

# 3과목-데이터베이스 구축

(Part 2. 물리 데이터베이스 설계 - Ⅱ)

# 데이터베이스 구축 총 파트

---

데이터베이스 구축 3과목은 총 5Part로 이루어져 있다.

1장 논리 데이터베이스 설계(49.72%)

**2장 물리 데이터베이스 설계(22.91%)**

3장 SQL 응용(26.26%)

4장 SQL 활용(1.12%)

5장 데이터 전환(0.00%)

# 물리 데이터베이스 설계

물리 데이터베이스 설계 Part는 16개의 섹션으로 구성되어 있다.

01 사전 조사분석

02 데이터베이스 저장공간설계

03 트랜잭션 분석 / CRUD 분석

04 인덱스 설계

05 뷰(View) 설계

06 클러스터 설계

07 파티션 설계

08 데이터베이스 용량 설계

09 분산 데이터베이스 설계

10 데이터베이스 이중화 / 서버 클러스터링

11 데이터베이스 보안 / 암호화

12 데이터베이스 보안 - 접근통제

13 데이터베이스 백업

14 스토리지

15 논리 데이터 모델의 물리 데이터 모델 변환

16 물리 데이터 모델 품질 검토

## 2. 물리 데이터베이스 설계-SEC\_09(분산 데이터베이스 설계)

### 1) 분산 데이터베이스 정의

; 분산 데이터베이스는 논리적으로는 하나의 시스템에 속하지만 물리적으로는 네트워크를 통해 연결된 여러 개의 컴퓨터 사이트(Site)에 분산되어 있는 데이터베이스를 말한다.

- 분산 데이터베이스는 데이터의 처리나 이용이 많은 지역에 데이터베이스를 위치시킴으로써 데이터의 처리가 가능한 해당 지역에서 해결될 수 있도록 한다.

### 2) 분산 데이터베이스 구성 요소

분산 처리기	자체적으로 처리 능력을 가지며, 지리적으로 분산되어 있는 컴퓨터 시스템을 말한다.
분산 데이터베이스	지리적으로 분산되어 있는 데이터베이스로서 해당 지역의 특성에 맞게 데이터베이스가 구성된다.
통신 네트워크	분산 처리기들을 통신망으로 연결하여 논리적으로 하나의 시스템처럼 작동할 수 있도록 하는 통신 네트워크를 말한다.

## 2. 물리 데이터베이스 설계-SEC\_09(분산 데이터베이스 설계)

### 3) 분산 데이터베이스 설계 시 고려 사항

- 작업 부하(Work Load)의 노드별 분산 정책
- 지역의 자치성 보장 정책
- 데이터의 일관성 정책
- 사이트나 회선의 고장으로부터의 회복 기능
- 통신 네트워크를 통한 원격 접근 기능

### 4) 분산 데이터베이스의 목표

- **위치 투명성(Location Transparency)** : 액세스하려는 데이터베이스의 실제 위치를 알 필요 없이 단지 데이터베이스의 논리적인 명칭만으로 액세스할 수 있다.
- **중복 투명성(Replication Transparency)** : 동일 데이터가 여러 곳에 중복되어 있더라도 사용자는 마치 하나의 데이터만 존재하는 것처럼 사용하고, 시스템은 자동으로 여러 자료에 대한 작업을 수행한다.
- **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.

## 2. 물리 데이터베이스 설계-SEC\_09(분산 데이터베이스 설계)

### 5) 분산 데이터베이스의 장, 단점

장점	단점
<ul style="list-style-type: none"><li>•지역 자치성이 높다.</li><li>•자료의 공유성이 향상된다.</li><li>•분산 제어가 가능하다.</li><li>•시스템 성능이 향상된다.</li><li>•중앙 컴퓨터의 장애가 전체 시스템에 영향을 끼치지 않는다.</li><li>•효용성과 융통성이 높다.</li><li>•신뢰성 및 가용성이 높다.</li><li>•점진적 시스템 용량 확장이 용이하다.</li></ul>	<ul style="list-style-type: none"><li>•DBMS가 수행할 기능이 복잡하다.</li><li>•데이터베이스 설계가 어렵다.</li><li>•소프트웨어 개발 비용이 증가한다.</li><li>•처리 비용이 증가한다.</li><li>•잠재적 오류가 증가한다.</li></ul>

### 6) 분산 데이터베이스 설계

; 분산 데이터베이스 설계는 애플리케이션이나 사용자가 분산되어 저장된 데이터에 접근하게 하는 것을 목적으로 한다.

- 잘못 설계된 분산 데이터베이스는 복잡성 증가, 응답 속도 저하, 비용 증가 등의 문제가 발생한다.
- 분산 데이터베이스의 설계는 전역 관계망을 논리적 측면에서 소규모 단위로 분할한 후, 분할된 결과를 복수의 노드에 할당하는 과정으로 진행된다. 노드에 할당된 소규모 단위를 분할(Fragment)이라 부른다.
- 분산 설계 방법에는 테이블 위치 분산, 분할(Fragmentation), 할당(Allocation)이 있다.

## 2. 물리 데이터베이스 설계-SEC\_09(분산 데이터베이스 설계)

### 7) 테이블 위치 분산

; 테이블 위치 분산은 데이터베이스의 테이블을 각기 다른 서버에 분산시켜 배치하는 방법을 의미한다.

- 테이블 위치를 분산할 때는 테이블의 구조를 변경하지 않으며, 다른 데이터베이스의 테이블과 중복되지 않게 배치한다.
- 데이터베이스의 테이블을 각각 다른 위치에 배치하려면 해당 테이블들이 놓일 서버들을 미리 설정해야 한다.

예) 테이블 분산 위치

테이블 위치	사원	부서	상품	판매현황	거래업체
본사	○	○			○
지사			○	○	

## 2. 물리 데이터베이스 설계-SEC\_09(분산 데이터베이스 설계)

### 8) 분할(Fragment)

; 분할은 테이블의 데이터를 분할하여 분산시키는 것이다.

#### ● 분할 규칙

- 완전성(Completeness) : 전체 데이터를 대상으로 분할해야 한다.
- 재구성(Reconstruction) : 분할된 데이터는 관계 연산을 활용하여 본래의 데이터로 재구성할 수 있어야 한다.
- 상호 중첩 배제(Dis-jointness) : 분할된 데이터는 서로 다른 분할의 항목에 속하지 않아야 한다.  
예) 수평 분할한 경우 각각의 분할에 포함된 튜플들이 상호 중복되지 않아야 하고, 수직 분할한 경우는 각각의 분할에 포함된 속성들이 중복되지 않아야 한다.

#### ● 주요 분할 방법

- 수평 분할 : 특정 속성의 값을 기준으로 행(Row) 단위로 분할
- 수직 분할 : 데이터 컬럼(속성) 단위로 분할



## 2. 물리 데이터베이스 설계-SEC\_09(분산 데이터베이스 설계)

### 9) 할당(Allocation)

; 할당은 동일한 분할을 여러 개의 서버에 생성하는 분산 방법으로, 중복이 없는 할당(Allocation)과 중복이 있는 할당(Allocation)으로 나뉜다.

- 비중복 할당 방식

- 최적의 노드를 선택해서 분산 데이터베이스의 단일 노드에서만 분할이 존재하도록 하는 방식이다.
- 일반적으로 애플리케이션에는 릴레이션을 배타적 분할로 분리하기 힘든 요구가 포함되므로 분할된 테이블 간의 의존성은 무시되고 비용 증가, 성능 저하 등의 문제가 발생할 수 있다.

- 중복 할당 방식 : 동일한 테이블을 다른 서버에 복제하는 방식으로, 일부만 복제하는 부분 복제와 전체를 복제하는 완전 복제가 있다.

# 물리 데이터베이스 설계-SEC\_09(분산 데이터베이스 설계) 기출 및 출제 예상 문제

## 기출 및 출제 예상 문제(분산 데이터베이스 설계)

### 1. 분산 데이터베이스 시스템과 관련한 설명으로 틀린 것은?

- ① 물리적으로 분산된 데이터베이스 시스템을 논리적으로 하나의 데이터베이스 시스템처럼 사용할 수 있도록 한 것이다.
- ② 물리적으로 분산되어 지역별로 필요한 데이터를 처리할 수 있는 지역 컴퓨터(Local Computer)를 분산 처리기(Distributed Processor)라고 한다.
- ③ 분산 데이터베이스 시스템을 위한 통신 네트워크 구조가 데이터 통신에 영향을 주므로 효율적으로 설계해야 한다.
- ④ 데이터베이스가 분산되어 있음을 사용자가 인식할 수 있도록 분산 투명성(Distribution Transparency)을 배제해야 한다.

**분산 데이터베이스**는 논리적으로 하나의 시스템에 속하지만 물리적으로는 네트워크를 통해 연결된 여러 개의 컴퓨터 사이트에 분산되어 있는 데이터 베이스를 말한다.

- 분산 데이터베이스는 데이터의 처리나 이용이 많은 지역에 데이터베이스를 위치시킴으로써 데이터의 처리가 가능한 해당 지역에서 해결될 수 있도록 한다.

### 분산 처리기

### 3. 분산 데이터베이스의 특징에 대한 설명으로 틀린 것은?

- ① 지역 서버의 고유 데이터에 대한 작업은 중앙 서버의 통제 없이 자유롭게 수행할 수 있다.
- ② 새로운 지역 서버를 추가하거나 장비를 추가하는 등의 작업이 용이하다.
- ③ 위치 투명성, 중복 투명성, 병행 투명성, 장애 투명성을 목표로 한다.
- ④ 데이터베이스 설계 및 소프트웨어 개발이 쉽고, 전반적인 시스템의 성능이 향상된다.

### 장점

- 1. 지역 자치성이 높다.
- 2. 자료의 공유성이 향상된다.
- 3. 분산 제어가 가능하다.
- 4. 시스템 성능이 향상된다.
- 5. 중앙 컴퓨터의 장애가 전체 시스템에 영향을 끼치지 아니한다.
- 6. 효율성과 융통성이 높다.
- 7. 신뢰성 및 가용성이 높다.
- 8. 점진적 시스템 용량 확장이 용이하다.

### 단점

# 물리 데이터베이스 설계-SEC\_09(분산 데이터베이스 설계) 기출 및 출제 예상 문제

## 기출 및 출제 예상 문제(분산 데이터베이스 설계)

5. 분산 데이터베이스 목표 중 "데이터베이스의 분산된 물리적 환경에서 특정 지역의 컴퓨터 시스템이나 네트워크에 장애가 발생해도 데이터 무결성이 보장된다."는 것과 관계 있는 것은?

- ① 장애 투명성                      ② 병행 투명성
- ③ 위치 투명성                      ④ 중복 투명성

6. 분산 데이터베이스의 장점으로 거리가 먼 것은?

- ① 데이터베이스의 설계와 개발이 용이
- ② 신뢰성(Reliability)과 가용성(Availability) 향상
- ③ 질의 처리(Query Processing) 시간의 단축
- ④ 데이터의 공유성 향상

분산 데이터베이스는 여러 곳에 분산되어 있는 데이터베이스를 하나의 데이터베이스처럼 사용할 수 있도록 만들어야 하기 때문에 중앙 집중식에 비해 설계와 개발이 어렵고 개발 비용이 많이 든다.

7. 다음 중 분산 데이터베이스 시스템의 목적이라 볼 수 없는 것은?

- ① 생산성의 향상
- ② 지역적 정보처리의 효율성 증진
- ③ 보안성 확보
- ④ 신뢰도와 가용성 증진

## 분산 데이터베이스의 목적

1. 원격 자원과 데이터를 사용함으로써 사용자의 생산성 향상과 시스템 가용성 증대.
2. 지역적 정보 처리의 효율성 증진
3. 시스템의 확장성과 변경이 용이하다.
4. 신뢰성 증진

## 2. 물리 데이터베이스 설계-SEC\_10(데이터베이스 이중화/서버 클러스터링)

### 1) 데이터베이스 이중화(Database Replication)

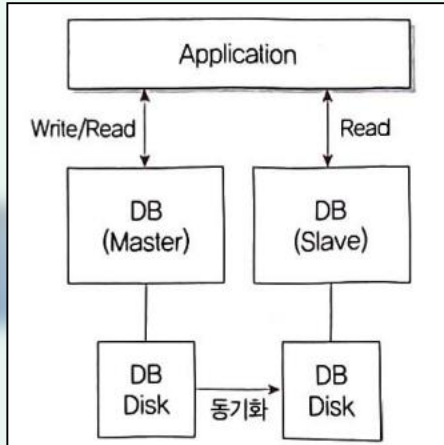
; 데이터베이스 이중화는 시스템 오류로 인한 데이터베이스 서비스 중단이나 물리적 손상 발생 시 이를 복구하기 위해 동일한 데이터베이스를 복제하여 관리하는 것이다.

- 데이터베이스 이중화를 수행하면 하나 이상의 데이터베이스가 항상 같은 상태를 유지하므로 데이터베이스에 문제가 발생하면 복제된 데이터베이스를 이용하여 즉시 문제를 해결할 수 있다.
- 데이터베이스 이중화는 여러 개의 데이터베이스를 동시에 관리하므로 사용자가 수행하는 작업이 데이터베이스 이중화 시스템에 연결된 다른 데이터베이스에도 동일하게 적용된다.
- 데이터베이스 이중화는 애플리케이션을 여러 개의 데이터베이스로 분산시켜 처리하므로 데이터베이스의 부하를 줄일 수 있다.
- 데이터베이스 이중화를 이용하면 손쉽게 백업 서버를 운영할 수 있다.

## 2. 물리 데이터베이스 설계-SEC\_10(데이터베이스 이중화/서버 클러스터링)

### 1) 데이터베이스 이중화(Database Replication)

예) MySQL의 이중화 방식



- 동일한 데이터베이스를 복제하여 관리하되, 데이터를 읽고 쓸 수 있는 마스터 DB와 읽기만 가능한 슬레이브 DB로 구분하여 사용한다.
- 슬레이브 DB는 마스터 DB의 로그를 이용하여 변경된 데이터를 동기화한다.

## 2. 물리 데이터베이스 설계-SEC\_10(데이터베이스 이중화/서버 클러스터링)

### 2) 데이터베이스 이중화의 분류

; 데이터베이스 이중화는 변경 내용의 전달 방식에 따라 Eager 기법과 Lazy 기법으로 나뉜다.

<b>Eager 기법</b>	트랜잭션 수행 중 데이터 변경이 발생하면 이중화된 모든 데이터베이스에 즉시 전달하여 변경 내용이 즉시 적용되도록 하는 기법
<b>Lazy 기법</b>	트랜잭션의 수행이 종료되면 변경 사실을 새로운 트랜잭션에 작성하여 각 데이터베이스에 전달되는 기법으로 데이터베이스마다 새로운 트랜잭션이 수행되는 것으로 간주된다.

### 3) 데이터베이스 이중화 구성 방법

; 데이터베이스의 이중화 구성 방법에는 활동-대기(Active-Standby) 방법과 활동-활동(Active-Active) 방법이 있다.

<b>활동-대기(Active-Standby) 방법</b>	<ul style="list-style-type: none"><li>•한 DB가 활성 상태로 서비스하고 있으면 다른 DB는 대기하고 있다가 활성 DB에 장애가 발생하면 대기 상태에 있던 DB가 자동으로 모든 서비스를 대신 수행한다.</li><li>•구성 방법과 관리가 쉬워 많은 기업에서 이용된다.</li></ul>
<b>활동 활동(Active-Active) 방법</b>	<ul style="list-style-type: none"><li>•두 개의 DB가 서로 다른 서비스를 제공하다가 둘 중 한쪽 DB에 문제가 발생하면 나머지 다른 DB가 서비스를 제공한다.</li><li>•두 DB가 모두 처리를 하기 때문에 처리율이 높지만 구성 방법 및 설정이 복잡하다.</li></ul>

## 2. 물리 데이터베이스 설계-SEC\_10(데이터베이스 이중화/서버 클러스터링)

### 4) 클러스터링(Clustering)

; 클러스터링은 두 대 이상의 서버를 하나의 서버처럼 운영하는 기술이다.

- 클러스터링은 서버 이중화 및 공유 스토리지를 사용하여 서버의 고가용성을 제공한다.
- 클러스터링에는 고가용성 클러스터링과 병렬 처리 클러스터링이 있다.
  - 고가용성 클러스터링 : 하나의 서버에 장애가 발생하면 다른 노드(서버)가 받아 처리하여 서비스 중단을 방지하는 방식으로, 일반적으로 언급되는 클러스터링이 고가용성 클러스터링이다.
  - 병렬 처리 클러스터링 : 전체 처리율을 높이기 위해 하나의 작업을 여러 개의 서버에서 분산하여 처리하는 방식이다.

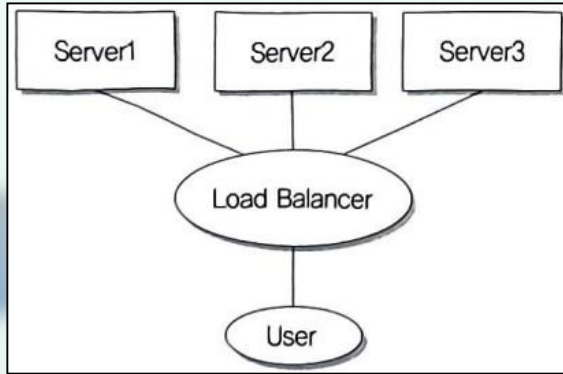
공유 스토리지(NAS; Network Attached Storage) : 공유 스토리지는 데이터 저장소를 네트워크로 연결하여 파일 및 데이터를 공유하는 것으로 다수의 사용자 또는 서버가 데이터를 안전하고 편리하게 공유할 수 있다.

고가용성(HA, High Availability) : 고가용성은 시스템을 오랜 시간 동안 계속해서 정상적으로 운영이 가능한 성질을 의미한다.

## 2. 물리 데이터베이스 설계-SEC\_10(데이터베이스 이중화/서버 클러스터링)

### 4) 클러스터링(Clustering)

예) 병렬 처리 클러스터링



- 사용자의 요청을 로드 밸런서(Load Balancer)가 여러 대의 서버로 분산한다.

로드 밸런서(Load Balancer) : 로드 밸런서는 특정 서버에 집중되는 부하를 덜기 위해 여러 개의 서버로 부하를 분산시키는 네트워크 서비스이다.



# 물리 데이터베이스 설계-SEC\_10(데이터베이스 이중화/서버 클러스터링) 출제 예상 문제

## 출제 예상 문제(데이터베이스 이중화/서버 클러스터링)

1. 다음 중 데이터베이스 이중화에 대한 설명으로 틀린 것은?

- ① 데이터베이스 이중화는 동일한 데이터베이스를 복제하여 관리한다.
- ② 데이터베이스 이중화가 시행 중일 때 사용자가 하나의 데이터베이스를 수정하면 복제된 데이터베이스에도 동일하게 적용된다.
- ③ 데이터베이스 이중화가 시행 중일 때 데이터베이스에 문제가 발생하면 복제된 데이터베이스를 이용하여 서비스를 계속해서 제공할 수 있다.
- ④ 데이터 변경이 발생하면 즉시 이중화된 데이터베이스에 전달 되어 변경 내용이 즉시 적용되는 기법은 Lazy 기법이다.

### Eager 기법

트랜잭션 수행 중 데이터 변경이 발생하면 이중화된 모든 데이터베이스에 즉시 전달하여 변경 내용이 즉시 적용되도록 하는 기법

### Lazy 기법

트랜잭션의 수행이 종료되면 변경 사실을 새로운 트랜잭션에 작성하여 각 데이터베이스에 전달되는 기법으로 데이터베이스마다

3. 다음 중 이중화와 클러스터링에 대한 설명으로 틀린 것은?

- ① 이중화와 클러스터링은 모두 고가용성을 위한 방법이다.
- ② 이중화의 활동-활동(Active-Active) 방법은 두 DB가 서로 다른 서비스를 제공하다가 둘 중 한쪽 DB에 문제가 발생하면 나머지 다른 DB가 서비스를 대신 수행한다.
- ③ 고가용성 클러스터링은 전체 처리율을 높이기 위해 하나의 작업을 여러 개의 서버에서 분산하여 처리한다.
- ④ 클러스터링은 두 대 이상의 서버를 하나의 서버처럼 운영하기 위한 기술이다.

데이터베이스 이중화는 시스템 오류로 인한 데이터베이스 서비스 중단이나 물리적 손상 발생 시 이를 복구하기 위해 동일한 데이터베이스를 복제하여 관리하는 것이다.

- 데이터베이스 이중화를 수행하면 하나 이상의 데이터베이스가 항상 같은 상태를 유지하므로 데이터베이스에 문제가 발생하면 복제된 데이터베이스를 이용하여 즉시 문제를 해결할 수가 있다.

- 데이터베이스 이중화는 여러 개의 데이터베이스를 동시에 관리하므로 사용자가 수행하는 작업이 데이터베이스 이중화 시스템에 연결된 다른 데이터베이스에도 동일하게 적용된다.

## 2. 물리 데이터베이스 설계-SEC\_11(데이터베이스 보안 / 암호화)

### 1) 데이터베이스 보안의 개요

; 데이터베이스 보안이란 데이터베이스의 일부분 또는 전체에 대해서 권한이 없는 사용자가 액세스하는 것을 금지하기 위해 사용되는 기술이다.

- 보안을 위한 데이터 단위는 테이블 전체로부터 특정 테이블의 특정한 행과 열 위치에 있는 특정한 데이터 값에 이르기까지 다양하다.
- 데이터베이스 사용자들은 일반적으로 서로 다른 객체에 대하여 다른 접근 권리 또는 권한을 갖게 된다.

### 2) 암호화(Encryption)

; 암호화는 데이터를 보낼 때 송신자가 지정한 수신자 이외에는 그 내용을 알 수 없도록 평문을 암호문으로 변환하는 것이다.

- 암호화(Encryption) 과정 : 암호화되지 않은 평문을 정보 보호를 위해 암호문으로 바꾸는 과정
- 복호화(Decryption) 과정 : 암호문을 원래의 평문으로 바꾸는 과정
- 암호화 기법에는 개인키 암호 방식과 공개키 암호 방식이 있다.

무결성(Integrity)과 보안(Security)

- 무결성은 권한이 있는 사용자로부터 데이터베이스를 보호하는 것이고, 보안은 권한이 없는 사용자로부터 데이터베이스를 보호하는 것이다.
- 보안은 데이터베이스 사용자들이 데이터베이스를 사용하고자 할 때 언제든지 사용할 수 있도록 보장하는 것이고, 무결성은 정확하게 사용할 수 있도록 보장하는 것이다.

## 2. 물리 데이터베이스 설계-SEC\_11(데이터베이스 보안 / 암호화)

### 3) 개인키 암호 방식(Private Key Encryption) = 비밀키 암호 방식

; 비밀키 암호화 기법은 동일한 키로 데이터를 암호화하고 복호화 한다.

- 비밀키 암호화 기법은 대칭 암호 방식 또는 단일키 암호화 기법이라고도 한다.
- 비밀키는 제3자에게는 노출시키지 않고 데이터베이스 사용 권한이 있는 사용자만 나누어 가진다.
- 종류: 전위 기법, 대체 기법, 대수 기법, 합성 기법(DES, LUCIFER)

### 4) 공개키 암호 방식(Public Key Encryption)

; 공개키 암호화 기법은 서로 다른 키로 데이터를 암호화하고 복호화 한다.

- 데이터를 암호화할 때 사용하는 키(공개키, Public Key)는 데이터베이스 사용자에게 공개하고, 복호화 할 때의 키(비밀키, Secret Key)는 관리자가 비밀리에 관리하는 방법이다.
- 공개키 암호화 기법은 비대칭 암호 방식이라고도 하며, 대표적으로 RSA(Rivest Shamir Adleman)가 있다.

DES(Data Encryption Standard) 기법 : DES 기법은 개인키 암호 방식의 대표적인 알고리즘으로 64Bit의 평문 블록을 56Bit의 16개 키를 이용하여 16회의 암호 계산 단계를 거쳐 64Bit의 암호문을 얻습니다.

루시퍼(Lucifer)는 민간에서 개발한 최초의 블록 암호들에 붙은 이름으로, 1970년대 IBM 소속의 호르스트 파이스텔 등에 의해 제작되었다.

RSA 암호는 공개키 암호 시스템의 하나로, 암호화뿐만 아니라 전자서명이 가능한 최초의 알고리즘으로 알려져 있다. RSA가 갖는 전자서명 기능은 인증을 요구하는 전자 상거래 등에 RSA의 광범위한 활용을 가능하게 하였다.

# 물리 데이터베이스 설계-SEC\_11(데이터베이스 보안 / 암호화) 기출 및 출제 예상 문제

## 기출 및 출제 예상 문제(데이터베이스 보안 / 암호화)

1. 암호화 기법 중 암호화 알고리즘과 암호화 키는 공개해서 누구든지 평문을 암호문으로 만들 수 있지만, 해독 알고리즘과 해독시키는 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

## 개인키 암호 방식(Private Key Encryption) = 비밀키 암호 방식

비밀키 암호화 기법은 동일한 키로 데이터를 암호화하고 복호화 한다.

- 비밀키 암호화 기법은 대칭 암호 방식 또는 단일키 암호화 기법이라고도 한다.

- 비밀키는 제 3자에게 노출시키지 않고 데이터베이스 사용 권한이 있는 사용자만 나누어 가진다.

- 종류 : 전위 기법, 대체 기법, 대수 기법, 합성 기법(DES, LUCIFER)

**DES(Data Encryption Standard)** : DES 기법은 개인키 암호 방식의 대표적인 알고리즘으로 64Bit의 평문 블록을 56Bit의 16회의 암호 계산 단계를 거쳐 64Bit의 암호문을 얻는 기법이다.

## 3. 정보보호를 위한 암호화에 대한 설명으로 틀린 것은?

- ① 평문 - 암호화되기 전의 원본 메시지
- ② 암호문 - 암호화가 적용된 메시지
- ③ 복호화 - 평문을 암호문으로 바꾸는 작업
- ④ 키(Key) - 적절한 암호화를 위하여 사용하는 값

암호화는 데이터를 보낼 때 송신자가 지정한 수신자 이외에는 그 내용을 알 수 없도록 평문을 암호문으로 변환하는 것이다.

- 암호화 과정 : 암호화 되지 않은 평문을 정보 보호를 위해 암호문으로 바꾸는 과정

- 복호화 과정 : 암호문을 원래의 평문으로 바꾸는 과정

## 4. 데이터베이스의 보안 기법이 아닌 것은?

- ① 비밀번호                      ② 권한 부여
- ③ 데이터 암호화                ④ 시스템 체크 포인트

## 보안 기법의 종류

대조 확인 : 비밀번호, 키, 목소리나 지문으로 대조 확인

권한 부여 규정 : 사용자 프로필의 유지

암호화 : 암호키와 해독키를 사용

시스템 체크 포인트는 장애 발생 시 회복을 위해 갱신 내용이나 시스템

## 2. 물리 데이터베이스 설계-SEC\_12(데이터베이스 보안 / 접근통제)

### 1) 접근 통제

; 접근 통제는 데이터가 저장된 객체와 이를 사용하려는 주체 사이의 정보 흐름을 제한하는 것이다.

- 접근 통제는 데이터에 대해 다음과 같은 통제를 함으로써 자원의 불법적인 접근 및 파괴를 예방한다.
  - 비인가된 사용자의 접근 감시
  - 접근 요구자의 사용자 식별
  - 접근 요구의 정당성 확인 및 기록
  - 보안 정책에 근거한 접근의 승인 및 거부 등
- 접근 통제 기술에는 임의 접근통제(DAC), 강제 접근통제(MAC), 역할기반 접근 통제(RBAC)가 있다.

## 2. 물리 데이터베이스 설계-SEC\_12(데이터베이스 보안 / 접근통제)

### 1) 접근 통제

<b>임의 접근 통제(DAC; Discretionary Access Control)</b>	<ul style="list-style-type: none"><li>•임의 접근통제는 데이터에 접근하는 사용자의 신원에 따라 접근 권한을 부여하는 방식이다.</li><li>•데이터 소유자가 접근 통제 권한을 지정하고 제어한다.</li><li>•객체를 생성한 사용자가 생성된 객체에 대한 모든 권한을 부여 받고, 부여된 권한을 다른 사용자에게 허가할 수도 있다.</li><li>•임의 접근 통제에 사용되는 SQL 명령어에는 GRANT와 REVOKE가 있다.</li></ul>
<b>강제 접근통제(MAC; Mandatory Access Control)</b>	<ul style="list-style-type: none"><li>•강제 접근 통제는 주체와 객체의 등급을 비교하여 접근 권한을 부여하는 방식이다.</li><li>•시스템이 접근 통제 권한을 지정한다.</li><li>•데이터베이스 객체별로 보안 등급을 부여할 수 있고, 사용자 별로 인가 등급을 부여할 수 있다.</li><li>•주체는 자신보다 보안 등급이 높은 객체에 대해 읽기, 수정, 등록이 모두 불가능하고, 보안 등급이 같은 객체에 대해서는 읽기, 수정, 등록이 가능하고, 보안 등급이 낮은 객체는 읽기가 가능하다.</li></ul>
<b>역할기반 접근통제 (RBAC; Role Based Access Control)</b>	<ul style="list-style-type: none"><li>•역할 기반 접근 통제는 사용자의 역할에 따라 접근 권한을 부여하는 방식이다.</li><li>•중앙관리자가 접근 통제 권한을 지정한다.</li><li>•임의 접근 통제와 강제 접근 통제의 단점을 보완하였으며, 다중 프로그래밍 환경에 최적화된 방식이다.</li><li>•중앙관리자가 역할마다 권한을 부여하면, 책임과 자질에 따라 역할을 할당 받은 사용자들은 역할에 해당하는 권한을 사용할 수 있다.</li></ul>

객체 : 테이블, 컬럼 등과 같은 데이터베이스 개체들을 의미한다.

주체 : 일반적으로 객체를 사용하기 위해 접근을 시도하는 사용자를 의미한다.

GRANT : 객체에 대한 권한을 부여하는 명령이다.

예) GRANT RESOURCE TO KORA; : 사용자 ID가 KORA인 사람에게 데이터베이스 및 테이블을 생성할 수 있는 권한을 부여한다.

REVOKE : 객체에 부여된 권한을 취소하는 명령이다.

예) REVOKE UPDATE ON 고객 FROM JULIA; : 사용자 ID가 JUUA인 사람에게 부여한 권한 중 갱신(UPDATE) 권한을 취소한다.

## 2. 물리 데이터베이스 설계-SEC\_12(데이터베이스 보안 / 접근통제)

### 2) 강제 접근 통제(MAC)의 보안모델

<b>벨 라파둘라 모델</b> (Bell-LaPadula Model)	<ul style="list-style-type: none"><li>•군대의 보안 레벨처럼 정보의 기밀성에 따라 상하 관계가 구분된 정보를 보호하기 위해 사용한다.</li><li>•보안 취급자의 등급을 기준으로 읽기 권한과 쓰기 권한이 제한된다.</li><li>•자신의 보안 레벨 이상의 문서를 작성할 수 있고, 자신의 보안 레벨 이하의 문서를 읽을 수 있다.</li></ul> 예) 보안 레벨이 2급인 보안 취급자의 읽기 권한과 쓰기 권한 <ul style="list-style-type: none"><li>- 2, 3급 비밀 문서를 조회할 수 있다.</li><li>- 1·2급 비밀 문서를 작성할 수 있다.</li></ul>
<b>비바 무결성 모델</b> (Biba Integrity Model)	<ul style="list-style-type: none"><li>•벨 라파둘라 모델을 보완한 수학적 모델로, 무결성을 보장하는 최초의 모델이다.</li><li>•비인가자에 의한 데이터 변형을 방지한다.</li></ul>
<b>클락-윌슨 무결성 모델</b> (Clark-Wilson Integrity Model)	무결성 중심의 상업용 모델로 사용자가 직접 객체에 접근할 수 없고 프로그램에 의해 접근이 가능한 보안 모델이다.
<b>만리장성 모델</b> (Chinese Wall Model)	서로 이해 충돌 관계에 있는 객체 간의 정보 접근을 통제하는 모델이다.

- 접근 통제의 3요소는 접근 통제 정책, 접근 통제 메커니즘, 접근 통제 보안모델이다.



## 2. 물리 데이터베이스 설계-SEC\_12(데이터베이스 보안 / 접근통제)

### 3) 접근 통제 정책

; 접근 통제 정책은 어떤 주체가(Who)가 언제(When), 어디서(Where), 어떤 객체(What)에게, 어떤 행위(How)에 대한 허용 여부를 정의하는 것으로, 신분 기반 정책, 규칙 기반 정책, 역할 기반 정책이 있다.

신분 기반 정책	<ul style="list-style-type: none"><li>•주체나 그룹의 신분에 근거하여 객체의 접근을 제한하는 방법으로, IBP와 GBP가 있다.</li><li>•IBP(Individual-Based Policy) : 최소 권한 정책으로, 단일 주체에게 하나의 객체에 대한 허가를 부여한다.</li><li>•GBP(Group-Based Policy) : 복수 주체에 하나의 객체에 대한 허가를 부여한다.</li></ul>
규칙 기반 정책	<ul style="list-style-type: none"><li>•주체가 갖는 권한에 근거하여 객체의 접근을 제한하는 방법으로, MLP와 CBP가 있다.</li><li>•MLP(Multi-Level Policy) : 사용자 및 객체별로 지정된 기밀 분류에 따른 정책</li><li>•CBP(Compartment-Based Policy) : 집단별로 지정된 기밀 허가에 따른 정책</li></ul>
역할 기반 정책	<p>GBP의 변형된 정책으로, 주체의 신분이 아니라 주체가 맡은 역할에 근거하여 객체의 접근을 제한하는 방법이다.</p> <p>예) 인사 담당자, DBA 등</p>



## 2. 물리 데이터베이스 설계-SEC\_12(데이터베이스 보안 / 접근통제)

### 4) 접근 통제 매커니즘

; 접근 통제 매커니즘은 정의된 접근 통제 정책을 구현하는 기술적인 방법으로, 접근 통제 목록, 능력 리스트, 보안 등급, 비밀번호, 암호화 등이 있다.

- 접근 통제 목록(Access Control List) : 객체를 기준으로 특정 객체에 대해 어떤 주체가 어떤 행위를 할 수 있는지를 기록한 목록이다.
- 능력 리스트(Capability List): 주체를 기준으로 주체에게 허가된 자원 및 권한을 기록한 목록이다.
- 보안 등급(Security Level) : 주체나 객체 등에 부여된 보안 속성의 집합으로, 이 등급을 기반으로 접근 승인 여부가 결정된다.
- 비밀번호 : 주체가 자신임을 증명할 때 사용하는 인증 방법이다.
- 암호화 : 데이터를 보낼 때 지정된 수신자 이외에는 내용을 알 수 없도록 평문을 암호문으로 변환하는 것으로, 무단 도용을 방지하기 위해 주로 사용된다.

## 2. 물리 데이터베이스 설계-SEC\_12(데이터베이스 보안 / 접근통제)

### 5) 접근 통제 보안 모델

; 접근 통제 보안 모델은 보안 정책을 구현하기 위한 정형화된 모델로, 기밀성 모델, 무결성 모델, 접근 통제 모델이 있다.

#### ● 기밀성 모델

- 기밀성 모델은 군사적인 목적으로 개발된 최초의 수학적 모델로, 기밀성 보장이 최우선인 모델이다.
- > 기밀성 모델은 군대 시스템 등 특수 환경에서 주로 사용된다.
- > 제약 조건
  - 단순 보안 규칙 : 주체는 자신보다 높은 등급의 객체를 읽을 수 없다.
  - ★(스타)-보안 규칙 : 주체는 자신보다 낮은 등급의 객체에 정보를 쓸 수 없다.
  - 강한 ★(스타) 보안 규칙 : 주체는 자신과 등급이 다른 객체를 읽거나 쓸 수 없다.

Level	단순 보안 규칙	★(스타)-보안 규칙	강한 ★(스타) 보안 규칙
	읽기 권한	쓰기 권한	읽기/쓰기 권한
높은 등급	통제	가능	통제
같은 등급	가능	가능	가능
낮은 등급	가능	통제	통제

기밀성(Confidentiality) : 기밀성은 시스템 내의 정보와 자원은 인가된 사용자에게만 접근이 허용되는 것으로 정보가 전송 중에 노출되더라도 데이터를 읽을 수 없는 것을 의미한다.

★(스타)-보안 규칙 : ★스타-보안 규칙은 높은 등급의 주체가 자신이 접근 가능한 정보를 낮은 등급으로 복사하여 유출하는 행위를 방지하기 위한 것으로 정보의 기밀성 보호를 위한 제약이다.

## 2. 물리 데이터베이스 설계-SEC\_12(데이터베이스 보안 / 접근통제)

### 5) 접근 통제 보안 모델

#### ● 무결성 모델

- 무결성 모델은 기밀성 모델에서 발생하는 불법적인 정보 변경을 방지하기 위해 무결성을 기반으로 개발된 모델이다.
- > 무결성 모델은 데이터의 일관성 유지에 중점을 두어 개발되었다.
- > 무결성 모델은 기밀성 모델과 동일하게 주체 및 객체의 보안 등급을 기반으로 한다.
- > 제약 조건
  - 단순무결성 규칙 : 주체는 자신보다 낮은 등급의 객체를 읽을 수 없다.
  - ★(스타)-무결성 규칙 : 주체는 자신보다 높은 등급의 객체에 정보를 쓸 수 없다.

Level	단순 무결성 규칙	★(스타)-무결성 규칙
	읽기 권한	쓰기 권한
높은 등급	가능	통제
같은 등급	가능	가능
낮은 등급	통제	가능

## 2. 물리 데이터베이스 설계-SEC\_12(데이터베이스 보안 / 접근통제)

### 5) 접근 통제 보안 모델

#### ● 접근 통제 모델

- 접근통제 모델은 접근 통제 메커니즘을 보안 모델로 발전시킨 것으로, 대표적으로 접근 통제 행렬 (Access Control Matrix)이 있다.
- 접근 통제 행렬(Access Control Matrix)
  - > 임의적인 접근 통제를 관리하기 위한 보안 모델로, 행은 주체, 열은 객체 즉, 행과 열로 주체와 객체의 권한 유형을 나타낸다.
    - 행 : 주체로서 객체에 접근을 시도하는 사용자이다.
    - 열 : 객체로서 접근 통제가 이뤄지는 테이블, 컬럼, 뷰 등과 같은 데이터베이스의 개체이다.
    - 규칙 : 주체가 객체에 대하여 수행하는 입력, 수정, 삭제 등의 데이터베이스에 대한 조작이다.

이름	사원	부서	상품	판매현황	거래업체
김상민	ALL	ALL	R	R	R
우지경	R/W	R/W	R	R	R
하석민	R	R	R	R/W	R/W
양구성	R	R	R/W		

## 2. 물리 데이터베이스 설계-SEC\_12(데이터베이스 보안 / 접근통제)

### 6) 접근 통제 조건

; 접근 통제 조건은 접근통제 매커니즘의 취약점을 보완하기 위해 접근통제 정책에 부가하여 적용할 수 있는 조건이다.

- 값 종속 통제(Value-Dependent Control) : 일반적으로는 객체에 저장된 값에 상관없이 접근통제를 동일하게 허용하지만 객체에 저장된 값에 따라 다르게 접근 통제를 허용해야 하는 경우에 사용한다.

예) 납입한 금액에 따라 보안 등급이 설정되고, 보안 등급에 따라 접근 여부가 결정되는 경우

- 다중 사용자 통제(Multi-User Control) : 지정된 객체에 다수의 사용자가 동시에 접근을 요구하는 경우에 사용된다.

예) 여러 명으로 구성된 한 팀에서 다수결에 따라 접근 여부가 결정되는 경우

- 컨텍스트 기반 통제(Context-Based Control) : 특정 시간, 네트워크 주소, 접근 경로, 인증 수준 등에 근거하여 접근을 제어하는 방법으로, 다른 보안 정책과 결합하여 보안 시스템의 취약점을 보완할 때 사용된다.

예) 근무시간, 즉 월요일~금요일 9:00~18:00 사이에만 접근 가능

## 2. 물리 데이터베이스 설계-SEC\_12(데이터베이스 보안 / 접근통제)

### 7) 감사 추적

; 감사 추적은 사용자나 애플리케이션이 데이터베이스에 접근하여 수행한 모든 활동을 기록하는 기능이다.

- 감사 추적은 오류가 발생한 데이터베이스를 복구하거나 부적절한 데이터 조작을 파악하기 위해 사용된다.
- 감사 추적 시 실행한 프로그램, 사용자, 날짜 및 시간, 접근한 데이터의 이전 값 및 이후 값 등이 저장된다.

# 물리 데이터베이스 설계-SEC\_12(데이터베이스 보안 / 접근통제) 기출 및 출제 예상 문제

## 기출 및 출제 예상 문제(데이터베이스 보안 / 접근통제)

1. 다음은 정보의 접근 통제 정책에 대한 설명이다. (ㄱ)에 들어갈 내용으로 옳은 것은?

정책	( ㄱ )	DAC	RBAC
권한 부여	시스템	데이터 소유자	중앙 관리자
접근 결정	보안등급 (Label)	신분 (Identity)	역할 (Role)
정책 변경	고정적 (변경 어려움)	변경 용이	변경 용이
장점	안정적 중앙 집중적	구현 용이 유연한	관리 용이

- ① NAC                      ② MAC
- ③ SDAC                    ④ AAC

### 임의 접근 통제(DAC; Discretionary Access Control)

- 임의 접근 통제는 데이터에 접근하는 사용자의 신원(신분)에 따라 접근 권한을 부여하는 방식이다.
- 데이터 소유자가 접근 통제 권한을 지정하고 제어한다.
- 객체를 생성한 사용자가 생성된 객체에 대한 모든 권한을 부여 받고, 부여된 권한을 다른 사용자에게 허가할 수도 있다.

3. 접근 통제 방법 중 조직 내에서 직무, 직책 등 개인의 역할에 따라 결정하여 부여하는 접근 정책은?

- ① RBAC                      ② DAC
- ③ MAC                        ④ QAC

4. 정보 시스템 내에서 어떤 주체가 특정 개체에 접근하려 할 때, 양쪽의 보안 레이블(Security Label)에 기초하여 높은 보안 수준을 요구하는 정보(객체)가 낮은 보안 수준의 주체에게 노출되지 않도록 하는 접근 제어 방법은?

- ① Mandatory Access Control
- ② User Access Control
- ③ Discretionary Access Control
- ④ Data-Label Access Control

# 물리 데이터베이스 설계-SEC\_12(데이터베이스 보안 / 접근통제) 기출 및 출제 예상 문제

## 기출 및 출제 예상 문제(데이터베이스 보안 / 접근통제)

### 5. 다음 내용이 설명하는 접근 제어 모델은?

- 군대의 보안 레벨처럼 정보의 기밀성에 따라 상하 관계가 구분된 정보를 보호하기 위해 사용
- 자신의 권한보다 낮은 보안 레벨 권한을 가진 경우에는 높은 보안 레벨의 문서를 읽을 수 없고 자신의 권한보다 낮은 수준의 문서만 읽을 수 있다.
- 자신의 권한보다 높은 보안 레벨의 문서에는 쓰기가 가능하지만 보안 레벨이 낮은 문서의 쓰기 권한은 제한한다.

- ① Clark-Wilson Integrity Model
- ② PDCA Model
- ③ Bell-Lapadula Model
- ④ Chinese Wall Model

### 벨 라파둘라 모델(Bell-Lapadula Model)

- 군대의 보안 레벨처럼 정보의 기밀성에 따라 상하 관계가 구분된 정보를 보호하기 위해 사용한다.
- 보안 취급자의 등급을 기준으로 읽기 권한과 쓰기 권한이 제한된다.
- 자신의 보안 레벨 이상의 문서를 작성할 수 있고, 자신의 보안 레벨

### 7. 다음 중 접근 통제 정책이 아닌 것은?

- ① 신분 기반 정책
- ② 업무 기반 정책
- ③ 규칙 기반 정책
- ④ 역할 기반 정책

### 신분 기반 정책

- 주체나 그룹의 신분에 근거하여 객체의 접근을 제한하는 방법으로, IBP와 GBP가 있다.

### 규칙 기반 정책

- 주체가 갖는 권한에 근거하여 객체의 접근을 제한하는 방법으로, MLP와 CBP가 있다.

### 역할 기반 정책

- GBP의 변형된 정책으로, 주체의 신분이 주체가 맡은 역할에 근거하여 객체의 접근을 제한하는 방법이다.

### 8. 다음 중 접근 통제 보안 모델에 대한 설명으로 틀린 것은?

- ① 접근 통제 보안 모델은 보안 정책을 구현하기 위한 정형화된 모델이다.
- ② 무결성 모델은 군대 시스템 등 특수 환경에서 주로 사용된다.
- ③ 무결성 모델은 데이터의 일관성 유지에 중점을 두어 개발되었다.



## 2. 물리 데이터베이스 설계-SEC\_13(데이터베이스 백업)

### 1) 데이터베이스 백업

; 데이터베이스 백업은 전산 장비의 장애에 대비하여 데이터베이스에 저장된 데이터를 보호하고 복구하기 위한 작업으로, 치명적인 데이터 손실을 막기 위해서는 데이터베이스를 정기적으로 백업해야 한다.

- 데이터베이스 관리 시스템(DBMS)은 데이터베이스 파괴 및 실행 중단이 발생하면 이를 복구할 수 있는 기능을 제공한다.

### 2) 데이터베이스 장애 유형

; 데이터베이스의 장애 유형을 정확히 파악하고 장애에 따른 백업 전략을 세워야 장애 발생 시 복구가 가능하다.

- 사용자 실수 : 사용자의 실수로 인해 테이블이 삭제되거나 잘못된 트랜잭션이 처리된 경우
- 미디어 장애 : CPU, 메모리, 디스크 등 하드웨어 장애나 데이터가 파손된 경우
- 구문 장애 : 프로그램 오류나 사용 공간의 부족으로 인해 발생하는 장애
- 사용자 프로세스 장애 : 프로그램이 비정상적으로 종료되거나 네트워크 이상으로 세션이 종료되어 발생하는 오류
- 인스턴스 장애 : 하드웨어 장애, 정전, 시스템 파일 파손 등 비정상적인 요인으로 인해 메모리나 데이터베이스 서버의 프로세스가 중단된 경우

## 2. 물리 데이터베이스 설계-SEC\_13(데이터베이스 백업)

### 3) 로그 파일

; 로그 파일은 데이터베이스의 처리 내용이나 이용 상황 등 상태 변화를 시간의 흐름에 따라 모두 기록한 파일로, 데이터베이스의 복구를 위해 필요한 가장 기본적인 자료이다.

- 로그 파일을 기반으로 데이터베이스를 과거 상태로 복귀(UNDO)시키거나 현재 상태로 재생(REDO)시켜 데이터베이스 상태를 일관성 있게 유지할 수 있다.
- 로그 파일은 트랜잭션 시작 시점, Rollback시점, 데이터 입력, 수정, 삭제 시점 등에서 기록된다.
- 로그 파일 내용 : 트랜잭션이 작업한 모든 내용, 트랜잭션 식별, 트랜잭션 레코드, 데이터 식별자, 갱신 이전 값(Before Image), 갱신 이후 값(After Image) 등

## 2. 물리 데이터베이스 설계-SEC\_13(데이터베이스 백업)

### 4) 데이터베이스 복구 알고리즘

; 데이터베이스 복구 알고리즘은 동기적/비동기적 갱신에 따라 NO-UNDO/REDO, UNDO/NO-REDO, UNDO/REDO, NO-UNDO/NO-REDO 방법으로 분류된다.

<b>NO-UNDO/ REDO</b>	<ul style="list-style-type: none"><li>•데이터베이스 버퍼의 내용을 비동기적으로 갱신한 경우의 복구 알고리즘</li><li>•NO-UNDO : 트랜잭션 완료 전에는 변경 내용이 데이터베이스에 기록되지 않으므로 취소할 필요가 없다.</li><li>•REDO : 트랜잭션 완료 후 데이터베이스 버퍼에는 기록되어 있고, 저장매체에는 기록되지 않았으므로 트랜잭션 내용을 다시 실행해야 한다.</li></ul>
<b>UNDO/ NO-REDO</b>	<ul style="list-style-type: none"><li>•데이터베이스 버퍼의 내용을 동기적으로 갱신한 경우의 복구 알고리즘</li><li>•UNDO: 트랜잭션 완료 전에 시스템이 파손되었다면 변경된 내용을 취소한다.</li><li>•NO-REDO: 트랜잭션 완료 전에 데이터베이스 버퍼 내용을 이미 저장 매체에 기록했으므로 트랜잭션 내용을 다시 실행할 필요가 없다.</li></ul>
<b>UNDO/ REDO</b>	<ul style="list-style-type: none"><li>•데이터베이스 버퍼의 내용을 동기/비동기적으로 갱신한 경우의 복구 알고리즘</li><li>•데이터베이스 기록 전에 트랜잭션이 완료될 수 있으므로 완료된 트랜잭션이 데이터베이스에 기록되지 못했다면 다시 실행해야 한다.</li></ul>
<b>NO-UNDO/ NO-REDO</b>	<ul style="list-style-type: none"><li>•데이터베이스 버퍼의 내용을 동기적으로 저장 매체에 기록하지만 데이터베이스와는 다른 영역에 기록한 경우의 복구 알고리즘</li><li>•NO-UNDO : 변경 내용은 데이터베이스와 다른 영역에 기록되어 있으므로 취소할 필요가 없다.</li><li>•NO-REDO : 다른 영역에 이미 기록되어 있으므로 트랜잭션을 다시 실행할 필요가 없다.</li></ul>

동기적 갱신(Synchronous I/O) : 트랜잭션이 완료되기 전에 데이터베이스 버퍼 내용을 동시적으로 저장매체에 기록하는 것

비동기적 갱신(Asynchronous I/O) : 트랜잭션이 완료된 내용을 일정 주거나 작업량에 따라 시간 차이를 두고 저장매체에 기록하는 것

UNDO : 기록한 내용을 취소함

REDO : 트랜잭션 내용을 다시 실행함

## 2. 물리 데이터베이스 설계-SEC\_13(데이터베이스 백업)

### 5) 백업 종류

; 백업 종류는 복구 수준에 따라서 운영체제를 이용하는 물리 백업과 DBMS 유틸리티를 이용하는 논리 백업으로 나뉜다.

- 물리 백업 : 데이터베이스 파일을 백업하는 방법으로, 백업 속도가 빠르고 작업이 단순하지만 문제 발생 시 원인 파악 및 문제 해결이 어렵다.
- 논리 백업 : DB 내의 논리적 객체들을 백업하는 방법으로, 복원 시 데이터 손상을 막고 문제 발생 시 원인 파악 및 해결이 수월하지만 백업 / 복원 시 시간이 많이 소요된다.

구분	설명	복구 수준
물리 백업	로그 파일 백업 실시	완전 복구
	로그 파일 백업 없음	백업 시점까지 복구
논리 백업	DBMS 유틸리티	

# 물리 데이터베이스 설계-SEC\_13(데이터베이스 백업) 출제 예상 문제

## 출제 예상 문제(데이터베이스 백업)

1. 다음 중 데이터베이스 백업에 대한 설명으로 틀린 것은?

- ① DBMS는 데이터베이스 파괴 시 복구하는 기능을 제공한다.
- ② 데이터베이스 백업은 정기적으로 수행하는 것이 좋다.
- ③ 데이터베이스의 상태를 일관성 있게 유지시키기 위해 로그 파일이 이용된다.
- ④ 로그 파일은 데이터베이스의 상태 변화에 대한 내용을 작업 순서에 따라 기록한 파일이다.

## 데이터베이스 백업

데이터베이스 백업은 전산 장비의 장애에 대비하여 데이터베이스 에 저장된 데이터를 보호하고 복구하기 위한 작업으로, 치명적인 데이터 손실을 막기 위해서는 데이터베이스를 정기적으로 백업해야 한다.

- 데이터베이스 관리 시스템(DBMS)은 데이터베이스 파괴 및 실행 중단이 발생하면 이를 복구할 수 있는 기능을 제공한다.

## 로그 파일

로그 파일은 데이터베이스의 처리 내용이나 이용 상황 등 상태 변화를 시간의 흐름에 따라 모두 기록한 파일로, 데이터베이스의

3. 데이터베이스 복구 알고리즘의 종류로 틀린 것은?

- ① NO-UNDO/REDO    ② UNDO/NO-REDO
- ③ UNDO/REDO        ④ DO/REDO

## NO-UNDO / REDO

- 데이터베이스 버퍼의 내용을 비동기적으로 갱신한 경우 복구 알고리즘
- NO-UNDO : 트랜잭션 완료 전에는 변경 내용이 데이터베이스에 기록되지 않으므로 취소할 필요가 없다.
- REDO : 트랜잭션 완료 후 데이터베이스 버퍼에는 기록되어 있고, 저장매체에는 기록되지 않았으므로 트랜잭션 내용을 다시 실행해야 한다.

## UNDO / NO-READ

- 데이터베이스 버퍼의 내용을 동기적으로 갱신한 경우 복구 알고리즘
- UNDO : 트랜잭션 완료 전에 시스템이 파손되었다면 변경된 내용을 취소한다.
- NO-READ : 트랜잭션 완료 전에 데이터베이스 버퍼 내용을 이미 저장매체에 기록했기에 트랜잭션 내용을 다시 실행할 필요가 없다.

## UNDO / REDO

- 데이터베이스 버퍼의 내용을 동기/비동기적으로 갱신한 경우 복구 알고리즘

## 2. 물리 데이터베이스 설계-SEC\_14(스토리지)

### 1) 스토리지(Storage)의 개요

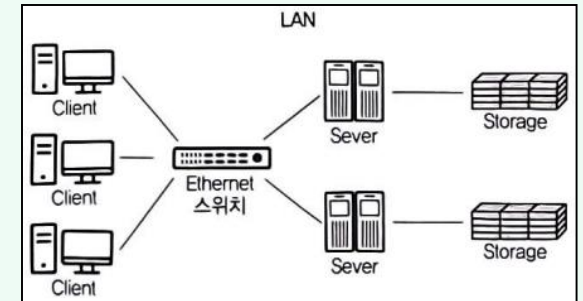
; 스토리지는 단일 디스크로 처리할 수 없는 대용량의 데이터를 저장하기 위해 서버와 저장장치를 연결하는 기술이다.

- 스토리지의 종류에는 DAS, NAS, SAN이 있다.

### 2) DAS(Direct Attached Storage)

; DAS는 서버와 저장장치를 전용 케이블로 직접 연결하는 방식으로, 일반 가정에서 컴퓨터에 외장하드를 연결하는 것이 여기에 해당된다.

- 서버에서 저장장치를 관리한다.
- 저장장치를 직접 연결하므로 속도가 빠르고 설치 및 운영이 쉽다.
- 초기 구축 비용 및 유지보수 비용이 저렴하다.
- 직접 연결 방식이므로 다른 서버에서 접근할 수 없고 파일을 공유할 수 없다.
- 확장성 및 유연성이 상대적으로 떨어진다.
- 저장 데이터가 적고 공유가 필요 없는 환경에 적합하다.



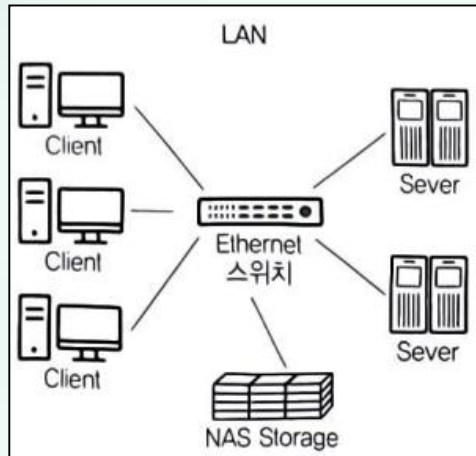
서버에 스토리지가 직접 연결되어 있어 속도는 빠르나 다른 서버에서 스토리지에 접근하여 사용할 수 없다.

## 2. 물리 데이터베이스 설계-SEC\_14(스토리지)

### 3) NAS(Network Attached Storage)

; NAS는 서버와 저장장치를 네트워크를 통해 연결하는 방식이다.

- 별도의 파일 관리 기능이 있는 NAS Storage가 내장된 저장장치를 직접 관리한다.
- Ethernet 스위치를 통해 다른 서버에서도 스토리지에 접근할 수 있어 파일 공유가 가능하고, 장소에 구애 받지 않고 저장장치에 쉽게 접근할 수 있다.
- DAS에 비해 확장성 및 유연성이 우수하다.
- 접속 증가 시 성능이 저하될 수 있다.



파일 관리 기능이 있는 NAS Storage가 네트워크 상에 독립적으로 연결되어 있으므로 서버들이 자유롭게 스토리지에 접근하여 파일을 공유할 수 있다.

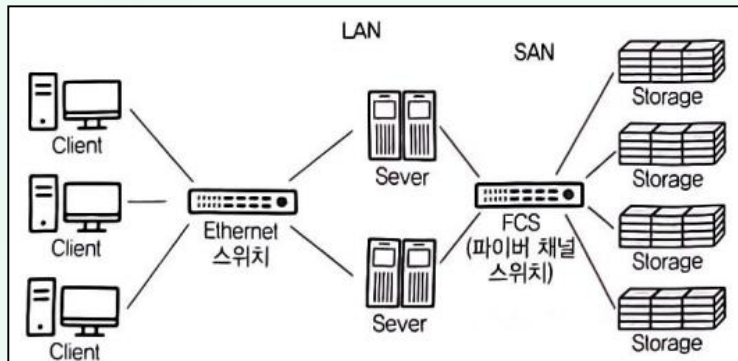


## 2. 물리 데이터베이스 설계-SEC\_14(스토리지)

### 4) SAN(Storage Area Network)

; SAN은 DAS의 빠른 처리와 NAS의 파일 공유 장점을 혼합한 방식으로, 서버와 저장장치를 연결하는 전용 네트워크를 별도로 구성하는 방식이다.

- 광 채널(FC) 스위치를 이용하여 네트워크를 구성한다.
- 광 채널 스위치는 서버나 저장장치를 광케이블로 연결하므로 처리 속도가 빠르다.
- 저장장치를 공유함으로써 여러 개의 저장장치나 백업 장비를 단일화시킬 수 있다.
- 확장성, 유연성, 가용성이 뛰어나다.
- 높은 트랜잭션 처리에 효과적이나 기존 시스템의 경우 장비의 업그레이드가 필요하고, 초기 설치시에는 별도의 네트워크를 구축해야 하므로 비용이 많이 든다.



광 채널(Fiber Channel, FC) : 광 채널은 컴퓨터 장치 간 데이터의 전송 속도를 기가 바이트로 높이기 위한 네트워크 기술이다. 초기에는 슈퍼 컴퓨터 분야에서 주로 사용되었으나 최근에는 SAN의 표준 연결 방식에도 사용되고 있다. 서버와 스토리지가 광 채널 스위치로 연결된 별도의 전용 스토리지 네트워크를 구성하는 방식으로 서버들이 저장장치 및 파일을 자유롭게 공유할 수 있다.



# 물리 데이터베이스 설계-SEC\_14(스토리지) 기출 및 출제 예상 문제

## 기출 및 출제 예상 문제(스토리지)

### 1. 다음 내용이 설명하는 것은?

- 네트워크상에 광채널 스위치의 이점인 고속 전송과 장거리 연결 및 멀티 프로토콜 기능을 활용
- 각기 다른 운영체제를 가진 여러 기종들이 네트워크상에서 동일 저장장치의 데이터를 공유하게 함으로써, 여러 개의 저장장치나 백업 장비를 단일화 시킨 시스템

- ① SAN                      ② MBR
- ③ NAC                      ④ NIC

**SAN**은 DAS의 빠른 처리와 NAS의 파일 공유 장점을 혼합한 방식으로, 서버와 저장장치를 연결하는 전용 네트워크를 별도로 구성하는 방식이다.

- **광 채널(FC) 스위치**를 이용하여 네트워크를 구성한다.
- 광 채널 스위치는 서버나 저장장치를 광케이블로 연결하므로 처리 속도가 빠르다.
- 저장장치를 공유함으로써, 여러 개의 저장장치나 백업 장비를 단일화 시킬 수 있다.
- 확장성, 유연성, 가용성이 뛰어나다

### 3. 다음 중 스토리지에 대한 설명으로 틀린 것은?

- ① 스토리지는 서버와 저장장치를 연결하는 하나의 기술이다.
- ② **DAS는 서버와 저장장치를 전용 케이블로 직접 연결하는 방식으로, 확장성과 유연성이 상대적으로 우수하다.**
- ③ NAS는 서버와 저장장치를 네트워크를 통해 연결하는 방식으로 접속 증가 시 성능이 저하될 수 있다.
- ④ SAN은 별도의 네트워크를 구성해 데이터를 관리하는 방식으로, 초기 별도의 네트워크를 구축해야 하므로 비용이 많이 든다.

**NAS**는 서버와 저장장치를 네트워크를 통해 연결하는 방식이다.

- 별도의 파일 관리 기능이 있는 NAS Storage가 내장된 저장장치를 직접 관리한다.
- Ethernet 스위치를 통해 다른 서버에서도 스토리지 접근할 수 있어 파일 공유가 가능하고, 장소에 구애 받지 않고 저장장치에 쉽게 접근할 수 있다.
- DAS에 비해서는 확장성 및 유연성이 우수하다.
- 접속 증가 시 성능이 저하될 수 있다.

### 4. 다음 중 단일 디스크로 처리할 수 없는 대용량의 데이터를 저장 하기 위해 서버와 저장장치를 연결하는 스토리지가 아닌 것은?

- ① SSD                      ② NAS

## 2. 물리 데이터베이스 설계-SEC\_15(논리 데이터 모델의 물리 데이터 모델 변환)

## 1) 테이블(Table)

; 테이블은 데이터를 저장하는 데이터베이스의 가장 기본적인 오브젝트이다.

- 테이블은 컬럼(Column, 열)과 로우(Row, 행)로 구성되며, 컬럼에는 지정된 유형에 따라 데이터가 저장된다.
- 테이블의 구성 요소

<b>로우(Row)</b>	튜플, 인스턴스, 어커런스라고도 한다.
<b>컬럼(Column)</b>	각 속성 항목에 대한 값을 저장한다.
<b>기본키 (Primary key)</b>	<ul style="list-style-type: none"> <li>•기본키는 후보키 중에서 선택한 주키(Main Key)이다.</li> <li>•한 릴레이션에서 특정 튜플을 유일하게 구별할 수 있는 속성이다.</li> </ul>
<b>외래키 (Foreign key)</b>	<ul style="list-style-type: none"> <li>•다른 릴레이션의 기본키를 참조하는 속성 또는 속성들의 집합을 의미한다.</li> <li>•한 릴레이션에 속한 속성 A와 참조 릴레이션의 기본키인 B가 동일한 도메인 상에서 정의되었을 때의 속성 A를 외래키라고 한다.</li> </ul>

	사원번호	이름	직급	연봉	부서번호
	S001	신은혁	과장	50,000,000	B02
	S002	이혜숙	차장	62,000,000	B01
	S003	김숙자	이사	90,000,000	B03

테이블 구성 요소

기본키 (사원번호)

외래키 (부서번호)

## 2. 물리 데이터베이스 설계-SEC\_15(논리 데이터 모델의 물리 데이터 모델 변환)

### 2) 엔티티(Entity)를 테이블로 변환

; 논리 데이터 모델에서 정의된 엔티티를 물리 데이터 모델의 테이블로 변환하는 것이다.

- 엔티티를 테이블로 변환한 후 테이블 목록 정의서를 작성한다.
  - 테이블 목록 정의서 : 전체 테이블을 목록으로 요약 관리하는 문서로, 테이블 목록이라고도 한다.

예) 테이블 목록 정의서

테이블 ID	테이블명	타입	분류	테이블 스페이스	파티션 여부	발생 주기	총 건수	수정일
user_item	제품	STANDARD						2020/12/02
sell_list	판매 목록	STANDARD						2020/12/09
user_name	사용자	STANDARD	key					2020/12/01



변환규칙

논리적 설계(데이터 모델링)	물리적 설계
엔티티(Entity)	테이블(Table)
속성(Attribute)	컬럼(Column)
주 식별자(Primary Identifier)	기본키(Primary Key)
외부 식별자(Foreign Identifier)	외래키(Foreign Key)
관계(Relationship)	관계(Relationship)

## 2. 물리 데이터베이스 설계-SEC\_15(논리 데이터 모델의 물리 데이터 모델 변환)

### 2) 엔티티(Entity)를 테이블로 변환

예) 엔티티를 테이블로 변환



#### ● 변환 시 고려사항

- 일반적으로 테이블과 엔티티 명칭은 동일하게 하는 것을 권고한다.
- 엔티티는 주로 한글명을 사용하지만 테이블은 소스 코드의 가독성을 위해 영문명을 사용한다.
- 메타 데이터 관리 시스템에 표준화된 용어가 있을 때는 메타에 등록된 단어를 사용하여 명명한다.

메타 데이터 관리 시스템 : 메타 데이터 관리 시스템은 메타 데이터를 수집하거나 여러 사람이 메타 데이터를 편리하게 사용할 수 있도록 제공하는 시스템이다.

메타 데이터(metadata or metainformation)는 데이터(data)에 대한 데이터이다. 이렇게 흔히들 간단히 정의하지만, 캐런 코일(Karen Coyle)에 의하면 '어떤 목적을 가지고 만들어진 데이터(constructed data with a purpose)'라고도 정의한다. 즉, 다른 데이터를 정의하고 기술하는 데이터(data that defines and describes other data)이다. 가령 도서관에서 사용하는 서지 기술용으로 만든 것이 그 대표적인 예이다. 지금은 온톨로지의 등장과 함께 기계가 읽고 이해할 수 있는(Machine Actionable) 형태의 메타 데이터가 많이 사용되고 있다. 설명 메타데이터, 구조화 메타 데이터로 구분된다.

## 2. 물리 데이터베이스 설계-SEC\_15(논리 데이터 모델의 물리 데이터 모델 변환)

### 3) 슈퍼타입/서브타입을 테이블로 변환

; 슈퍼타입/서브타입은 논리 데이터 모델에서 이용되는 형태이므로 물리 데이터 모델을 설계할 때는 슈퍼타입/서브타입을 테이블로 변환해야 한다.

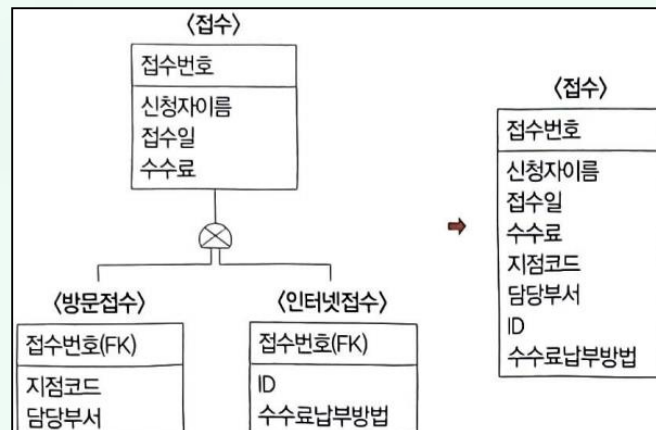
- 슈퍼타입/서브타입 모델을 테이블로 변환하는 방법에는 슈퍼타입 기준 테이블 변환, 서브타입 기준 테이블 변환, 개별타입 기준 테이블 변환이 있다.

- 슈퍼타입 기준 테이블 변환

- 슈퍼타입 기준의 테이블 변환은 서브타입을 슈퍼타입에 통합하여 하나의 테이블로 만드는 것이다.

- > 서브타입에 속성이나 관계가 적을 경우에 적용하는 방법으로, 하나로 통합된 테이블에는 서브타입의 모든 속성이 포함되어야 한다.

예) 슈퍼타입 기준 테이블 변환



서브타입의 <방문접수> 개체에 있는 '지점코드', '담당부서' 와 <인터넷접수> 개체에 있는 'ID', '수수료 납부방법'이 슈퍼타입인 <접수> 개체에 통합되어 <접수> 테이블로 변환 된다.

## 2. 물리 데이터베이스 설계-SEC\_15(논리 데이터 모델의 물리 데이터 모델 변환)

### 3) 슈퍼타입/서브타입을 테이블로 변환

#### ● 장·단점

장점	<ul style="list-style-type: none"><li>•데이터의 액세스가 상대적으로 용이하다.</li><li>•뷰를 이용하여 각각의 서브타입만을 액세스하거나 수정할 수 있다.</li><li>•서브타입 구분이 없는 임의의 집합에 대한 처리가 용이하다.</li><li>•여러 테이블을 조인하지 않아도 되므로 수행 속도가 빨라진다.</li><li>•SQL 문장 구성이 단순해진다.</li></ul>
단점	<ul style="list-style-type: none"><li>•테이블의 컬럼이 증가하므로 디스크 저장 공간이 증가한다.</li><li>•처리마다 서브타입에 대한 구분(TYPE)이 필요한 경우가 많이 발생한다.</li><li>•인덱스 크기의 증가로 인덱스의 효율이 떨어진다.</li></ul>

#### ● 서브타입 기준 테이블 변환

- 서브타입 기준의 테이블 변환은 슈퍼타입 속성들을 각각의 서브타입에 추가하여 서브타입들을 개별적인 테이블로 만드는 것이다.

- 서브타입에 속성이나 관계가 많이 포함된 경우 적용한다.

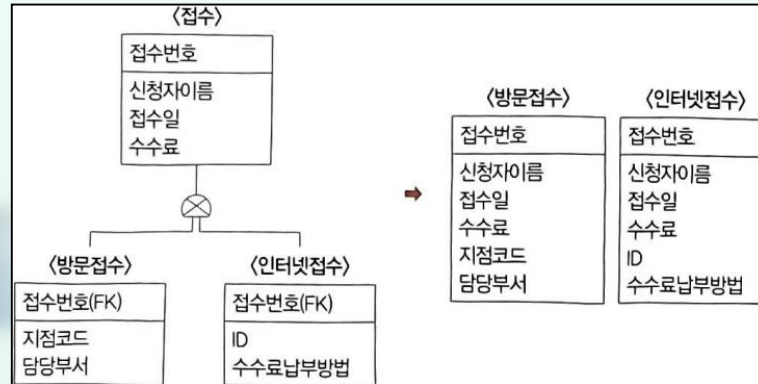
스캔 방식은 테이블의 데이터를 검색할 때 앞에서부터 순차적으로 행(Row)을 비교하여 원하는 데이터를 가져오는 접근 방법이다. 전체 스캔(Full Scan)은 전체 테이블을 범위 스캔(Range Scan)은 테이블의 특정 부분만을 검색하는 방법임.

장점	<ul style="list-style-type: none"><li>•각 서브타입 속성들의 선택 사양이 명확한 경우에 유리하다.</li><li>•처리할 때마다 서브타입 유형을 구분할 필요가 없다.</li><li>•여러 개의 테이블로 통합하므로 테이블당 크기가 감소하여 전체 테이블 스캔 시 유리하다.</li></ul>
단점	<ul style="list-style-type: none"><li>•수행 속도가 감소할 수 있다.</li><li>•복잡한 처리를 하는 SQL의 통합이 어렵다.</li><li>•부분 범위에 대한 처리가 곤란해진다.</li><li>•여러 테이블을 통합한 뷰는 조회만 가능하다.</li><li>•UID(Unique Identifier, 식별자)의 유지 관리가 어렵다.</li></ul>

## 2. 물리 데이터베이스 설계-SEC\_15(논리 데이터 모델의 물리 데이터 모델 변환)

### 3) 슈퍼타입/서브타입을 테이블로 변환

예) 서브타입 기준 테이블 변환



슈퍼타입인 <접수> 개체에 있는 '신청자 이름', '접수', '수수료'가 서브타입인 <방문접수> 개체와 <인터넷 접수> 개체에 각각 추가되어 <방문접수>와 <인터넷접수> 테이블로 변환된다.

## 2. 물리 데이터베이스 설계-SEC\_15(논리 데이터 모델의 물리 데이터 모델 변환)

### 3) 슈퍼타입/서브타입을 테이블로 변환

#### ● 개별타입 기준 테이블 변환

- 개별타입 기준의 테이블 변환은 슈퍼타입과 서브타입들을 각각의 개별적인 테이블로 변환하는 것이다.
  - > 슈퍼타입과 서브타입 테이블들 사이에는 각각 1:1 관계가 형성된다.
  - > 개별타입 기준 테이블 변환을 적용하는 경우
    - 전체 데이터에 대한 처리가 빈번한 경우
    - 서브타입의 처리가 대부분 독립적으로 발생하는 경우
    - 통합하는 테이블의 컬럼 수가 많은 경우
    - 서브타입의 컬럼 수가 많은 경우
    - 트랜잭션이 주로 슈퍼타입에서 발생하는 경우
    - 슈퍼타입의 처리 범위가 넓고 빈번하게 발생하여 단일 테이블 클러스터링이 필요한 경우

#### ● 장, 단점

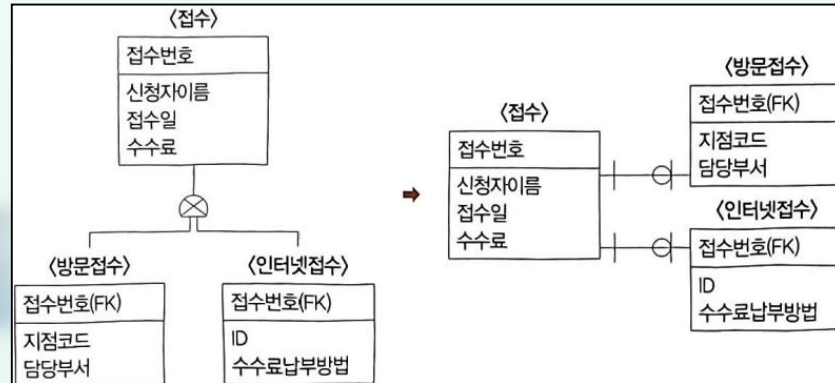
장점	•저장 공간이 상대적으로 작다. •슈퍼타입 또는 서브타입 각각의 테이블에 속한 정보만 조회하는 경우 문장 작성이 용이하다.
단점	슈퍼타입 또는 서브타입의 정보를 이 처리하면 항상 조인이 발생하여 성능이 저하된다.



## 2. 물리 데이터베이스 설계-SEC\_15(논리 데이터 모델의 물리 데이터 모델 변환)

### 3) 슈퍼타입/서브타입을 테이블로 변환

예) 개별타입 기준 테이블 변환



슈퍼타입의 <접수> 개체와 서브타입의 <방문접수>, <인터넷접수> 개체가 각각 <접수>, <방문접수>, <인터넷 접수> 테이블로 변환된다.

## 2. 물리 데이터베이스 설계-SEC\_15(논리 데이터 모델의 물리 데이터 모델 변환)

### 4) 속성을 컬럼으로 변환

; 논리 데이터 모델에서 정의한 속성을 물리 데이터 모델의 컬럼으로 변환한다.

#### ● 일반 속성 변환

- 속성과 컬럼은 명칭이 반드시 일치할 필요는 없으나, 개발자와 사용자 간 의사소통을 위하여 가능한 표준화된 약어를 사용하여 일치시키는 것이 좋다.
- 컬럼명은 SQL의 예약어(Reserved Word) 사용을 피한다.
- 컬럼명은 SQL의 가독성을 높이기 위해 가능한 한 짧게 지정한다.
- 복합 단어를 컬럼명으로 사용할 때는 미리 정의된 표준을 따른다.
- 테이블의 컬럼을 정의한 후에는 한 로우(Row)에 해당하는 샘플 데이터를 작성하여 컬럼의 적합성을 검증한다.

예) 일반 속성 변환

〈사원〉		〈사원〉					
사원번호	→	empnum	depnum	name	add	phon	email
부서번호(FK)		AK2015	Z01	홍길동	서울 마포구	010-2510-****	zmz512
이름		A003	R01	이춘식	서울 영등포구	010-9502-****	love104
주소		AK2018	T01	성봉호	경북 포항	010-1234-****	5120mama
전화번호							
이메일							

<사원> 엔티티의 '부서번호', '이름', '주소', '전화번호', '이메일' 속성이 <사원> 테이블의 각각의 컬럼으로 변환되었으며, 예시를 위한 데이터가 들어있다.

## 2. 물리 데이터베이스 설계-SEC\_15(논리 데이터 모델의 물리 데이터 모델 변환)

### 4) 속성을 컬럼으로 변환

- Primary UID를 기본키로 변환
  - 논리 데이터 모델에서의 Primary UID는 물리 데이터 모델의 기본키로 만든다.
- Primary UID(관계의 UID Bar)를 기본키로 변환
  - 다른 엔티티와의 관계로 인해 생성된 Primary UID는 물리 데이터 모델의 기본키로 만든다.
- Secondary(Alternate) UID를 유니크 키로 변환
  - 논리 모델링에서 정의된 Secondary UID 및 Alternate Key는 물리 모델에서 유니크 키로 만든다.

#### Primary UID

UID(Unique Identifier)는 식별자, Primary UID는 주 식별자를 의미한다.

UID Bar란 엔티티에 포함된 고유한 속성의 식별자(UID)가 아니라 다른 엔티티와의 관계로 인해 생성된 식별자(UID)를 의미한다.

유니크 키는 해당 속성에 입력된 값이 유일하다는 것을 보장하기 위한 제약 조건인 유니크(Unique) 속성이 설정된 키이다.

## 2. 물리 데이터베이스 설계-SEC\_15(논리 데이터 모델의 물리 데이터 모델 변환)

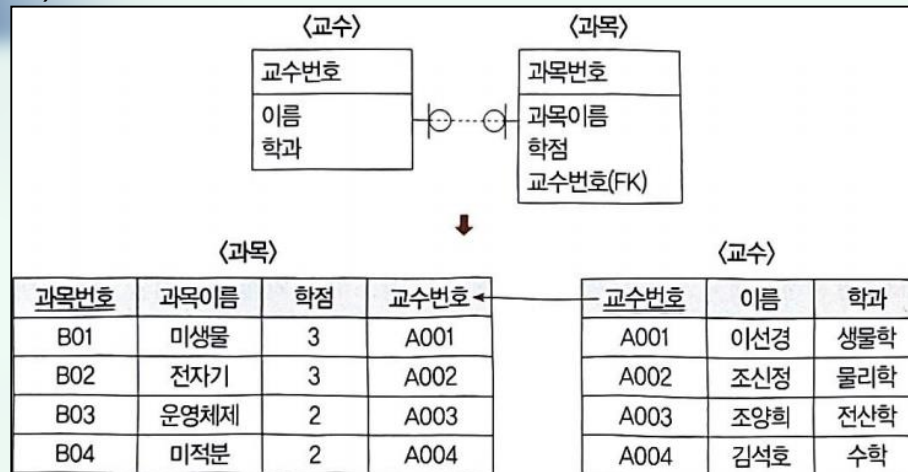
### 5) 관계를 외래키로 변환

; 논리 데이터 모델에서 정의된 관계는 기본키와 이를 참조하는 외래키로 변환한다.

● 다음은 개체 A, B로 이루어진 E-R 모델을 관계 테이블로 변환하는 방법이다.

- 1:1 관계 : 개체 A의 기본키를 개체 B의 외래키로 추가하거나 개체 B의 기본키를 개체 A의 외래키로 추가하여 표현한다.

예) 1:1 관계 변환



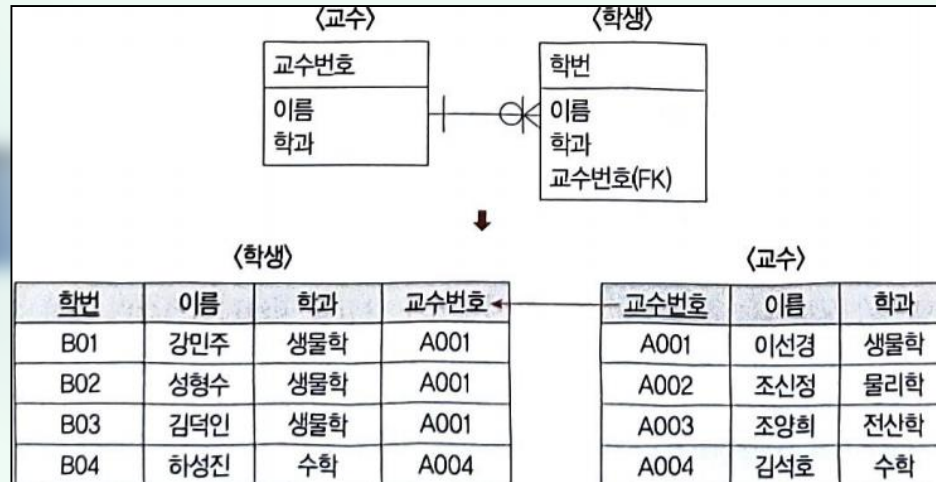
<교수> 테이블의 기본키인 '교수번호' 필드를 <과목> 테이블의 외래키로 추가한다.

## 2. 물리 데이터베이스 설계-SEC\_15(논리 데이터 모델의 물리 데이터 모델 변환)

### 5) 관계를 외래키로 변환

- 1:M 관계 : 개체 A의 기본키를 개체 B의 외래키로 추가하여 표현하거나 별도의 테이블로 표현한다.

예) 1:M 관계 변환



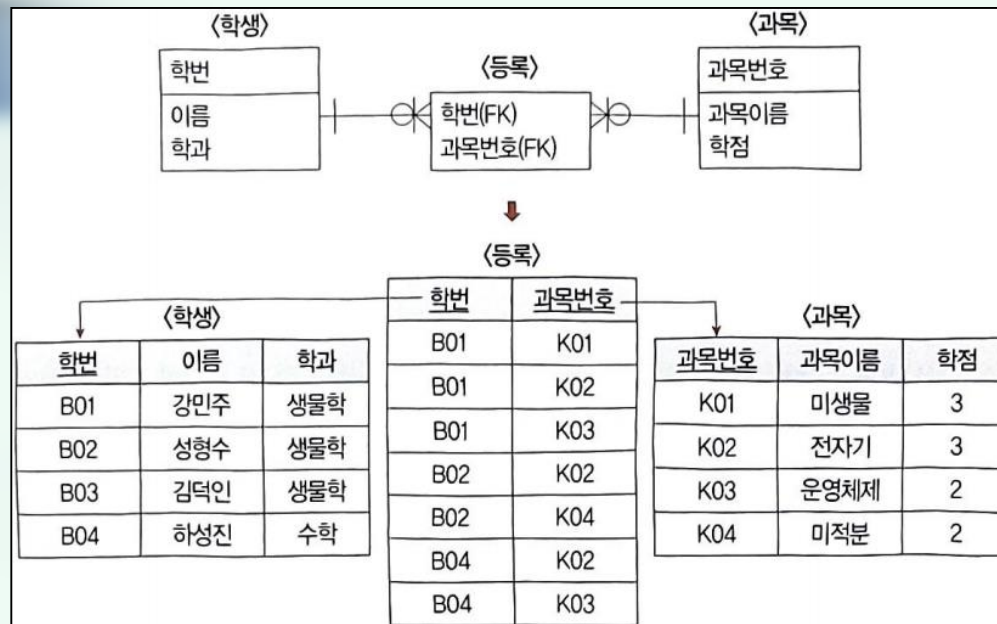
<교수> 테이블의 기본키인 '교수번호' 필드를 <학생> 테이블의 외래키로 추가한다.

## 2. 물리 데이터베이스 설계-SEC\_15(논리 데이터 모델의 물리 데이터 모델 변환)

### 5) 관계를 외래키로 변환

- N:M 관계 : 릴레이션 A와 B의 기본키를 모두 포함한 별도의 릴레이션으로 표현한다. 이때 생성된 별도의 릴레이션을 교차 릴레이션(Intersection Relation) 또는 교차 엔티티(Intersection Entity)라고 한다.

예) N:M 관계 변환



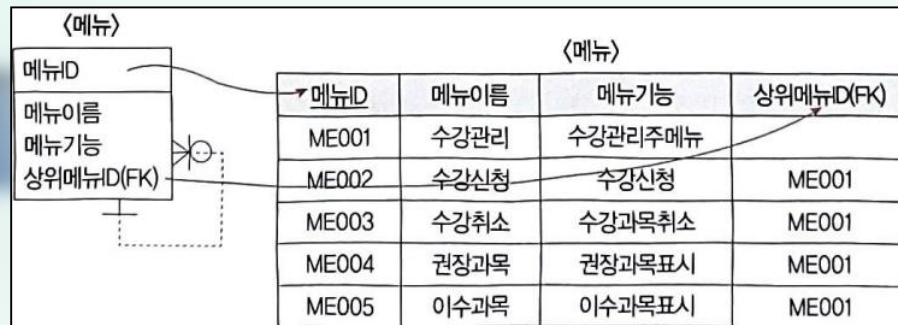
N:M의 관계는 관계를 별도의 테이블로 구성해서 표시해야 한다. <학생> 테이블의 기본키인 '학번' 필드와 <과목> 테이블의 기본키인 '과목번호' 필드를 이용하여 <등록> 테이블을 만든다.

## 2. 물리 데이터베이스 설계-SEC\_15(논리 데이터 모델의 물리 데이터 모델 변환)

### 5) 관계를 외래키로 변환

- 1:M 순환 관계: 개체 A에 개체 A의 기본키를 참조하는 외래키 컬럼을 추가하여 표현한다. 데이터의 계층 구조를 표현하기 위해 주로 사용된다.

예) 1:M 순환 관계 변환



<메뉴> 테이블의 기본키인 '메뉴 ID' 필드를 참조하는 외래키 '상위메뉴ID' 필드를 추가한다.

### 6) 관리 목적의 테이블/컬럼 추가

; 논리 데이터 모델에는 존재하지 않는 테이블이나 컬럼을 데이터베이스의 관리 혹은 데이터베이스를 이용하는 프로그래밍의 수행 속도를 향상시키기 위해 물리 데이터 모델에 추가할 수 있다.

예) 시스템 등록 일자, 시스템 번호 등

## 2. 물리 데이터베이스 설계-SEC\_15(논리 데이터 모델의 물리 데이터 모델 변환)

### 7) 데이터 타입 선택

; 논리 데이터 모델에서 정의된 논리적인 데이터 타입을 물리적인 DBMS의 물리적 특성과 성능을 고려하여 최적의 데이터 타입과 데이터의 최대 길이를 선택한다.

- 주요 타입에는 문자 타입(Character Type), 숫자 타입(Numeric Type), 날짜 타입(Date Type)이 있다.
- Oracle에서 자주 사용되는 데이터 유형

<b>CHAR</b>	고정길이 문자열 Data 최대 2,000Byte까지 저장 가능
<b>VARCHAR2</b>	가변길이 문자열 Data 최대 4,000Byte까지 저장 가능
<b>NUMBER</b>	38자릿수의 숫자 저장 가능
<b>DATE</b>	날짜 저장



# 물리 데이터베이스 설계-SEC\_15(논리 데이터 모델의 물리 데이터 모델 변환) 출제 예상 문제

## 출제 예상 문제(논리 데이터 모델의 물리 데이터 모델 변환)

1. 다음 중 논리 데이터베이스에서 설계한 객체들을 물리 데이터베이스로 변환하는 과정에 대한 설명으로 틀린 것은?

- ① 엔티티는 테이블로 변환한다.
- ② 속성은 컬럼으로 변환한다.
- ③ 주 식별자는 외래키로 변환한다.
- ④ 테이블명은 영문으로 지정한다.

### 변환 규칙

논리적 설계(데이터 모델링)	물리적 설계
엔티티(Entity)	테이블(Table)
속성(Attribute)	컬럼(Column)
주 식별자(Primary Identifier)	기본키(Primary Key)
외부 식별자(Foreign Identifier)	외래키(Foreign Key)
관계(Relationship)	관계(Relationship)

2. 논리 데이터 모델을 물리 데이터 모델로 변환하는 과정에서 슈퍼 타입과 서브 타입을 테이블로 변환하는 방법이 아닌 것은?

- ① 슈퍼 타입을 기준으로 하나의 테이블로 변환
- ② 서브 타입을 기준으로 하나의 테이블로 변환

3. 다음 중 논리 데이터 모델을 물리 데이터 모델로 변환할 때 슈퍼 타입/서브 타입 데이터 모델의 변환 방법으로 가장 부적절한 것은?

- ① 슈퍼 타입과 서브 타입을 각각의 독립적인 테이블로 변환하여 테이블간에 1:M 관계가 유지되도록 변환할 수 있다.
- ② 슈퍼 타입과 서브 타입을 통합하여 하나의 테이블로 변환할 수 있다.
- ③ 슈퍼 타입 속성들을 서브 타입에 각각 추가하여 서브 타입마다 하나의 테이블로 변환할 수 있다.
- ④ 슈퍼 타입을 기준으로 하나의 테이블로 변환하면 데이터 검색이 수월하고 조인이 감소한다.

### 1. 슈퍼 타입으로 변환하는 경우의 장단점

#### 장점

- 데이터의 액세스가 상대적으로 용이하다.
- 뷰를 이용하여 각각의 서버 타입만을 액세스하거나 수정할 수 있다.
- 서브 타입 구분이 없는 임의 집합에 대한 처리가 용이하다.
- 여러 테이블을 조인하지 않아도 되므로 수행 속도가 빨라진다.
- SQL문장 구성이 단순해진다.

#### 단점

- 테이블의 컬럼이 증가하므로 디스크 저장 공간이 증가한다.

## 물리 데이터베이스 설계-SEC\_15(논리 데이터 모델의 물리 데이터 모델 변환) 출제 예상 문제

출제 예상 문제(논리 데이터 모델의 물리 데이터 모델 변환)

5. 다음 중 논리 데이터 모델의 속성을 컬럼으로 변환하는 방법으로 틀린 것은?

- ① 가독성을 높이기 위해 SQL의 예약어를 컬럼명으로 사용한다.
- ② 컬럼명은 될 수 있으면 짧게 지정한다.
- ③ 복합 단어를 컬럼명으로 사용할 때는 미리 정의된 표준을 따른다.
- ④ 실제 테이블을 검증하기 위해 표본 데이터를 입력시킨다.

## 2. 물리 데이터베이스 설계-SEC\_16(물리 데이터 모델 품질 검토)

### 1) 물리 데이터 모델 품질 검토

; 물리 데이터 모델의 품질 검토는 물리 데이터 모델을 설계하고 데이터베이스 객체를 생성한 후 개발 단계로 넘어가기 전에 모델러와 이해관계자들이 모여 수행한다.

- 물리 데이터 모델은 시스템 성능에 직접적인 영향을 미치므로 향후 발생할 문제에 대해 면밀히 검토해야 한다.
- 물리 데이터 모델 품질 검토의 목적은 데이터베이스의 성능 향상과 오류 예방이다.
- 물리 데이터 모델을 검토하려면 모든 이해관계자가 동의하는 검토 기준이 필요하다.

### 2) 물리 데이터 모델 품질 기준

; 대표적인 물리 데이터 모델의 품질 기준으로는 정확성, 완전성, 준거성, 최신성, 일관성, 활용성이 있다.

정확성	데이터 모델이 요구사항이나 업무 규칙, 표기법에 따라 정확하게 표현되었음을 의미한다.
완전성	데이터 모델이 데이터 모델의 구성 요소를 누락 없이 정의하고 요구사항이나 업무 영역을 누락 없이 반영하였음을 의미한다.
준거성	데이터 모델이 데이터 표준, 표준화 규칙, 법적 요건 등을 정확하게 준수하였음을 의미한다.
최신성	데이터 모델이 최근의 이슈나 현행 시스템을 반영하고 있음을 의미한다.
일관성	데이터 모델이 표현상의 일관성을 유지하고 있음을 의미한다.
활용성	작성된 모델과 설명을 사용자가 충분히 이해할 수 있고, 업무 변화에 따른 데이터 구조의 변경이 최소화 될 수 있도록 설계되었음을 의미한다.

- 물리 데이터 모델의 품질 기준은 조직 혹은 업무 상황에 따라 가감하거나 변형하여 사용한다.

## 2. 물리 데이터베이스 설계-SEC\_16(물리 데이터 모델 품질 검토)

### 3) 물리 데이터 모델 품질 검토 항목

; 물리 데이터 모델의 품질 검토 항목은 물리 데이터 모델의 특성을 반영한 품질 기준을 작성한 후 이를 기반으로 작성한다.

- 물리 데이터 모델에 정의된 테이블, 컬럼, 무결성 제약 조건 등 물리 데이터 모델의 주요 구성 요소와 반정규화, 인덱스, 스토리지 등 물리 데이터 모델의 전반적인 것을 검토 항목으로 작성한다.
- 체크리스트는 물리 데이터 모델의 품질 검토를 용이하게 수행할 수 있도록 한다.

예) 데이터 품질 체크리스트

대상	검토 항목	검토 내용	평가	비고
테이블	명명	명명 규칙을 준수하였는가?		
		의미전달이 명확한 명칭을 사용하였는가?		
		테이블 한글명은 엔티티 명칭과 일치하는가?		
	설명	데이터 집합 구성상의 특징이 설명되어 있는가?		
	정의	테이블 형태는 성능을 고려하여 결정되었는가?		
		테이블 생성 관련 파라미터들은 적절하고 충분하게 정의되었는가?		
	권한	테이블 생성/변경/삭제 시 메타 데이터 권한을 정의하였는가?		
		테이블에 대한 접근 권한을 정의하였는가?		

## 2. 물리 데이터베이스 설계-SEC\_16(물리 데이터 모델 품질 검토)

### 4) 물리 데이터 모델의 품질 검토 순서

- ① 데이터 품질 정책 및 기준을 확인한다.
- ② 물리 데이터 품질의 특성에 따라 품질 기준을 작성한다.
- ③ 데이터 품질 기준에 따라 체크리스트를 작성한다.
- ④ 논리 데이터 모델과 물리 데이터 모델을 비교한다.
- ⑤ 각 모델링 단계의 모델러와 이해관계자가 품질 검토를 수행한다.
- ⑥ 모델러와 이해관계자가 작성한 체크리스트 내용을 종합하여 물리 데이터베이스 모델의 품질 검토 보고서를 작성한다.

# 물리 데이터베이스 설계-SEC\_16(물리 데이터 모델 품질 검토) 출제 예상 문제

## 출제 예상 문제(물리 데이터 모델 품질 검토)

1. 다음 중 물리 데이터 모델의 품질 검토에 대한 내용으로 가장 옳지 않은 것은?

- ① 물리 데이터 모델 품질 검토의 목적은 데이터베이스의 성능 향상과 오류 예방이다.
- ② 물리 데이터 모델을 검토하려면 모든 이해관계자가 동의하는 검토 기준이 필요하다.
- ③ 물리 데이터 모델의 품질 기준에는 정확성, 완전성, 준거성 등 이 있다.
- ④ 데이터 품질 기준 및 정책이 수립된 경우에는 물리 데이터 모델의 품질 기준을 변경하거나 수정해서는 안된다.

### 물리 데이터 모델 품질 검토

; 물리 데이터 모델의 품질 검토는 물리 데이터 모델을 설계하고 데이터베이스 객체를 생성한 후 개발 단계로 넘어가기 전에 모델러와 관련된 이해관계자들이 모여서 수행한다.

- 물리 데이터 모델은 시스템 성능에 직접적인 영향을 미치므로 향후 발생할 문제에 대해 면밀히 검토해야 한다.
- 물리 데이터 모델 품질 검토의 목적은 데이터베이스의 성능 향상과

3. 다음 중 물리 데이터 모델의 품질 검토 항목이 아닌 것은?

- ① 트리거
- ② 테이블
- ③ 무결성 제약 조건
- ④ 컬럼

트리거란 특정 테이블의 데이터에 삽입, 삭제, 수정과 같은 이벤트가 일어나면 묵시적으로 자동으로 수행되는 저장 프로시저라고 볼 수 있다.

### 물리 데이터 모델 품질 검토 항목

; 물리 데이터 모델의 품질 검토 항목은 물리 데이터 모델의 특성을 반영한 품질 기준을 작성한 후 이를 기반으로 작성한다.

- 물리 데이터 모델에 정의된 테이블, 컬럼, 무결성 제약조건 등 물리 데이터 모델의 주요 구성 요소와 반정규화, 인덱스, 스토리지 등 물리 데이터 모델의 전반적인 것을 검토 항목으로 작성한다.

- 체크리스트는 물리 데이터 모델의 품질 검토를 용이하게 수행할 수 있도록 한다.

A close-up, low-angle shot of a white car's side mirror and door handle against a light blue sky. The car is on the left side of the frame, and the sky is on the right. The text "감사합니다." is overlaid on the right side of the image.

**감사합니다.**