이터베이스 내용의 정확성과 안정성을 유지할 수 있는 제어 기능
  ④ 데이터 조작어로 데이터베이스를 조작할 수 있는 조작 기능

정답 2

스키마

- 스키마의 정의

✓데이터베이스의 구조와 제약 조건에 관한 전반적인 명세를 기술한 메타데이타(Meta-Data)의 집합이다.

✓데이터베이스를 구성하는 데이터 객체, 이들의 성질, 이들 간에 존재하는 관계, 그리고 데이터의 조작 또는 이들 데이터 값들이 갖는 제약 조건 등에 관해 전반적으로 정의한다.

- 스키마의 특징

✓데이터 사전(Data Dictionary)에 저장된다.

✓다른 이름으로 메타데이터(Meta-Data)라고도 한다.

✓데이터베이스 구조(개체, 속성, 관계)에 대한 정의이다.

- 스키마의 3계층
  - ✓외부 스키마(External Schema) = 서브 스키마 = 사용자 뷰(View)
    - ✓ 사용자나 응용 프로그래머가 각 개인의 입장에서 필요로 하는 데이터베이스의 논리적 구조를 나나낸 것
    - ✓ 전체 데이터베이스의 한 논리적인 부분으로 볼 수 있다.
    - ✓ 하나의 베이터베이스 시스템에는 여러 개의 외부 스키마가 존재할 수 있다.
  - ✓개념 스키마(Conceptual Schema) = 전체적인 뷰(View)
    - ✓ 데이터베이스의 전체적인 논리적 구조로서, 모든 응용 시스템과 사용자들이 필요로 하는 데이터를 통합한 조직 전체의 데이터베이스로 하나만 존재한다.
    - ✓ 개체 간의 관계와 제약 조건을 나타내고 데이터베이스의 접근 권한, 보안 및 무결성 규칙에 관한 명세가 있는 스키마이다.
    - ✓ 데이터베이스 파일에 저장되는 데이터의 형태를 나타내는 것으로 단순 스키마라고도 한다.
  - ✓내부 스키마(Internal Schema) = 저장 스키마(Storage Schema)
    - ✓ 물리적 저장장치의 입장에서 본 데이터베이스 구조로서 실제로 데이터베이스에 저장될 레코드의 물리적 구조를 정의하고, 저장 데이터 항목의 표현 방법, 내부 레코드의 물리적 순서 등을 나타내는 스키마이다.

문제

- 개체 간의 관계와 제약 조건을 나타내고 데이터베이스의 접근 권한, 보안 및 무결성 규칙 명세가 있는 스키마는?
  ① 내부 스키마
  ② 외부 스키마
  ③ 개념 스키마
  ④ 서브 스키마

정답 3

정보처리기사

제1과목 데이터베이스

04 데이터베이스의 개념

문혜영교수

목차

1 데이터베이스 언어

2 데이터베이스 사용자

3 DBA

4 데이터 모델

데이터베이스 언어

- 데이터베이스 언어

✓데이터베이스를 구축(정의)하고 이용하기 위한 데이터베이스 시스템과의 통신 수단이다

✓데이터베이스 언어는 기능과 사용 목적에 따라 데이터 정의 언어, 데이터 조작 언어, 데이터 제어 언어로 구분된다.

- 데이터 정의 언어(DDL : Data Definition Language)

✓데이터베이스를 구축(정의)하거나 수정할 목적으로 사용하는 언어
✓번역한 결과가 데이터 사전(Data Dictionary)에 저장된다.

✓데이터 정의 언어의 기능
  ✓외부 스키마 명세를 정의
  ✓데이터 베이스의 논리적, 물리적 구조 및 구조 간의 사상을 정의
  ✓스키마에 사용되는 제약 조건에 대한 명세를 정의

- 데이터 조작 언어(DML : Data Manipulation Language, 서브 언어

  ✓사용자로 하여금 데이터를 처리할 수 있게 하는 도구로서 사용자(응용 프로그램)와 DBMS 간의 인터페이스를 제공한다.

  ✓사용자가 데이터 모델에 적합하게 조직된 데이터를 접근하고 조작할 때 사용하는 언어이다.

- 데이터 제어 언어(DCL : Data Control Language)

✓무결성, 보안 및 권한 제어, 회복 등을 하기 위한 언어이다.

✓데이터 제어 언어의 기능

- ✓불법적인 사용자로부터 데이터를 보호하기 위한 데이터 보안(Security)
- ✓데이터 정확성을 위한 무결성(Integrity) 유지
- ✓시스템 장치에 대비한 데이터 회복과 병행수행 제어

문제

- 데이터 제어어(DCL)의 역할이 아닌 것은?
  ① 불법적인 사용자로부터 데이터를 보호하기 위한 데이터 보안(Security)
  ② 데이터 정확성을 위한 무결성(Integrity) 유지
  ③ 시스템 장애에 대비한 데이터 회복과 병행 수행
  ④ 데이터의 검색, 삽입, 삭제, 변경

정답 4

- 데이터 베이스 사용자 : DBA, 응용 프로그래머, 일반 사용자
- DBA(DataBase Administrator, 데이터베이스 관리자)
  - ✓데이터베이스 시스템의 모든 관리와 운영에 대한 책임을 지고 있는 사람이나 그룹
- DBA의 역활
  - ✓데이터베이스 설계와 조작에 대항 책임
    - ✓ 개념 스키마와 내부 스키마의 정의
    - ✓ 데이터베이스의 저장 구조 및 접근 방법 정의
    - ✓ 보안 및 데이터베이스의 접근 권한 부여 정책 수립
    - ✓ 장애에 대비한 예비(Back Up) 조치와 회복(Recovery)에 대한 전략 수립
    - ✓ 무결성을 위한 제약 조건의 지정
    - ✓ 사용자의 변화 요구와 성능 향상을 위한 데이터베이스의 재구성
  - ✓책임 행정
    - ✓ 사용자의 요구와 불평의 청취 및 해결
  - ✓시스템 감시 및 성능 분석

문제

- DBA의 수행 역할에 관한 설명으로 거리가 먼 것은?
  ① 데이터베이스의 구축
  ② 응용 프로그램 개발
  ③ 사용자 요구 정보 결정 및 효율적 관리
  ④ DBMS의 관리

정답 2

- 데이터 모델의 정의

✓현실 세계의 정보들을 컴퓨터에 표현하기 위해서 단순화, 추상화하여 체계적으로 표현한 개념적 모형이다.

✓현실 세계를 데이터베이스에 표현하는 중간 과정 즉, 데이터베이스 설계 과정에서 데이터의 구조를 논리적으로 표현하기 위해 사용되는 도구이다.

- 데이터 모델의 종류
  - ✓개념적 데이터 모델
    - ✓속성들로 기술된 개체 타입과 이 개체 타입들 간의 관계를 이용하여 현실 세계를 표현한다.
    - ✓현실 세계에 존재하는 개체를 인간이 이해할 수 있는 정보 구조로 표현하기 때문에 정보 모델이라고도 한다.
  - ✓논리적 데이터 모델
    - ✓필드로 기술된 데이터 타입과 이 데이터 타입들 간의 관계를 이용하여 현실 세계를 표현한다.

❖ 데이터 모델링 : 현실 세계에 존재하는 개체를 컴퓨터 세계의 데이터 구조로 기술하는 것.

❖ 정보 모델링 : 현실 세계에 존재하는 개체를 인간이 이해할 수 있는 정보 구조로 표현하는 것.

- 데이터 모델에 표시할 요소

✓구조(Structure) : 논리적으로 표현된 개체 타입들 간의 관계

✓연산(Operation) : 데이터베이스에 저장된 실제 데이터를 처리하는 작업에 대한 명세로서 데이터베이스를 조작하는 기본 도구

✓제약 조건(Constraint)

정보처리기사

제1과목 데이터베이스

05 데이터 모델링 설계

문혜영교수

목차

1 E-R 모델

2 관계형 데이터모델

3 계층형 데이터 모델

4 망형 데이터모델

- 데이터 모델의 구성 요소
  - ✓개체(Entity)
    - ✓데이터베이스에 표현하려고 하는 현실 세계의 대상체이다.
    - ✓실세계에 독립적으로 존재하는 유형, 무형의 정보로서 서로 연관된 몇 개의 속성으로 구성된다.
    - ✓파일 시스템의 레코드에 대응하는 것으로 어떤 정보를 제공하는 역할을 수행한다.
  - ✓속성(Attribute)
    - ✓데이터의 가장 작은 논리적 단위로서 파일 구조상의 데이터 항목 도는 데이터 필드에 해당한다.
  - ✓관계(Relationship)
    - ✓관계의 형태
      - ✓ 일 대 일(1:1)
      - ✓ 일 대 다(1:n)
      - ✓ 다 대 다(n:n) : 개체 집합 A의 각 원소는 개체 집합 B의 원소 여러 개와 대응하고, 개체 집합 B의 각 원소도꺠체 집합 A의 원소 여러 개에 대응하는 관계

문제

- 데이터 모델의 구성 요소가 아닌 것은?
  ① 논리적으로 표현된 데이터 구조
  ② 구성 요소의 연산
  ③ 구성 요소의 제약 조건
  ④ 물리적 저장 구조

정답 4

- 개체-관계(Entity-Relationship, E-R) 모델의 개요

  ✓피터 첸(Peter Chen)에 의해서 제안되었다.

  ✓개체와 개체 간의 관계를 기본 요소로 이용하여 현실 세계의 무질서한 데이터를 개념적인 논리 데이터로 표현하기 위한 방법이다.

  ✓개체 타입(Entity Type)과 이들 간의 관계 타입(Relationship Type)을 이용해 현실 세게를 개념적으로 표현한다.

  ✓데이터를 개체(Entity), 관계(Relationship), 속성(Attribute)으로 묘사한다.

  ✓E-R 다이어그램으로 표현하며, 1:1, 1:N, N:M 등의 관계 유형을 제한 없이 나타낼 수 있다.

- 개체-관계(Entity-Relationship, E-R) 다이어그램

  ✓사각형 : 개체(Entity) 타입
  ✓마름모 : 관계(Relationship) 타입
  ✓타원 : 속성(Attribute)
  ✓이중 타원 : 다중값 속성(복합 속성)
  ✓선 링크 : 개체 타입과 속성 연결

문제

- 개체-관계(E-R) 모델에 대한 설명으로 잘못된 것은?
  ① E-R 다이어그램으로 표현하며 P. Chen이 제안했다.
  ② 일 대 일(1:1) 관계 유형만을 표현할 수 있다.
  ③ 개체 타입과 이들 간의 관계 타입을 이용해 현실 세계를 개념적으로 표현한 방법이다.
  ④ E-R 다이어그램은 E-R 모델을 그래프 방식으로 표현한 것이다.

정답 2

- 관계형 데이터 모델(Relational Data Model)의 개념

  ✓개체 집합에 대한 속성 관계를 표현하기 위해 개체를 2차원적인 표(Table)를 사용한다.

  ✓개체 집합들 사이의 관계를 공통 속성으로 연결하는 독립된 형태의 데이터 모델이다.

  ✓기본키(Premary Key)와 이를 참조하는 외래키(Foreign Key)로 데이터 간의 관계를 표현한다.

문제

- 개체 집합에 대한 속성 관계를 표현하기 위해 개체를 테이블(Table)로 사용하고 개체 집합들 사이의 관계를 공통 속성으로 연결하는 독립된 형태의 데이터 모델은?
  ① 망 데이터 모델
  ② 계층 데이터 모델
  ③ 관계 데이터 모델
  ④ 객체지향 데이터 모델

정답 3

- 계층형 데이터 모델(Hierarchical Data Model)의 구성 형태
  - ✓전체적인 구조가 트리 형태이며 두 레코드 타입 간에 하나의 관계만 허용되는 데이터 모델이다.
  - ✓개체 집합에 대한 속성 관계를 표시하기 위해 개체를 노드로 표현하고 개체 집합들 사이의 관계를 링크로 연결한다.

- 계층형 데이터 모델의 장 · 단점
  - ✓데이터의 독립성이 보장된다.
  - ✓다 대 다(M:N) 관계를 처리하기 어렵다.

- 계층형 데이터 모델(Hierarchical Data Model)의 특징

✓일 대 다(1:N) 대응 관계만 존재한다.

✓개체 타입들 간에는 사이클(Cycle)이 허용되지 않는다.

✓계층을 정의하는 트리는 하나의 루트 개체 타입과 다수의 종속되는 개체 타입으로 구성된 순서 트리이다.

✓두 개체 간에는 하나의 관계만 허용된다.

문제

- 전체적인 구조가 트리 형태로 되어 있고, 두 레코드 타입 간에는 하나의 관계만 허용되는 데이터 모델은?
  ① 관계 데이터 모델
  ② 네트워크 데이터 모델
  ③ 계층 데이터 모델
  ④ 객체-관계 데이터 모델

정답 3

- 망형 데이터 모델(Network Data Model)의 개요

  ✓CODASYL.DBTG 모델이라고도 한다.

  ✓그래프를 이용해서 데이터 논리 구조를 표현한 데이터 모델이다.

  ✓상위(Owner)와 하위(Member) 레코드 사이에서 다 대 다(N:M) 대응 관계를 만족하는 구조이다.

- 망형 데이터 모델(Network Data Model)의 표현

  ✓Entity군 : 동종의 Entity그룹 예. 학과, 교수, 학생

  ✓Entity SET : 주종관계에 있는 Entity군들의 그룹

  ✓SET Membership Type : 일 대 다(1:N) 관계에 연관된 레코드 타입들을 각각 오너(Owner), 멤버(Member)라고 한다.

    - 오너(Owner)
    - 멤버(Member)

문제

- 데이터베이스의 논리적 구조 표현을 그래프 형태로 표현하며, 일 대 다(1:N) 관계에 연관된 레코드 타입들을 각각 오너(Owner), 맴버(Member)라 하고, 이들의 관계를 오너-맴버 관계라고도 일컫는 데이터 모델은?
  ① 관계형 데이터 모델
  ② 네트워크 데이터 모델
  ③ 계층적 데이터 모델
  ④ 객체지향적 데이터 모델

정답 2

정보처리기사

제1과목 데이터베이스

06 데이터 모델링 설계

문혜영교수

목차

1 데이터베이스 설계

2 설계단계별 특징

3 관계형 데이터베이스

4 릴레이션 구조

- 데이터베이스 설계 시 고려사항

  ✓무결성
  ✓일관성
  ✓회복(Recovery)
  ✓보안
  ✓효율성
  ✓데이터베이스 확장

- 데이터베이스 설계
- 1. 요구조건 분석
- 2. 개념적 설계(정보 모델링, 개념화)

❖ 데이터베이스 설계 순서

요구 분석 → 개념적 설계 → 논리적 설계 → 물리적 설계 → 구현

✓데이터베이스 설계에 있어 개념 스키마 모델링과 트랜잭션 모델링을 병행적으로 수행한다.

✓요구 분석 단계에서 나온 결과(요구 조건 명세)를 DBMS에 독립적인 E-R 다이어그램(개체 관계도)으로 작성한다.

✓DBMS에 독립적인 개념 스키마를 설계한다.

- 3. 논리적 설계(데이터 모델링)

❖ 데이터베이스 설계 순서

요구 분석 → 개념적 설계 → 논리적 설계 → 물리적 설계 → 구현

✓특정 DBMS가 지원하는 논리적 자료 구조로 변환시키는 과정이다.

✓개념 스키마를 평가 및 정재하고 DBMS에 따라 서로 다른 논리적 스키마를 설계하는 단계이다.

✓트랜잭션의 인터페이스를 설계한다.

- 4. 물리적 설계(데이터 구조화)

❖ 데이터베이스 설계 순서

요구 분석 → 개념적 설계 → 논리적 설계 → 물리적 설계 → 구현

✓물리적 저장장치에 저장할 수 있는 물리적 구조의 데이터로 변환하는 과정이다.

✓다양한 데이터베이스 응용에 대해 처리 성능을 얻기 위해 데이터베이스 파일의 저장 구조 및 액세스 경로를 결정한다.

✓수행해야 할 사항은 저장 레코드의 양식 설계, 레코드 집중(Record Clustering)의 분석 및 설계, 접근 경로 설계 등이다.

- 물리적 설계 시 고려 사항
  - ✓인덱스의 구조
  - ✓성능 향상을 위한 개념 스키마의 변경 여부 검토
  - ✓비번한 질의와 트랜잭션들의 수행속도를 높이기 위한 고려

- 물리적 설계 옵션 선택 시 고려 사항
  - ✓반응 시간(Response Time)
  - ✓공간 활용도(Space Utilization)
  - ✓트랜잭션 처리량(Transaction Throughput)

- 6. 데이터베이스 구현

✓특정 DBMS의 DDL을 이요하여 데이터베이스 스키마를 작성한다.
✓응용 프로그램을 위한 트랜잭션을 작성한다.

❖ 데이터베이스 설계 순서

요구 분석 → 개념적 설계 → 논리적 설계 → 물리적 설계 → 구현

문제

- 데이터베이스 설계 시 물리적 설계 단계에서 수행하는 사항이 아닌 것은?
  ① 저장 레코드의 양식 설계
  ② 레코드 집중의 분석 및 설계
  ③ 접근 경로 설계
  ④ 목표 DBMS에 맞는 스키마 설계

정답 4

- 관계형 데이터베이스의 Relation 구조

✓릴레이션(Relation)
  ✓구조를 나타내는 릴레이션 스키마와 실제 값들인 릴레이션 인스턴스로 구성된다.
✓튜플(Tuple)
  ✓릴레이션을 구성하는 각각의 행을 말한다.
  ✓튜플의 수를 카디널러티(Cardinality)또는 기수 대응수라고 한다.
✓속성(Attribute)
  ✓데이터베이스를 구성하는 가장 작은 논리적 단위이다.
  ✓파일 구조상의 데이터 항목 또는 데이터 필드에 해당한다.
  ✓속성의 수를 디그리(Degree) 또는 차수라고 한다.
✓도메인(Domain)
  ✓하나의 애트리뷰트가 취할 수 있는 같은 타입의 원자(Atomic)값들의 집합이다.

- 릴레이션의 특징

✓한 릴레이션에는 똑같은 튜플이 포함될 수 없으므로 릴레이션에 포함된 튜플들은 모두 상이하다.

✓한 릴레이션에 포함된 튜플 사이에는 순서가 없다.

✓속서들 간의 순서는 중요하지 않다.

✓속성의 값은 원자값만을 저장한다.

✓각 속성은 릴레이션 내에서 유일한 이름을 가진다.

문제

- 릴레이션에서 속성의 수와 튜플의 수를 의미하는 것으로 순서대로 옳게 짝지어진 것은?
  ① Cardinality, Degree
  ② Domain, Degree
  ③ Degree, Cardinality
  ④ Degree, Domain

정답 3

정보처리기사

제1과목 데이터베이스

07 관계데이터베이스 모델

문혜영교수

목차

1 키의 종류

2 키의 특징

3 무결성

4 관계 대수

- 키(Key)의 종류
  - ✓ 후보키(Candidate Key)
    - ✓ 릴레이션을 구성하는 속성들 중에서 튜플을 유일하게 식별하기 위해 사용하는 속성들의 부분집합. 기본키로 사용할 수 있는 속성들을 말한다.
    - ✓ 후보키는 유일성과 최소성을 만족시켜야 한다.
  - ✓ 기본키(Primary Key)
    - ✓ 후보키 중에서 선택한 주키(Main Key)로서 레코드를 검색하거나 정렬할 때 사용
    - ✓ 한 릴레이션에서 특정 튜플을 유일하게 구별할 수 있는 속성
  - ✓ 대체키(Alternate Key)
    - ✓ 후보키가 둘 이상일 때 기본키를 제외한 나머지 후보키를 말한다.
  - ✓ 슈퍼키(Super Key)
    - ✓ 한 릴레이션 내에 있는 속성들의 집합으로 구성된 키
    - ✓ 릴레이션에 있는 모든 튜플에 대해 유일성은 만족시키지만 최소성은 만족시키지 못한 키
  - ✓ 외래키(Foreign Key)
    - ✓ 참조되는 릴레이션의 기본키와 대응되어 릴레이션 간에 참조 관계를 표현하는데 중요한 도구
    - ✓ 관계(Relationship)를 맺고 있는 릴레이션 R1, R2에서 릴레이션 R1이 참조하고 있는 릴레이션 R2의 기본키와 같은 R1의 릴레이션의 속성을 외래키라 한다.

- 무결성

✓개체 무결성

✓릴레이션에서 기본키를 구성하는 속성은 널(Null) 값이나 중복값을 가질 수 없다.

✓참조 무결성

✓릴레이션은 참조할 수 없는 외래키 값을 가질 수 없다.

문제

- 키는 개체 집합에서 고유하게 개체를 식별할 수 있는 속성이다. 데이터베이스에서 사용되는 키의 종류에 대한 설명 중 옳지 않은 것은?
  ① 후보키(Candidate Key) : 개체들을 고유하게 식별할 수 있는 속성
  ② 슈퍼키(Super Key) : 한 개 이상의 속성들의 집합으로 구성된 키
  ③ 외래키(Foreign Key) : 다른 테이블의 기본키로 사용되는 속성
  ④ 대체키(Alternative Key) : 후보키 중에서 대표로 선정된 키

정답 4

- 관계대수의 개요

✓관계형 데이터베이스에서 원하는 정보와 그 정보를 검색하기 위해서 어떻게 유도하는가를 기술하는 절차적인 언어이다.

✓순수 관계 연산자 : Select, Project, Join, Division

✓일반 집합 연산자 : UNION(합집합), INTERSECTION(교집합), DIFFERENCE(차집합), CARTESIAN PRODUCT(교차곱)

- 순수 관계 연산자

  ✓Select : 조건을 만족하는 릴레이션의 수평적 부분집합으로 구성하며, 연산자의 기호는 그리스 문자 시스마(σ)를 사용한다 .

  ✓Project : 테이블에서 특정 속성에 해당하는 열을 선택하는데 상요되며 결과로는 릴레이션의 수직적 부분집합에 해당한다.

  ✓Join

  ✓Division

정보처리기사

제1과목 데이터베이스

08 관계데이터베이스 모델

문혜영교수

목차

1 집합연산자

2 관계해석

3 이상현상

4 정규화

- 관계대수-집합연산자-합집합

합집합(union)

① 합병 가능한 두 릴레이션 R과 S의 합집합은 R∪S로 나타낸다.

② 릴레이션 R에 속하거나 릴레이션 S에 속하는 모든 튜플로 결과 릴레이션이 구성된다.

R

번호	이름
100	정수현
200	김치국
300	고아라

S

번호	이름
100	정수현
101	윤서윤
102	이수진

R∪S

번호	이름
100	정수현
200	김치국
300	고아라
101	윤서윤
102	이수진

- 관계대수-집합연산자-교집합

교집합(intersection)

① 합병 가능한 두 릴레이션 R과 S의 교집합을 R∩S로 나타낸다.

② 릴레이션 R과 릴레이션 S에 모두 속하는 튜플로 결과 릴레이션이 구성된다.

R

번호	이름
100	정수현
200	김치국
300	고아라

S

번호	이름
100	정수현
101	윤서윤
102	이수진

R∩S →

번호	이름
100	정수현

• 관계대수-집합연산자-차집합

차집합(difference)

① 합병 가능한 두 릴레이션 R과 S의 차집합은 R-S로 나타낸다.

② 릴레이션 R에는 존재하고 릴레이션 S에는 존재하지 않는 튜플로 결과 릴레이션이 구성된다.

R

번호	이름
100	정수현
200	김치국
300	고아라

S

번호	이름
100	정수현
101	윤서윤
102	이수진

R-S →

번호	이름
200	김치국
300	고아라

R

번호	이름
100	정수현
200	김치국
300	고아라

S

번호	이름
100	정수현
101	윤서윤
102	이수진

S-R →

번호	이름
101	윤서윤
102	이수진

• 관계대수-집합연산자-카디션 프로덕트

카티션 프로덕트(cartesian product)

① 두 릴레이션 R과 S의 카티션 프로덕트는 R×S로 나타낸다.

② 릴레이션 R에 속한 각 튜플과 릴레이션 S에 속한 각 튜플을 모두 연결하여 만들어진 새로운 튜플로 결과 릴레이션이 구성된다.

R

번호	이름
100	정수현
200	김치국
300	고아라

S

번호	수량
100	25
101	35
102	45

R-S

R번호	R이름	S번호	S수량
100	정수현	100	25
100	정수현	101	35
100	정수현	102	45
200	김치국	100	25
200	김치국	101	35
200	김치국	102	45
300	고아라	100	25
300	고아라	101	35
300	고아라	102	45

- 관계 해석

✓수학의 Predicate Calculus(술어 해석)에 기반을 두고 있다.
✓튜플 관계해석과 도메인 관계해석이 있다.
✓기본적으로 관계해석과 관계대수는 관계 데이터베이스를 처리하는 기능과 능력 면에서 동등하다.
✓관계해석으로 표현한 식은 관계대수로 표현할 수 있다.
✓관계해석은 원하는 정보가 무엇이라는 것만 정의하는 비절차적 특성을 지닌다.

문제

- 관계해석(Relational Calculus)에 대한 설명으로 잘못된 것은?
  ① 튜플 관계해석과 도메인 관계해석이 있다.
  ② 원하는 정보와 그 정보를 어떻게 유도하는가를 기술하는 절차적인 특성을 지니다.
  ③ 기본적으로 관계해석과 관계대수는 관계 데이터베이스를 처리하는 기능과 능력 면에서 동등하다.
  ④ 수학의 Predicate Calculus에 기반을 두고 있다.

정답 2

- 정규화의 개요
  - ✓하나의 종속성이 하나의 릴레이션에 표현될 수 있도록 분해해가는 과정이다.
  - ✓더 작은 테이블로 분해해 가면서 종속성을 제거하는 것이다.
  - ✓정규형은 차수가 높아질수록 만족시켜야 할 제약 조건이 늘어난다.

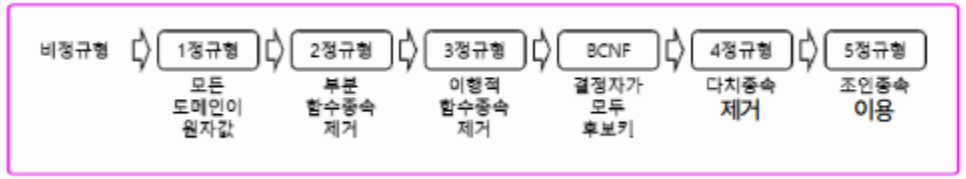
- 정규화의 목적
  - ✓어떠한 릴레이션이라도 데이터베이스 내에서 표현 가능하게 만든다.
  - ✓효과적인 검색 알고리즘을 생성할 수 있다.
  - ✓중복을 배제하여 삽입, 삭제, 갱신 이상의 발생을 방지한다.
  - ✓데이터 삽입 시 릴레이션을 재구성할 필요성을 줄인다.

- Anomaly(이상)의 개념 및 종류
  - ✓Anomaly(이상)은 정규화(Normalization)를 거치지 않으면 데이터베이스 내에 데이터들이 불필요하게 중복되어 저장되면 릴레이션 조작 시 예기치 못한 곤란한 현상이 발생하는 것으로 삽입 이상, 삭제 이상, 갱신 이상이 있다.

  - ✓삽입 이상(Insertion Anomaly) : 데이터를 삽입할 때 불필요한 데이터가 함께 삽입되는 현상
  - ✓삭제 이상(Delettion Anomaly) : 한 튜플을 삭제함으로써 연쇄 삭제로 인해 정보의 손실을 발생시키는 현상
  - ✓갱신 이상(Update Anomaly) : 튜플 중에서 일부 속성을 갱신함으로써 정보의 모순성이 발생하는 현상

❖ 정규화는 Anomaly(이상)을 제거하기 위해서 중복성 및 종속성을 배제시키는 방법으로 사용된다.

- 정규화 과정



✓1NF(제1정규형) : 릴레이션에 속한 모든 도메인이 원자값만으로 되어 있는 릴레이션이다.

✓2NF(제2정규형) : 반드시 1NF를 만족해야하고, 키가 아닌 모든 속성이 기본키에 대하여 완전 함수적 종속 관계를 만족한다.

✓3NF(제3정규형) : 반드시 2NF를 만족시켜야하고, 키가 아닌 모든 에트리뷰트가 기본키에 대해 이행적 종속 관계를 이루지 않도록 제한한 관계형이다.

✓BCNF(Boyce-Codd 정규형) : 강한 제3정규형이라고도 하며, 릴레이션 R에서 결정자가 모두 후보키인 관계형이다.

✓4NF(제4정규형)

✓5NF(제5정규형, PJ/NF) : 릴레이션 R의 모든 조인 종속성(JD)의 만족이 R의 후보키를 통해서만 만족될 때 그 릴레이션 R은 제5정규형 또는 PJ/NF에 속한다.

문제

- 제2정규형에서 제3정규형이 되기 위한 조건은?
  ① 이행적 함수 종속 제거
  ② 부분적 함수 종속 제거
  ③ 다치 종속 제거
  ④ 결정자이면서 후보 키가 아닌 것 제거

정답 1

정보처리기사

제1과목 데이터베이스

09 관계데이터베이스 모델

문혜영교수

목차

1 SQL

2 DDL

3 DML

4 DCL

- SQL(Structured Query Language)의 개요
  - ✓국제표준 데이터베이스 언어이며, 관계형 데이터베이스를 지원하는 언어로 채택하고 있으며 데이터 구조의 정의, 데이터 조작, 데이터 제어 기능이 있다.
  - ✓ SQL은 관계대수와 관계해석을 기초로 한 데이터베이스 언어로서, 정의어, 조작어, 제어어가 있다.

  - ✓데이터 정의어(DDL)는 스키마를 정의하거나 수정, 삭제하기 위해 사용되는 명령어로, CREATE, ALTER, DROP명령이 해당된다.
  - ✓데이터 조작어(DML) : 데이터의 삽입, 삭제, 수정, 검색 등의 처리를 요구하기 위한 것으로, INSERT, DELETE, UPDATE, SELECT 명령이 있다.
  - ✓데이터 제어어(DCL) : 내부적으로 필요한 규칙이나 기법을 정의하기 위해 사용되는 것으로 COMMIT, ROLLBACK, GRANT, REVOKE 명령이 있다.

문제

- 다음 SQL 명령 중 데이터 정의어(DDL)에 포함되지 않는 명령어는?
  ① CREATE
  ② ALTER
  ③ SELECT
  ④ DROP

정답 3

- DDL(데이터 정의어)
  - ✓CREATE SCHEMA
  - ✓CREATE DOMAIN
  - ✓CREATE TABLE : 기본 테이블을 정의하는 명령문
  - ✓CREATE INDEX

```
CREATE TABLE 테이블이름
(속성명 데이터타입 [NOT NULL]
PRIMARY KEY(기본키),
UNIQUE(대체키...),
FOREIGN KEY(외래키) REFERENCES 참조 테이블(기본키),
CONSTRAINT 제약조건명 CHECK(조건식) )
```

① NOT NULL은 특정 필드값에 대하여 NULL을 허용하지 않는다는 의미이다.

② 기본키는 반드시 NOT NULL로 지정되어야 한다.

③ PRIMARY KEY : 기본키 속성명을 지정한다.

④ UNIQUE : 대체키로 사용할 속성명들을 지정한다.

⑤ FOREIGN KEY는 외래키 지정시 참조 무결성을 유지하기 위해 사용되며, REFERENCES는 외래키가 참조하는 테이블과 그 테이블의 기본키를 지정한다.

사용 예

CREATE TABLE 학생 학번 VARCHAR(6), 이름 VARCHAR(10) NOT NULL, 전공 VARCHAR(10), PRIMARY KEY(학번), UNIQUE(이름),	학생테이블을 생성 학번은 6자의 문자형. 이름은 10자의 문자형, NULL값 안됨 전공은 10자의 문자형 학번은 기본키 이름은 대체키
FOREIGN KEY(전공) REFERENCES 학과(학과코드)	전공은 학과테이블의 학과코드를 참조하는 외래키
CONSTRAINT 성별제약 CHECK(성별='남')	'성별' 속성의 값에서 '남'만 저장할 수 있게 지정한 제약조건의 이름은 '성별제약'이다.

ALTER

① 테이블의 정의를 변경하는 명령문이다.

```
ALTER TABLE 테이블명 ADD 추가할 속성명 데이터타입;
ALTER TABLE 테이블명 ALTER 속성명[SET DEFAULT '기본값'];
ALTER TABLE 테이블명 DROP 삭제할 속성명;
```

② ALTER TABLE 학생 ADD 전화번호 CHAR(15);

⇒ 학생테이블의 전화번호 속성을 추가

③ ALTER TABLE 학생 ALTER 주소 SET DEFAULT '서울시';

⇒ 학생테이블의 주소의 기본값을 '서울시'로 변경

④ ALTER TABLE 학생 DROP 주소;

⇒ 학생테이블의 주소 속성을 제거

DROP

① 스키마, 도메인, 테이블, 뷰, 인덱스를 제거하는 명령이다.

```
DROP TABLE 테이블명 [CASCADE | RESTRICT];
```

② CASCADE : 참조하는 모든 테이블도 함께 제거한다.

③ RESTRICT : 다른 테이블이 참조 중이면 삭제하지 않는다.

문제

- SQL 구문과 의미가 잘못 연결된 것은?
  ① CREATE – 테이블 생성
  ② DROP – 레코드 삭제
  ③ UPDATE – 자료 갱신
  ④ DESC – 내림차순 정렬

정답 2

INSERT

① 새로운 튜플을 삽입하는 명령이다.

```
INSERT INTO 테이블명(속성1, 속성2...) VALUE(데이터1, 데이터2..)
```

INSERT INTO 학생(번호, 이름, 학년, 학과) VALUES('010', '정수현', 2, '사이버 안보')

학생 테이블의 번호, 이름, 학년, 학과 속성에 '010', '정수현', 2, '사이버 안보'를 삽입하시오.

명령	내용
INSERT INTO 학생 VALUES('홍길동', '컴퓨터', '서울')	학생테이블에 홍길동, 컴퓨터, 서울을 삽입하시오.
INSERT INTO 학생(이름, 수강료) VALUES (홍길동, 120)	학생테이블의 이름과 수강료에 홍길동, 120을 추가하시오.
INSERT INTO 수강정보(이름, 과목, 수강료) SELECT 이름, 과목, 수강료 FROM 학생 WHERE 주소='서울'	학생 테이블에서 주소가 서울인 데이터의 이름, 과목, 수강료의 내용을 수강정보 테이블의 이름, 과목, 수강료에 추가하시오.

DELETE

① 테이블의 특정 튜플을 삭제하는 명령이다.

```
DELETE  FROM 테이블명 WHERE 조건
```

② DELETE FROM 학생 WHERE 번호='101'

학생테이블에서 번호가 101 튜플을 삭제

DELETE FROM 학생 WHERE 과목='컴퓨터'	학생 테이블에서 과목이 컴퓨터인 튜플을 삭제하시오.

UPDATE 학생 SET 이름='홍길동' WHERE 번호='101'

⇒ 학생테이블에서 번호가 '101'인 튜플의 이름을 '홍길동'으로 수정하시오.

UPDATE 학생 SET 수강과목='컴퓨터' WHERE 이름='홍길동'	학생 테이블에서 이름이 홍길동인 데이터의 수강과목을 컴퓨터로 변경하시오.
UPDATE 학생 SET 수강료=수강료+10000 WHERE 수강과목='컴퓨터'	학생 테이블에서 수강과목이 컴퓨터인 데이터의 수강료에 만 원을 더하시오.

SELECT명령

① 특정한 조건을 만족하는 튜플을 검색하는 명령이다.

```
SELECT [DISTINCT] 속성명
FROM 테이블명
WHERE 조건
ORDER BY 정렬할 속성명 [ASC] 또는 [DESC]
GROUP BY 그룹화 속성명
HAVING 그룹조건;
```

② DISTINCT는 중복된 데이터를 제거하고 표시할 때 사용한다.

③ FROM절은 질의에 필요한 테이블을 나열한다.

④ WHERE절은 추출할 조건을 기술한다.

⑤ ORDER BY절은 필드별로 오름차순, 내림차순을 지정할 수 있다.

⑥ GROUP BY절은 그룹화할 기준 필드들을 기술한다.

⑦ HAVING절은 그룹들 중에서 선택할 조건을 기술하며 내장된 그룹함수와 함께 기술한다.

<사원>테이블의 모든 튜플을 검색하시오.

```
Select * From 사원;
Select 사원.* From 사원;
Select 이름, 부서, 생일, 주소, 급여 From 사원;
Select 사원.이름, 사원.부서, 사원.생일, 사원.주소, 사원.급여 From 사원;
```

<사원>

이름	부서	생일	주소	급여
홍병학	기획	1990-03-05	효월동	220
임희상	인터넷	1979-01-09	석월동	180
황재원	편집	1985-07-21	후월동	200
김지용	편집	1973-10-22	효월동	190
성희원	기획	1984-02-20	후월동	200
장수빈	편집	1987-03-11	고월동	220
일수현	기획	1978-04-29	원월동	210
강지현	인터넷	1980-12-11		190

<활동>

이름	특기	경력
김지용	골프	10
성희원	수영	5
일수현	댄스	15
임희상	등산	8

<사원>테이블에서 주소속성만 나타내는데 같은 주소는 한번만 출력하시오.

```
Select Distinct 주소
From 사원;
```

주소
효월동
석월동
후월동
고월동
원월동

<사원>

이름	부서	생일	주소	급여
홍병학	기획	1990-03-05	효월동	220
임희상	인터넷	1979-01-09	석월동	180
황재원	편집	1985-07-21	후월동	200
김지용	편집	1973-10-22	효월동	190
성희원	기획	1984-02-20	후월동	200
장수빈	편집	1987-03-11	고월동	220
일수현	기획	1978-04-29	원월동	210
강지현	인터넷	1980-12-11		190

<활동>

이름	특기	경력
김지용	골프	10
성희원	수영	5
일수현	댄스	15
임희상	등산	8

<사원>테이블에서 부서가 '편집'부인 모든 튜플을 검색하시오.

```
Select *
From 사원
Where 부서='편집';
```

이름	부서	생일	주소	급여
황재원	편집	1985-07-21	후월동	200
김지용	편집	1973-10-22	효월동	190
장수빈	편집	1987-03-11	고월동	220

<사원>

이름	부서	생일	주소	급여
홍병학	기획	1990-03-05	효월동	220
임희상	인터넷	1979-01-09	석월동	180
황재원	편집	1985-07-21	후월동	200
김지용	편집	1973-10-22	효월동	190
성희원	기획	1984-02-20	후월동	200
장수빈	편집	1987-03-11	고월동	220
일수현	기획	1978-04-29	원월동	210
강지현	인터넷	1980-12-11		190

<활동>

이름	특기	경력
김지용	골프	10
성희원	수영	5
일수현	댄스	15
임희상	등산	8

; <사원>테이블에서 부서가 '편집'부이면서 주소가 '후월동'인 튜플을 검색하시오.

```
Select *
From 사원
Where 부서='편집' And 주소='후월동';
```

이름	부서	생일	주소	급여
황재원	편집	1985-07-21	후월동	200

<사원>

이름	부서	생일	주소	급여
홍병학	기획	1990-03-05	효월동	220
임희상	인터넷	1979-01-09	석월동	180
황재원	편집	1985-07-21	후월동	200
김지용	편집	1973-10-22	효월동	190
성희원	기획	1984-02-20	후월동	200
장수빈	편집	1987-03-11	고월동	220
일수현	기획	1978-04-29	원월동	210
강지현	인터넷	1980-12-11		190

<활동>

이름	특기	경력
김지용	골프	10
성희원	수영	5
일수현	댄스	15
임희상	등산	8

<사원>테이블에서 성이 '홍'인 사람의 튜플을 검색하시오

```
Select *
From 사원
Where 이름 Like '홍%';
```

이름	부서	생일	주소	급여
홍병학	기획	1990-03-05	효월동	220

<사원>

이름	부서	생일	주소	급여
홍병학	기획	1990-03-05	효월동	220
임희상	인터넷	1979-01-09	석월동	180
황재원	편집	1985-07-21	후월동	200
김지용	편집	1973-10-22	효월동	190
성희원	기획	1984-02-20	후월동	200
장수빈	편집	1987-03-11	고월동	220
일수현	기획	1978-04-29	원월동	210
강지현	인터넷	1980-12-11		190

<활동>

이름	특기	경력
김지용	골프	10
성희원	수영	5
일수현	댄스	15
임희상	등산	8

<사원>테이블에서 생일이 '1970-01-01'~'1980-12-31'사이인 튜플을 검색하시오.

```
Select *
From 사원
Where 생일 Between #1970-001-01# And #1980-12-31#;
```

이름	부서	생일	주소	급여
임희상	인터넷	1979-01-09	석월동	180
김지용	편집	1973-10-22	효월동	190
일수현	기획	1978-04-29	원월동	210
강지현	인터넷	1980-12-11		190

<사원>

이름	부서	생일	주소	급여
홍병학	기획	1990-03-05	효월동	220
임희상	인터넷	1979-01-09	석월동	180
황재원	편집	1985-07-21	후월동	200
김지용	편집	1973-10-22	효월동	190
성희원	기획	1984-02-20	후월동	200
장수빈	편집	1987-03-11	고월동	220
일수현	기획	1978-04-29	원월동	210
강지현	인터넷	1980-12-11		190

<활동>

이름	특기	경력
김지용	골프	10
성희원	수영	5
일수현	댄스	15
임희상	등산	8

. <사원>테이블에서 주소가 NULL인 튜플을 검색하시오.

```
Select *
From 사원
Where 주소 Is Null
```

이름	부서	생일	주소	급여
강지현	인터넷	1980-12-11		190

<사원>

이름	부서	생일	주소	급여
홍병학	기획	1990-03-05	효월동	220
임희상	인터넷	1979-01-09	석월동	180
황재원	편집	1985-07-21	후월동	200
김지용	편집	1973-10-22	효월동	190
성희원	기획	1984-02-20	후월동	200
장수빈	편집	1987-03-11	고월동	220
일수현	기획	1978-04-29	원월동	210
강지현	인터넷	1980-12-11		190

<활동>

이름	특기	경력
김지용	골프	10
성희원	수영	5
일수현	댄스	15
임희상	등산	8

. <사원>테이블에서 주소를 기준으로 오름차순 정렬시켜서 상위 2개의 튜플만 표시하시오.

```
Select Top 2
From 사원
Order By 주소 Asc;
```

이름	부서	생일	주소	급여
장수빈	편집	1987-03-11	고월동	220
임희상	인터넷	1979-01-09	석월동	180

<사원>

이름	부서	생일	주소	급여
홍병학	기획	1990-03-05	효월동	220
임희상	인터넷	1979-01-09	석월동	180
황재원	편집	1985-07-21	후월동	200
김지용	편집	1973-10-22	효월동	190
성희원	기획	1984-02-20	후월동	200
장수빈	편집	1987-03-11	고월동	220
일수현	기획	1978-04-29	원월동	210
강지현	인터넷	1980-12-11		190

<활동>

이름	특기	경력
김지용	골프	10
성희원	수영	5
일수현	댄스	15
임희상	등산	8

문제

- SQL 문장에서 GROUP BY절에 의해 선택된 그룹의 탐색 조건을 지정할 수 있는 것은?
  ① HEAVING
  ② ORDER BY
  ③ UNION
  ④ JOIN

정답 1

문제

- SQL에서 DELETE 명령에 대한 설명으로 옳지 않은 것은?
  ① 테이블의 행을 삭제할 때 사용한다.
  ② WHERE 조건절이 없는 DELETE 명령을 수행하면 DROP TABLE 명령을 수행했을 때와 같은 효과를 얻을 수 있다.
  ③ SQL을 사용 용도에 따라 분류할 경우 DML에 해당한다.
  ④ 기본 사용 형식은 “DELETE FROM 테이블 [WHERE 조건];” 이다.

정답 2

정보처리기사

제1과목 데이터베이스

10 관계데이터베이스 모델

문혜영교수

목차

1 내장 SQL

2 뷰

3 뷰 장단점

4 카탈로그

- 내장 SQL
  - ✓내장 SQL(Embedded SQL)의 정의
    - ✓응용 프로그램 내에 SQL 문장을 내포하여 프로그램이 실행될 때 함께 실행되도록 호스트 프로그램 언어로 만든 프로그램에 삽입된 SQL이다.
  - ✓내장 SQL의 특징
    - ✓호스트 언어에서 실행문이 나타날 수 있는 곳이면 프로그램의 어느 곳에서나 사용할 수 있다.
    - ✓내장 SQL 프로그램은 컴파일보다 우선하는 전처리기에 의해 처리된다.
    - ✓호스트 변수와 데이터베이스 필드의 이름은 같아도 된다.
    - ✓내장 SQL문에 사용된 호스트 변수의 데이터 타입은 이에 대응하는 데이터베이스 필드의 SQL 데이터 타입과 일치하여야 한다.

- ✓내장 SQL
  - ✓호스트 언어의 실행문과 구분시키는 방법
    - ✓C/C++에서 내장 SQL문은 $와 세미콜론(;)문자 사이에 기술한다.
    - ✓Visual Basic에서는 내장 SQL문 앞에 'EXEC SQL'을 기술한다.
    - ✓내장 SQL에서 사용하는 호스트 변수는 변수 앞에 콜론(: 문자)를 붙인다.
  - ✓커서(Cursor)
    - ✓내장 SQL문의 수행결과로 반환될 수 있는 복수의 튜플들을 액세스 할 수 있도록 해주는 개념이다.
    - ✓DECLARE
    - ✓OPEN

문제

- 삽입 SQL(Embedded SQL)에 대한 설명으로 옳지 않은 것은?
  ① 응용 프로그램에 삽입되어 상용되는 SQL이다.
  ② SQL 문장의 식별자로서 EXEC SQL을 앞에 기술한다.
  ③ 호스트 변수와 데이터베이스 필드의 이름은 같아도 무방하다.
  ④ 호스트 언어의 변수는 SQL 변수와 구별하기 위하여 앞에 % 기호를 붙인다.

정답 4

- 뷰(View)의 개요
  - ✓사용자에게 접근이 허용된 자료만을 제한적으로 보여주기 위해 하나 이상의 기본 테이블로부터 유도된 이름을 가지는 가상 테이블이다.
  - ✓저장장치 내에 물리적으로 존재하지 않지만 사용자에게는 있는 것처럼 간주한다.
- 뷰(View)의 특징
  - ✓가상 테이블이기 때문에 물리적으로 구현되어 있지 않다.
  - ✓필요한 데이터만 뷰로 정의해서 처리할 수 있기 때문에 관리가 용이하고 명령문이 간단해진다.
  - ✓정의된 뷰는 다른 뷰의 정의에 기초가 될 수 있다.
  - ✓뷰를 삭제하면 그 뷰를 기초로 정의된 다른 뷰도 자동으로 삭제된다.

- 뷰(View)의 장 · 단점
  - ✓장점
    - ✓논리적 데이터 독립성을 제공한다.
    - ✓여러 사용자의 상이한 요구를 지원해 준다.
    - ✓사용자의 데이터 관리를 간단하게 해준다.
    - ✓접근 제어를 통한 자동 보안이 제공된다.
  - ✓단점
    - ✓독립적인 인덱스를 가질 수 없다.
    - ✓ALTER VIEW문을 사용할 수 없다. 즉, 뷰의 정의를 변경할 수 없다.
    - ✓삽입, 갱신, 삭제 연산에는 제약이 따른다.

- 뷰(View) 정의문
  - ✓CREATE VIEW 명령으로 정의한다.
  - ✓뷰의 정의는 ALTER문을 이용하여 변경할 수 없다.

- 뷰(View) 삭제문
  - ✓뷰(View)를 삭제할 때 사용하는 명령은 DROP이다.
  - ✓뷰는 ALTER문을 이용하여 변경할 수 없으므로 필요한 경우는 삭제한 후 재생성한다.
    - ✓RESTRICT 옵션 : 뷰를 다른 곳에서 참조하고 있으면 삭제가 취소된다.
    - ✓CASCADE 옵션 : 뷰를 참조하는 다른 뷰나 제약 조건가지 모두 삭제된다.

문제

- 뷰(View)에 관한 설명 중 잘못된 것은?
  ① 뷰는 SQL에서 CREATE VIEW 명령으로 정의된다.
  ② 뷰는 하나 이상의 기본 테이블로부터 유도되어 만들어지는 가상 테이블이다.
  ③ 뷰는 INSERT, DELETE, UPDATE 등을 이용한 삽입, 삭제, 갱신 연산을 항상 허용한다.
  ④ 뷰의 정의는 ALTER문을 이용하여 변경할 수 없다.

정답 3

- 시스템 카탈로그(System Catalog)의 의미

  ✓시스템 자신이 필요로 하는 여러 가지 개체에 관한 정보를 포함한 시스템 데이터베이스이다.

  ✓개체들로서는 기본 테이블, 뷰, 인덱스, 패키지, 접근 권한 등이 있다.

  ✓시스템 카탈로그 내의 각 테이블은 데이터베이스에 포함되는 모든 데이터 객체들에 대한 정의나 명세에 관한 정보를 유지 관리하는 시스템이다.

  ✓카탈로그를 좁은 의미로 자료 사전(Data Dictionary)이라고도 한다.

  ✓카탈로그(Data Dictionary)에 저장된 정보를 메타 데이터(Meta-Data)라고 한다.

- 시스템 카탈로그(System Catalog)의 특징

✓카탈로그 자체도 시스템 테이블로 구성되어 있어 일반 이용자도 SQL을 이용하여 내용을 검색해 볼 수 있다.

✓데이터베이스 시스템에 따라 상이한 구조를 갖는다.

✓DBMS가 스스로 생성하고 유지하는 데이터베이스 내의 특별한 테이블의 집합체이다.

✓카탈로그 갱신 : 사용자가 SQL문을 실행시켜 기본 테이블, 뷰, 인덱스 등에 변화를 주면 시스템이 자동으로 갱신한다.

- 카탈로그을 참조하기 위한 DBMS 내의 모듈 시스템
  - ✓데이터 정의어 번역기(DDL Compiler)
  - ✓데이터 조작어 번역기(DML Compiler)
  - ✓Data Directory
    - ✓데이터 사전에 수록된 데이터를 실제로 접근하는 데 필요한 정보를 관리 유지하는 시스템이다.
    - ✓시스템 카탈로그는 사용자와 시스템 모두 접근할 수 있지만 데이터 디렉토리는 시스템만 접근할 수 있다.
  - ✓질의 최적화기
  - ✓트랜잭션 처리기

문제

- 시스템 카탈로그에 대한 설명으로 옳지 않은 것은?
  ① 데이터베이스의 무결성 유지를 위하여 사용자가 직접 시스템 카탈로그를 갱신해야 한다.
  ② 일반 질의어를 이용해 그 내용을 검색할 수 있다.
  ③ DBMS가 스스로 생성하고, 유지하는 데이터베이스 내의 특별한 테이블의 집합이다.
  ④ 데이터베이스 스키마에 대한 정보를 제공한다.

정답. 1

정보처리기사

제1과목 데이터베이스

11 데이터베이스 고급기능

문혜영교수

목차

1 트랜잭션

2 연산

3 회복

4 병행제어

- 트랜잭션의 정의

✓데이터베이스의 상태를 변환시키는 하나의 논리적 기능을 수행하기 위한 작업의 단위이다.

✓회복 작업 시 처리되는 작업의 논리적 단위이다.

✓하나의 트랜잭션은 Commit되거나 Rollback된다.

- 트랜잭션의 특성
  - ✓Atomicity(원자성)
    - ✓데이터베이스에 모두 반영되도록 완료(Commit)되든지 아니면 전혀 반영되지 않도록 복구(Rollback)되어야 한다.
    - ✓트랜잭션은 완벽히 수행되어야 하며, 모두가 완벽히 수행되지 않고 어느 하나라도 오류가 발생하면 트랜잭션 전부가 취소되어야 한다.
  - ✓Consistency(일관성)
    - ✓시스템의 고정 요소는 트랜잭션 수행 전과 트랜잭션 수행 완료 후의 상태가 같아야 한다.
  - ✓Isolation(독립성, 격리성, 순차성)
    - ✓트랜잭션이 실행 중에 다른 트랜잭션의 연산이 끼어들 수 없다.
  - ✓Durability(영속성, 지속성)
    - ✓완료된 트랜잭션의 결과는 시스템이 고장나더라도 영구적으로 기억되는 성질

- Commit, Rollback 연산
  - ✓Commit 연산
    - ✓한 작업의 논리적 단위가 성공적으로 끝나고, 데이터베이스가 다시 일관된 상태에 있으며, 이 트랜잭션이 행한 갱신 연산이 완료된 것을 트랜잭션 관리자에게 알려주는 연산이다.
  - ✓Rollback 연산
    - ✓트랜잭션 처리가 비정상적으로 종료되어 이 트랜잭션이 행한 모든 연산을 취소(Undo)하는 연산이다.

- 트랜잭션의 상태
  - ✓활동(Active)
  - ✓실패(Failed)
  - ✓철회(Aborted) : 트랜잭션 처리가 비정상적으로 종료되어 Rollback 연산을 수행한 상태
  - ✓부분 완료(Partially Committed) : 트랜잭션의 마지막 연산까지 실행했지만 Commit 연산이 실행되기 직전의 상태
  - ✓완료(Committed) : 트랜잭션이 성공적으로 종료되어 Commit 연산을 실행한 후의 상태

문제

- 트랜잭션(Transaction)의 특성에 대한 설명으로 옳지 않은 것은?
  ① 원자성(Atomicity)은 트랜잭 의 일부만 수행된 상태로 종료될 수 있다는 특성을 의미한다.
  ② 일관성(Consistency)은 시스템의 고정 요소는 트랜잭션 수행 전과 수행 완료 후에 같아야 한다는 특성을 의미한다.
  ③ 고립성(Isolation) 은 트랜잭션이 실행될 때마다 다른 트랜잭션의 간섭을 받지 않아야 한다는 성질을 의미한다.
  ④ 지속성(Durability)은 트랜잭션의 완료 결과가 데이터베이스에 영구히 기억되는 성질을 의미한다.

정답 1

- 회복(Recovery)의 정의
  - ✓트랜잭션들을 수행하는 도중 장애가 발생하여 데이터베이스가 손상되었을 때 손상되기 이전의 정상 상태로 복구하는 작업이다.

- 회복 관리기(Recovery Management)
  - ✓트랜잭션이 데이터베이스에 생성했던 모든 변화를 취소(Undo)시키고 트랜잭션 수행 이전의 원래 상태로 복구하는 역할을 담당한다.

문제

- 트랜잭션들을 수행하는 도중 장애로 인해 손상된 데이터베이스를 손산되기 이전의 정상적인 상태로 복구시키는 작업은?
  ① Recovery
  ② Restart
  ③ Commit
  ④ Abort

정답 1

- 병행제어의 목적
  - ✓데이터베이스의 공유를 최대화한다.
  - ✓시스템의 활용도를 최대화한다.
  - ✓데이터베이스의 일관성을 유지한다.
  - ✓사용자에 대한 응답시간을 최소화한다.

- 병행수행의 문제점
  - ✓병행제어(Concurrency Control) 기법에 의한 제어 없이 트랜잭션들이 데이터베이스에 동시에 접근하도록 허용할 경우 발생하는 문제점
    - ✓갱신 분실
    - ✓비완료 의존성
    - ✓모순성
    - ✓연쇄 복귀

- 병행제어 기법의 종류
  - ✓로킹(Locking)
    - ✓ 로킹 단위(Locking Granularity)
      - ✓ 한거번에 로킹할 수 있는 단위
      - ✓ 데이터베이스, 파일, 레코드, 필드 등은 로킹 단위가 될 수 있다.
      - ✓ 로킹 단위가 크면 로크 수가 작아 관리하기 쉽지만 병행성 수준이 낮아지고 , 로킹 단위가 작으면 로크 수가 많아 관리하기 복잡해 오버헤드가 증가하지만 병행성 수준이 높아진다.
  - ✓타임 스템프 순서(Time Stamp Ordering)
    - ✓동시성 제어를 위한 직렬의 기법으로 트랜잭션 간의 순서를 미리 정하는 방법
  - ✓최적 병행수행(검증 기법, 확인 기법, 낙관적 기법)
  - ✓다중 버전 기법

문제

- 병행제어의 로킹(Locking) 단위에 대한 설명으로 옳지 않은 것은?
  ① 로킹 단위가 작아지면 병행성 수준이 낮아진다.
  ② 데이터베이스, 파일, 레코드 등은 로킹 단위가 될 수 있다.
  ③ 로킹 단위가 작아지면 로킹 오버헤드가 증가한다.
  ④ 한꺼번에 로킹할 수 있는 단위를 로킹 단위라고 한다.

정답 1

정보처리기사

제1과목 데이터베이스

12 데이터베이스 고급기능

문혜영교수

목차

1 무결성

2 보안

3 암호화 기법

4 분산데이터

- 무결성(Integrity)
  - ✓정확성
  - ✓검증 프로그램이 무결성을 검증하기 위해 무결성 규정을 사용한다.
  - ✓무결성 규정에는 규정 이름, 검사 시기(트리거 조건), 제약 조건(프레디킷), 위반 조치 등을 명시한다.
- 무결성의 종류
  - ✓널 무결성
  - ✓고유 무결성
  - ✓참조 무결성
  - ✓도메인 무결성
  - ✓키 무결성
  - ✓관계 무결성
  - ✓개체 무결성

문제

- 릴레이션은 참조할 수 없는 외래키 값을 가질 수 없음을 의미하는 제약 조건은?
  ① 개체 무결성
  ② 참조 무결성
  ③ 보안 무결성
  ④ 정보 무결성

정답 2

- 데이터베이스 보안의 개요
  - ✓ 보안을 위한 데이터 단위는 테이블 전체로부터 특정 테이블의 특정한 행과 열 위치에 있는 특정한 데이터 값에 이르기까지 다양하다.
  - ✓ 각 사용자들은 일반적으로 서로 다른 객체에 대하여 다른 접근 권리 또는 권한을 갖게 된다.
- 암호화 기법
  - ✓ 개인키 암호 방식 = 비밀키 암호 방식
    - ✓ DES : 평문을 64Bit로 블록화하고, 실제의 길이는 56Bit를 이용
  - ✓ 공개키 암호 방식
    - ✓ RSA : (공개키), (비밀키)

- ❖무결성(Integrity)과 보안(Security)
  - ❖무결성(Integrity)
    - ✓권한이 있는 사용자로부터 데이터베이스를 보호하는 것이고,
    - ✓보안은 권한이 없는 사용자로부터 데이터 베이스를 보호하는 것
  - ❖보안
    - ✓데이터베이스를 사용하고자 할 때 언제든지 사용할 수 있도록 보장하는 것이고, 무결성은 정확하게 사용할 수 있도록 보장하는 것이다.

문제

- 암호화 기법 중 암호화 알고리즘과 암호화키는 공개해서 누구든지 평문을 암호문으로 만들 수 있지만, 해독 알고리즘과 해독키는 비밀로 유지하는 기법을 무엇이라 하는가?
  ① DES(Data Encryption Standard) 기법
  ② 공중키(Public Key) 암호화 기법
  ③ 대체(Substitution) 암호화 기법
  ④ 전치(Transposed) 암호화 기법

정답 2

- 분산 데이터베이스 설계 시 고려사항
  - ✓작업부하(Work Load)의 노드별 분산 정책
  - ✓지역의 자치성 보장 정책
  - ✓데이터의 일관성 정책
- 분산 데이터베이스의 목표
  - ✓위치 투명성(Location Transparency)
    - ✓ 데이터베이스의 실제 위치를 알 필요 없이 단지 데이터베이스의 논리적인 명칭만으로 액세스할 수 있다.
  - ✓중복 투명성(Replication Transparency)
    - ✓ 트랜잭션이 데이터의 중복 개수나 중복 사실을 모르고도 데이터 처리가 가능
  - ✓병행 투명성(Concurrency Transparency)
    - ✓ 분산 데이터베이스와 관련된 다수의 트랜잭션들이 동시에 실현되더라도 그 트랜잭션의 결과는 영향을 받지 않는다.
  - ✓장애 투명성(Failure Transparency)
    - ✓ 트랜잭션, DBMS, 네트워크, 컴퓨터 장애에도 불구하고 트랜잭션을 정확하게 처리한다.

- 분산 데이터베이스의 장 · 단점
  - ✓장점
    - ✓지역 자치성이 높다
    - ✓자료의 고유성이 향상된다.
    - ✓시스템 성능이 향상된다.
    - ✓효용성과 융통성이 높다.
    - ✓신뢰성 및 가용성이 높다.
    - ✓점진적 시스템 용량 확장이 용이하다.
  - ✓단점
    - ✓데이터베이스 설계가 어렵다.
    - ✓소프트웨어 개발 비용이 증가한다.

문제

- 분산 데이터베이스의 장점으로 거리가 먼 것은?
  ① 지역 자치성이 높다.
  ② 효용성과 융통성이 높다.
  ③ 점진적 시스템 용량 확장이 용이하다.
  ④ 소프트웨어 개발 비용이 저렴하다.

정답. 4

정보처리기사

제1과목 데이터베이스

13 자료구조

문혜영교수

목차

1 자료구조

2 선형구조

3 스택

4 큐, 데큐

- 자료 구조의 분류
  - ✓선형 구조
    - ✓리스트 : 선형 리스트(배열), 연결 리스트
    - ✓스택(Stack)
    - ✓큐(Queue)
    - ✓데크(Deque)
  - ✓비선형 구조
    - ✓트리(Tree)
    - ✓그래프(Graph)

- 순차 리스트(Sequential)가 아닌 것은?
  - ① 배열(Array)
  - ② 트리(Tree)
  - ③ 데크(Deque)
  - ④ 스택(Stack)

정답 2

- 선형 리스트(Linear List)
  - ✓ 배열과 같이 연속되는 기억장소에 저장되는 리스트

- 선형 리스트의 특징
  - ✓ 기억장소 이용 효율이 가장 좋다.
  - ✓ 삽입, 삭제 시 자료의 이동이 필요하기 때문에 작업이 번거럽다.

- 연결 리스트(Linked List)
  - ✓ 자료 항목의 순서에 따라 노드의 포인트 부분을 이용하여 서로 연결

- 연결 리스트의 특징
  - ✓ 노드의 삽입, 삭제 작업이 용이하다.
  - ✓ 순차 리스트에 비해 기억공간 이용 효율이 좋지 않다.
  - ✓ 연결을 위한 포인터를 찾는 시간이 필요하기 때문에 접근 속도가 느리다.
  - ✓ 중간 노드 연결이 끊어지면 그 다음 노드를 찾기 힘들다.
  - ✓ 희소 행렬을 링크드 리스트로 표현하면 기억장소가 절약된다.

문제

- 희소 행렬을 링크드 리스트(Linked List)로 표현할 때 가장 큰 장점은?
  ① 기억장소가 절약된다.
  ② 임의 위치 액세스(Random Access)가 가능하다.
  ③ 이진 검색(Binary Search)이 가능하다.
  ④ 행렬 간의 연산 시간을 줄일 수 있다.

정답 1

- 스택(Stack)
  - ✓스택의 개념
    - ✓리스트의 내의 데이터 삽입, 삭제가 한쪽 끝에서 이루어지는 데이터 구조
    - ✓후입선출(LIFO; Last In First Out)
  - ✓자료의 삽입(Push)

```
Top=Top + 1
If Top > M Then
    Overflow
Else
    X(Top) ← item
```

✓자료의 삭제(Pop Up)

```
If Top = 0 Then
    Underflow
Else
    item ← X(Top)
    Top=Top – 1
```

Stack의 응용 분야

- ✓서브루틴의 복기 번지 저장
- ✓인터럽트 처리
- ✓수식의 계산

문제

- 리스트 내의 데이터 삽입, 삭제가 한쪽 끝에서 이루어지는 데이터 구조를 무엇이라 하는가?
  ① 스택(Stack)
  ② 큐(Queue)
  ③ 데크(Deque)
  ④ 원형 큐(Circular Queue)

정답 1

- 큐(Queue)
  - ✓선형 리스트의 한쪽에서는 삽입 작업이, 다른 한쪽에서는 삭제 작업이 수행되는 자료 구조
  - ✓선입선출(FIFO; First In First Out)
  - ✓ 운영체제의 작업 스케줄링에 사용한다.

- 데크(Deque)
  - ✓삽입과 삭제가 리스트의 양쪽 끝에서 모두 발생할 수 있는 자료 구조
  - ✓Stack과 Queue의 장점만 따서 구성한 것이다.
  - ✓입력 제한 데크 : Scroll
  - ✓출력 제한 데크 : Shelf

문제

- 운영체제의 작업 스케줄링 등에 응용되는 것으로 가장 적합한 자료 구조는?
  ① 스택(Stack)
  ② 큐(Queue)
  ③ 연결 리스트(Linked List)
  ④ 트리(Tree)

정답 2

정보처리기사

제1과목 데이터베이스

14 자료구조

문혜영교수

목차

1 트리

2 이진트리

3 트리 운행법

4 수식 표기법

- 트리 관련 용어
  - ✓ 근 노드(Root Node) : 트리의 맨 위에 있는 노드
  - ✓ 디그리(Degree, 차수) : 가지의 수
  - ✓ 단말 노드(Terminal Node) :Degree가 0인 노드
  - ✓ 부모 노드(Parent Node) : 이전 레벨의 노드
  - ✓ 형제 노드(Brother Node) : 동일한 부모 갖는 노드
  - ✓ Level : 근 도드의 Level을 1로 가정한 후 어떤 노드의 Level이 L이면 자식 노드의 Level은 L+1이다.
  - ✓ 트리의 디그리 : 노드들의 디그리 중에서 가장 많은 수

- 트리 구조에 대한 용어 설명 중 옳지 않은 것은?
  - ① 어떤 노드의 서브트리 수를 그 노드의 차수라고 한다.
  - ② 차수가 0인 노드를 단말 노드라고 한다.
  - ③ 같은 부모 노드를 가지는 노드를 형제 노드라고 한다.
  - ④ 모든 노드는 하나의 부모 노드를 가진다.

정답 4

- 이진 트리의 특성
  - ✓터미널 노드수 n_0, 차수가 2인 노드 수가 n_2라 할 때 $n_0 = n_2 + 1$

- 이진 트리의 종류
  - ✓정이진 트리(Full Binary Tree)
    - ✓깊이(.Depth)가 K일 때 전체 노드의 수가 $2^k - 1$개의 노드이고, 레벨 i마다 2^{i-1}개의 노드들로 꽉 찬 트리를 말한다.
  - ✓전이진 트리(Complete Binary Tree)
  - ✓사향 이진 트리(Skewed Binary Tree)
    - ✓왼쪽 사향 이진 트리
    - ✓오른쪽 사향 이진 트리

문제

- 깊이가 6인 이진 트리의 최대 노드 수는?
  ① 63
  ② 64
  ③ 65
  ④ 66

정답 1

- 트리의 운행법
  - ✓Preorder 운행 : Root → Left → Right
  - ✓Inorder 운행 : Left → Root → Right
  - ✓Postorder 운행 : Left → Right → Root

- 수식의 표기법
  - ✓전위 표기법(PreFix) : 연산자 → Left → Right
  - ✓중위 표기법(InFix) : Left → 연산자 →Right
  - ✓후위 표기법(PostFix) : Left → Right→ 연산자

- ✓Infix 표기를 Prefix로 변환하기
  - ✓연산 우선순위에 따라 괄호를 묶는다.
  - ✓연산자를 해당 괄호의 앞(왼쪽)으로 옮긴다.
  - ✓필요없는 괄호를 제거한다.
- ✓Infix 표기를 Postfix로 변환하기
- ✓Postfix로 표기된 수식을 Infix로 변환하기
- ✓Prefix로 표기된 수식을 Infix로 변환하기

- 스레드 이진 트리(Threaded Binary Tree)
  - ✓이진 트리(Binary Tree)에서 발생하는 널(Null) 링크를 트리 운행에 필요한 다른 노드의 포인터로 사용하도록 고안된 트리를 말한다.

- 다음과 같은 중위식(Infix)을 후위식(Postfix)으로 올바르게 표현한 것은?

  X = A + (B + C / D) × E – F

  ① X = A + B + C / D × E – F
  ② X A B C D / + E × + F – =
  ③ = X – + A × + B / C D E F
  ④ X A B C D E F = + + / × –

정답 2

정보처리기사

제1과목 데이터베이스

15 자료구조

문혜영교수

목차

4 내부정렬

3 정렬

2 그래프 표현

1 그래프

- 그래프의 정의
  - ✓통신망(Network), 교통망, 유리화학 구조식등에 응용된다.
- 인접행렬을 이용한 그래프의 표현 방법
- 최소 비용 신장 트리(MST, Minimum-Cost Spanning Tree)

문제

- 그래프로 표현하기에 적절치 않는 것은?
  ① 행렬
  ② 유기화학 구조식
  ③ 통신 연결망
  ④ 철도 교통망

정답 1

- 정렬(Sort)의 개요
  - ✓오름차순(Ascending), 내림차순(Descending)
- 정렬 방식
  - ✓내부 정렬-주기억장치
    - ✓선택법 : 힙 정렬
    - ✓삽입법 : 삽입 정렬, 쉘 정렬
    - ✓교환법 : 버블 정렬, 선택 정렬, 퀵 정렬
    - ✓병합법(합병법) : 2-Way Merge Sort
    - ✓분배법(분산법) : 기수 정렬(Radix Sort)
  - ✓외부 정렬-보조기억장치
    - ✓밸런스 병합 정렬, 캐스케이드 병합 정렬, 폴리파즈 병합 정렬, 오실레이팅 병합 정렬

- 정렬(Sort) 알고리즘 선택 시 고려 사항
  - ✓ 데이터의 양
  - ✓ 초기 데이터의 배열 상태
  - ✓ 키 값들의 분포 상태
  - ✓ 소요공간 및 작업시간
  - ✓ 사용 컴퓨터 시스템의 특성

- 주기억장치 내에서 정렬하는 방법에 해당하는 것은?
  - ① Cascade Merge Sort
  - ② Heap Sort
  - ③ Balanced Merge Sort
  - ④ Polyphase Merge Sort

정답 2

- 내부 정렬
  - ✓삽입 정렬(Insertion Sort)
    - ✓두 번째 키와 첫 번째 키를 비교
    - ✓세 번째 키를 첫 번째, 두 번째 키와 비교
    - ✓n번째 키를 앞의 (n-1)개의 키와 비교하여 알맞은 순서에 삽입
  - ✓선택 정렬(Selection Sort)
    - ✓n개의 레코드 중에서 최소값을 찾아 첫 번째 레코드 위치에 놓고,
    - ✓나머지 (n-1)개 중에서 다시 최소값을 찾아 두 번째 레코드 위치에 놓는 방식
  - ✓버블 정렬(Bubble Sort)
    - ✓인접한 두 개의 레코드 키 값을 비교

- ✓퀵 정렬(Quick Sort)
  - ✓하나의 파일을 부분적으로 나누어 가면서 정렬하는 방법
  - ✓키를 기준으로 작은 값은 왼쪽에, 큰 값은 오른쪽 서브파일로 분해시키는 방식
- ✓힙정렬
  - ✓전이진트리를 이용
- ✓2-Way 합병 정렬(Merge Sort)
  - ✓이미 정렬되어 있는 두 개의 파일을 한 개의 파일로 합병하는 정렬 방식이다.
- ✓기수 정렬(Radix Sort) = Bucket Sort
  - ✓레코드의 키 값을 분석하여 같은 수 또는 같은 문자끼리 그 순서에 맞는 버킷에 분배하였다가 버킷의 순서대로 레코드를 꺼내어 정렬한다.

문제

- 자료가 아래와 같이 주어졌을 때, 선택 정렬(Selection Sort)을 적용하여 오름차순으로 정렬할 경우 Pass 2를 진행한 후의 정렬된 값으로 옳은 것은?

9, 4, 5, 11, 8

① 4, 5, 9, 8, 11
② 4, 5, 9, 11, 8
③ 4, 5, 8, 11, 9
④ 4, 5, 8, 9, 11

정답 2

정보처리기사

제1과목 데이터베이스

16 자료구조

문혜영교수

목차

1 검색

2 이분검색

3 해싱

4 해싱함수

- 선형 검색(Linear Search)
  - ✓ 순차적으로 검색하는 방식
- 제어 검색(Control Search)
  - ✓ 반드시 순서화된 파일이어야 검색할 수 있다.
  - ✓ 이분 검색(이진 검색, Binary Search)
    - ✓ 비교 회수를 거듭할 때마다 검색 대상이 되는 데이터의 수가 절반으로 줄어든다.
  - ✓ 피보나치검색
  - ✓ 보간검색
  - ✓ 블록검색
  - ✓ 이진트리검색

문제

- 이진 검색(Binary Search) 기법을 적용하기 위한 선행 조건은?
  ① 자료의 개수가 짝수이어야 한다.
  ② 자료가 반드시 정렬되어야 한다.
  ③ 자료의 구성은 비순차적이어야 한다.
  ④ 자료의 구성은 홀수, 짝수 순으로 이루어져야 한다.

정답 2

- 해싱의 개요
  - ✓DAM(직접 접근) 파일을 구성할 때 사용된다.
  - ✓ 해시 테이블(Hash Table, 해시표)
    - ✓ 버킷(Bucket).
    - ✓ 슬롯(Slot
    - ✓ Collision(충돌 현상)
    - ✓ Synonym
    - ✓ Overflow
- 해싱 함수(Hashing Function)
  - ✓제산법
  - ✓제곱법
  - ✓폴딩법
  - ✓기수(Radix) 변환법
  - ✓대수적 코딩법
  - ✓계수 분석법(숫자 분석법)
  - ✓무작위법

문제

- 키 값을 여러 부분으로 분류하여 분류한 부분을 더하거나 XOR하여 주소를 계산하는 해싱 함수의 종류는?
  ① 제산(Divide) 함수
  ② 접지(Folding) 함수
  ③ 중간제곱(Mid-Square) 함수
  ④ 숫자 분석 함수

정답 2

정보처리기사

제1과목 데이터베이스

17 자료구조

문혜영교수

목차

1 인덱스

2 B-트리

3 파일편성

4 색인순차파일

- 인덱스의 개념
  - ✓데이터가 저장된 물리적 구조와 밀접한 관계가 있다.
  - ✓인덱스를 통해서 파일의 레코드에 대한 액세스를 빠르게 수행할 수 있다.
- B-트리

- B-트리에 대한 설명 중 옳지 않는 것은?
  ① 한 노드 안에 있는 키 값들은 오름차순을 유지한다.
  ② 탐색, 추가, 삭제는 루트로부터 시작한다.
  ③ 루트와 리프를 제외한 모든 노드는 적어도 2/3 이상 채워져 있다.
  ④ B-트리는 인덱스 파일에서 인덱스를 구성하는 방법 중의 하나이다.

정답 3

- 순차 파일(Sequential File) = 순서 파일
  - ✓레코드들이 순차적으로 처리되므로 일괄 처리에 적합한 구조이다.
  - ✓레코드 사이에 빈 공간이 존재하지 않으므로 기억공간을 효율적으로 사용할 수 있다.

✓순차 파일의 단점

- ✓새로운 레코드를 삽입하거나 삭제하는 경우 파일 재구성을 위해 전체를 복사해야 하므로 시간이 많이 소요된다.
- ✓데이터 검색 시 처음부터 순차적으로 검색하기 때문에 검색 효율이 낮다.

- 색인 순차 파일(Indexed Sequential File)
  - ✓순차 처리와 랜덤 처리가 모두 가능하다.
- 색인 순차 파일의 구성
  - ✓기본 구역(Prime Area)
  - ✓색인 구역(Index Area)
    - ✓트랙 색인 구역(Track Index Area)
    - ✓실린더 색인 구역(Cylinder Index Area)
    - ✓마스터 색인 구역(Master Index Area)
  - ✓오버플로 구역(Overflow Area)
    - ✓실린더 오버플로 구역(Cylinder Overflow Area)
    - ✓독립 오버플로 구역(Independent Overflow Area)

✓색인 순차 파일의 장점

- ✓순차 처리와 랜덤 처리가 모두 가능하다.
- ✓레코드의 삽입, 삭제, 갱신이 용이하다.
- ✓레코드를 추가 및 삽입하는 경우, 파일 전체를 복사할 필요가 없다.

✓색인 순차 파일의 단점

- ✓색인 구역과 오버플로 구역을 구성하기 위한 추가 기억공간이 필요하다.
- ✓파일 사용 중 오버플로 레코드가 많아지면 파일을 재편성해야 한다.
- ✓색인을 이용한 액세스를 하기 때문에 액세스 시간이 랜덤 편성 파일보다 느리다.

❖정적 인덱스

- ❖ 데이터 파일에 레코드가 삽입되거나 삭제되어도 인덱스의 구조가 변경하지 않고 내용만 변하는 구조

❖동적 인덱스

- ❖ 인덱스 파일이나 데이터 파일을 블록으로 구성하고 각 블록에는 추가로 삽입될 레코드를 감안하여 빈 공간을 미리 예비해 두는 인덱스 방법이다.

- VSAM 파일
  - ✓기본 구역과 오버플로 구역을 구분하지 않는다.
  - ✓레코드를 삭제하면 그 공간을 재사용할 수 있다.
  - ✓제어 구간에 가변 길이 레코드를 쉽게 수용할 수 있다.

- 직접 파일(Direct File)
  - ✓해시 함수(Hash Function)를 이용하여 이 키에 대한 보조기억장치의 물리적 상대 레코드 주소를 계산한 후 해당하는 주소에 레코드를 저장

- 파일 조직 기법 중 순차 파일에 대한 설명으로 옳지 않은 것은?
  - ① 레코드 사이에 빈 공간이 존재하지 않으므로 기억장치의 효율적 이용이 가능하다.
  - ② 레코드들이 순차적으로 처리되므로 대화식 처리보다 일괄 처리에 적합한 구조이다.
  - ③ 필요한 레코드를 삽입, 삭제하는 경우 파일을 재구성해야 하므로 파일 전체를 복사해야 한다.
  - ④ 데이터 검색 시 검색 효율이 높다.

정답 4