1과목 : 데이터베이스

Check 1. 데이터베이스 정의 [출제빈도: 下]

1. 데이터 베이스 정의 ★★★☆☆

-특정 조직이 업무 수행하는데 필요한 관련성 있는 자료들의 집합체
-통합(Integrated),저장(Stored),운영(Operational),공용(Shared)

2. 데이터베이스 시스템 도입 배경 ★★☆☆☆

-파일 시스템의 문제점을 해결

ex) 응용 p/g 1(인사) ——— 파일1
응용 p/g 1(인사) ——— 파일2
응용 p/g 1(인사) ——— 파일3
응용 p/g 1(인사) ——— 파일4

* **독립된 파일 단위로 업무와 관련한 데이터를 저장**하므로 **데이터 중복성**과 **데이터 종속성** 발생
⇒ 데이터 무결성 위배 가능성 높음

3. 데이터베이스 시스템 개념 ★★☆☆☆

1) 데이터 독립성
*물리적 데이터 독립성: 기존 응용 프로그램에 영행을 주지 않고 데이터의 물리적인 구조를 변경할 수 있는 것을 말함.

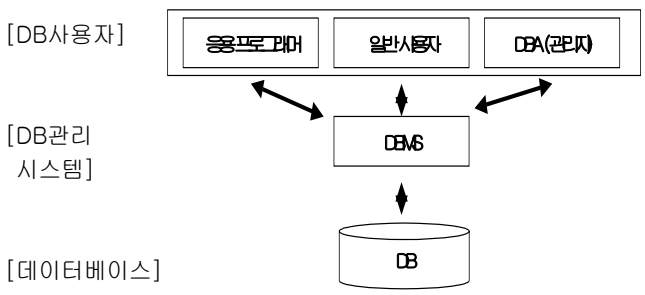
*논리적 데이터 독립성: 데이터의 논리적 구조를 변경시키더라도 응용 프로그램은 변경되지 않는다.

2) 데이터를 통합운영 하므로 중복성 감소, 불일치 감소
⇒ 데이터 일관성, 무결성 유지

4. 데이터베이스 특징 ★★★☆☆

1) 실시간 접근: 내가 원할 때 마다 언제든지 바로 접근해서 자료를 찾을 수 있다.
2) 계속적인 변화: 데이터의 삽입, 삭제, 갱신 작업으로 항상 최신의 데이터를 유지해야 한다.
3) 공용: 여러 사용자가 함께 쓸 수 있어야 한다.
4) 내용에 의한 참조: 위치나 주소가 아닌 데이터의 내용, 즉 값에 따라 참조 할 수 있다.

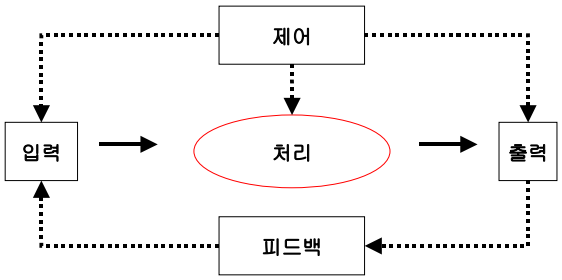
5. 데이터베이스 시스템 구성



Check 2. 정보시스템 [출제빈도: 下]

1. 정보시스템 ★★☆☆☆

-한 조직체의 데이터를 바탕으로 의사결정에 필요한 정보를 추출하고 생성하는 시스템



2. 자료처리 시스템

-정보 시스템에서 처리 과정을 의미함

3. 자료처리 시스템의 종류 ★☆☆☆☆

1) **일괄**처리시스템 (Batch Processing System)

-일괄시간 또는 일정량의 데이터를 한꺼번에 모아서 처리 (시스템중심)
-각 트랜잭션 당 처리비용이 적게든다.
-ex) 급여계산, 회계마감업무, 세무처리 등

2) **온라인 실시간 처리** 시스템 (Real-time Processing System)

-데이터가 발생하는 즉시 처리하여 결과를 산출하도록 하는 시스템(사용자중심)
-ex) 기차예매, 티겟예매, 은행업무 등

3) **분산** 처리 시스템

-컴퓨터들이 지리적으로 분산되어 있지만, 실제 사용자들이 볼 때는 논리적으로 하나로 연결되어 있는 것처럼 보여져서 처리되는 시스템

4. 데이터웨어하우스(Datawatehouse)

-기간 업무 시스템에서 추출되어 새로이 생성된 데이터베이스로서 **의사결정자원 시스템을 지원**하는 주체적, 통합적, 시간적 데이터의 집합체

Check 3. DBNS [출제빈도: 上]

1. DBMS 정의 ★☆☆☆☆

-응용프로그램(사용자)와 데이터베이스 사이에서 사용자의 요구에 따라 DB생성, 관리해주는 S/W
-응용 프로그램과 데이테베이스 사이의 중재자

2. DBMS 필수 기능 ★

1) **정의**: DB 자료형, 데이터 구조, 이용방법, 제약조건을 명시

-다양한 응용 P/G과 DB가 서로 인터페이스를 할 수 있는 방법을 제공
-데이터의 논리적 구조와 물리적 구조 사이의 변환이 가능하도록 두 구조 사이의 사상(Mapping)을 명세

2) **조작**: 검색, 저장, 삭제, 갱신 기능
3) **제어**: 데이터의 무결성, 보안, 정확성, 병행수행, 안정성유지

3. DBMS의 장·단점 ★☆☆☆☆

장 점	단 점
①독립성 보장 (논리적 or 물리적)	①전문가가 부족
②데이터 중복이 X	②전산화 비용이 증가
③공동으로 자료 이용	③Acess할 때 오버헤드 발생
④일관성을 유지	④시스템이 복잡
⑤데이터의 무결성을 유지	→예비(Backup)와 회복(Recovery)이 어려움
⑥데이터 표준화	⑤자료처리 복잡
⑦데이터를 통합하여 관리	
⑧최신의 데이터 유지	
⑨데이터 실시간 처리	

Check 4. 스키마(Schema) [출제빈도: 上]

1. 스키마 정의 ★☆☆☆☆

-데이터베이스의 구조와 제약조건에 대한 명세(Specification)를 기술(Description)한 것
-데이터베이스를 구성하는 데이터 객체, 이들의 성질, 이들 간에 존재하는 관계, 그리고 데이터의 조작 또는 이들 데이터 값들이 갖는 제약조건에 관한 정의를 총칭하는 용어.

2. 스키마 특징 ★☆☆☆☆

1) 데이터사전(Data Dictionary =시스템카탈로그)에 저장

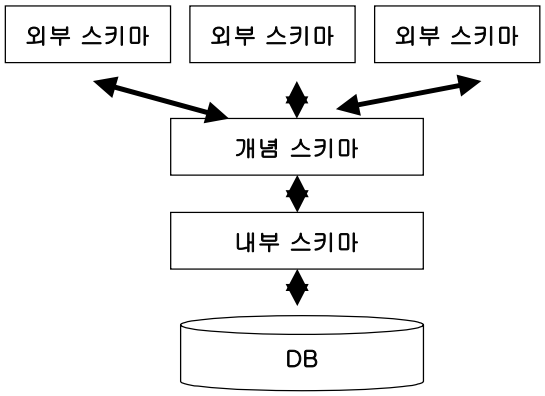
-DB에 저장되어 있는 모든 데이터 개체들에 대한 정보를 유지, 관리하는 시스템

2) 데이터베이스의 구조(개체, 속성, 관계)에 대한 정의

3) 다른 이름으로 메타데이터(데이터의 데이터)라고 함

3. 스키마 3계층 ★

1) 외부스키마 (=서브스키마=사용자 뷰) : **사용자**가 보는 관점 (사용자에 따라 다름, 여러 개 존재)
2) 개념스키마 (=스키마 =전체적인 뷰 =범 기관적, 총괄적 입장) : DB **전체적인 논리적구조** 개체간의 관계와 제약조건을 나타내고 DB의 접근 권한, 보안 및 무결성 규칙을 명세화 한다.
3) 내부스키마 (실제 데이터 저장) : DB 전체적인 물리적구조, DBA 관리



Check 5. 데이터 언어 [출제빈도: 中]

1. 데이터 언어(=데이터베이스 언어)

-DBMS의 3가지 기능을 구현하기 위한 언어

2. 데이터 언어 종류 ★★★★☆

1) **데이터 정의어** (DDL) Definition -DB형태,구조,DB의 저장에 관한 내용 정의, 및 변경 →사용자(응용프로그램)과 DB간의 인터페이스 제공	**DBA**
2) **데이터 조작어** (DML) Manipulation -사용자의 요구에 따라 검색,갱신,삽인, 삭제 등을 지원하는 기능 →사용자(응용프로그램)과 DBMS 간의 인터페이스 제공	**응용 프로그래머, 사용자**
3) **데이터 제어어** (DCL) Control -정확성과 안정성을 유지하는 기능 (무결성유지, 보안, 권한, 병행수행제어, 회복)	**DBA**

*질의어(Query) : 데이터 언어 중에서 터미널에서 주로 이용하는 비절차적 데이터 조작어

3. 데이터 사용자 ★☆☆☆☆

1) 응용프로그래머	DB활용, 사용자인터페이스 제공
2) 일반 사용자	
3) DB 관리자 (DBA: Administrator)	* 역할 ★ · DB 설계와 조직에 대한 책임 · DDL, DCL을 사용 · DB스키마 정의 · 보안정책과 무결성 유지 · DB 설계와 운영 · 사용자의 요구와 불평 청취/해결 · 시스템 감시 및 성능 분석 (사용자요구변화분석, 장비성능감시 데이터사용추세분석) · DBMS관리 · DB 구조관리, DD(데이터사전) 구성 · 저장구조와 액세스 방법 정의 · DB의 이상 현상 감시 ※ 역할이 아닌것~!! (주의) · 응용프로그램 개발 (X) · 주로 DML(조작어)를 이용(X) · DB 자원활용(사용) 및 사용자의 인터페이스 제공 (X) · 데이터를 저장하고 저장된 데이터를 사용(X) · 사용자 통제 및 감시 (X) · 정보추출을 위한 DB 접근 (X)

Check 6. 데이터베이스 설계 [출제빈도: 上]

1. 데이터베이스 설계란?

-현실세계의 업무적인 프로세스를 컴퓨터세계로 DB화 하기위한 과정

2. 설계순서 ★ (순서 암기!)

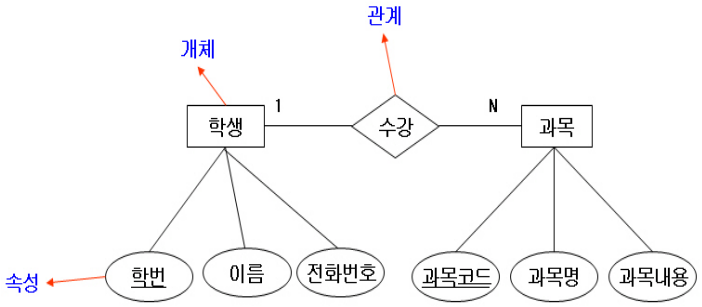
①요구조건분석→②개념적설계→③논리적설계→④물리적설계→⑤구현→⑥운영→⑦감시 및 개선

①요구조건분석: 업무프로세스 분석→요구조건 명세서작성

*요구조건명세서
한국대학교의 주된 개체는 학생과 과목이다. 학생은 고유학번이 부여되며, 추가로 이름, 전화번호 정보를 가진다. ~~~~

②개념적 설계 ★★☆☆☆

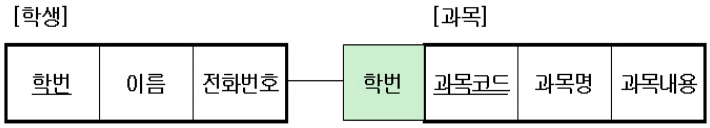
-개체 타입과 이들 간의 관계타입을 이용해 현실세계를 개념적으로 표현. (산출물: 개체관계도= ER 다이어그램)
-DBMS에서 독립적인 **개념스키마 모델링**
-**트랜잭션 모델링**



개체 Entity	DB에 표현하려는 현실 세계의 대상체	학생, 교수, 학과, 과목
속성 Attribute	개체의 성질, 분류, 식별, 수량, 상태 등을 나타냄.	학생-학번, 이름, 전화번호
관계 Relation ship	두 개체에 의미 있는 연결	학생은 과목을 수강한다. 과목은 학생에게 수강되어진다.

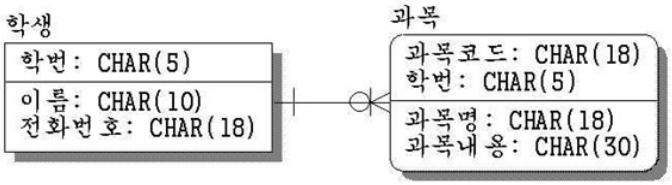
③논리적설계 ★★☆☆☆

-목표 DBMS에 맞추어 논리적 모델로 설계 (관계형, 계층형, 망형 모델)
-**트랜잭션 인터페이스 설계**
-**스키마**의 평가 및 정제
-정규화 과정 수행



④물리적설계 ★★★☆☆

-저장레코드 양식의 설계 및 물리적 구조 데이터 표현
-설계 시 고려사항: 응답시간, 저장공간의 효율성, 트랜잭션의 처리양
-어떤 인덱스를 만들 것인지에 대한 고려
-성능 향상을 위한 개념 스키마의 변경 여부 검토
-빈번한 질의와 트랜잭션들의 수행속도를 높이기 위한 고려
-접근경로설계
-레코드 집중의 분석 및 설계
-트랜잭션 세부사항 설계



⑤구현 ★☆☆☆☆

-목표 DBMS DDL로 스키마를 작성(정의), 응용 P/G을 위한 트랜잭션을 작성하는 단계

Check 7. 데이터모델 [출제빈도: 中]

1. 데이터 모델 정의 ★★★☆☆

-현실세계의 데이터 구조를 컴퓨터 세계의 데이터 구조로 기술하는 개념적인 도구이다.

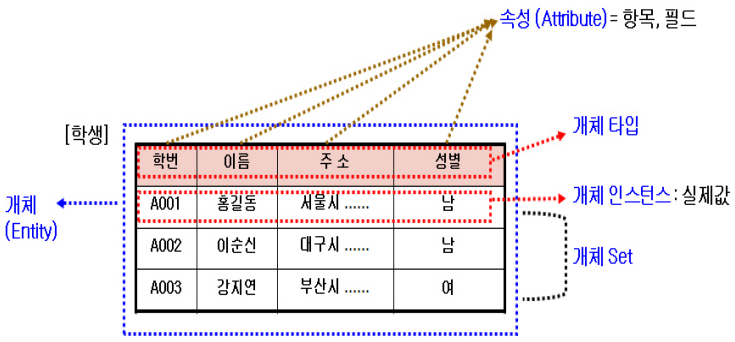
2. 데이터 모델 종류 ★☆☆☆☆

개념적 모델	현실세계를 추상적으로 표현	E-R 모델
논리적 모델	개념적 모델을 컴퓨터가 이해할 수 있도록 표현	관계, 계층, 네트워크(망) 모델

3. 데이터 모델 구성요소 ★★★★★

1)**구조(Structure):** 개체들 간의 관계
2)**연산(Operation):** 데이터 처리하는 방법
3)**제약조건(Constraint):** 실제 데이터의 논리적인 제약조건

4. 데이터모델 용어



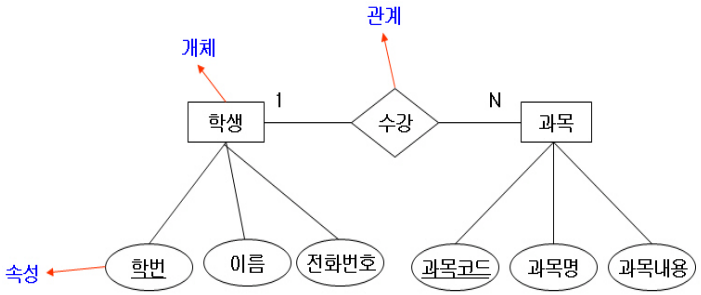
Check 8. E-R 모델 [출제빈도: 上]

1. E-R (Entity-Relationship, 개체관계도) 모델

-개체와 개체간의 관계를 도식화
-1976년 P.Chan에 의해 처음제안

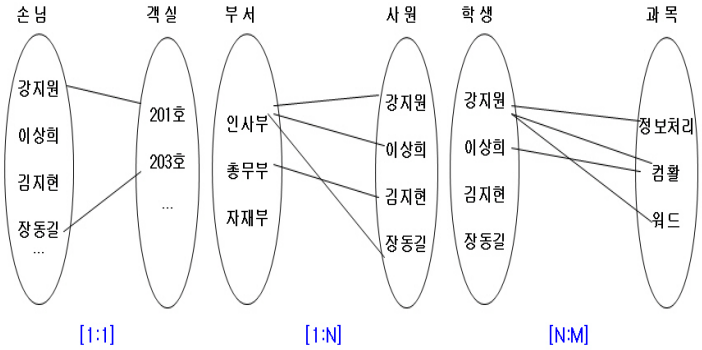
의미	개체	관계	속성	기본키 속성	연결,링크
기호	사각형	마름모	타원,원		선

기본키 : 고유의 속성(ex. 아이디,학번)



2. 관계의 종료 (관계 타입) ★☆☆☆☆

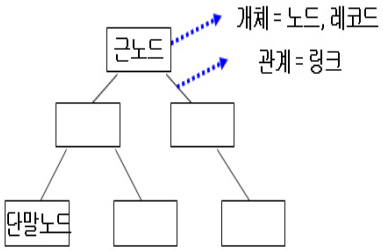
1) **1:1 관계** : E1에 있는 한 개의 데이터는 E2에 있는 한 개의 데이터와 일치하는 관계이다.
2) **1:N 관계**: E1에 있는 각각의 데이터는 E2에 있는 하나이상 데이터와 일치하나 E2에 있는 데이터는 E1에 있는 데이터와 단지 하나만이 일치하는 관계이다.
3) **N:M 관계**: E1에 있는 각각의 데이터는 E2에 있는 하나 이상의 데이터와 일치하고 E2에 있는 데이터도 E1에 있는 하나 이상의 데이터와 일치하는 관계이다.



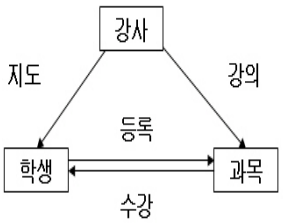
Check 8. 논리적데이터 모델 [출제빈도: 上]

1. 논리적 데이터 모델의 종류 ★★★☆☆

종류	구조	관계표현	특징
관계형	표 =Table =Relation	키(기본키, 외래키) 1:1, 1:N, N:M	*SQL 가장多사용되는모델
계층형	트리	부모-자식관계 N:M 직접 표현 불가	-사이클 허용 X -개체 삭제 시 연쇄 삭제 발생
네트워크형	그래프, 망	오너-멤버관계 N:M 직접 표현 불가	-사이클 허용 -상/하위 개체 복수대응



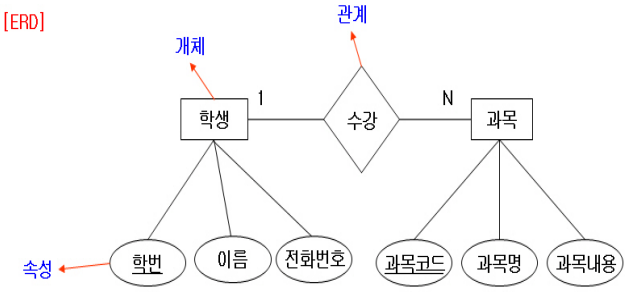
[계층형]

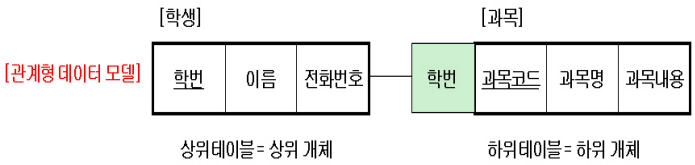


[네트워크형]

2. 요구분석→개념적 설계→논리적 설계

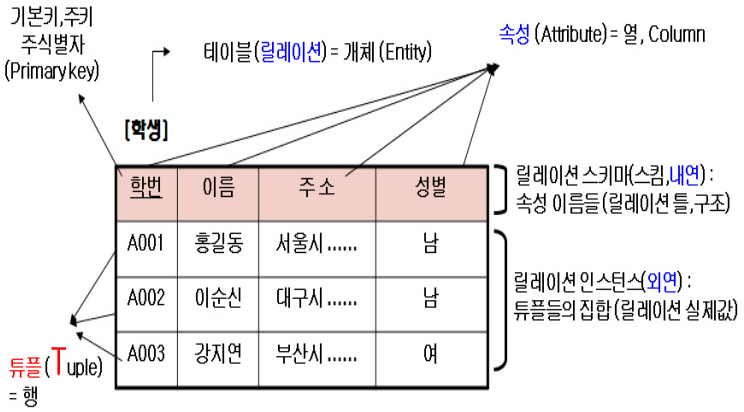
[요구분석]한국학원의 주된 개체는 학생과 과목이다. 학생은 고유의 학번이 부여되며, 추가로 이름, 전화번호 정보를 가진다. 과목은 고유의 과목코드가 부여되며, 추가로 과목명, 과목내용을 가진다. 한 명의 학생은 여러 개의 과목을 수강할 수 있다.





Check 9. 관계데이터 모델 [출제빈도: 上]

1. 관계형 데이터베이스의 릴레이션 구조 ★



* 도메인(Domain) : 한 속성에 나타날 수 있는 값들의 범위(집합)
* 차수(Degree) : 속성들의 수
* 카디날리티(cardinality) : 튜플들의 수
* 널(Null) : "해당없음" 등의 이유로 정보 부재를 나타내기 위해 사용하는 특수한 데이터 값
- 공백이나 0(zero) X

2. 릴레이션(Relation)의 특징 ★★★★★ (선이해, 후암기)

1) 한 R에 정의된 튜플들을 모두 다르다.
2) 한 R에 정의된 튜플들은 순서에 무관하다.
3) 튜플들은 시간에 따라 변한다.
4) R 스키마를 구성하는 속성들도 순서에 무관하다.
5) 속성의 명칭은 유일해야하지만, 속성의 값은 동일해도 된다.
6) 속성은 더 이상 쪼갤 수 없는 원자값으로 구성된다.
(속성의 값은 분해X, 다중값X)
7) R을 구성하는 튜플을 유일하게 식별하기 위한 속성들의 부분 집합을 키(Key)로 설정한다.

3. 키(Key)의 개념 및 종류 ★★★☆☆

-DB에서 조건에 만족하는 튜플을 찾거나 순서대로 정렬할 때 다른 튜플들과 구별할 수 있는 유일한 기준이 되는 속성이다.

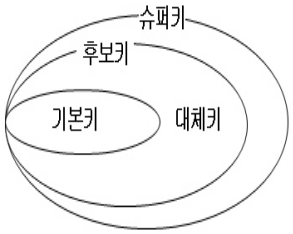
1)슈퍼키(Super Key)
: 한 R 내에 있는 속성들의 집합으로 구성된 키 (유일성)

2)후보키(Candidate Key)
: 한 R 내에 있는 모든 튜플들을 유일하게 식별할 수 있는 하나 또는 몇 개의 에트리뷰트 집합
(최소 슈퍼키: 유일성 + 최소성)

3)기본키(Pramary Key)
: 후보키 중에 선택한 키
(중복되어서는 안되며, Null 값을 가질 수 없다.)

4)대체키(Alternate Key)
: 후보키 중에서 기본키를 제외한 속성들

5)외래키(Foreign Key)
: 어떤 R에서 다른R을 참조할 때 참조 기준이 되는 속성으로서 참조하고자 하는 R의 기본키와 동일



[회원]

학번	이름	주소	주민번호	전공
A001	홍길동	서울시	123456	컴퓨터
A002	이순신	대구시	222222	수학
A003	강감찬	부산시	333333	물리

4. 제약조건 (무결성, Integrity) ★

개체 무결성	한 R의 **기본키**를 구성하는 어떠한 속성 값도 널(NULL) 값이나 중복 값을 가질 수 없다.
참조 무결성	R은 참조할 수 없는 **외래키**값을 가질 수 없음을 의미하는 제약조건
도메인 무결성	각 속성 값은 반드시 정의된 도메인에 속한 값이어야 한다.

성적관리 테이블

학번	이름	핸드폰번호	주소	선택과목	담당교수	성적
K001	홍길동	011-111-1111	서울시 구로구 구로본동 234-24	자료구조	박세리	A
K002	이순신	016-111-1111	서울시 영등포구 여의도동 234-2	데이터베이스	이미연	B
K003	강감찬	018-111-1111	서울시 강남구 역삼동 234-2			
K004	선동열	017-111-1111	서울시 서초구 서초동 234-2			
K005	박찬호	019-111-1111	서울시 종로구 관철동 234-2	컴퓨터구조	강동원	C

관계형성을통한상위테이블의PK전이

Primary key (기본키) → Foreign key (외래키)

학생관리 테이블

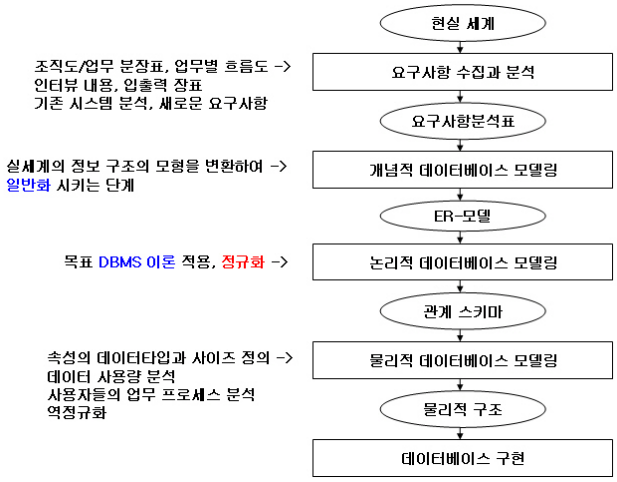
학번	이름	핸드폰번호	주소
K001	홍길동	011-111-1111	서울시 구로구 구로본동 234-24
K002	이순신	016-111-1111	서울시 영등포구 여의도동 234-2
K003	강감찬	018-111-1111	서울시 강남구 역삼동 234-2
K004	선동열	017-111-1111	서울시 서초구 서초동 234-2
K005	박찬호	019-111-1111	서울시 종로구 관철동 234-2

성적관리 테이블

학번	선택과목	담당교수	성적
K001	자료구조	박세리	A
K002	데이터베이스	이미연	B
K005	컴퓨터구조	강동원	C

Check 10. 정규화 [출제빈도: 中]

1. 관계형 데이터베이스 모델링 ★★★☆☆



2. 정규화 (Normalization) ★

-정규화를 하는 이유는 데이터의 중복을 방지하고 보다 효율적으로 데이터를 저장하기 위함.
(릴레이션 분리 →삽입, 삭제, 이상의 발생가능성을 줄이는 것)
-**단점:** 연산 시간이 증가됨.

-특징

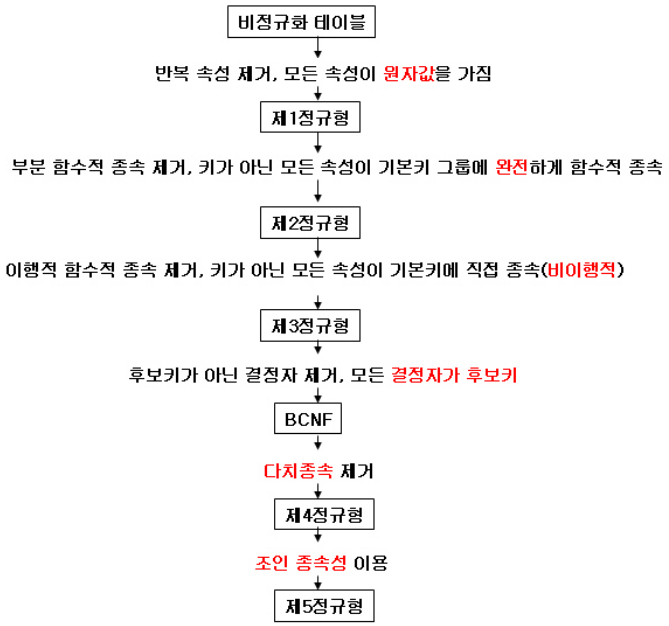
①예를 들어 현재 테이블이 3정규형 상태라면 1,2 정규형은 자동으로 만족해야 한다.
②정규형들은 차수가 높이질수록(1→5정규형)만족시켜야 할 제약조건이 증가된다.
③정규화는 논리적 처리 및 품질에 큰 영향을 미친다.
④정규화의 목적은 논리적DB 구조상에 있어 삽입,수정,삭제 결과 생기는 이상현상을 제거하는데 있다.
⑤레코드들의 관련 속성들 간의 종속성을 최소화하기 위한 구성 기법이다.
⑥정규화가 잘못되면 데이터의 불필요한 중복을 야기하여 릴레이션 조작 시 문제를 일으킨다.
⑦정규화되지 못한 릴레이션의 조작시 발생하는 이상현상의 근본적인 원인은 여러 가지 종류의 사실들이 하나의 릴레이션에 표현되기 때문이다.

3. 이상(Anomaiy) ★★☆☆☆

-릴레이션에서 일부 속성들의 종속으로 인해 데이터의 중복이 발생하여 테이블 조작 시 불일치가 발생하는 것.

갱신이상 (Update)	반복된 데이터 중에 일부만 수정하면 데이터의 불일치가 발생
이상희라는 사람의 주소를 변경할 경우 모든 속성(칼럼)의 주소를 변경해야 한다. 만약 하나만 변경할 경우 데이터의 불일치가 발생한다.	
삽입이상 (Deletion)	불필요한 정보를 합계 저장하지 않고는 정보를 저장하는 것이 불가능
MOS라는 과목을 추가할 경우, 불필요한 회원정보까지 추가해야 한다.	
삭제이상 (Deletion)	유용한 정보를 함께 삭제하지 않고는 어떤 정보를 삭제하는 것이 불가능
이상희라는 사람의 데이터를 삭제하고자 할 경우, 정보처리라는 과목까지 삭제되어 버린다.	

4. 정규화 과정 ★★☆☆☆



Check 11. 관계데이터연산 [출제빈도: 中]

1. 관계 데이터 연산 ★★☆☆☆

관계대수	관계해석
1.절차적 언어(절차중심) -원하는 정보를 **'어떻게'유도**하는가를 연산자와 연산규칙 이용하여 기술 2.분류: 순수관계연산자, 일반집합연산자 3. SQL의 이론적인 기초	**1. 비절차적 언어(결과중심)** -원하는 정보가 **'무엇'**이라는 것만 정의 2. 분류: 튜플관계해석과 도메인관계해석

-기본적으로 관계해석과 관계대수는 관계 데이터베이스를 처리하는 기능과 능력면에서 동등하다.
-관계해석으로 표현한 식은 관계대수로 표현할 수 있다.

2. 관계대수 종류 ★★★☆☆

1) 순수관계 연산자

Select(σ)	-릴레이션에서 주어진 조건을 만족하는 **튜플**들을 검색하는 것으로 기호는 그리스문자의 **시**그마를 이용 **-(행, 수평적 연산)**
Project(π)	-릴레이션에서 주어진 조건을 만족하는 **속성**들을 검색하는 것으로 기호는 그리스문자의 **파**이를 이용 **-(열, 수직적 연산)**
Join(⋈)	-두개의 릴레이션 A와 B에서 공통된 속성을 연결
Division (÷)	-나누어지는 릴레이션인 A는 릴레이션B의 모든 내용을 포함한 것이 결과 릴레이션이 된다.

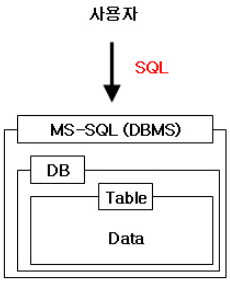
2) 일반 집합 연산자

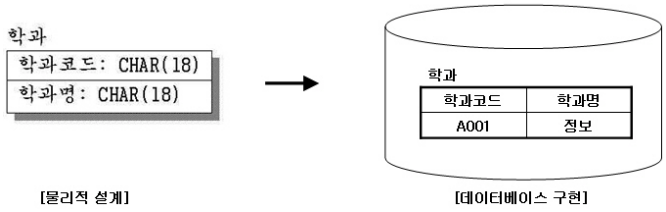
합집합 (U)	릴레이션 A 또는 B에 속하는 튜플들로 구성된 릴레이션 (Union)
교집합 (∩)	릴레이션 A 와 B에 광통적으로속하는 튜플들로 구성된 릴레이션 (Intersection)
차집합 (-)	릴레이션 A에만 있고 B에는 없는 튜플들로 구성된 릴레이션 (Difference)
카디션 프로덕트 (X)	A에 속한 각 튜플 a에 대해하여 B에 속한 튜플 b를 모두 접속시킨 튜플들(a b)로 구성된 릴레이션 (cartesian product)

Check 12. SQL [출제빈도: 上]

1. SQL (Structured Query Language) 특징 ★

1) 관계대수와 관계해석을 기초로 한 고급데이터 언어
2) 이해하기 쉬운 형태
3) 대화식 질의어로 사용 가능
4) 데이터 정의, 조작, 제어 기능 제공
5) COBOL, C, PASCAL 등의 언어에 삽입→ 내장 SQL
6) 레코드 집합 단위로 처리
7) DBMS에서 사용되는 비절차적 대화형 Language





[물리적 설계] [데이터베이스 구현]

2. SQL 구분 ★

* 각 구문은 반드시 책으로 복습합시다. (6강 SQL 구문에 해당)

1) DDL(데이터 정의어)

-스키마, 도메인, 테이블, 뷰, 인덱스를 정의하거나 변경 또는 삭제 할 때 사용하는 언어.

CREATE (정의)	스키마, 도메인, 데이블, 뷰, 인덱스를 정의
ALTER (변경)	데이블에 대한 정의를 변경하는데 사용
DROP (제거)	스키마, 도메인, 데이블, 뷰, 인덱스를 삭제

2) DML(데이터 조작어)

-데이터베이스 사용자와 관리시스템간의 인터페이스를 제공

SELECT (검색)	데이블에서 조건에 맞는 튜플을 검색함
INSERT (삽입)	데이블에 새로운 튜플을 삽입함
DELETE (삭제)	데이블에서 조건에 맞는 튜플을 삭제함
UPDATE (갱신)	데이블에서 조건에 맞는 튜플의 내용을 변경함

3)DCL(데이터 제어어)

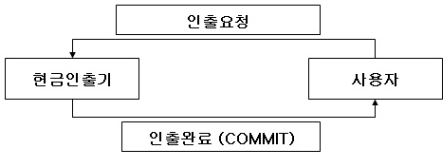
-데이터의 보안, 무결성, 데이터 회복, 병행수행제어 등을 정의하는데 사용하는 언어.
-데이터베이스 관리자가 데이터관리를 목적으로 사용

COMMIT	명령에 의해 수행된 결과를 실제 물리적 디스크로 저장하고, DB 조작 작업이 정상적으로 완료 되었음을 관리자에게 알려줌 (트랜잭션 완료→DB 적용)
ROLLBACK	DB조작 작업이 비정상적으로 종료 되었을 때 원래의 상태로 복구함 (트랜잭션 취소→DB 적용 안됨)
GRANT	DB 사용자에게 사용 권한을 부여함
REVOKE	DB 사용자의 사용 권한을 취소함

COMMIT/ROLLBACK

* COMMIT : 트랜잭션의 성공했을 경우 그 결과를 DB에 적용하여 완료 시킴.
* ROLLBACK : 트랜잭션의 실패로 작업을 취소하고, 이전 상태로 되돌림.

확인 취소



* 인출과정 전체를 Transaction 이라고 한다. -> 작업의 논리적인 단위

* 모든 작업이 성공한 경우 Commit 을 해주고, 중간에 조금이라도 실수가 있었다면 Roll Back 을 하게 됩니다.

* 롤백을 하면 Transaction 을 하기 전까지의 상태로 돌릴 수가 있습니다.

Check 13. 시스템 카탈로그 [출제빈도: 上]

1. 시스템 카탈로그(= 데이터 사전) ★

-시스템 자신이 필요로 하는 여러 가지 객체에 관한 정보를 포함하고 있는 시스템 데이터베이스

-특징

①DB시스템에 따라 상이한 구조를 지님
②사용자도 SQL을 이용하여 검색가능
(DBMS만 스스로 갱신 유지할 수 있고, 사용자 갱신 안됨)
③객체들로서는 기본테이블, 뷰, 인텍스, 데이터베이스, 패키지, 접근 권한 등이 있다.
④데이터베이스 스키마에 대한 정보를 제공
⑤개체들에 대한 정의나 명세에 관한 정보를 유지 관리하는 시스템
⑥**데이터 디렉토리**: 데이터 사전에 수록된 데이터를 실제로 접근하는데 필요한 정보를 관리 유지하는 시스템만이 접근할 수 있는 구역

Check 14. 뷰(View) [출제빈도: 上]

1. 뷰(View) ★

-사용자에게 접근이 허용된 자료만을 제한적으로 보여주기 위해서 하나 이상의 기본 테이블로부터 유도된 가상 테이블 (물리적 X, 논리적 O)

-특징

① 구조가 기본테이블과 경의 유사
② 물리적으로 구현되지 않음
③ 논리적 독립성 제공
④ 필요한 데이터로만 구성 → 관리수월, 명령간단
⑤ 데이터보호 효율적 → 자동보안
⑥ 삽입,삭제,갱신 연산이 가능하지만 제한적
⑦ 다른 VIEW정의에 기초
⑧ 하나의 VIEW를 삭제→그 VIEW를 기초로 만들어진 VIEW도 자동 삭제
⑨ 독립적인 인덱스를 가질수 없음
⑩ 뷰에 대한 검색은 일반 테이블과는 같음
⑪ VIEW의 정의 변경 불가

Check 15. 내장 SQL [출제빈도: 上]

1. 내장 SQL (Embedded-SQL) ★★☆☆☆

-호스트 언어(C, C++, 비주얼 베이직 등)에 삽입된 SQL
-목적: 일괄처리, 동일업무 반복 시

```
EXEC SQL BEGIN DECLARE SECTION;
        char 물품번호;
        char 이름;
        int 단가;
EXEC SQL END DECLARE SECTION;
        물품번호='A001';
EXEC SQL SELECT 이름, 단가
        INTO :이름, :단가
        FROM Product
        WHERE 물품번호 = :물품번호;
END EXEC
```

***변수선언**

-테이블의 속성을 변수로 선언
-변수타입은 속성타입과 같아야 함

***삽입되는 SQL 문**

-INTO 절에는 검색 결과가 저장될 변수가 지정됨.

: 이름→ SQL 내 속성명과 구분하기 위해 호스트 변수를 이용할 경우 **' : ' 을 붙인다.**

내장 SQL 문장 끝은 호스트 언어의 종류에 따라 종료를 표시하는 방법이 다르다. (일반적으로 세미콜론 ;)

2. SQL CODE (SQL 상태) ★☆☆☆☆

-내장 SQL에서 SQLCODE(정수 타입으로 선언해야 한다.)라는 변수 사용시는 SQL 명령문이 실행되고 나서 **이 변수에 0값 SET 되면 성공적 실행 상태를** 나타내는 것이고, **1000l SET 되게 되면 경고, 음수가 SET되게 되면 에러**를 나타낸다.

Check 16. 트랜젝션(Transaction) [출제빈도: 上]

1. 정의

-데이터베이스의 상태를 변화시키는 논리적 연산의 집합

2. 트랜잭션 특징 ★★★★★

원자성 (Atomicity)	-모두 반영 되거나 아니면 전혀 반영되지 않아야 함 (부분 실행 안됨)
일관성 (Consistency)	-트랜잭션이 그 실행을 성공적으로 완료하면 언제나 일관성 있게 DB상태로 변환 -시스템이 가지고 있는 고정요소는 트랜잭션 수행 전과 트랜잭션 수행 완료 후에 같아야 한다.
독립성,격리성 (Isolation)	-둘 이상의 트랜잭션이 동시에 병행 실행되고 있을 때 또 다른 하나의 트랜잭션의 연산이 끼어들 수 없다.
영속성,지속성 (Durability)	-트랜잭션의 결과는 영구적으로 반영

3. 연산의 종류 ★★☆☆☆

Commit	-한 작업의 논리적 단위가 성공적으로 끝났고, DB가 다시 일관된 상태에 있으며 이 트랜잭션이 행한 갱신 연산이 완료된 것을 트랜잭션 관리자에게 알려주는 연산
Rollback	-트랜잭션의 실행이 실패하였음을 알리는 연산자로 트랜잭션이 수행한 결과를 원래의 상태로 완성 복귀시키는 연산

4. 회복 정의

-트랜잭션 수행도중 장애가 발생하여 DB가 손상 입었기에 손상되기 이전 상태로 복구하는 작업

5. 장애 유형 ★☆☆☆☆

1)트랜잭션 장애 2)시스템 장애 3) 미디어장애
→트랜잭션 장애가 DBdp 손상을 줄 가능성이 가장 적은 장애

Check 17. 보안, 암호화 [출제빈도: 下]

1. 개인키와 공개키 암호화 알고리즘 ★★☆☆☆

개인키 (Privata)	-DES(암호화키=복호화키) -동일한 키를 이용하는 방식→보안수준↓ →알고리즘이 단순하고 빠름
공개키 (Public)	-RSA(암호화키<>복호화키) -서로다른 키를 사용하는 비대칭 암호화 방식 →보안수준↑→속도가 느리고 알고리즘 복잡, 파일크기 大

암호화키 복호화키(해독키)

***보안과 무결성**

-무결성은 권한이 있는 사용자로부터 DB를 보호하는 것
-보안은 권한이 없는 사용자로부터 DB를 보호하는 것이다.

*보안을 위한 사용자들의 권한부여는 관리자의 정책에 의해 결정

Check 18. 병행제어 [출제빈도: 中]

1. 병행제어 정의

-동시에 여러 개 수행할 때, DB 일관성을 유지를 위해 트랜잭션 간의 상호 작용을 제어

2. 병행제어 목적 ★☆☆☆☆

1) DB 공유 최대화
2) 시스템 활용도 최대화
3) DB 일관성 유지
4) 사용자에 대한 응답시간 최소화

3. 병행제어 기법 ★★★☆☆

1) 로킹(Locking)

-하나의 트랜잭션이 데이터를 액세스하는 동안 다른 트랜잭션이 그 데이터 항목을 액세스할 수 없도록 하는 방법

2) 로킹 단위

-병행제어에서 한꺼번에 로킹 할 수 있는 단위
-로킹 단위가 大→로크수가 小→관리수월, 병행성 수준 ↓
-로킹 단위가 小→로크수가 大→관리복잡, 병행성 수준 ↑

4. 병행 수행 허용시 발생하는 문제점 ★☆☆☆☆

1) 갱신 분실: 2개 이상의 트랜잭션이 같은 자료를 갱신할 때 일부가 없어지는 현상
2) 비완료 의존성: 하나의 트랜잭션이 실패한 후 다른 트랜잭션이 실패한 갱신 결과를 참조하는 현상
3) 불일치: 원치 않는 자료를 이용하는 현상

Check 19. 분산 데이타베이스 [출제빈도: 中]

1. 분산 데이터베이스 정의

-컴퓨터 네트워크 상에 물리적으로 분산된 DB를 논리적으로는 1개로 인식하는 기법

2. 분산 데이터베이스 4대 목표 ★★★☆☆

1) 위치 투명성: 사용자가 물리적으로 저장되어 있는 곳을 알 필요 없이 논리적인 입장에서 데이터가 모두 자신의 사이트에 있는 것처럼 처리
2) 중복(복제) 투명성: 트랜잭션이 데이터의 중복 개수나 중복 사실을 모르고도 데이터 처리가 가능
3) 병행 투명성: 분산 DB와 관련된 다수의 트랜잭션들이 동시에 실현 되더라도 그 트랜잭션의 결과는 영향을 안 받음
4) 장애 투명성: 트랜잭션, DBMS, 네트워크, 컴퓨터 장애에도 불구 하고 트랜잭션을 정확하게 처리함

3. 분산 데이터베이스의 특징 ★★★★★

1) 자료공유 용이
2) 시스템 성능 향상
3) 점증적 시스템 용량 확장 용이
4) 설계가 어렵고 소프트웨어 개발 비용 증가
5) 오류발생 가능성 높음

Check 20. 자료구조(선형, 비선형구조) [출제빈도: 上]

1. 자료구조 분류 ★

-자료를 기억장치 내에 저장하는 방법

선형구조	순차리스트(스택, 큐, 데크, 배열), 연결리스트
비선형구조	Tree, Graph

2. 선형 구조 ★☆☆☆☆

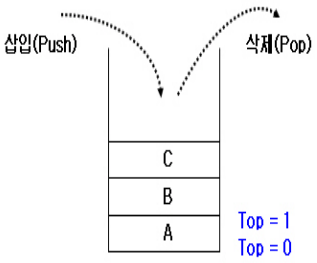
1) **순차리스트(선형, Sequential List)**: 연속적인 기억장소에 저장
-아파트 계단으로 연속적으로 이동
-**특징**: 구조간단, 기억장소 이용 효율↑, 삽입/삭제 어려움, 연결리스트에 비해 검색 빠름

2) **연결리스트(Linked List)**: 비연속적으로 저장
-아파트를 엘리베이터로 비연속적으로 이동(포인터)
-특징: 기억장소 이용 효율 낮음, 삽입/삭제 용이 순차리스트에 비해 검색 느림

3. 선형구조 - 순차리스트 종류

1) 스택(stack) ★★★★★

-삽입/삭제가 한쪽에서 이루어지는 데이터
(LIFO: Last In FIRST OUT)



*Top Point: 가장 최근 삽입된 자료 EH는 가장 먼저 삭제될 자룔를 가르키는 스택 포인터
-삽입: Top값 증가
-삭제: Top값 감소

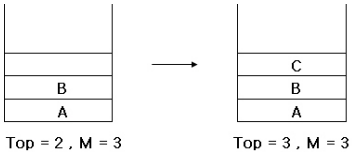
-스택 응용분야 ★

①인터럽트의 처리
②수식의 계산
③서브루틴의 복귀번지 저장
④부프로그램(sub p/g)의 호출+ 함수호출의순서제어

⑤운영체제의 작업스케줄링 (X)

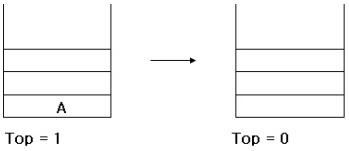
*삽입 알고리즘

```
Top = Top + 1
If(Top > M) Then
      Stack_overflow
Else
      Stack(Top) ← data
```



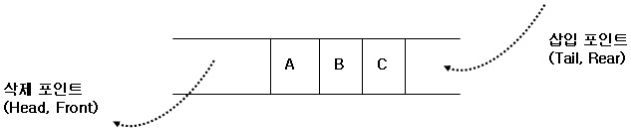
*삭제 알고리즘

```
If(Top = 0) Then
      Stack_Empty
Else
      data ← Stack(Top)
      Top = Top - 1
```



2) 큐(Queue) ★★★☆☆

-노드의 삽입 작업은 선형 리스트의 한 쪽 끝에서,
제거 작업은 다른 쪽 끝에서 수행되는 자료구조
(FIFO: First IN FIRST OUT)



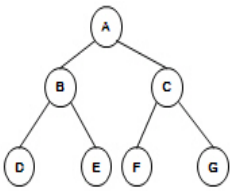
-응용분야

①운영체제의 작업 스케줄링
②키보드 버퍼 이용 시
③스풀(spool) 운용 시

4. 비선형구조 종류

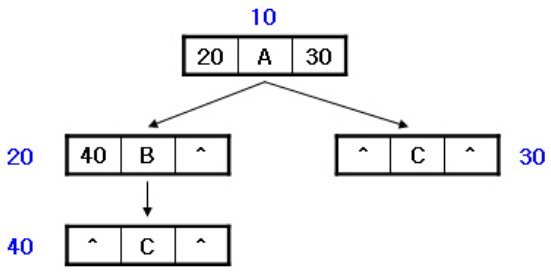
1) 트리(Tree) ★★★☆☆

-노드와 간선으로 구성되어져 있고, 사이클이 없다.
(족보, 부모자식 관계)



*** 용어**
-Node(노드)
-Root Node(근노드): 최상위 노드
-노드의 차수(Degree): 어떤 노드의 서브트리 수
-트리의 차수: 노드의 차수 값 중 최대값
-단말노드(Terminal, Leaf): 차수가 0인 노드
-형제노드: 같은 부모 노드를 가지는 노드
-레벨(Leavel): 노드의 깊이

*** 특징**: 연결 리스트 구조로 표현(포인터 이용)



***트리 종류**

① 이진트리 ★☆☆☆☆

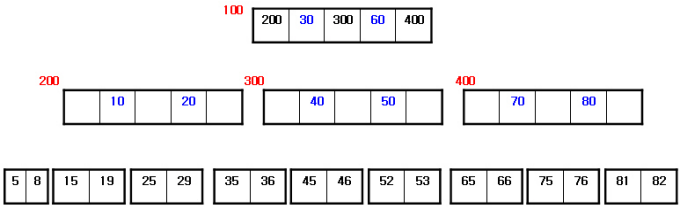
- 차수가 2 이하로 구성된 트리
- 이진 트리의 레벨 l에서 최대 노드 수: $2^i - 1$

② 스레드(threade) 이진 트리 ★☆☆☆☆

-기억 공간의 낭비 원인이 되는 널 링크 부분을 트리 순회 시 이용되도록 구성한 트리
-널 링크를 다른 노드를 가리키는 포인터로 대체한다.
-스택의 도움 없이 트리를 순회할 수 있는 장점이 있다.
-실제 포인터와 스레드를 구별하기가 어렵다.

③ B- 트리 ★☆☆☆☆

-인덱스 파일에서 인덱스를 구성하는 방법 중 하나
-한 노드 안에 있는 키 값은 오름차순을 유지한다.
-모든 리프노드(단노드)는 같은 레벨에 있다. (균형유지)
-루트노드는 리프가 아닌 이상 적어도 두 개의 서브트리를 갖는다.
-탐색, 추가, 삭제는 푸트로부터 시작한다.
-인덱스 파일에서 인덱스를 구성하는 방법 중 하나다.



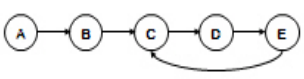
2) 이진 트리 운행법 ★★★★☆

-트리를 구성하는 노드들을 찾아가는 방법

Preorder (전위)	Root→Left→Right
Inorder (중위)	Left→Root→Right
Postorder(후위)	Left→Right→Root

3) Graph ★★☆☆☆

-노드와 간선으로 구성되어져 있고, 사이클이 있다.
-그래프를 인접행렬로 표시



	A	B	C	D	E
A	0	1	0	0	0
B	0	0	1	0	0
C	0	0	0	1	0
D	0	0	0	0	1
E	0	0	1	0	0

5. 수식 표기법 변환 ★★★★☆

-산술을 계산하기 위해 기억공간에 기억시키는 방법
-이진트리를 많이 사용

PreFix (전위)	연산자→Left피연상자→Right피연산자
InFix (중위)	Left피연산자연산자→Right피연산자
PostFix(후위)	Left피연산자→Right피연산자→연산자

Check 21. 자료구조-정렬 [출제빈도: 下]

1. 정렬 ★★★☆☆

-오름차순 또는 내림차순으로 데이터를 나열함

내부정렬 (주기억장치 사용)	선택정렬(select sort), 버블정렬(bubble sort), 삽입정렬(shell sort), 힙정렬 (heap sort)
	쉘 정렬(shell sort), 퀵 정렬(quick sort), 2-way 병합 정렬(2-way merge sort), 기수정렬(radix sort)
외부정렬 (보조기억장치 사용)	벨런스드 병합정렬(Balanced merge sort), 캐스캐이드 병합정렬(CAscade merge sort),
	폴리파즈 병합정렬(Polyphase merge sort), 오실레이팅 병합정렬(Oscillation merge sort)

* 병합정렬이라는 말을 빼고, 보기를 주는 경우가 있음

2. 정렬 알고리즘 선택 시 고려사항 ★☆☆☆☆

1) 데이터의 양
2) 초기 데이터의 배열상태
3) 키 값들의 분포상태
4) 소요공간 및 작업시간
5)운영체제의 종류, 액세스 빈도, 증가 데이터의 배열상태 (x)

*선택정렬, 버블정렬, 삽입정렬, 2-way 병합정렬 방법 교재 참고 (DB 9강)

Check 22. 자료구조-검색 [출제빈도: 下]

1. 검색

- 원하는 데이터를 찾음

1) 선형 검색(순차검색)

-모든 레코드들을 대상으로 순차적 검색, 자료가 정렬되지 않을 때, 처리속도가 느리다.

2) 이분(이진)검색 ★★☆☆☆

-자료가 정렬되어 있어야 함, 중간값을 비교 검색, 많은 레코드 검색 시 효율적

10	15	31	45	62	63	74	87	91	92

3) 보간 검색: 찾고자 하는 레코드 키가 있음직한 위치를 추정

4) 트리검색, 블록검색, 피보나치검색

2. 해싱검색(hasing) 검색 ★★★★☆

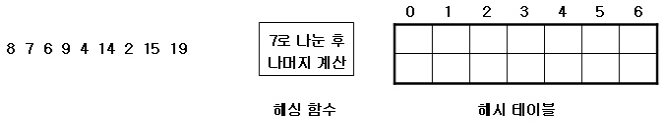
-해싱 함수를 이용하여 자료를 검색하는 방법.
-데이터를 해시 테이블이라는 배열에 저장하고 해싱함수를 이용하여 데이터가 위치한 곳으로 주소를 찾기 때문에 신속하게 원하는 자료를 검색할 수 있는 키-주소 변환방법

1) 특징

-DAM(직접접근, Direct Access Method) 파일을 구성할 때 해싱이 사용
-삽입, 삭제 작업의 빈도가 많을 때 유리한 방식
-검색은 가장 빠르지만 기억공간의 낭비 발생

2) 용어정리

-해싱함수: 해시테이블의 주소를 생성해 내는 함수
-해시 테이블: 해싱 함수에 의하여 참조되는 테이블
-버킷(bucket): 하나의 주소를 갖는 파일의 한 구역
-슬롯(slot): n개의 슬롯이 모여 하나의 버킷을 형성
-충돌(collision): 서로 다른 2개 이상의 레코드가 같은 주소를 갖는 현상
-시노잉(synonym): 같은 주소를 갖는 레코드의 집합
-오버플로: 베켓 냉에 기억 공간이 없는 현상



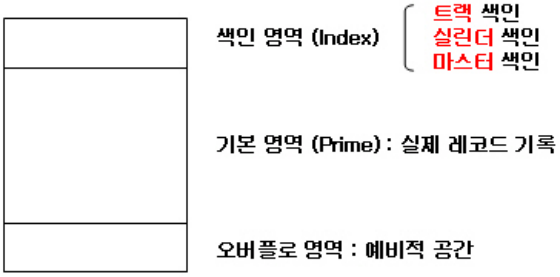
Check 23. 순차파일(SAM) [출제빈도: 下]

1. 순차파일 (SAM: Sequential Access Method) : 목차 없는 책 ★☆☆☆☆

-파일 내의 각 레코드를 논리적 순서에 따라 물리적으로 연속된 위치에 기록한 파일
-기억장소의 낭비가 없다.
-색인순차파일에 비해 삽입, 삭제, 검색이 어렵다.

2. 색인순차파일 (ISAM: Index Sequential Access Method) : 목차 있는 책(정적 인덱스) ★★★☆☆

-인덱스를 통한 랜덤 처리와 데이터의 순차 처리를 병행할 수 있는 파일.
-삽입, 삭제, 갱신, 검색 용이
-삽입 시 기본영역에 추가 공간이 없을 경우 오버플로 영역에 저장
-재사용이 안되므로 삽입, 삭제가 빈번할 경우 기억공간 낭비 발생하므로 재구성이 필요



3. 직접파일 (DAM : Direct Access Method) ★☆☆☆☆

-해싱 함수를 계산해서 물리적 주소를 직접 접근
(대화형 처리 가능)
-순서에 관계없이 저장
-레코드 주소의 변환과정의 시간소요
-기억공간 효율 저하

4. VSAM (Cirtual Sequential Access Method) : 동적인덱스 ★☆☆☆☆

-동적 인텍스 방법을 이용한 색인 순차 파일
-기본 구역과 오버플로우 구역을 구분하지 않음
(기본 구역 내에 예비 공간을 두어 추가로 삽입될 경우 이용)
-레코드를 삭제하면 그 공간을 재사용할 수 있음
(정적 인덱스는 재사용이 안됨)
-제어구간에 가변길이 레코드를 쉽게 수용할 수 있음

5. 역파일 : 찾아보기 기능 ★☆☆☆☆

-특정파일을 여러개의 색인으로 만들어 항목별 특성에 맞게 작업하도록 구성
-질의으답 시간이 단축되고 처리가 쉽다.
-색인의 각 항목의 길이가 가변

6. 인덱스 ★☆☆☆☆

-인덱스를 통해서 테이블의 레코드에 대한 액세스를 빠르게 수행할 수 있다.
-인덱슨느 하나 이상의 필드로 만들어도 된다.
-레코드의 삽입과 삭제가 수시로 일어나는 경우는 인덱스를 최대화 한다.