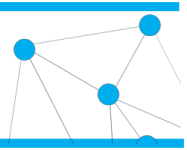
CHAPTER

09

데이터베이스와 빅데이터(2)

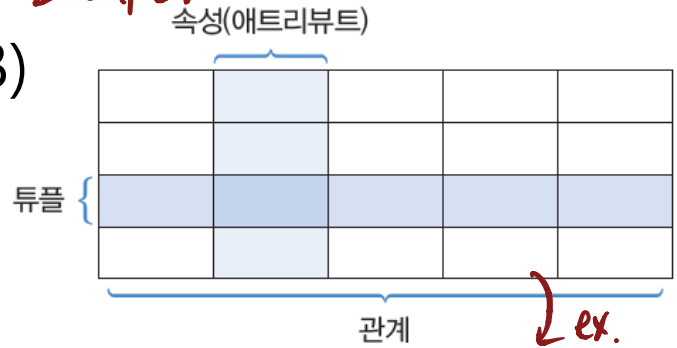
9.3 관계형 데이터베이스



- 관계형 데이터베이스의 구조
- 관계 대수
- 질의어(SQL)

9.3.1 관계형 데이터베이스의 구조

- 테이블은 열(Column)과 행(Row)으로 구성
  - 열 = 필드(field), 행 = 레코드(record)
    = 속성(attribute) = 튜플(tuple)
- 관계형 데이터베이스(RDB)
  - 테이블 = 관계(Relation)
  - 필드 = 속성(Attribute)
  - 레코드 = 튜플(Tuple)
  - 차수(degree) : 어떤 관계가 소유하는 속성의 수
  - 관계가 가지는 튜플의 수를 Cardinality(카디널리티)
  - 수학적 관계(Relation) 및 집합(Set) 이론에 근거



ex. 위의 relation의경우 차수는 5, 카디널리티는 4

RDB

- 관계형 데이터베이스의 구성 예
  - STUDENT(학생) 관계
    - 속성 : 학생 ID, 이름, 학과, 전화번호 (차수가 4)

STUDENT

학생 ID	이름	학과	전화번호

차수4

PROFESSOR

교수 ID	이름	학과	주소	전화번호

차수5

COURSE_RECORD

과목 코드	과목명	학점	교수 ID	학생 ID	점수

차수6

  - 도메인(domain) : 관계의 각 속성이 가질 수 있는 값의 영역
    - STUDENT의 '학생ID' 속성값은 0~9 숫자로만 구성되며, '학과'는 대학에 속하는 학과 이름 중에서만 값을 취할 수 있음

- 관계의 특성

DB설계 → 정규화

  - 다음 4가지 특성을 모두 만족시키는 관계를 정규화 관계(Normalized Relation)라고 함
    1) 한 관계에 포함된 튜플은 모두 상이 (튜플의 유일성)
    2) 한 관계에 포함된 튜플 사이에는 순서가 없음 (튜플의 무순서성)
    3) 한 관계를 구성하는 속성 사이에 순서가 없음 (속성의 무순서성)
    4) 모든 속성값은 더 이상 분해할 수 없는 원자값을 가짐

ex. 주소 = 우편번호 + 도로명주소 (최소단위로 쪼개기)

  - 정규화 관계에서 각 튜플은 항상 유일하므로 몇 개 속성값을 이용해서 튜플을 유일하게 식별할 수 있음
    - **키(Key)**: 튜플을 유일하게 식별할 수 있는 속성의 집합

ex. STUDENT relationship에서 key는 학번

학번	이름	학과	학년	주소	...
✓					

- 관계 $R = R(A_1, A_2, \ldots, A_n)$에 대해 속성의 집합 $A = \{A_1, A_2, \ldots, A_n\}$일 때, R의 한 속성의 집합 $K \subset A$가 다음 두 성질을 만족한다면, K를 후보키(Candidate Key)라고 부름.
  1) **유일성**: 관계 R의 모든 튜플에 대해 K의 값은 서로 상이하고 유일함
  2) **최소성**: 유일성의 특성을 가진 K가 둘 이상의 속성으로 구성되어 있을 때 어느 한 속성을 제거하면 튜플의 유일성이 깨짐. 즉, 후보 키는 모든 튜플을 유일하게 식별하기 위한 최소의 속성만을 가짐

  예) {주민번호, 이름}는 최소성을 만족시키지 못함 → 주민번호가 없어지면
  ⇒ {주민번호}가 최소성을 만족하게 되므로 후보키

➤ 모든 관계는 적어도 하나의 후보키를 반드시 가짐

- **기본키(Primary)** : 관계에서 튜플을 유일하게 식별하기 위해 기본적으로 지정해 놓은 후보키

  예) STUDENT 관계에서 {학생 ID}

  PROFESSOR 관계에서 {교수 ID}

  ↳ relation 마다 반드시하나 여러개있을경우 그중하나가 기본키

- 대체키(Alternate Key) : 후보키의 조건을 만족시키지만, 기본 키가 아닌 후보 키를 말함
  - 예) PROFESSOR 관계에서 {교수이름, 학과}는 대체키가 될 수 있음 (한 학과에 동일한 이름을 가진 교수가 없다고 가정할 때)

- 외래키(Foreign Key) : 한 관계 R에 속한 속성 집합 F가 다른 관계 S의 기본키가 될 때 F를 관계 R의 외래키라고 함 → relation 끼리의 관계연결.

R
orders의 외래키

orders

order_d...	product_id	product_title	qty	user_id
2019-10-01	p131	coke	5	marco171
2019-10-01	p132	sprite	10	marco171
2019-10-02	p131	coke	1	marco171
2019-10-02	p132	sprite	5	marco171

F

S
users의 기본키

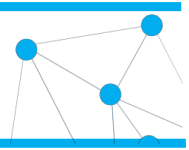
users

user_id	user_pwd	name	gender	address
marco117	xxx	Marco	male	451 Sierra St.

→ R(orders)의 속성집합 F (user_id)가 다른 relation인 S(users)의 기본키

→ user_id는 orders의 외래키.

9.3.2 관계 대수



- 관계형 데이터베이스(RDB)에서 관계는 튜플의 집합
  - 관계대수 : RDB의 관계를 처리하기 위한 연산을 지원
  - 집합 연산:
    - 합집합 연산(Union)
    - 교집합 연산(Intersection)
    - 차집합 연산(Difference)
    - 카티션 프로덕트 연산(Cartesian Product)
  - 순수 관계 연산
    - 선택 연산(Select)
    - 추출 연산(Project)
    - 조인 연산(Join)

집합연산과 관계연산 사례

A =

K	L	M	N
r	6	a	3
s	5	c	7
t	3	d	4
p	8	c	7

- 선택 연산(SELECT)

  SELECT from A where A.N = 7

K	L	M	N
s	5	c	7
p	8	c	7

- 추출 연산(PROJECT)

  PROJECT M, N from A

M	N
a	3
c	7
d	4

- 조인 연산(JOIN)

두개의 relation으로 새로운 relation

JOIN A and B where A.L = B.P

A.K	A.L	A.M	A.N	B.P	B.Q	B.R
s	5	c	7	5	p	s
t	3	d	4	3	u	r

A =

K	L	M	N
r	6	a	3
s	5	c	7
t	3	d	4
p	8	c	7

B =

P	Q	R
5	p	s
4	t	p
3	u	r
7	v	a

- 합집합 연산(UNION) : B ∪ C

5	p	s
4	t	p
3	u	r
7	v	a
2	a	f

C =

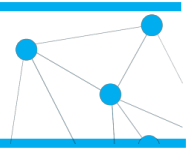
X	Y	Z
2	a	f
3	u	r
4	t	p

- 교집합 연산(INTERSECTION) : B ∩ C

4	t	p
3	u	r

9.3.3 질의어(SQL)

DBMS : DB정의, 제어, 조작



- SQL(Structured Query Language)
  - 데이터베이스의 개념적 구조를 정의하고 데이터를 제어하기 위해 표준 질의어가 필요
    - 데이터베이스 정의 기능 DDL(Data Definition Language),
    - 데이터 조작 기능 DML(Data Manipulation Language),
    - 데이터 제어 기능 DCL(Data Control Language)
  - 비절차적 언어로 사용자는 자신이 원하는 데이터를 선언함으로써 정보를 검색
  - SQL은 관계해석(Relational Calculus)에 기반

해석학, 집합론

1) 데이터 정의어(DDL)

- 사용자가 관계를 생성/제거하고, 새로운 속성을 추가/삭제하며, 관계의 기본 키를 정의하는 기능을 담당
- SQL의 CREATE, ALTER, DROP, RENAME 등

생성 속성수정 삭제 이름바꾸기

```
CREATE TABLE DEPARTMENT
    (DEPTNO    NUMBER    NOT NULL,
    DEPTNAME   CHAR(12),
    BUILDING   CHAR(20)
    PRIMARY KEY (DEPTNO));
```

table name

2) 데이터 조작어(DML)

- 데이터베이스로부터 데이터를 검색하고 데이터를 수정, 추가, 삭제하는 기능 → 연산
- SQL의 SELECT, SORTING, INSERT, DELETE, UPDATE
- SELECT 문의 사례:

```
SELECT ST_ID, NAME
from STUDENT
where DNO = 3, YEAR = 4
```

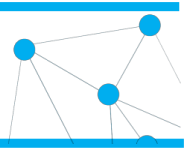
학과번호 학년 → 여기에 일치하는 ID 아이템

3) 데이터 제어어(DCL)

- 관계에 대한 접근 권한을 부여하거나 취소하는 기능

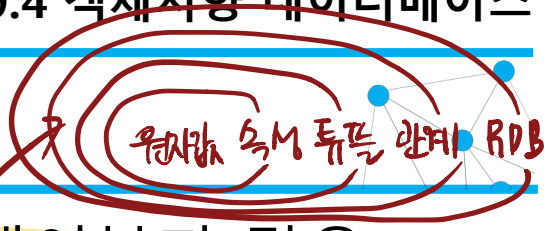
일반사용자X ex. GRANT, DENY, REVOKE, ..

9.4 객체지향 데이터베이스



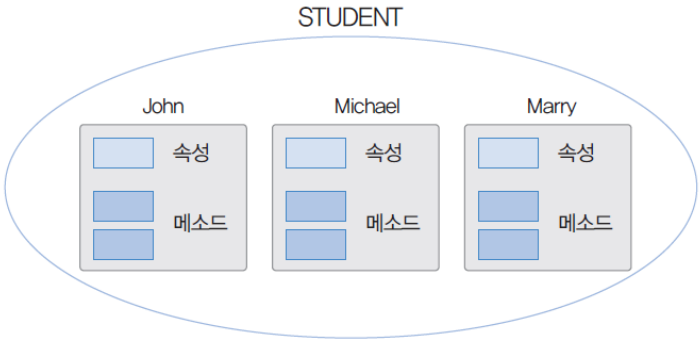
- 객체지향의 개념
- 멀티미디어 데이터베이스

9.4.1 객체지향의 개념



- RDB는 현실세계의 개체들을 테이블과 같은 정형화된 형태로 표현하는 경우 적합
  - RDB는 관계로 구성 – 각 관계는 같은 유형의 튜플들로 이루어지고, 각 튜플은 몇 개의 속성을 가짐
  - 속성은 숫자, 문자열 등 고정 길이의 작은 원자값을 가짐
- 현실 세계의 모든 개체들이 다 관계로 표현되지는 못함
  - 단순한 테이블 형태가 아니고 복잡한 구조를 갖는 비정형화된 데이터가 존재

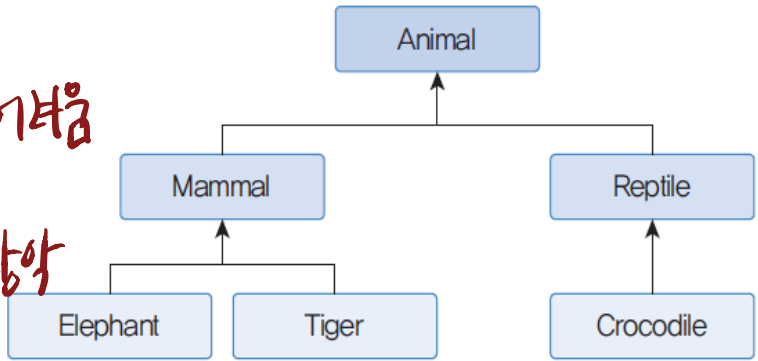
- 객체지향데이터베이스(OODB)는 객체지향 개념에 기반
- 객체와 메소드 (데이터+연산)
  - 현실 세계의 개체를 객체로 정의
  - 객체 식별자, 속성(애트리뷰트), 메소드, 클래스, 클래스 계층 및 상속, 복합 객체 개념을 모두 포함
  - 연산은 모두 메시지(Message)를 통해 수행



| 그림 9-16 클래스의 표현과 객체 및 메소드

- SQL과 유사한 OQL(Object Query Language)를 지원함으로써 객체 데이터베이스에 대해 비절차적 접근을 수행
- 클래스의 상속 관계
  - 슈퍼클래스/서브클래스 : 계층구조
  - 슈퍼클래스의 속성과 메소드를 상속받는다.

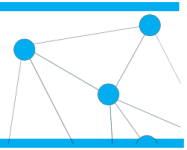
1990 등장
만들고 고작하게 이거임
+
지금 RDB가 시장장악
→ 사라짐.



| 그림 9-17 클래스 계층과 상속

- 객체-관계 데이터베이스(ORDB)
  - 1990년대 후반 등장
  - 관계형 데이터베이스에 객체지향 개념이 통합된 개념으로, 기존의 RDB를 확장하여 멀티미디어와 같이 보다 복잡한 데이터를 처리할 수 있도록 객체지향 개념을 첨가시킨 데이터 모델
  - ORDB는 관계, 질의어, 객체, 메소드, 클래스, 상속, 복합 객체를 지원
  - ORDBMS를 Universal DBMS라 부르기도 함

9.4.2 멀티미디어 데이터베이스



- 멀티미디어 데이터는 이미지, 사운드, 비디오, 텍스트/문서, 애니메이션, 그래픽과 같은 데이터가 복합적으로 존재하는 것
  - 과거에는 멀티미디어 데이터를 외부에 파일 형태로 저장
  - 멀티미디어 데이터의 양이 급증하면서, DB를 이용하여 저장, 검색, 관리해야 하는 단계
  - 멀티미디어 데이터의 특성 : 제작자, 제작일시, 카테고리, 용량 등과 같은 서술적 속성도 가지면서, 데이터의 구조가 복잡하고 용량이 매우 크다

- 관계형 데이터베이스: 정형화 데이터베이스
- 멀티미디어 데이터: 비정형화 데이터 ⇒ 검색의 난이도가 증가

텍스트만 X

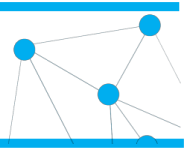
- 내용기반 검색(CBIR: Contents Contents-Based Information Retrieval)
  - 어떤 내용이나 행위를 담고 있는 멀티미디어 데이터를 검색하는 것

9.5 빅데이터와 데이터 분석

- 인터넷의 활용이 급증하고 모바일 환경으로 변화하면서 다양한 소스를 통해 수많은 양의 데이터가 생성
- 현대의 데이터는 의사결정 및 미래 예측에 매우 중요한 역할

  - 빅데이터의 소개
  - 빅데이터 분석과 활용

9.5.1 빅데이터의 소개

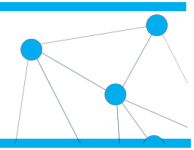


- 빅데이터의 등장 배경
  - 인터넷, 모바일 인터넷, 소셜미디어, GPS 등의 센서, 디지털 카메라 등 IoT 기술의 발전으로 다양한 소스를 통해 엄청난 양의 빅데이터 생성
  - 통신 인프라 구축과 인터넷 속도의 증가
  - 프로세서 가격 하락과 처리 속도의 가파른 증가로 데이터를 실시간 분석하고 활용할 수 있게 됨
    - 메모리 가격의 하락과 용량 증가

- 빅데이터의 특성 3V
  - 데이터양(Volume)
    - TB = 10^{12} 바이트, FB = 10^{15} 바이트
  - 다양성(Variety)
    - 기업의 데이터, 인터넷 & SNS의 텍스트/이미지/동영상 데이터, 위치정보, 센서 데이터 등
  - 속도(Velocity)
    - 발생 빈도와 갱신 속도가 매우 빠름
    - 예) 매장에서 발생하는 POS 데이터, 24시간 발생하는 전자상거래 데이터, 감시 카메라로부터 영상 데이터는 지속적으로 발생

➢ 끊임없이 발생하는 다량의 데이터를 분석/처리하여 적절하게 활용하는 것은 매우 도전적인 과제

9.5.2 빅데이터의 분석과 활용



- 빅데이터 기술 : 하둡과 NoSQL
  - RDBMS는 일반적인 기업의 데이터 업무처리에 적합하지만, 빅데이터는 비구조적 데이터이므로 정형화된 기술은 부적합
  - 비구조적 데이터에 스키마를 미리 지정해 두는 것은 비현실적
  - 빅데이터는 대규모 분산처리 기술인 하둡(Hadoop)을 주로 이용하므로, 분산되어 저장/처리되어 데이터 간의 일관성 유지가 어려움
    - NoSQL('Not only SQL') 데이터베이스가 개발되어 빅데이터 분석에 사용

- 빅데이터 분석 기술
  - 기계 학습이나 데이터 마이닝 기술 적용
  - 기계학습 ex. 쇼핑몰의 추천system
    - 인공지능 기술의 하나
    - 인간의 학습능력을 컴퓨터를 통해 구현한 것
    - 빅데이터를 분석해서 그 데이터로부터 유용한 규칙, 지식 표현, 판단 기준 등을 도출하는 것
  - 데이터 마이닝
    - 대량의 데이터를 분석하여 데이터 속에 내재되어 있는 변수 사이의 상호관계를 규명하여 일정한 패턴을 찾는 기법
    - 클러스터링, 신경망 네트워크, 회귀 분석, 결정 트리 및 연관분석 등의 방법

ex. AI 기자가 쓴 뉴스기사

- 자연어 처리 기술을 이용하여 대용량 소셜 미디어의 텍스트 데이터로부터 유용한 정보 추출

- 빅데이터의 활용
  - 검색 엔진, 패턴 인식, 번역 서비스, 음성 인식 서비스 등의 분야
  - 전자상거래의 추천 시스템, 사용자 행동 분석, 고객 구매행동 예측