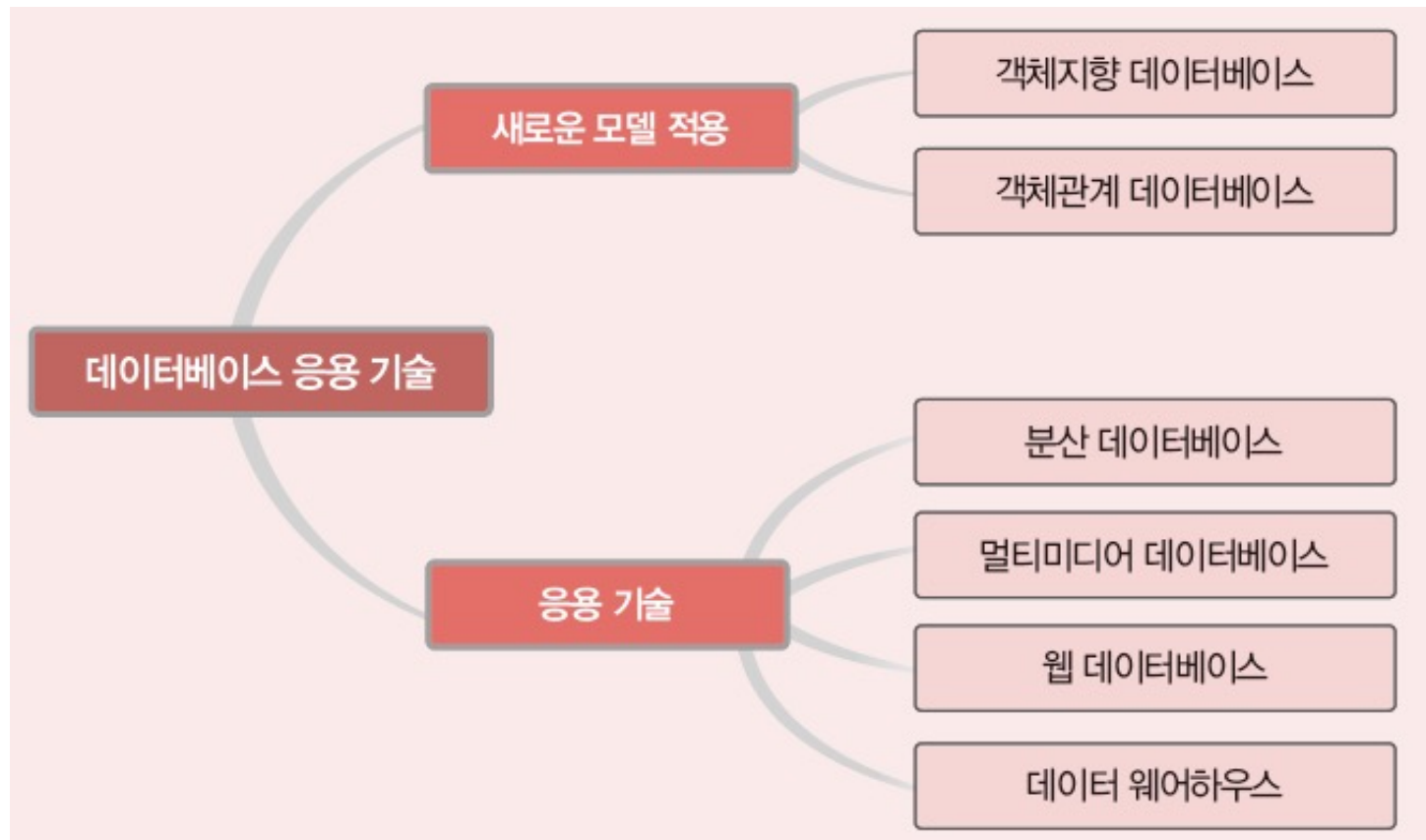


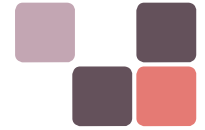


# 12장. 데이터베이스 응용 기술

- 객체지향 데이터베이스
- 객체관계 데이터베이스
- 분산 데이터베이스 시스템
- 멀티미디어 데이터베이스 시스템
- 기타 데이터베이스 응용 기술

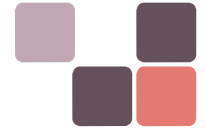


- ❖ 새로 제안된 데이터 모델의 특징을 관계 데이터 모델과 비교하여 알아본다.
- ❖ 데이터베이스 분야의 최신 응용 기술을 살펴본다.



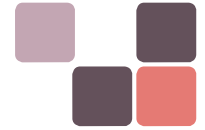
## ❖ 객체지향 데이터 모델

- 객체지향 개념에 기반을 둔 데이터 모델
  - 객체와 객체 식별자, 속성과 메서드, 클래스, 클래스 계층 및 상속, 복합 객체 등을 지원
- 다양한 응용 분야의 데이터 모델링을 위한 새로운 요구 사항을 지원
- 의미적으로 관계가 있는 데이터베이스 구조를 표현하기 위한 강력한 설계 기능 제공
- 특수한 몇몇 분야에서 사용됨



## ❖ 객체지향 데이터 모델 – 객체(object)

- 현실 세계에 존재하는 개체를 추상적으로 표현한 것
- 객체지향 데이터 모델을 구성하는 기본 요소
- 각 객체는 시스템 전체에서 유일하게 식별될 수 있는 객체 식별자(OID; Object Identifier)를 가짐
- 객체 간의 관계는 객체 식별자를 사용해 참조할 수 있음
- 객체는 상태를 의미하는 속성과 상태를 조작하는 메서드로 구성



## ❖ 객체지향 데이터 모델 – 속성(attribute)

- 관계 데이터 모델의 속성과 같은 의미
- 관계 데이터 모델의 속성은 기본으로 제공되는 데이터 타입을 도메인으로 하는 단일 값만 가질 수 있지만, 객체지향 데이터 모델의 속성은 값을 여러 개 가질 수 있음
- 사용자가 정의한 클래스뿐만 아니라 해당 클래스의 하위 클래스도 도메인으로 정의 가능

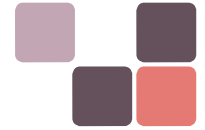


## ❖ 객체지향 데이터 모델 – 메서드(method)

- 객체에 수행할 수 있는 연산
- 객체의 속성 값을 검색하거나 추가, 삭제, 수정하는데 주로 사용
- 프로그래밍 언어의 함수와 유사한 개념

## ❖ 객체지향 데이터 모델 – 메시지(message)

- 객체에 접근하기 위한 공용 인터페이스 역할 담당
- 특정 객체의 속성과 메서드에 접근하려면 메시지를 사용해야 함
  - 예) 객체의 속성 값을 수정하려면 그에 해당하는 메서드를 실행시키는 메시지를 해당 객체에 보내야 함



## ❖ 객체지향 데이터 모델 – 클래스(class)

- 속성과 메서드를 공유하는 유사한 성질의 객체들을 하나로 그룹화한 것
- 객체는 클래스의 구성원
  - 객체 = 클래스 인스턴스(class instance) 또는 객체 인스턴스(object instance)
- 클래스 내부에 해당 클래스의 객체를 위한 데이터 구조와 메서드 구현의 세부 사항을 기술

# 01 객체지향 데이터베이스

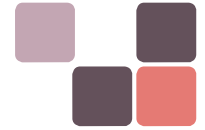


## ❖ 객체지향 데이터 모델 – 클래스와 객체



그림 12-1 사원 클래스와 객체 인스턴스





## ❖ 객체지향 데이터 모델 – 클래스 계층(class hierarchy)

- 클래스를 세분화(specialization)하면 클래스 간의 계층 관계가 발생하여 결과적으로 클래스 계층이 형성됨
- 상위클래스와 하위클래스
  - 상위클래스(superclass) : 클래스 계층에서 상위에 위치하는 클래스
  - 하위클래스(subclass) : 클래스 계층에서 하위에 위치하는 클래스
  - 상위클래스와 하위클래스는 일반적으로 IS-A 관계가 성립

# 01 객체지향 데이터베이스



## ❖ 객체지향 데이터 모델 - 클래스 계층

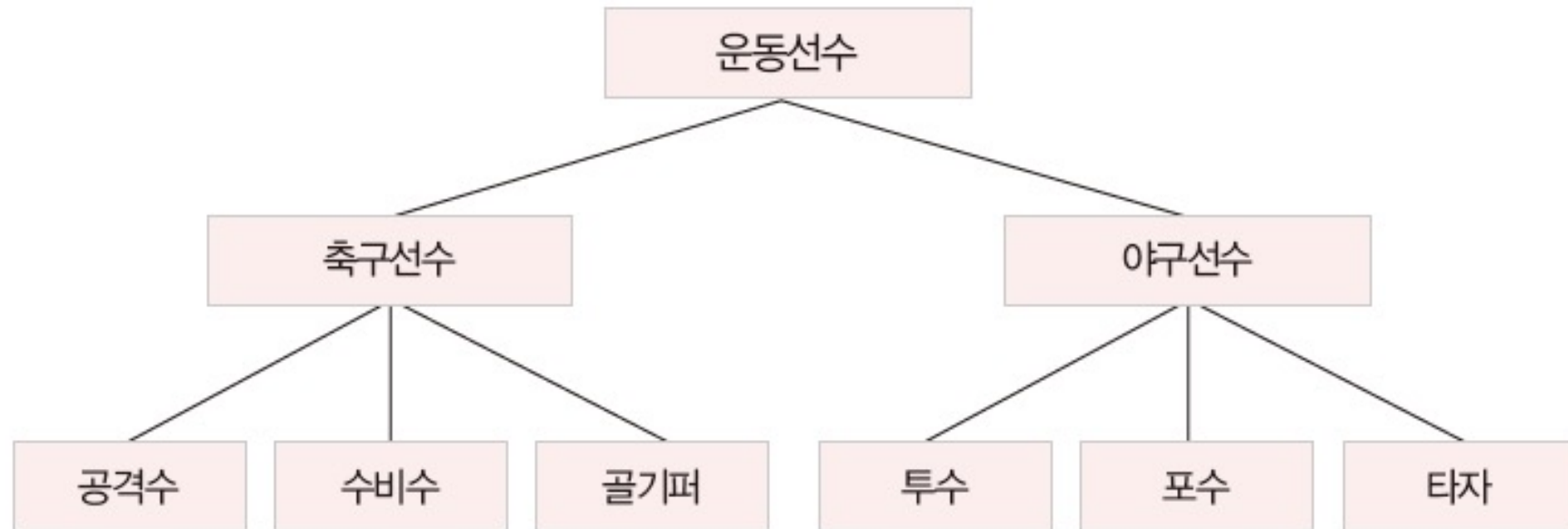


그림 12-2 클래스 계층의 예



## ❖ 객체지향 데이터 모델 – 상속(inheritance)

- 상위클래스의 속성과 메서드를 자신의 모든 하위클래스에 물려주는 개념
- 단일 상속 vs. 다중 상속
  - 단일 상속 : 하위클래스가 단 하나의 상위클래스로부터 상속받는 것
  - 다중 상속 : 하위 클래스가 여러 개의 상위클래스로부터 상속받는 것



그림 12-3 다중 상속의 예



## ❖ 객체지향 데이터 모델 – 복합 객체(composite object)

- 시스템에서 기본적으로 제공하지 않는 사용자 정의 클래스(user-defined class)를 도메인으로 하는 속성을 가진 객체
- 사용자 정의 클래스가 도메인인 속성은 객체 식별자를 값으로 가지므로 속성 값으로 다른 객체를 참조할 수 있음
- 일반적으로 Is-Part-Of 관계가 있는 객체를 표현하는 데 사용

# 01 객체지향 데이터베이스



## ❖ 객체지향 데이터 모델 – 복합 객체 클래스

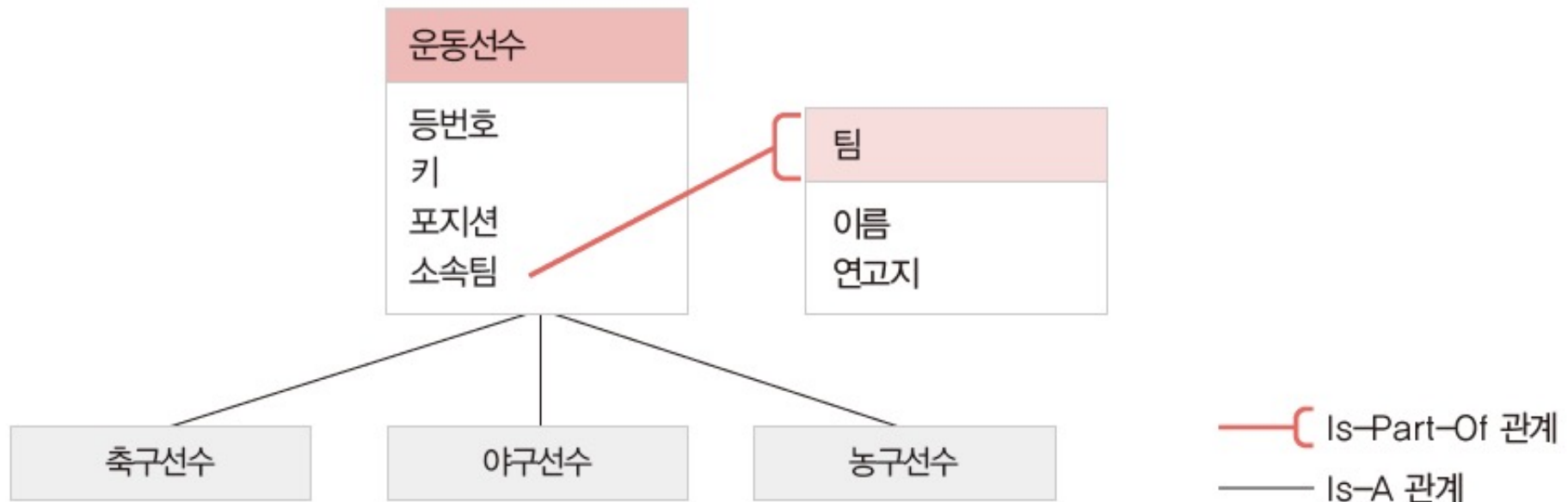


그림 12-4 복합 객체 클래스의 예



## ❖ 객체지향 질의 모델

- 질의 대상은 클래스, 질의 결과는 클래스에 속하는 객체 집합
- 클래스, 속성, 메서드, 객체 등 객체지향 개념에 기반을 두고 질의를 표현
- 단일 오퍼랜드(single operand) 질의
  - 하나의 클래스 또는 하나의 클래스와 그 클래스의 하위클래스 전체를 대상으로 하는 질의
- 다중 오퍼랜드(multiple operand) 질의
  - 여러 클래스를 대상으로 하는 질의
- 객체지향 개념을 완벽히 표현하면서 쉽게 사용할 수 있는 표준 질의어가 없음



## ❖ 객체지향 질의 모델

- 예) [그림 12-4]에서 정의한 운동선수 클래스에 대해 키가 180이상이고, 소속팀의 연고지가 서울인 모든 운동선수 객체를 검색하는 질의문

예) SELECT P

FROM P : 운동선수

WHERE P.키 >= 180 AND P.소속팀.연고지 = '서울';

키가 180 이상이고, 소속팀의 연고지가  
서울인 모든 운동선수 검색



### ❖ 객체관계 데이터 모델

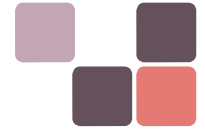
- 객체지향 개념과 관계 데이터 모델의 개념을 통합한 것
  - 릴레이션, 객체, 메서드, 클래스, 상속, 캡슐화, 복합 객체 등을 모두 지원
- SQL을 표준 질의어로 채택하여 계속 발전시킴
  - 1999년 발표된 SQL3부터는 객체지향 개념을 지원하기 때문에 객체관계 데이터베이스에 적용 가능
  - 기본 질의 기능과 함께 사용자 정의 타입, 객체, 객체 식별자, 메서드 등과 같은 객체지향 특성도 가지고 있음





### ❖ 객체지향 데이터베이스 vs. 객체관계 데이터베이스

- 기능적 유사성은 많지만, 기본 철학과 구현 방식이 달라 데이터베이스의 설계나 조작 방법 등에 차이가 있음
- 객체지향 데이터베이스
  - 객체지향 프로그래밍 개념에 기반을 두고 데이터베이스의 기능을 추가하는데 목적을 두고 있음
- 객체관계 데이터베이스
  - 관계 데이터베이스에 기반을 두고 사용자가 다양한 데이터 타입을 추가할 수 있도록 하는데 목적을 두고 있음



### ❖ 중앙 집중식 데이터베이스 시스템

- 데이터베이스 시스템을 물리적으로 한 장소에 설치하여 운영하는 것

### ❖ 분산 데이터베이스 시스템(distributed database system)

- 물리적으로 분산된 데이터베이스 시스템을 네트워크로 연결해 사용자가 논리적으로는 하나의 중앙 집중식 데이터베이스 시스템처럼 사용할 수 있도록 한 것

# 03 분산 데이터베이스 시스템



## ❖ 분산 데이터베이스 시스템의 구성

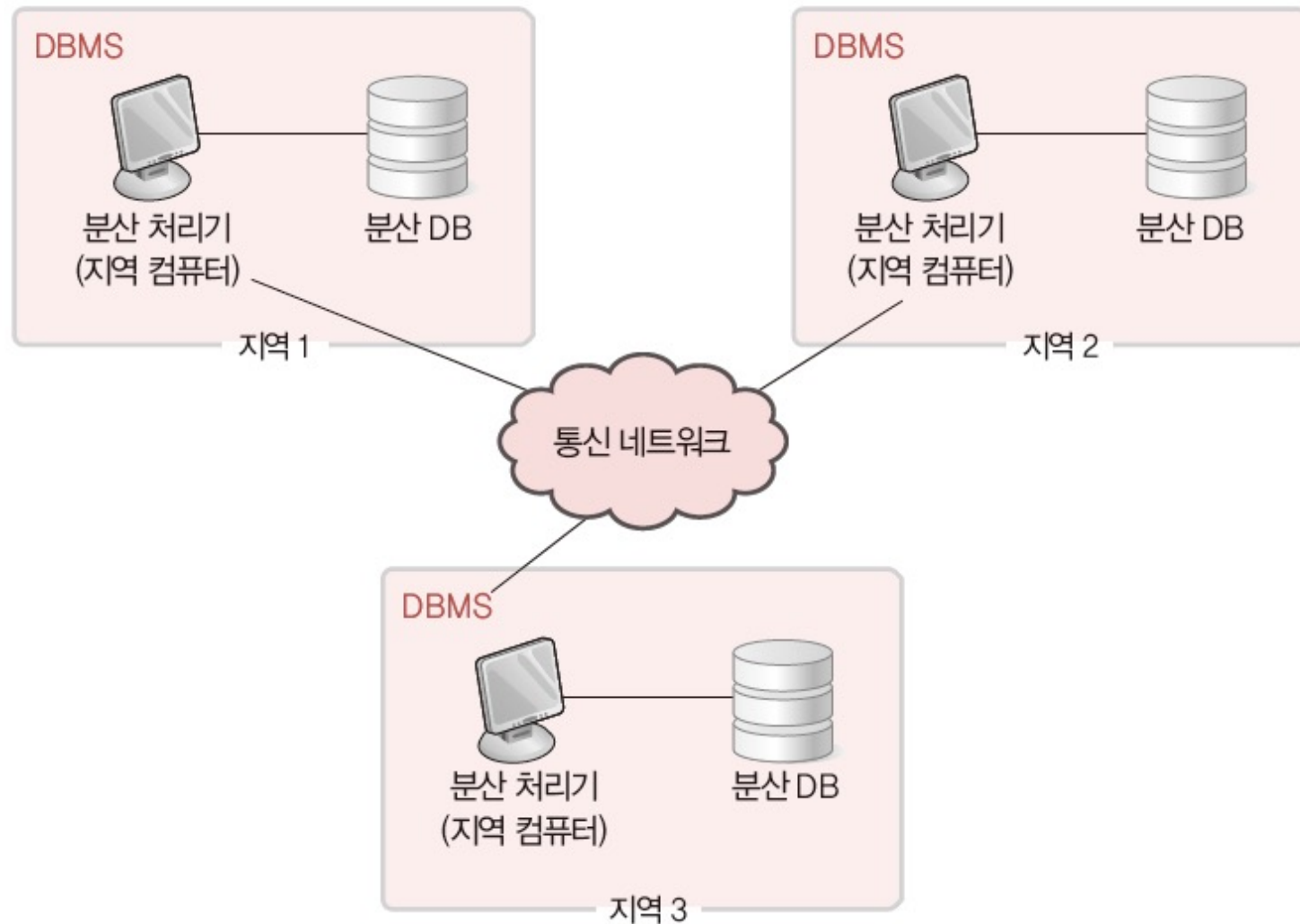


그림 12-5 분산 데이터베이스 시스템의 구성



## ❖ 분산 데이터베이스 시스템의 주요 구성 요소

- 분산 처리기(distributed processor)
  - 지역별로 필요한 데이터를 처리할 수 있는 지역 컴퓨터(local computer)
  - 각 지역의 데이터베이스를 자체적으로 관리하는 DBMS를 별도로 가지고 있음
- 분산 데이터베이스(distributed database)
  - 물리적으로 분산된 지역 데이터베이스(local database)
  - 해당 지역에서 가장 많이 사용하는 데이터를 저장
- 통신 네트워크
  - 분산 처리기는 통신 네트워크를 통해 자원을 공유
  - 통신 네트워크에 있는 모든 분산 처리기는 특정 통신 규약에 따라 데이터를 전송하고 수신함

## 03 분산 데이터베이스 시스템



### ❖ 데이터베이스 분산 저장 방법

- 1) 데이터가 중복되지 않게 분할하여 지역의 분산 데이터베이스에 저장
- 2) 지역의 분산 데이터베이스에 데이터를 중복해서 저장(주로 이용)

표 12-1 데이터 중복의 장단점

구분	내용
장점	<ul style="list-style-type: none"><li>• 한 지역에서 문제가 발생해도 동일한 데이터가 저장되어 있는 다른 지역에서 작업을 계속 수행할 수 있으므로 신뢰성과 가용성이 높아진다.</li><li>• 동일한 데이터가 저장된 여러 지역에서 병렬 처리를 수행할 수 있어 데이터 처리 성능이 향상된다.</li><li>• 데이터 처리 요청이 한 지역에 집중되지 않고 여러 지역에 분산되므로 처리 부담을 줄일 수 있다.</li></ul>
단점	<ul style="list-style-type: none"><li>• 동일한 데이터를 중복 저장하므로 저장 공간을 많이 사용한다.</li><li>• 데이터를 변경하려면 중복 저장된 데이터를 모두 함께 변경해야 하므로 비용이 증가하고, 변경 도중에 문제가 생겨 데이터 불일치가 발생할 수도 있다.</li></ul>



### ❖ 단편화(fragmentation)

#### ■ 의미

- 하나의 릴레이션을 더 작은 조각(단편)으로 나누고 각 조각을 별개의 릴레이션으로 처리하는 것

#### ■ 장점

- 각 조각이 전체 릴레이션의 일부가 되기 때문에 저장 공간을 적게 사용하고, 관리할 데이터도 줄어듦
- 데이터 중복의 장점은 그대로 취하면서 데이터 중복의 단점을 보완할 수 있음

# 03 분산 데이터베이스 시스템



## ❖ 단편화 방법 - 수평적 단편화

- 릴레이션을 수평적으로 단편화하는 것, 릴레이션을 튜플(행) 단위로 나눔

제품 릴레이션

제품번호	제품명	재고량	단가	제조업체
p01	그냥만두	5000	4500	대한식품
p02	매운짬면	2500	5500	민국푸드
p03	쿵떡파이	3600	2600	한빛제과

제품번호	제품명	재고량	단가	제조업체
p01	그냥만두	5000	4500	대한식품
p02	매운짬면	2500	5500	민국푸드

지역 1

제품번호	제품명	재고량	단가	제조업체
p03	쿵떡파이	3600	2600	한빛제과

지역 2

그림 12-7 수평적 단편화의 예

# 03 분산 데이터베이스 시스템



## ❖ 단편화 방법 - 수직적 단편화

- 릴레이션을 수직적으로 단편화하는 것, 릴레이션을 속성(열) 단위로 나눔

제품 릴레이션

제품번호	제품명	재고량	단가	제조업체
p01	그냥만두	5000	4500	대한식품
p02	매운짬면	2500	5500	민국푸드
p03	콩떡파이	3600	2600	한빛제과



그림 12-8 수직적 단편화의 예





### ❖ 단편화 방법 - 혼합 단편화

- 수평적 단편화와 수직적 단편화를 모두 사용하여 릴레이션을 나눔

### ❖ 단편화 수행 조건

완전성	전체 릴레이션의 모든 데이터는 어느 한 조각에는 꼭 속해야 한다.
회복성	단편화된 조각들로부터 원래의 전체 릴레이션을 회복할 수 있어야 한다.
분리성	전체 릴레이션의 모든 조각을 서로 중복되지 않게 분리해야 한다.

그림 12-6 단편화 수행 조건



### ❖ 분산 데이터베이스 시스템의 주요 목표

- 분산 데이터 독립성(distributed data independency)
  - 데이터베이스가 분산되어 있음을 사용자가 인식하지 못하게 하는 것
  - 분산 투명성(distribution transparency)이 보장되어 함
    - 위치 투명성
    - 중복 투명성
    - 단편화 투명성
    - 병행 투명성
    - 장애 투명성



### ❖ 분산 투명성 - 위치 투명성(location transparency)

- 사용자가 접근하려는 데이터의 실제 저장 위치를 알 필요 없이 논리적 이름만으로 데이터에 접근할 수 있음
- 시스템 카탈로그에서 데이터의 모든 위치 정보를 관리하다가 데이터 접근 요구가 발생하면 이 위치 정보를 통해 데이터를 제공함
- 다른 지역에 있는 데이터에 대한 접근 요청을 처리하는 방법
  - 다른 지역에 있는 데이터를 가져와 처리함
  - 데이터 접근 요청을 한 트랜잭션을 데이터가 있는 지역으로 보내 처리한 후, 결과 데이터만 가져옴
  - 두 가지 방법을 모두 사용해서 처리함



### ❖ 분산 투명성 - 중복 투명성(replication transparency)

- 동일한 데이터가 여러 지역에 중복 저장되더라도 사용자가 중복을 인식하지 못하고, 하나의 데이터베이스 시스템에 데이터가 저장된 것처럼 사용하는 것
- 데이터를 중복하여 저장하는 방법
  - 완전 중복
    - 동일한 데이터를 둘 이상의 지역에 있는 분산 데이터베이스에 저장하는 것
  - 부분 중복
    - 일부 데이터만 중복하여 저장하는 것



### ❖ 분산 투명성 - 단편화 투명성(fragmentation transparency)

- 단편화된 데이터를 여러 지역에 나누어 저장하지만 사용자는 데이터가 단편화된 것을 인식할 수 없도록 하는 것

### ❖ 분산 투명성 - 병행 투명성(concurrency transparency)

- 분산 데이터베이스와 관련된 트랜잭션들이 동시에 수행되더라도 결과는 항상 일관성을 유지하는 것

### ❖ 분산 투명성 - 장애 투명성(failure transparency)

- 특정 지역 시스템에 문제가 발생하더라도 전체 시스템이 작업을 계속 수행할 수 있는 것

# 03 분산 데이터베이스 시스템



## ❖ 분산 데이터베이스의 기본 구조

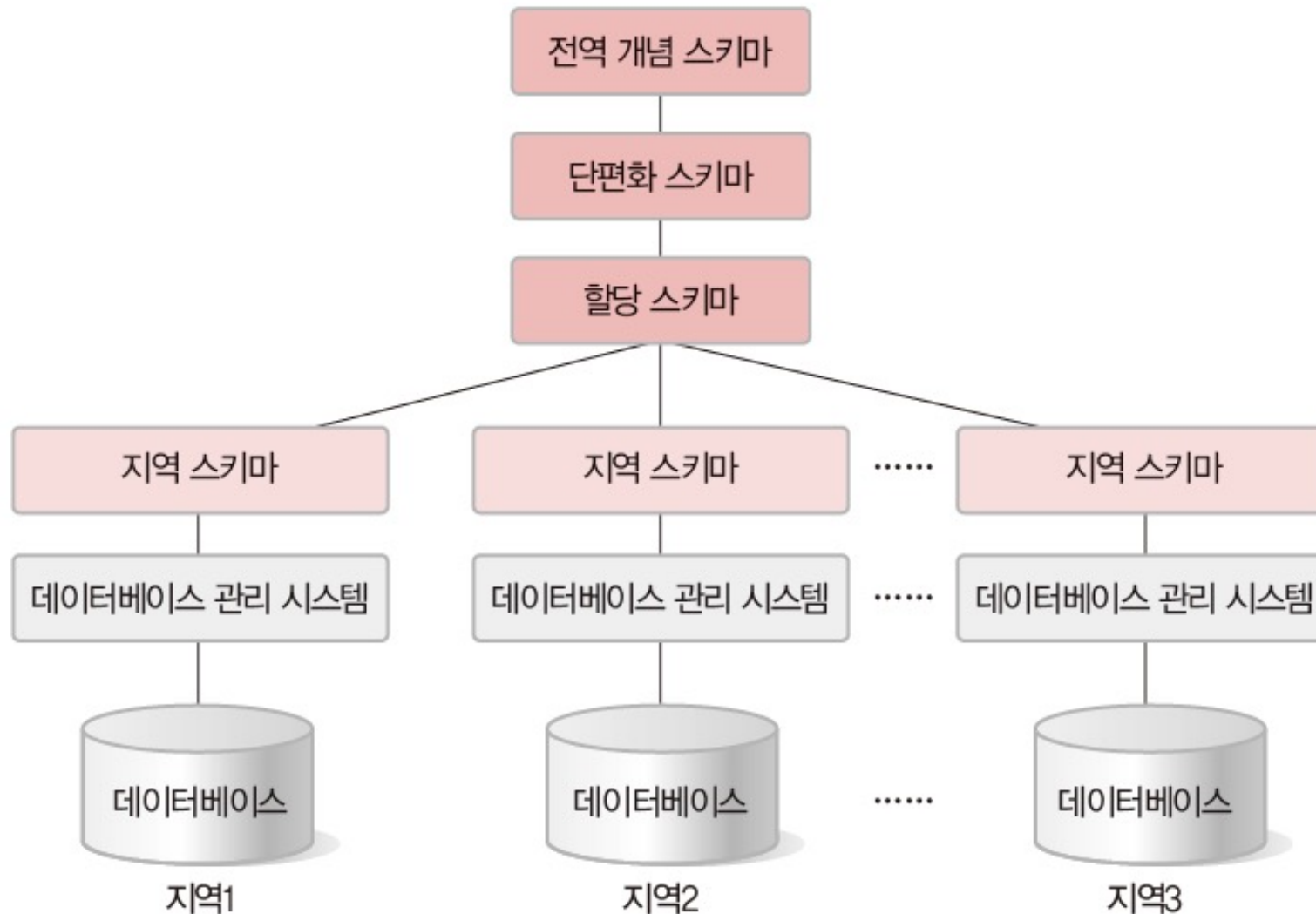


그림 12-9 분산 데이터베이스의 일반 구조



## ❖ 분산 데이터베이스의 기본 구조

- 전역 개념 스키마(global conceptual schema)
  - 분산 데이터베이스에 저장할 모든 데이터 구조와 제약조건을 정의
    - 데이터베이스 안에 존재하는 모든 릴레이션 스키마의 집합
  - 데이터 분산은 고려하지 않고 정의함
- 단편화 스키마(fragmentation schema)
  - 전역 개념 스키마를 논리적으로 분할하는 방법인 단편화를 정의
  - 전역 개념 스키마와 각 조각 스키마의 대응 관계도 정의
- 할당 스키마(allocation schema)
  - 각 조각 스키마의 인스턴스를 물리적으로 저장해야 되는 지역을 정의
- 지역 스키마(local schema)
  - 지역별로 저장하고 있는 데이터 구조와 제약조건을 정의



### ❖ 분산 데이터베이스의 질의 처리

- 질의 처리 전략의 선택 기준
  - 디스크 접근 횟수
  - 네트워크에서 데이터를 전송하는 비용
  - 하나의 질의문을 분해하여 여러 지역에서 병렬 처리함으로써 얻는 성능상 이점



## 03 분산 데이터베이스 시스템



### ❖ 분산 데이터베이스 시스템의 장점

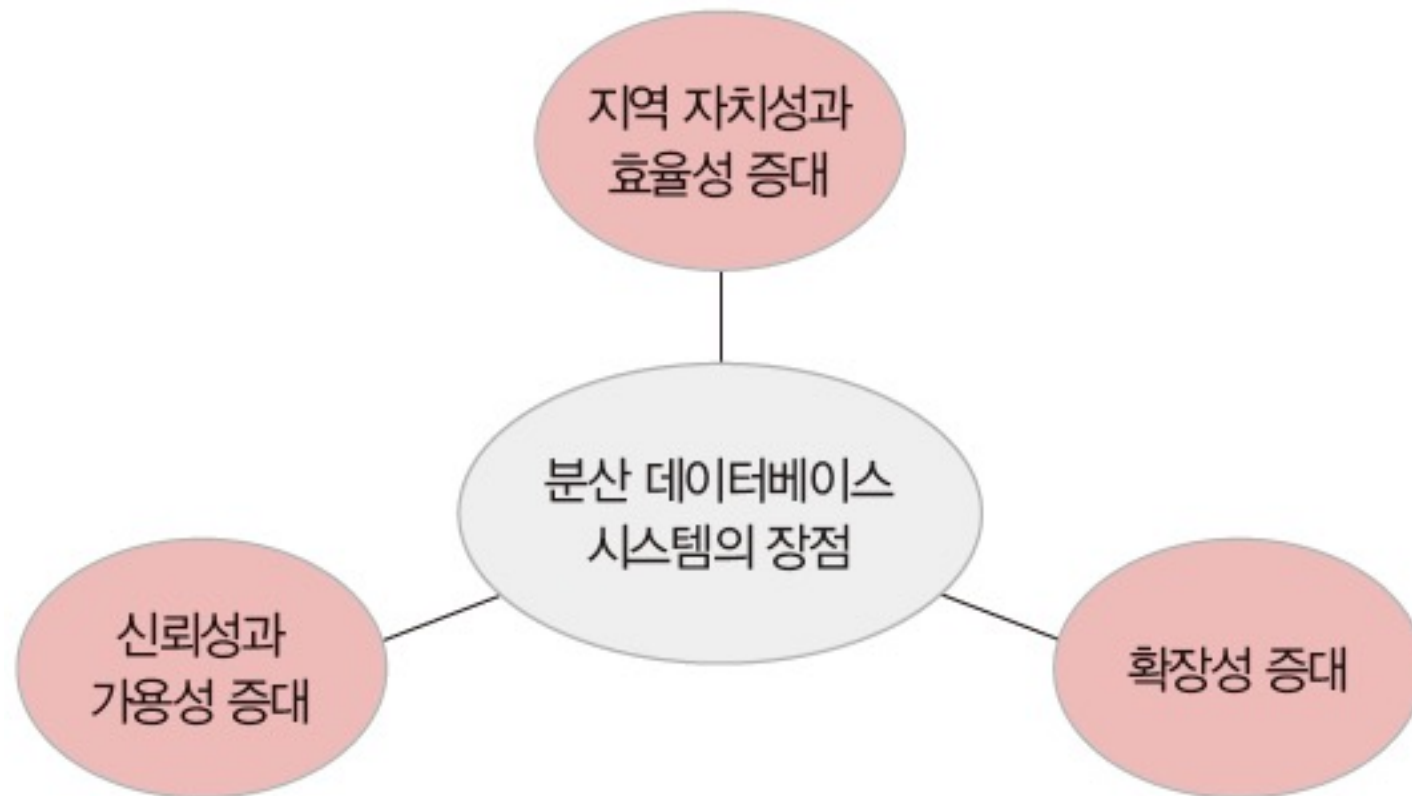
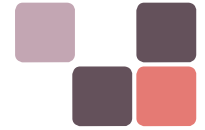


그림 12-10 분산 데이터베이스 시스템의 장점



### ❖ 분산 데이터베이스 시스템의 장점

- 신뢰성과 가용성 증대
  - 장애 발생 시 다른 지역의 데이터베이스를 이용해 작업을 계속 수행할 수 있음
- 지역 자치성과 효율성 증대
  - 데이터베이스를 지역별로 독립적으로 관리
  - 데이터 요청에 대한 응답 시간을 줄이고 통신 비용도 절약됨
- 확장성 증대
  - 처리할 데이터 양이 증가하면 새로운 지역에 데이터베이스를 설치하여 운영하면 됨



### ❖ 분산 데이터베이스 시스템의 단점

- 중앙 집중식 시스템에 비해 설계 및 구축 비용이 많이 발생함
- 여러 지역에 대한 관리가 복잡하고 비용도 많이 발생함
- 중앙 집중식 시스템에 비해 추가적인 통신 비용이나 처리 비용이 발생함



## ❖ 미디어(media)

- 데이터의 각 타입
  - 예) 문자 타입, 숫자 타입, 그래픽 타입, 이미지 타입, 비디오 타입, 오디오 타입 등

## ❖ 멀티미디어 데이터(mutimedia data)

- 여러 미디어의 조합으로 이루어진 데이터

표 12-2 멀티미디어 데이터의 유형

유형	의미
텍스트	문자로 구성된 데이터
그래픽	수학 공식을 기반으로 제작된 벡터 이미지 데이터
이미지	정적 이미지나 사진과 같이 픽셀 단위로 표현되는 비트맵 이미지
비디오	동영상, 애니메이션
오디오	음성, 소리, 음악



### ❖ 멀티미디어 데이터의 특성

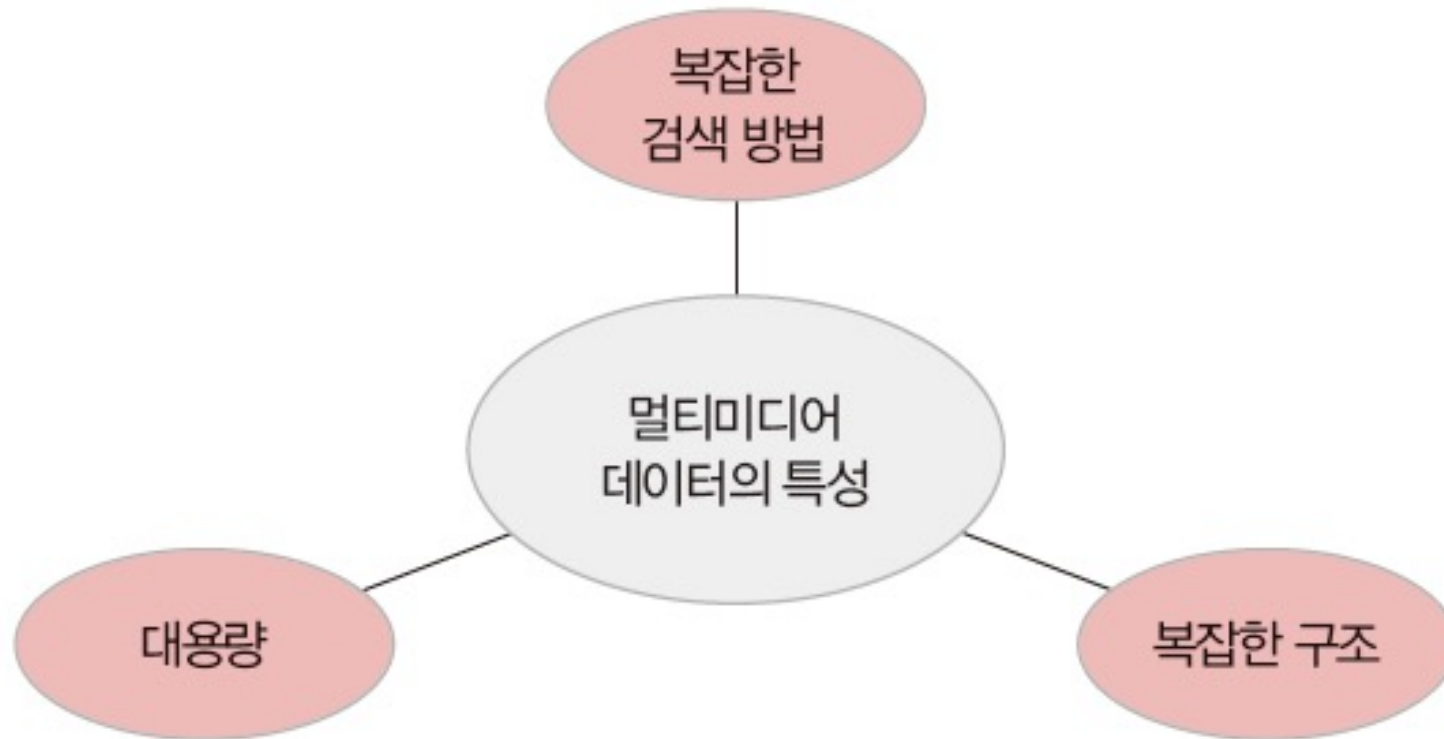
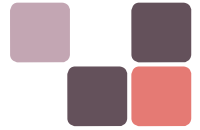


그림 12-11 멀티미디어 데이터의 특성



### ❖ 멀티미디어 데이터의 특성 – 대용량 데이터

- 일반적으로 크기가 수 킬로바이트에서 수십 메가바이트 이상
- 압축해서 저장해야 하므로 일반 데이터와는 다른 구조로 별도의 저장 공간을 구성해 관리해야 함



## ❖ 멀티미디어 데이터의 특성 – 검색 방법이 복잡한 데이터

### ■ 멀티미디어 데이터의 검색 방법

- 설명 기반 검색(description-based retrieval)
  - 멀티미디어 데이터의 특성을 나타내는 키워드나 자세한 설명을 멀티미디어 데이터와 함께 저장해두었다가 검색에 이용
- 내용 기반 검색(content-based retrieval)
  - 멀티미디어 데이터의 실제 내용을 검색에 이용
  - 특정 객체를 포함한 멀티미디어 데이터 검색
  - 예) 손흥민(객체)이 포함된 비디오를 검색



## ❖ 멀티미디어 데이터의 특성 – 구조가 복잡한 데이터

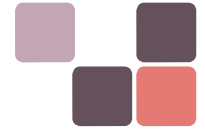
- 멀티미디어 데이터는 원시 데이터, 등록 데이터, 서술 데이터 등으로 구성
  - 원시 데이터(raw data)
    - 텍스트, 그래픽, 이미지, 비디오, 오디오 등 기본 타입의 데이터
  - 등록 데이터(registration data)
    - 멀티미디어 데이터의 특성과 필요한 정보를 별도로 추출한 데이터
    - 예) 이미지의 해상도, 픽셀 수, 색상, 크기, 포맷 등의 정보
  - 서술 데이터(description data)
    - 멀티미디어 데이터를 검색할 때 사용되는 데이터
    - 멀티미디어 데이터에 지정된 키워드나 자세한 설명 등
- 공간 및 시간적으로 복잡한 관계성을 표현하고 관리할 수 있는 기술 필요





## ❖ 멀티미디어 데이터베이스의 발전 과정

- 관계 데이터베이스에서의 멀티미디어 데이터 처리
  - 관계 데이터베이스에 멀티미디어 데이터를 위한 새로운 데이터 타입을 추가하여 멀티미디어 데이터를 저장하고 처리하는 방법
    - 이진 대형 객체(BLOB; Binary Large Object) 데이터 타입
  - 관계 데이터베이스가 제공하는 이론과 다양한 기법을 그대로 이용할 수 있음
  - 멀티미디어 데이터가 가진 시공간적인 특성 표현, 다양한 미디어 데이터들의 통합 모델링 기능, 사용자의 요구에 맞는 다양한 연산 표현 및 조작 기능 등을 거의 제공 못함
  - 예) GENESIS, STAIRS 등



## ❖ 멀티미디어 데이터베이스의 발전 과정

- 객체지향 데이터베이스에서의 멀티미디어 데이터 처리
  - 멀티미디어 데이터를 객체와 클래스로 표현하고 데이터 추상화, 캡슐화, 상속 등의 개념을 지원
  - 멀티미디어 데이터의 복잡하고 다양한 모델링 요구 사항을 완벽히 만족시키지 못함
  - 예) ORION, MULTOS, MINOS 등



## ❖ 멀티미디어 데이터베이스 관리 시스템의 구성

- 멀티미디어 데이터베이스 시스템의 요구 사항
  - 데이터베이스 시스템의 기본 기능을 제공하면서 멀티미디어 데이터 특성에 따른 새로운 사항도 고려해야 함
  - 대용량이나 시공간적 연속성과 같은 멀티미디어 데이터만의 특성을 지원하려면 데이터베이스 관리 시스템의 역할이 중요
- 멀티미디어 데이터 관리 기능을 제공하는 데이터베이스 관리 시스템 예
  - UniSQL, 오라클, 인포믹스, O2, DB2 UDB 등



## ❖ 멀티미디어 데이터베이스 관리 시스템의 구성

- 파일 시스템을 이용하는 방식
  - 초기에 많이 사용하던 방식
  - 응용 프로그램에 필요한 멀티미디어 데이터를 파일로 저장하고 관리
  - 프로그래밍 언어로 데이터를 처리하는 코드를 직접 작성하여 응용 프로그램에 포함시킴
  - 응용 프로그램 개발이 어렵고, 복잡한 멀티미디어 데이터를 파일의 단순한 저장 구조에 저장하기 어려움
  - 데이터의 동시 공유, 회복, 보안 등 데이터베이스 관리 시스템의 고급 기능을 제공하기 어려움



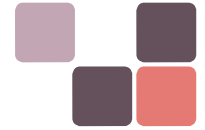
## ❖ 멀티미디어 데이터베이스 관리 시스템의 구성

- 관계 데이터베이스 관리 시스템을 이용하는 방식
  - 텍스트 같은 일반 데이터는 관계 데이터베이스에 저장하고, 이미지나 비디오 같은 데이터는 파일에 저장
  - 지리 정보 시스템(GIS; Geographical Information System) 등에서 많이 사용
  - 파일에 저장된 데이터에 대한 처리 요청을 프로그래밍 언어로 작성하고, 관계 데이터베이스에 저장된 데이터에 대한 처리 요청은 SQL로 작성
  - 파일에 저장된 멀티미디어 데이터에 데이터베이스 관리 시스템의 고급 기능을 제공할 수 없음



## ❖ 멀티미디어 데이터베이스 관리 시스템의 구성

- 확장된 관계 데이터베이스 관리 시스템을 이용하는 방식
  - 텍스트뿐만 아니라 멀티미디어 데이터를 모두 저장할 수 있도록 기존의 관계 데이터베이스 관리 시스템을 확장
    - 예) 대용량 멀티미디어 데이터를 위한 이진 대형 객체(BLOB) 데이터 타입을 추가
  - 멀티미디어 데이터에 데이터베이스 관리 시스템의 고급 기능을 제공할 수 있지만, 완벽히 지원하기 어렵고 멀티미디어 데이터의 특성을 반영한 처리 요청을 SQL로 표현하기 쉽지 않음



### ❖ 멀티미디어 데이터베이스 관리 시스템의 구성

- 객체지향 데이터베이스 관리 시스템을 이용하는 방식
  - 객체지향 개념을 지원하는 데이터베이스 관리 시스템을 이용해 멀티미디어 데이터를 처리하는 데 필요한 다양한 기능 제공
  - 기존 관계 데이터베이스 관리 시스템에서 제공하는 동시성 제어, 질의 최적화, 회복 기능 등의 고급 기능을 제공하지 못함



## ❖ 멀티미디어 데이터의 질의

### ■ 특징

- 데이터 자체에 대한 질의보다는 데이터에 포함된 특정 객체, 데이터에 대한 설명이나 키워드를 이용한 질의를 주로 사용
- 미디어에 따라 다양한 유형의 질의가 존재

### ■ 질의 유형

- 텍스트 질의
  - 사용자가 제시한 키워드를 포함하는 문서를 검색하는 질의
  - 예) ‘한빛’과 ‘데이터베이스’ 키워드를 포함하는 문서를 모두 검색하는 질의
- 비디오 질의
  - 장면을 대상으로 하는 검색 질의
  - 예) ‘미녀’와 ‘야수’가 식사하는 장면을 검색하는 질의





## ❖ 멀티미디어 데이터의 질의

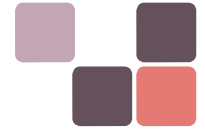
### ■ 질의 유형

#### • 이미지 질의

- 사용자가 제시한 키워드와 관련 있는 이미지를 검색하는 내용 검색이나 사용자가 제시한 이미지와 유사한 이미지를 검색하는 유사도 검색 질의
- 예) ‘개’를 포함하는 이미지를 검색하는 질의
- 예) 제시한 이미지와 유사한 이미지를 검색하는 질의

#### • 공간 질의

- 주어진 범위 조건에 맞는 특정 위치를 검색하는 질의
- 예) ‘한빛아카데미’를 기준으로 5km 이내에 있는 식당을 검색하는 질의
- 예) ‘한빛아카데미’에서 가장 가까운 식당을 검색하는 질의



## ❖ 멀티미디어 데이터의 질의 처리 기법

- 매칭(matching) 기법
  - 저장된 데이터와 질의 조건으로 주어진 데이터 간의 유사도를 수학 함수로 계산하여, 유사도가 높은 데이터를 검색
- 랭킹(ranking) 기법
  - 질의 조건과의 관련 정도에 따라 정렬하여, 관련성이 높은 결과부터 제공
- 필터링(filtering) 기법
  - 질의 조건과의 관련성이 적은 데이터를 단계적으로 제거하여 검색 범위를 줄여가면서 검색
- 인덱스(index) 기법
  - 인덱스 구조를 이용해 질의 조건에 적합한 데이터를 검색



## ❖ 웹 데이터베이스(web database)

### ■ 필요성

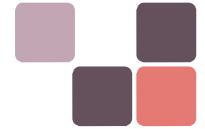
- 새로운 유형의 웹 서비스에서 대용량 데이터를 효율적으로 관리하기 위해 데이터베이스 시스템의 기능이 필요

### ■ 개념

- 웹 서비스의 특성과 데이터베이스 시스템의 데이터 관리 기능을 통합한 것

### ■ 주요 구성 요소

- 미들웨어(middleware)
  - 웹 서비스와 데이터베이스 시스템을 연결해주는 역할을 담당
  - 데이터베이스 통로(database gateway)라고도 함
  - 미들웨어를 통해 데이터베이스에 접근하는 프로그램을 웹 서버 쪽에 두는 서버 확장 방법이나 클라이언트 쪽에 두는 클라이언트 확장 방법으로 구현



### ❖ 데이터 웨어하우스(data warehouse)의 개념

- 데이터베이스 시스템에서 의사 결정에 필요한 데이터를 미리 추출하여, 이를 원하는 형태로 변환하고 통합한 읽기 전용의 데이터 저장소
- 데이터베이스에 저장된 수많은 데이터 중에서 의사 결정에 도움이 되는 데이터를 빠르고 정확히 추출할 수 있는 방법 중 하나
  - 의사 결정 지원 시스템(DSS; Decision Support System) 구축에 활용 가능
- 여러 개의 데이터베이스 시스템을 대상으로 할 수도 있음



### ❖ 데이터 웨어하우스의 개념

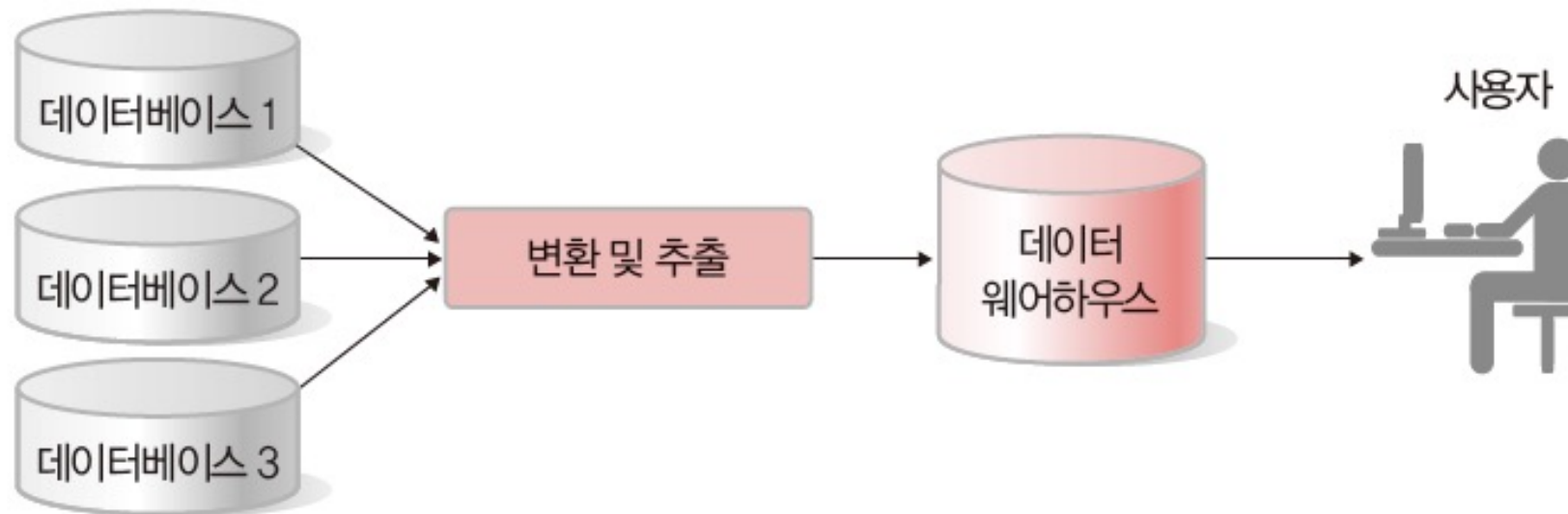


그림 12-12 데이터 웨어하우스 개념



### ❖ 데이터 웨어하우스의 특징

- 주제 지향적(subject-oriented) 내용
  - 일반 데이터베이스는 업무 처리 중심의 데이터로 구성되지만 데이터 웨어하우스는 의사 결정에 필요한 주제를 중심으로 데이터를 구성
- 통합된(integrated) 내용
  - 데이터 웨어하우스는 내부적으로 데이터가 항상 일관된 상태를 유지하도록 여러 데이터베이스에서 추출한 데이터를 통합하여 저장
- 비소멸성(nonvolatile)을 가진 내용
  - 일반 데이터베이스의 데이터는 삽입, 삭제, 수정 작업이 자주 발생하지만 데이터 웨어하우스는 검색 작업만 수행되는 읽기 전용의 데이터를 유지



## ❖ 데이터 웨어하우스의 특징

- 시간에 따라 변하는(time-variant) 내용
  - 일반 데이터베이스는 현재의 데이터만 유지하지만 데이터 웨어하우스는 데이터 간의 시간적 관계나 동향을 분석해 의사 결정에 반영할 수 있도록 현재와 과거 데이터를 함께 유지
  - 각 시점의 데이터를 의미하는 스냅샷(snapshot)을 주기적으로 유지

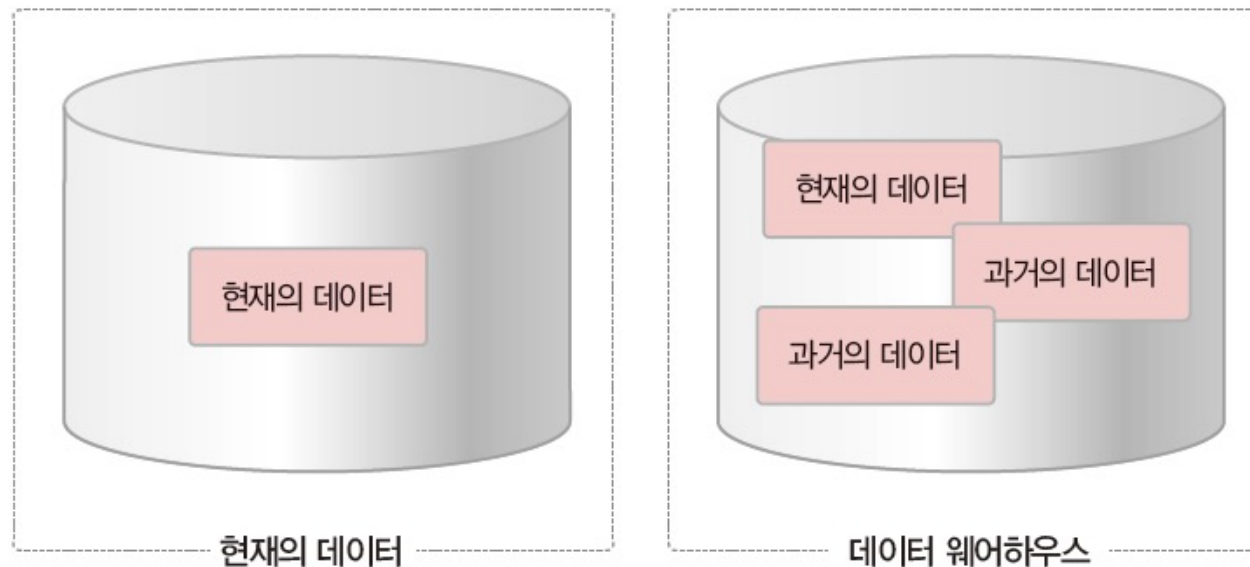
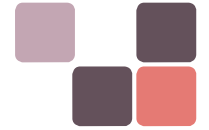


그림 12-13 시간에 따라 변화하는 데이터 웨어하우스



Thank You