

121 회 기출풀이

정보관리기술사

kpc SMART KPC
한국생산성본부

교육 문의 및 상담 : 황 의 선



- Tel : 02) 724-1831
- Fax : 02) 724-1875
- E-mail : kpcitpe@kpc.or.kr
- Web Site : www.kpc.or.kr

cafe.naver.com/81th



[토요일 명품심화반]

- 단합반(SPP 반) (안경환 PE @ KPC)
→ luverleon@naver.com
- FB(Future Builders) (윤정호 PE&이영룡 PE@ KPC)
→ hayangomiya@naver.com / 010-4126-8161
- 정 주 행 (서정훈 PE&문광석 PE @ KPC)
→ neom@naver.com / 010-6244-6843
- ITPE Makers (박제일 PE @ KPC)
→ kftc@naver.com / 010-2267-4539
- 공 감 반 (공수재 PE @ KPC)
→ Ksujae22@naver.com / 010-8917-5742
- KPC95 반 (채 PE&전지웅 PE@ KPC)
→ lovevani1@naver.com

[일요일 명품심화반]

- T.O.P 반 (유술사 PE @ KPC)
→ itpe_you@naver.com / 010-5862-8469
- N S 반 (강정배 PE&박주형 PE @ 강남아지트)
→ joohyung1002@naver.com / 010-6700-6883

[평일 명품심화반]

- 강남평일야간반 (강정배 PE&전일 PE @ 강남아지트/ 화, 금)
→ nikki6@hanmail.net / 010-2353-7203



국가기술자격 기술사 시험문제

기술사 제 121 회

제 2 교시 (시험시간: 100분)

분야	정보통신	종목	정보관리기술사	수험번호		성명	
----	------	----	---------	------	--	----	--

* 다음 문제 중 4문제를 선택하여 설명하시오. (각25점)

1. 기존의 데이터웨어하우스와 빅데이터의 핵심 기술인 하둡을 통합해 추상화한 정보분석 아키텍처로서 로지컬 데이터웨어하우스(LDW : Logical Data Warehouse)에 대하여 검토하고 있다. 다음에 대하여 설명하시오.
 - 가. LDW의 정의 및 특징
 - 나. LDW의 구성요소
 - 다. LDW의 아키텍처
 - 라. LDW의 활용방안
2. DBMS 병행수행(Concurrency)제어가 되지 않는 경우 발생할 수 있는 3가지 문제와 그 해결방법에 대하여 예시를 들어 각각 설명하시오.
3. 침해사고 대응 측면과 디지털 감사(Audit) 측면에서 디지털포렌식의 필요성을 설명하고, 디지털포렌식 절차 및 활용되는 기술을 설명하시오.

국가기술자격 기술사 시험문제

기술사 제 121 회

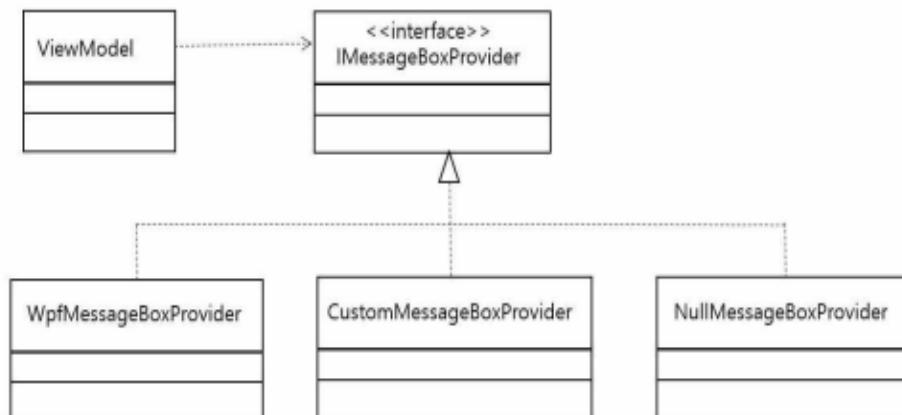
제 2 교시 (시험시간: 100분)

분야	정보통신	종목	정보관리기술사	수험번호	성명

4. 소프트웨어 재사용성과 유지보수 향상을 위하여 객체지향 설계 5대 원칙을 적용하고 있다. 다음에 대하여 답하시오.

- 가. 의존성 역전 원칙(Dependency Inversion Principle)을 설명하시오.
- 나. 의존성 주입(Dependency Injection)을 구현하는 3가지 방식을 설명하고 각 방식별 아래의 조건을 고려하여 구현 예시를 작성하시오.

조건)



국가기술자격 기술사 시험문제

기술사 제 121 회

제 2 교시 (시험시간: 100분)

분야	정보통신	종목	정보관리기술사	수험번호		성명	

5. 최근 부하 분산을 위하여 클라우드 서비스에서 Auto Scale 기능을 많이 활용한다. 다음에 대하여 설명하시오.

- 가. Auto Scale Up과 Auto Scale Out 아키텍처 비교
- 나. 스케줄 기반의 Auto Scale과 부하 기반의 Auto Scale 기능 비교

6. CISO(Chief Information Security Officer)의 자격요건과 겸직 여부 등을 개정한 「정보통신망 이용촉진 및 정보보호 등에 관한 법률」이 2020년 1월 1일부터 적용되고 있다. 다음에 대하여 설명하시오.

- 가. CISO의 직무
- 나. CISO의 지정·신고 의무대상자
- 다. CISO의 겸직 제한 대상
- 라. CISO의 일반 자격요건

1	LDW(Logical Data Warehouse)
문제	<p>1. 기존의 데이터웨어하우스와 빅데이터의 핵심 기술인 하둡을 통합해 추상화한 정보분석 아키텍처로서 로지컬 데이터웨어하우스(LDW: Logical Data Warehouse)에 대하여 검토하고 있다. 다음에 대하여 설명하시오.</p> <p>가. LDW 의 정의 및 특징 나. LDW 의 구성요소 다. LDW 의 아키텍처 라. LDW 의 활용방안</p>
도메인	디지털 서비스
정의	정형, 비정형 데이터의 저장 및 병렬 처리를 위해 기존 EDW 와 Hadoop 환경을 통합 하여 구성한 데이터 관리 아키텍처
키워드	EDW, Taxonomy, Ontology, Hadoop, Datalake
출제의도분석	기존 DW 보다 확장된 개념인 LDW 에 대한 지식 검증
답안작성 전략	물어본 것에 정확히 답변 모르는 경우 DW 에서 개념적으로 유추 하여 방어적으로 작성
참고문헌	Logical Data Warehouse(LDW) (https://blogs.gartner.com/henry-cook/2017/05/23/building-the-logical-data-warehouse-ldw/)
풀이 기술사님	서경석 기술사 (제 119 회 정보관리 기술사 / akslemf@naver.com)

1. 빅데이터 시대의 효율적 정보 처리 아키텍처, LDW의 정의 및 특징

가. LDW(Logical Data Warehouse)의 정의

- 정형, 비정형 데이터의 저장 및 병렬 처리를 위해 기존 EDW 와 Hadoop 환경을 통합 하여 구성한 데이터 관리 아키텍처
- 데이터의 재배치나 변환 없이 조회 가능하도록 지원 하는 기존 DW 에 빅데이터를 포함하는 확장된 개념의 아키텍처

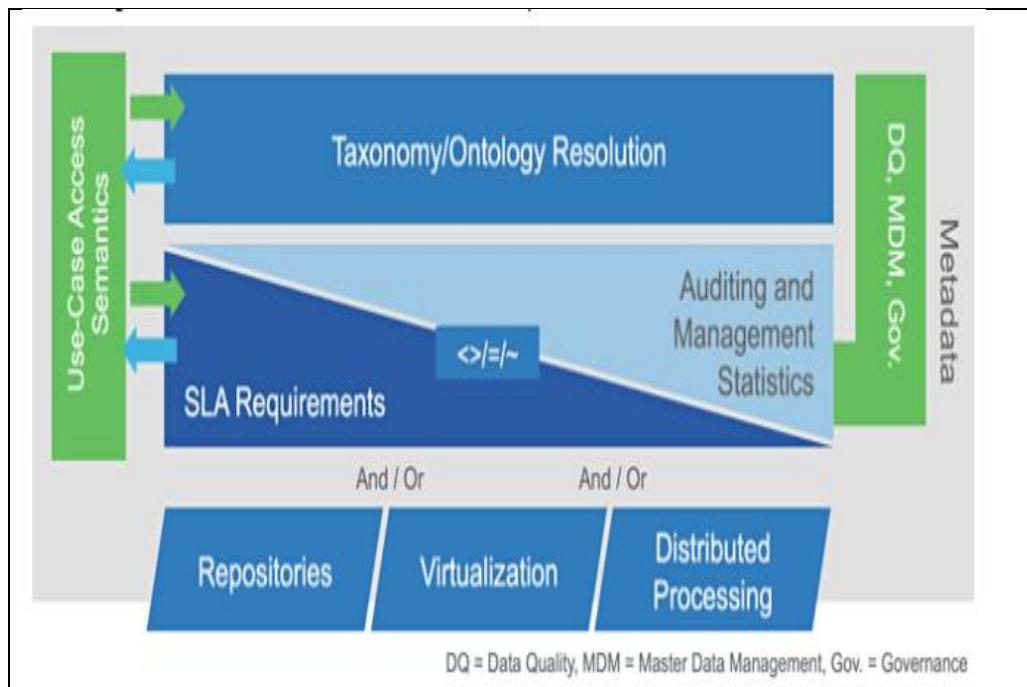
나. LDW의 특징

특징	설명
비용 절감	<ul style="list-style-type: none"> - 오픈 소스 기반의 Hadoop 을 이용 하여 관리 비용 절감 - 하드웨어/소프트웨어 증설에 따른 비용 절감 가능
처리 속도 증가	<ul style="list-style-type: none"> - 데이터 병렬 처리 환경 구축에 따른 처리 속도 증가
분석의 적시성	<ul style="list-style-type: none"> - 대용량 데이터 분산 병렬 처리 통한 작업 시간 단축 - 분석에 필요한 데이터 적시 제공 가능
Scale out 가능	<ul style="list-style-type: none"> - 하둡 기반 빅데이터 시스템들의 Scale out 용이
데이터 효용성 증대	<ul style="list-style-type: none"> - DW 에서 관리 되는 고품질 정형 데이터들과 비정형 데이터를 통합 - 통합 데이터 분석을 통한 부가 가치 창출 가능

- LDW 는 기존 DW 보다 대용량 데이터 처리 지원에 적합한 형태의 DW 개념

2. LDW의 구성도 및 구성 요소

가. LDW의 구성도



- LDW는 단순한 데이터의 저장 장소가 아닌 데이터의 처리, 관리까지 지원하는 통합 체계

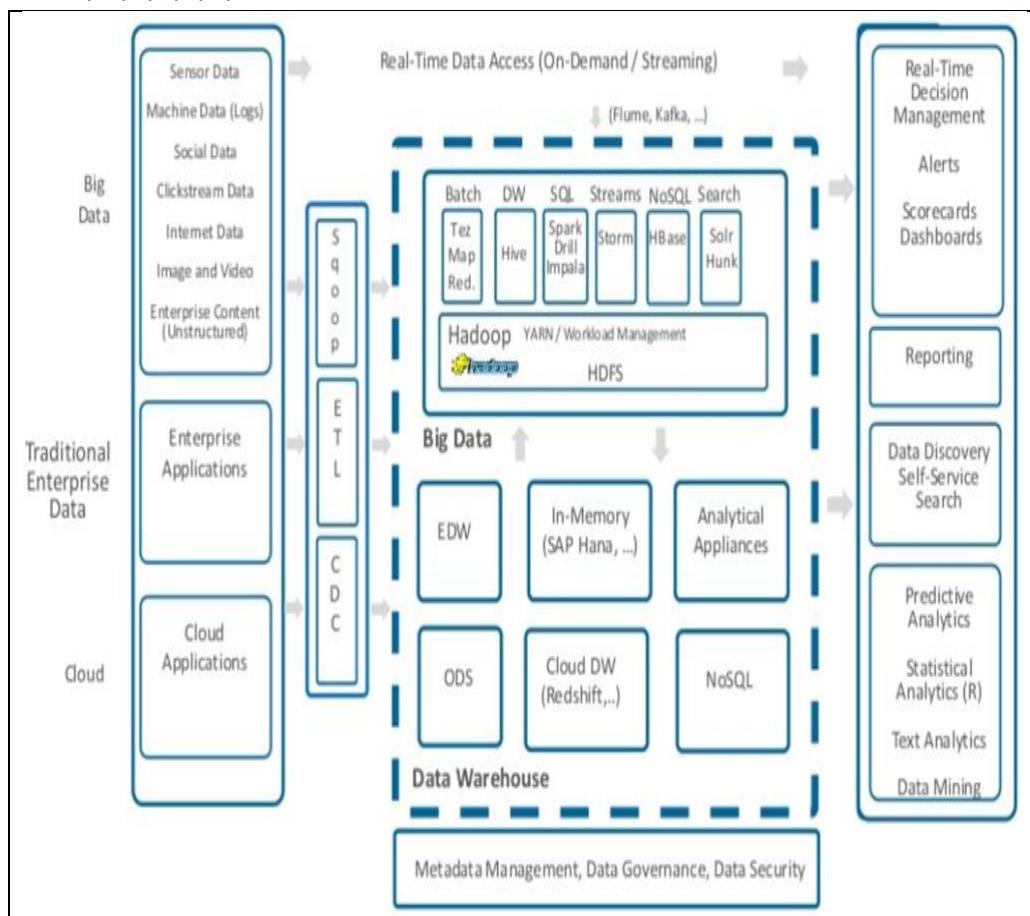
나. LDW의 구성 요소

구분	세부	설명
저장, 처리	Repository Management	<ul style="list-style-type: none"> - 대량의 정형, 비정형 데이터의 효율적 관리 위한 저장소 - 기존 DW 환경에 HDFS 환경 통합 사용
	Data Virtualization	<ul style="list-style-type: none"> - 모든 데이터 탑재, 구조에 대해 View를 제공 하는 기술 - 원본 데이터를 가상의 View를 통한 데이터 활용 지원
	Distributed Processing	<ul style="list-style-type: none"> - 대량의 데이터에 대한 서버 자원 및 데이터에 대한 분산 처리를 통해 대량 데이터 처리가 가능 - 분산 처리 완료 후 결과값을 취합하여 요청자에게 전달
기준 데이터	Metadata Management	<ul style="list-style-type: none"> - 대량의 데이터 활용을 위한 관리 기준 필요 - 기술적, 업무적, 정보 메타 데이터로 구분 관리 - 품질관리, 데이터 거버넌스, 마스터 데이터 관리에 사용
	Taxonomy/Ontology resolution	<ul style="list-style-type: none"> - 연관 데이터 결합, 분석을 위해 필요한 데이터 간 연관 정보, 데이터 집합에 대한 분류 체계 - Metadata management의 정보 메타 데이터에 해당
운영	Auditing and performance services	<ul style="list-style-type: none"> - 원천 데이터로부터 데이터를 추출, 가공, 저장하는 작업과 처리 작업의 성능 개선 작업 - 모니터링 및 로그 관리 통한 서비스 개선 활용
	SLA management	<ul style="list-style-type: none"> - 사용자들의 서비스 수준 관리 - 분석 성능, 리포트 조회 성능, 데이터 정합성 수준 등

- 데이터의 양과 형태가 다양해짐에 따라 기준 데이터 및 성능 향상이 더욱 강조됨.

3. LDW의 아키텍처

가. LDW의 아키텍처 구성도



- 원천 데이터, 분산 처리 환경, 시각화, 분석에 이르는 포괄적인 범위의 아키텍처

나. LDW의 아키텍처 기술 요소

구분	기술 요소	설명
원천 데이터	소셜 데이터	- 다양한 플랫폼으로부터 발생하는 Social Data
	스트리밍 데이터	- 실시간성으로 처리가 필요한 유형의 데이터
	센싱 데이터	- IoT 플랫폼, 스마트 팩토리 등에서 발생하는 대량 데이터
수집	CDC	- DB에서 변경이 발생하는 데이터를 지속적으로 수집
	ETL	- DW의 데이터를 추출, 변환 후 적재 하는 기술
저장, 처리	HDFS	- 파일을 블록 단위로 나누어 다중 노드에 분산 저장, 처리
	ODS	- 원천 데이터 가공하여 운영 보고, 제어, 의사 결정 활용
	In-memory DB	- 메인 메모리를 활용하여 고속 처리가 가능한 DB
	No SQL	- 빅데이터를 처리하기 위한 비관계형 데이터 관리 체계
응용	BI	- 데이터 분석 작업을 통한 기업 의사 결정 지원 기술
	Data Analysis	- 원천 데이터 분석, 마이닝, 분류화 작업 통한 가치 창출
	Visualization	- 데이터 활용도 향상 위한 대시보드 구현 등 가시화 기술

- 대용량 정형, 비정형 데이터 처리를 위한 분산 처리 및 실시간 처리를 위한 기술 적용 필요

4. LDW의 활용 방안

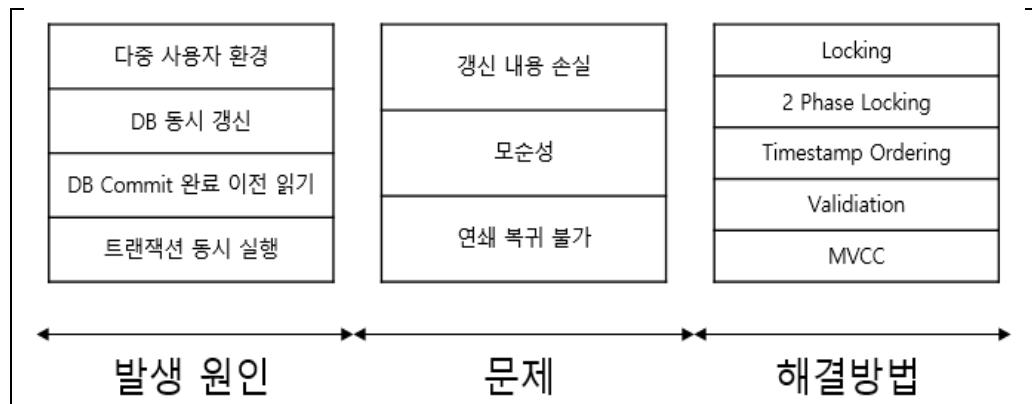
구분	설명
IoT 플랫폼	- 센서, 스마트 디바이스로부터 발생하는 빅데이터 활용 플랫폼 구성 - M2M 기반 서비스, 헬스케어, 실시간 제어 서비스 등
Hybrid DW	- 기존 DW의 성능, 비용적 한계 극복을 위한 추가적인 DW 구성 - 대량 데이터를 위한 LDW와 정제된 데이터를 위한 DW를 Hybrid 형태로 구성 가능
Data Analytics	- 기존에 분석 하지 못하던 비정형, 대량의 데이터 분석 가능 - 분석 영역 확장에 따른 기업 이익 극대화 기대
Data Virtualization	- 데이터 이미징 작업, 별도의 가상화 도구를 활용하여 원본 데이터는 유지한 상태로, 필요 시점마다 데이터를 가상화 활용 가능 - 가상화한 데이터를 활용한 DR 센터 구축, 데이터 백업 작업 등 활용

- LDW는 데이터의 지속적인 증가에 따라 데이터 레이크 등 연계 기술의 지속 발전 예상.

"끝"

2	DBMS 병행 수행 제어(Concurrency Control)
문제	2. DBMS 병행수행(Concurrency)제어가 되지 않는 경우 발생할 수 있는 3 가지 문제와 그 해결방법에 대하여 예시를 들어 각각 설명하시오.
도메인	DB
정의	- 다중 사용자 환경을 지원하는 데이터베이스 시스템에서 여러 트랜잭션들이 동시에 실행 되어도 데이터의 일관성을 유지 할 수 있도록 지원하는 기능
키워드	갱신 손실, 모순성, 연쇄 복귀 불가, 현황 파악 오류, Locking, 2PL, Timestamp, Validation, MVCC
출제의도분석	DB 빈출 토픽으로써 기본적인 내용에 대한 습득 여부 확인
답안작성 전략	문제점 및 해결 방안의 예시 부분 상세히 작성
참고문헌	동시성제어(http://www.dbguide.net/db.db?cmd=view&boardUid=148217&boardConfigUid=9&boardIdx=138&boardStep=1)
풀이 기술사님	서경석 기술사 (제 119 회 정보관리 기술사 / akslemf@naver.com)

1. 다중 사용자 환경 데이터 베이스의 무결성 확보, DBMS 병행 수행 제어의 개요



- 다중 사용자 환경을 지원하는 데이터베이스 시스템에서 여러 트랜잭션들이 동시에 실행되어도 데이터의 일관성을 유지 할 수 있도록 지원하는 기능

2. DBMS 병행 수행 제어가 되지 않는 경우 발생할 수 있는 3 가지 문제 및 예시

가. 3 가지 문제의 개념 및 발생 시점

구분	개념	발생 시점
갱신 내용 손실	- 트랜잭션들이 동일 데이터를 동시에 갱신할 경우 발생하는 문제	- 트랜잭션1이 데이터를 갱신한 후 Commit하기 전에 트랜잭션2가 갱신 값을 덮어쓰는 경우
모순성	- 두 트랜잭션이 동시에 실행할 때 DB가 모순된 상태로 남는 문제	- 복수의 사용자가 동시에 DB를 Access 하여 갱신한 결과 값이 상호 일치하지 않거나 모순된 경우

연쇄 복귀 불가	- 특정 트랜잭션이 처리를 취소할 경우 다른 트랜잭션이 선처리한 부분에 대해 취소 불가능한 문제	- 트랜잭션1, 트랜잭션2가 종료 후 트랜잭션1을 Rollback 하는 경우 취소 불가능
----------------	---	---

- 동시성 제어가 되지 않을 경우 추가적으로 현황 파악 오류도 발생 가능.

나. 3 가지 문제 예시 및 설명

구분	예시	설명
갱신 내용 손실	<p>Time sequence:</p> <ul style="list-style-type: none"> T1: Read(X) → X = 500 T1: X = X + 100 → X = 600 T2: Read(X) → X = 600 T2: X = X * 2 → X = 1000 	<ul style="list-style-type: none"> - T1과 T2가 순서 없이 동시에 갱신을 시도하여 늦게 일어난 T2의 Write 연산으로 인해 T1의 갱신이 무효화되었음
모순성	<p>Time sequence:</p> <ul style="list-style-type: none"> T1: Read(X) → X = 500, Y = 500 T1: X = X + 100 → X = 600, Y = 500 T2: Read(X) → X = 600, Y = 500 T2: X = X * 2 → X = 1200, Y = 500 T1: Read(Y) → Y = 500 T1: Y = Y + 100 → Y = 1000 T2: Read(Y) → Y = 1000 T2: Y = Y * 2 → Y = 1100 	<ul style="list-style-type: none"> - T1의 예상 결과값은 x=600, y=600이며, T2의 예상 값은 x=1000, y=1000 이었으나, 실제 최종 결과 값은 x=1200, y=1100으로 모순되는 결과 가져옴
연쇄 복귀 불가	<p>Time sequence:</p> <ul style="list-style-type: none"> T1: Read(X) → X = 500, Y = 500 T1: X = X + 100 → X = 600, Y = 500 T2: Read(X) → X = 600, Y = 500 T2: X = X * 2 → X = 1200, Y = 500 T1: Read(Y) → Y = 500 Rollback T1 	<ul style="list-style-type: none"> - T1의 Read(Y) 이후에 Fail시 Roll back 해야 하는 경우 발생 가정 시 T1의 개시 최초 상태인 x=500, y=500인 상태로 복귀해야 하지만 T2가 이미 트랜잭션을 완료하고 시스템을 떠난 상태이기 때문에 초기 상태로 복귀 불가함

- 병행 수행 제어가 되지 않을 경우 DB 일관성에 문제가 발생 하므로 해결 방안 필요.

3. DBMS 병행 수행 제어 문제점 해결 방법의 개요 및 예시

가. DBMS 병행 수행 제어 문제점 해결 방법의 개요

구분	개념	비고
Locking	<ul style="list-style-type: none"> - 트랜잭션이 사용하는 자원에 대하여 상호 배제 기능을 제공하는 기법 - 상호 배제는 특정 트랜잭션이 Lock을 설정하면, Unlock 할 때까지 독점적으로 사용할 수 있는 기법 	<ul style="list-style-type: none"> - Shared Lock 공유 Lock 트랜잭션은 다른 트랜잭션도 Read만 실행 가능 - Exclusive Lock 다른 트랜잭션은 Read와 Write 모두 불가능
2 Phase Locking	<ul style="list-style-type: none"> - 트랜잭션의 Lock과 Unlock 연산을 확장단계와 수축 단계로 구분하여 수행하는 기법 	<ul style="list-style-type: none"> - 확장단계 : 트랜잭션은 lock만 수행, unlock 수행 불가 - 수축단계 : 트랜잭션은 unlock만 수행 가능, lock은 수행 불가
Timestamp Ordering	<ul style="list-style-type: none"> - 트랜잭션에 식별자인 타임 스탬프를 지정하여 트랜잭션 간의 직렬화를 보장 하는 기법 	<ul style="list-style-type: none"> - 시스템 시계 사용법 시스템 시계 값을 타임스탬프 값으로 부여 - 논리적 계수기 (counter) 사용법 트랜잭션 발생 시 카운터의 값을 타임스탬프 값으로 부여
MVCC	<ul style="list-style-type: none"> - 트랜잭션 실행 시 해당 트랜잭션의 타임스탬프와 접근 대상 데이터의 여러 버전의 타임스탬프를 비교, 직렬가능성 보장 버전을 선택, 접근 하는 기법 	<ul style="list-style-type: none"> - read_TS(xk) 버전 xk를 성공적으로 읽은 트랜잭션 중 가장 최근 스탬프 - write_TS(xk) 버전 xk를 생성한 트랜잭션의 타임스탬프
Validation	<ul style="list-style-type: none"> - 트랜잭션 수행 동안은 어떠한 검사도 하지 않고, 트랜잭션 종료 시 일괄적으로 검사하는 기법 	<ul style="list-style-type: none"> - 판독 단계 트랜잭션의 모든 갱신은 사본 수행, 실제 DB는 수행하지 않음 - 확인 단계 판독 단계 트랜잭션 실행 결과의 직렬 가능성 위반여부 확인 - 기록 단계 트랜잭션의 실행 결과를 반영, 오류 발생 시실행 결과 취소 후 복귀

- 트랜잭션 발생 빈도, 수행 시간, 중요성 등을 고려 하여 적합한 해결 방법 적용 필요

나. DBMS 병행 수행 제어 문제점 해결 방법의 예시

구분	예시	설명																		
Locking	<pre> graph TD T1[Read(X) X = X + 100 Lock(X)] --> X1[X = 500] X1 --> T2[Read(X) X = X * 2] T2 --> X2[X = 600] X2 --> T1[Write(X) Unlock(X)] T1 --> X3[X = 1200] </pre> <p style="text-align: center;">시간</p>	<ul style="list-style-type: none"> - T1 실행 시 Lock 설정 후 T1 완료 후 Unlock 처리 하여 T2가 T1이 완료 되기 전에 반영 되지 않아 X가 정상적으로 1200으로 출력 																		
2 Phase Locking	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>시간</th> <th>트랜잭션 T₁</th> <th>트랜잭션 T₂</th> </tr> </thead> <tbody> <tr> <td rowspan="2" style="vertical-align: middle; text-align: center;">↓</td> <td> lock(X); read(X); X = X + 1000; (3000 + 1000) write(X); lock(Y); unlock(X); </td> <td></td> </tr> <tr> <td></td> <td> lock(X); read(X); X = X * 0.5; (4000 * 0.5) write(X); </td> </tr> <tr> <td rowspan="2" style="vertical-align: middle; text-align: center;">↓</td> <td> read(Y); Y = Y + 1000; (3000 + 1000) write(Y); unlock(Y); </td> <td> lock(Y); unlock(X); read(Y); Y = Y * 0.5; (4000 * 0.5) write(Y); unlock(Y); </td> </tr> </tbody> </table>	시간	트랜잭션 T ₁	트랜잭션 T ₂	↓	lock(X); read(X); X = X + 1000; (3000 + 1000) write(X); lock(Y); unlock(X);			lock(X); read(X); X = X * 0.5; (4000 * 0.5) write(X);	↓	read(Y); Y = Y + 1000; (3000 + 1000) write(Y); unlock(Y);	lock(Y); unlock(X); read(Y); Y = Y * 0.5; (4000 * 0.5) write(Y); unlock(Y);	<ul style="list-style-type: none"> - T1 수행 전 x에 lock 설정, x+1000 계산 후 write를 이용 하여 DB에 x의 변경된 값 입력 후 unlock 처리, T2도 lock, read, unlock 순으로 처리 하여 모순성이 발생 하지 않도록 처리 							
시간	트랜잭션 T ₁	트랜잭션 T ₂																		
↓	lock(X); read(X); X = X + 1000; (3000 + 1000) write(X); lock(Y); unlock(X);																			
		lock(X); read(X); X = X * 0.5; (4000 * 0.5) write(X);																		
↓	read(Y); Y = Y + 1000; (3000 + 1000) write(Y); unlock(Y);	lock(Y); unlock(X); read(Y); Y = Y * 0.5; (4000 * 0.5) write(Y); unlock(Y);																		
	Timestamp Ordering	<p style="text-align: center;">도착시간</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>T_s(T₃) = 1201</td> <td>T_s(T₂) = 1158</td> <td>T_s(T₁) = 1150</td> </tr> <tr> <td colspan="3" style="text-align: center;">Timestamp</td> </tr> </table> <p style="text-align: center;">스케줄링</p>	T _s (T ₃) = 1201	T _s (T ₂) = 1158	T _s (T ₁) = 1150	Timestamp			<ul style="list-style-type: none"> - T1, T2, T3의 트랜잭션에 시스템 시계나 Counter를 이용 하여 Timestamp 부여 후 스케줄러에서 해당 Timestamp 기준으로 직렬화 											
T _s (T ₃) = 1201	T _s (T ₂) = 1158	T _s (T ₁) = 1150																		
Timestamp																				
MVCC	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td>Txn#1 Tid: 20 Write(B)</td> <td>Txn#2 Tid: 30 Read(A)</td> </tr> </table> <table border="1" style="display: inline-table; vertical-align: middle; margin-left: 10px;"> <thead> <tr> <th></th> <th>Txn-id</th> <th>Begin-ts</th> <th>End-ts</th> </tr> </thead> <tbody> <tr> <td>Ax</td> <td>0</td> <td>10</td> <td>25</td> </tr> <tr> <td>Ax+1</td> <td>0</td> <td>25</td> <td>40</td> </tr> <tr> <td>Ax+2</td> <td>0</td> <td>50</td> <td>INF</td> </tr> </tbody> </table>	Txn#1 Tid: 20 Write(B)	Txn#2 Tid: 30 Read(A)		Txn-id	Begin-ts	End-ts	Ax	0	10	25	Ax+1	0	25	40	Ax+2	0	50	INF	<ul style="list-style-type: none"> - Txn#2는 오브젝트 A를 Read 할 때, 세가지 버전 Ax, Ax+1와 Ax+2중 visible하면서 최신
Txn#1 Tid: 20 Write(B)	Txn#2 Tid: 30 Read(A)																			
	Txn-id	Begin-ts	End-ts																	
Ax	0	10	25																	
Ax+1	0	25	40																	
Ax+2	0	50	INF																	

Notes

		버전인 Ax+1을 Read
Validation	<p>Read Phase Validation Phase Execution Phase</p> <p>T1 → Start time : 0516 Finish time : 0517</p> <p>T2 → Start time : 0519 Finish time : 0520</p> <p>T2의 Start time이 T1의 Finish time 이후 이므로 두 트랜잭션 간의 간섭은 없는 것으로 확인</p> <p>T1 Commit T2 Commit</p>	- T1, T2를 수행 후 두 트랜잭션 간의 Start, Finish time을 통한 검증 후 실행 단계에서 Write 및 Commit 진행

- 병행 수행 제어 중 교착 상태가 발생 하지 않도록 점검 후 적용 필요

4. DBMS 병행 수행 제어를 위한 고려 사항

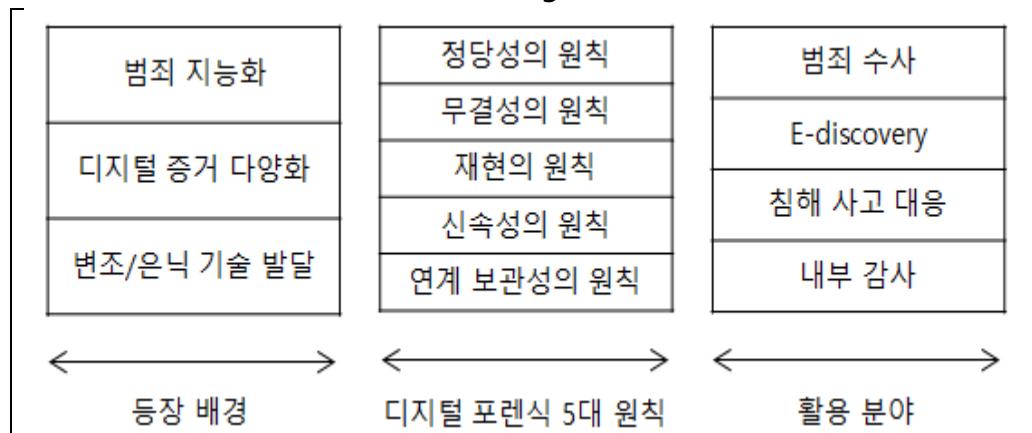
고려 사항	세부	설명
인프라	DBMS	<ul style="list-style-type: none"> - 시스템에 적합한 DBMS 선택 필요 - Implicit Lock, 자동 Rollback, 자연 갱신 기능 지원 확인
	분산 처리	<ul style="list-style-type: none"> - 다중 사용자, 트랜잭션 처리 위한 분산 처리 고려 - 수행 시간 단축 및 성능 향상 통한 문제 해결
관리	시스템 환경	<ul style="list-style-type: none"> - 시스템 사용 환경 분석을 통한 문제 발생 최소화 - 사용자 수, 동시 접속자 수, 부하 분석 등 진행
	인력 교육	<ul style="list-style-type: none"> - 개발 인력 교육을 통한 충돌 발생 최소화 Coding - 현업 교육을 통한 배치잡, DB 스케줄링 최적화

- 병행 수행 제어 문제 발생의 근본적 감소를 위한 대책 수립 필요

"끝"

3	디지털 포렌식(Digital Forensic)
문제	3. 침해사고 대응 측면과 디지털 감사(Audit) 측면에서 디지털 포렌식의 필요성을 설명하고, 디지털포렌식 절차 및 활용되는 기술을 설명하시오
도메인	보안
정의	- PC, USB, 이메일 등 디지털 증거물을 사법기관에 제출하기 위해 데이터를 수집, 분석, 처리 하기 위한 절차 및 기법
키워드	디지털 포렌식 5 원칙, 절차, 안티 포렌식
출제의도분석	- 디지털 증거의 중요도가 향상됨에 따라 관련 기술인 디지털 포렌식에 대한 문제 출제
답안작성 전략	- 물어 본 내용이 많으므로 분량 및 시간 배분에 주의 하여 작성
참고문헌	침해사고대응 모델 및 디지털 포렌식 모델의 프로세스 통합에 관한 연구(이승원), 디지털 포렌식의 개념 및 해외 적용사례(차경엽)
풀이 기술사님	서경석 기술사 (제 119 회 정보관리 기술사 / akslempf@naver.com)

1. 디지털 범죄 수사의 핵심, 디지털 포렌식(Digital Forensic)의 개요



- PC, USB, 이메일 등 디지털 증거물을 사법기관에 제출하기 위해 데이터를 수집, 분석, 처리 하기 위한 절차 및 기법

2. 침해 사고 대응 측면과 디지털 감사 측면에서 디지털 포렌식의 필요성

가. 침해 사고 대응 측면에서 디지털 포렌식의 필요성

구분	필요성	설명
사전 대응	모니터링	- 침해 행위 발생 여부 확인을 위해 모니터링 진행
	유입 경로 차단	- 침해 시도 발생 사전 차단을 위한 유입 경로 확인
침해 대응	시스템 변조 확인	- 침해 사고 후 기존 시스템 및 환경 변경 여부 확인
	저장 매체 확인	- 침해 사고 영향 유입, 유출 저장 매체 여부 확인
	시스템 봉쇄	- 추가적인 침해 사고 발생 방지 위한 시스템 봉쇄
사후 조치	악성 코드 제거	- 침해 사고 전/후 시스템 비교 통한 악성 코드 제거
	백업 자료 재설치	- 데이터 손실 발생 시 백업 데이터 비교 후 재설치
	문서화 및 보고	- 침해 사고 대응 관련 포렌식 진행 시 후속 절차

- 침해 사고 관련 사전, 사후 대응 서비스에 디지털 포렌식 적용 및 활용 가능

나. 디지털 감사 측면에서 디지털 포렌식의 필요성

구분	필요성	설명
감사 품질 제고	충분한 증거 확보	- 단기간 내 대량의 증거 확보 가능
	감사 증거의 신뢰성	- 무결성 원칙 준수 수집 증거로 신뢰성 확보
	법적 문제 해결	- 포렌식 절차 준수 통한 적법한 증거 수집
증거의 다양화	대용량 하드 디스크	- 대량의 데이터에 대한 증거 수집 효율화
	휘발성 데이터	- 휘발성 매체에 저장한 증거 수집 용이
	암호화 저장 매체	- 암호화 저장 매체에 대한 분석 필요
부패의 지능화	자료 은폐, 조작 대처	- 디지털 증거의 변조 여부 확인 가능
	정보 유출	- Log 분석, 접속 이력, N/W 트래픽 분석

- 내부 감사, 정보 보호 측면에서도 감사 업무에 디지털 포렌식 적용 가능

3. 디지털 포렌식 절차 및 활용 기술

가. 디지털 포렌식 절차 및 설명

구분	세부	설명
절차도	<pre> graph LR A[조사 준비] --> B[현장 대응] B --> C[증거 확보, 수집] C --> D[증거 운반, 확인] D --> E[조사 및 분석] E --> F[보고 및 증언] B1[조사 권한 획득] --- A B2[현장 통제, 보존] --- B B3[대상 매체 확보] --- C B4[증거물 확인] --- D B5[사본 생성] --- E B6[보고서 작성, 제출] --- F B21[조사 팀 구성] --- B2 B22[협조 요청] --- B3 B23[조사 대상 파악] --- B4 B24[증거물 포장, 봉인] --- B5 B25[원본 보관] --- B6 </pre>	
조사 준비	권한 획득 장비, 도구 준비	- 디지털 포렌식 수행 시 필요한 관련 권한 획득 - 디지털 포렌식 Tool kit 및 H/W, S/W 준비
현장 대응	현장 통제, 보존 협조 요청	- 현장 통제, 보존용 사진 촬영, 폴리스 라인 설치 - 수사 진행 시 필요 유관 부서, 담당자 협조 요청
증거 확보, 수집	데이터 선별 수집 증거물 포장, 봉인	- 수집된 데이터 중 디지털 증거 선별 작업 진행 - 증거 봉인, 목록 작성 통한 연계보관성 확보
증거 운반, 확인	증거물 등록 원본 보관	- 확보한 증거 등록 관리 통한 유실 방지 - 사건 관련 증거 보관소에 확보한 원본 증거 보관
조사 및 분석	사본 생성 상세 분석	- 수사 진행 시 증거 활용 위한 사본 증거 생성 - 사본 증거 대상으로 디지털 포렌식 기법 적용
보고 및 증언	보고서 작성, 제출 법정 증언	- 디지털 포렌식 진행 관련 보고서 작성, 법원 제출 - 포렌식 수행 결과 및 디지털 증거 증언 진행

- 디지털 포렌식 진행 시 관련 절차 및 원칙 준수를 통한 디지털 증거의 적법성 확보 필요

나. 디지털 포렌식 활용 기술

구분	기술	설명
증거 수집	디스크 이미징	- 디스크에 저장 되어 있는 데이터 복사, 생성 기술
	캡쳐 프로그램	- 시스템 내 화면, 웹페이지 캡쳐 통한 증거 확보
	Staganalysis	- 스테가노그래피 적용 여부 탐지 위한 기술

Notes

	인덱스 기반 탐지	- 모든 파일 대상 키워드 탐색, 파티션 영역등 검색
	BitWise	- 디스크 섹터, 슬랙 공간 비활당 영역 데이터 확보
	슬랙 공간 복구	- 유실 파일 헤더 재구성 통한 원천 파일 복구 기술
	스왑 영역 검색	- Swap 영역에 대한 데이터 존재 여부 확인
증거 분석	디스크 브라우징	- 저장 매체, 디스크 내부 구조, 파일 시스템 확인
	데이터 뷰잉	- 디지털 데이터 구조 파악 후 가시화 하여 출력
	타임라인	- 파일 시스템 저장 파일의 시간 정보 및 행위 추적
	데이터 마이닝	- 확보한 디지털 증거 통합 분석 통한 데이터 추출
	로그 분석	- N/W, 이메일, 클라우드 등의 Log 분석 작업 진행
증거 제출	해시함수	- 증거의 무결성 증명 위해 Message Digest 활용
	전자서명	- 공개키 암호화 방식 활용 전자 문서 무결성 확보

- 디지털 포렌식의 절차 및 원칙을 준수하도록 적합한 디지털 포렌식 기술 적용 필요

4. 디지털 포렌식 기술 위기 및 발전 방향

구분	세부	설명
기술 위기	데이터 추출, 분석 비용 증가	- 분석 대상 장치 증가 - 저장 장치 용량 증가
	웹 기술 발전	- 블로그, 까페, SNS 등 발전 - 클라우드 컴퓨팅 기술 발전
	안티 포렌식 기술 등장	- 와이파이, 암호화, 스테가노그래피등
발전 전망	데이터 추상화	- 다양한 형식의 데이터 적용 지원
	모듈화	- 다양한 언어, 플랫폼 지원 가능 - Plug-in 형태의 지원 목표
	휘발성 메모리 포렌식	- 사용자 시스템 활성 데이터 수집 - 무결성 문제 극복 및 데이터 수집

- 디지털 범죄가 지속적으로 증가함에 따라 디지털 포렌식 감정 건수도 해마다 증가 추세

"끝"

4	의존성 역전(Dependency Inversion)
문제	<p>4. 소프트웨어 재사용성과 유지보수 향상을 위하여 객체지향 설계 5 대 원칙을 적용하고 있다. 다음에 대하여 답하시오.</p> <p>가. 의존성 역전 원칙(Dependency Inversion Principle)을 설명하시오.</p> <p>나. 의존성 주입(Dependency Injection)을 구현하는 3 가지 방식을 설명하고 각 방식별 아래의 조건을 고려하여 구현 예시를 작성하시오.</p> <p>(조건)</p> <pre> classDiagram class ViewModel { <<interface>> IMESSAGEBOXPROVIDER } class IMESSAGEBOXPROVIDER { <<interface>> <<interface>> IMESSAGEBOXPROVIDER } class WpfMessageBoxProvider { <<interface>> IMESSAGEBOXPROVIDER } class CustomMessageBoxProvider { <<interface>> IMESSAGEBOXPROVIDER } class NullMessageBoxProvider { <<interface>> IMESSAGEBOXPROVIDER } ViewModel --> IMESSAGEBOXPROVIDER IMESSAGEBOXPROVIDER < -- WpfMessageBoxProvider IMESSAGEBOXPROVIDER < -- CustomMessageBoxProvider IMESSAGEBOXPROVIDER < -- NullMessageBoxProvider </pre>
도메인	S/W 공학
정의	- 의존 관계를 맺을 때 자주 변화 하는 것보다 거의 변화가 없는 것에 의존 해야 한다는 객체 지향 설계 원칙
키워드	SOLID, Setter Injection, Constructor Injection, Field Injection
출제의도분석	이론적 지식에 추가적으로 실무 지식에 대한 검증을 위한 구현 문제 출제
답안작성 전략	구현 예시를 작성 해야 하므로 정확한 구현에 중점
참고문헌	위키피디아(https://ko.wikipedia.org/wiki/SOLID_(%EA%B0%9D%EC%B2%B4_%E%A7%80%ED%96%A5_%EC%84%A4%EA%B3%84)) 의존관계 역전과 의존성 주입(https://dalgonal.github.io/posts/2019-03-26-dip-and-di.html)
풀이 기술사님	서경석 기술사 (제 119 회 정보관리 기술사 / akslemf@naver.com)

1. 재사용성과 유지보수 향상을 위한 객체지향 설계 5 대 원칙의 개요

가. 객체지향 설계(Object Oriented Design)의 정의

- 소프트웨어 작업에서 프로그래머가 소스 코드가 읽기 쉽고 확장하기 쉽게 될 때까지 소프트웨어 소스 코드를 리팩터링하여 코드 냄새를 제거하기 위해 적용 하는 설계 방식

나. 객체 지향 설계 5 대 원칙

머리문자	약어	원칙
S	SRP	단일 책임 원칙(Single Responsibility Principle)
O	OCP	개방-폐쇄 원칙(Open-Closed Principle)
L	LSP	리스코프 치환 원칙(Liskov Substitution Principle)
I	ISP	인터페이스 분리 원칙(Interface Segregation Principle)
D	DIP	의존성 역전 원칙(Dependency Inversion Principle)

- 프로그래머가 유지 보수와 확장이 쉬운 시스템을 만들기 위해 설계 원칙 적용

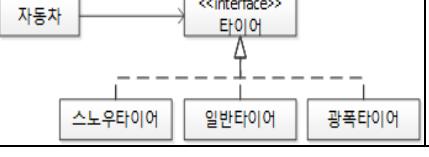
2. 객체 지향 설계 5 대 원칙 개념 및 의존성 역전 원칙 상세 설명

가. 객체 지향 설계 원칙 개념

원칙	개념	특징
SRP	- 모든 객체는 하나의 책임만을 가지며, 객체가 제공하는 서비스는 하나만의 책임 수행 설계 원칙	- 응집도 향상으로 유지보수성 향상 - 하나의 책임만을 수행하기 때문에 변화에 적응도 높음
OCP	- 소프트웨어 Entity는 확장에는 열려 있고 수정에는 닫혀있어야 한다는 설계 원칙	- 기존 코드 변경 없이 확장 통한 코드 변경 허용 - 기능 상속이 아닌 설계 유연성 강조
LSP	- 서브 타입은 언제나 기반 타입과 호환 되어야 한다는 설계 원칙	- 상/하위 클래스 간 호환성 향상 - 하위 클래스는 상위 클래스의 책임을 넘지 않음
ISP	- 클래스는 사용 하지 않는 인터페이스는 구현 하지 않는다는 설계 원칙	- 인터페이스의 단일 책임을 강조 - 불필요 기능 구현 방지 통한 클래스 복잡화 예방

- 객체 지향 설계 원칙 준수를 통한 이후 시스템 유지 보수 시간 및 비용 절감 가능

나. 의존성 역전 원칙 상세 설명

구분	설명	비고
개념	참조의 대상은 파생 클래스가 아닌 추상 클래스여야 한다	- 구현 클래스에 의존성 제거 통한 낮은 결합도 유지 - 추상화된 클래스 의존을 통한 확장성 확보 및 유지 보수성 향상
적용 전 사례		- 자주 변경 되는 파생클래스 참조
적용 후 사례		- 파생 클래스를 참조 하던 것을 추상 클래스를 의존 하도록 의존 관계 역전
기대 효과	- 파생 클래스 변경에 따른 추상 클래스 변경 최소화 - 복잡한 컴포넌트 간의 관계 단순화 가능	

- 해당 문제에서 ViewModel은 인터페이스인 IMessageBoxProvider를 통하여 Wpf, Custom, NullMessageBoxProvider 사용

3. 의존성 주입 구현 3 가지 방식 설명 및 각 방식별 구현 예시

가. 의존성 주입 구현 3 가지 방식 개념

구분	개념
생성자 사용	- 생성자의 매개변수에 생성되는 인스턴스가 의존하게 될 객체를 전달, 객체의 생성과 동시에 의존성을 주입하는 방식

속성 사용	- 클래스에서 노출하는 속성의 값을 설정하여 의존성을 주입하는 방식 - 클래스 외부에서 속성 접근 위해서는 loosely coupled 상태 구성 필요
메서드 사용	- 메서드의 매개변수에 객체를 전달하여 의존성을 주입하는 방식 - 인터페이스의 기능을 사용하는 메서드가 많거나, 한 메서드가 여러 개의 인터페이스에 의존하는 경우 복잡도 상승 가능

- 순환 참조 오류 방지, 테스트 용이성 등의 장점이 있어 생성자 사용 방식 사용 하도록 가이드.

나. 의존성 주입 방식별 구현 예시

구분	설명
생성자 사용	<pre>class ViewModel { protected IMESSAGEBOXPROVIDER MB { get; set; } public ViewModel(IMESSAGEBOXPROVIDER mb) { MB = mb; } } // ViewModel을 생성하는 곳에서 var vm = new ViewModel(new WpfMessageBoxProvider());</pre>
속성 사용	<pre>class ViewModel { public IMESSAGEBOXPROVIDER MessageBoxProvider { get; set; } // ViewModel을 생성하는 곳에서 var vm = new ViewModel(); vm.MessageBoxProvider = new WpfMessageBoxProvider();</pre>
메서드 사용	<pre>class ViewModel { public SomeMethod(IMESSAGEBOXPROVIDER mb) { if (mb.ShowYesNo("...", "...")) // ... } }</pre>

- Spring 환경에서는 속성 대신 필드 주입을 사용하고 메서드는 setter 메서드가 주로 사용

"끝"



5	Auto Scale
문제	5. 최근 부하 분산을 위하여 클라우드 서비스에서 Auto Scale 기능을 많이 활용한다. 다음에 대하여 설명하시오. 가. Auto Scale Up 과 Auto Scale Out 아키텍처 비교 나. 스케줄 기반의 Auto Scale 과 부하 기반의 Auto Scale 기능 비교
도메인	디지털 서비스
정의	클라우드 환경에서 인스턴스의 부하를 모니터링해서 특정 이벤트에 대해 인스턴스를 추가하거나 삭제 등의 관리를 위한 기능
키워드	Scale Up, Scale Out, Scale In, Scale Down, On-Demand, Load Balancer, Scheduler
출제의도분석	클라우드 서비스 사용자가 증가함에 따라 부하 분산을 위해 자주 활용 되고 있는 기술에 대한 지식 확인
답안작성 전략	비교 문제이므로 비교 대상에 대한 비교 항목 다양화 필요
참고문헌	AWS EC2 Auto Scaling (https://docs.aws.amazon.com/ko_kr/autoscaling/ec2/userguide/what-is-amazon-ec2-auto-scaling.html)
풀이 기술사님	서경석 기술사 (제 119 회 정보관리 기술사 / akslemf@naver.com)

1. 클라우드 환경의 능동적 자원 할당, 오토 스케일의 개요

가. 오토 스케일(Auto Scale)의 정의

- 클라우드 환경에서 요청이 늘어나면 이를 수용하기 위해 추가 자원의 할당, 회수를 통해 자원을 효율적으로 활용 하여 On-Demand 서비스를 지원 하는 기술
- 서비스 제공 시 관리 기준에 따라 자동으로 할당 자원을 추가 할당, 회수함으로써 자원을 효율적으로 활용 하기 위한 클라우드 서비스기반의 기술

나. Auto Scale 의 분류

구분	세부	설명
관리 기준	- 스케줄	- 서비스 관점의 스케줄 기준으로 관리
	- 시스템 부하	- 서비스 제공 시 부하 기준으로 관리
구현 방식	- 컨테이너 기반	- 컨테이너형 어플리케이션 플랫폼으로 구축
	- VM 기반	- VM 자체에서 제공 하는 기능 활용 구축
자원 할당	- Scale Up	- 인스턴스 자체의 성능 향상 자원 할당
	- Scale Down	- 인스턴스 성능 향상 자원 회수
	- Scale Out	- 추가 인스턴스 배정 통한 성능 향상
	- Scale In	- 추가 인스턴스 회수 통한 자원 효율화

- 관리 기준, 구현 방식, 자원 할당 등 다양한 방식의 Auto Scale 적용 가능

2. Auto Scale Up 과 Auto Scale Out 아키텍처 비교

가. Auto Scale Up 과 Auto Scale Out 의 개념도 및 개념 비교

Notes

구분	세부	설명
Auto Scale Up	개념도	<p>Diagram illustrating Auto Scale Up architecture:</p> <ul style="list-style-type: none"> VPC (10.0.0.0/16) connected to Internet. Load Balancer (10.0.1.0/24) distributing traffic to two Auto Scaling groups. Each Auto Scaling group contains two instances (e.g., Instance A, Instance B). Instances are located in different Availability Zones (Zone A and Zone B). Each instance has Memory, CPU, and Disk resources.
	개념	- 클라우드 환경에서 요청이 증가하는 경우 CPU, Memory, Disk 등의 자원을 증설 하는 기술
Auto Scale Out	개념도	<p>Diagram illustrating Auto Scale Out architecture:</p> <ul style="list-style-type: none"> VPC (10.0.0.0/16) connected to Internet. Load Balancer (10.0.1.0/24) distributing traffic to two Auto Scaling groups. Each Auto Scaling group contains two instances (e.g., Instance 1, Instance 2). Instances are located in different Availability Zones (Zone A and Zone B).
	개념	- 클라우드 환경에서 요청이 증가하는 경우 추가적인 Instance를 할당 하여 자원을 확보 하는 기술

- Auto Scale Up 과 Auto Scale Out 은 자원의 할당 방식에서 가장 큰 차이 보유

4. Auto Scale Up 과 Auto Scale Out 상세 비교

항목	Auto Scale Up	Auto Scale Out
구현 난이도	- 단순 추가 자원 할당으로 용이	- 상대적 구현 난이도 높음
비용	- 성능 증가에 따른 비용 증가 큼	- 성능에 따른 비용 부담 적음
자원 효율성	- 자원 할당, 회수 제약 있음	- 자원 할당, 회수 용이
장애 대응	- 장애 발생 시 대응 어려움	- 분산 처리 통한 장애 대응 가능
관리	- 관리 대상이 적어 관리 용이	- 다수의 인스턴스 관리 어려움
확장성	- 수직적 확장의 한계 존재	- 수평적 확장 통한 확장 용이
적용 사례	- 전통적 구조의 시스템	- MSA

- 확장성, 비용, 장애 대응 측면에서 유리한 Auto Scale Out 이 주로 사용되는 추세

3. 스케줄 기반의 Auto Scale 과 부하 기반의 Auto Scale 개념 및 기능 비교

가. 스케줄 기반의 Auto Scale 과 부하 기반의 Auto Scale 개념

구분	세부	설명
스케줄 기반	개념도	<p>The diagram illustrates a schedule-based auto scaling architecture. At the top is the Internet. Below it is a VPC network with a CIDR block of 10.0.0.0/16. Inside the VPC, there are two availability zones: Zone A and Zone B. Zone A contains two instances of 'Web Servers in Availability Zone A'. Zone B contains two instances of 'Web Servers in Availability Zone B'. An 'Auto Scaling group' is shown at the bottom, managing these instances. A 'Scheduler' is positioned between the VPC and the Auto Scaling group. Two locks are placed on the connections between the Scheduler and the Auto Scaling group, indicating that the Scheduler triggers scaling actions.</p>
	개념	<ul style="list-style-type: none"> - 이벤트 발생 일정을 미리 알고 있는 경우에 해당 이벤트 발생 시점에 Auto Scale을 Scheduler에 입력하여 자원 할당 기술
부하 기반	개념도	<p>The diagram illustrates a load-based auto scaling architecture. At the top is the Internet. Below it is a VPC network with a CIDR block of 10.0.0.0/16. Inside the VPC, there are two availability zones: Zone A and Zone B. Zone A contains two instances of 'Web Servers in Availability Zone A'. Zone B contains two instances of 'Web Servers in Availability Zone B'. An 'Auto Scaling group' is shown at the bottom, managing these instances. A 'Load Balancer' is positioned between the VPC and the Auto Scaling group. Two locks are placed on the connections between the Load Balancer and the Auto Scaling group, indicating that the Load Balancer triggers scaling actions based on current load levels.</p>
	개념	<ul style="list-style-type: none"> - 미리 스케줄링 하지 않고 부하 발생 시점에 Load Balancer 모니터링을 통해 동적으로 자원을 할당 하는 기술

- 시스템 환경이나 업무 특성에 맞춘 Auto Scale 기술 선택 필요

나. 스케줄 기반의 Auto Scale 과 부하 기반의 Auto Scale 기능 비교

항목	스케줄 기반	부하 기반
이벤트 예약	- 특정 일정 기준 예약 용이	- 추가적인 시스템 구성 필요
부하 관리	- 부하 관리 기능 부족	- Load Balancer 활용 부하 관리
모니터링	- 체계적인 모니터링 진행 가능	- 현재 발생 기준 모니터링 가능
실시간 대응	- 실시간 이벤트 대응 어려움	- 부하 관리 통한 대응 가능
필요 자원 산정	- 이벤트 시점 필요 자원 산정 가능	- 정확한 필요 자원 산정 어려움
장애 발생 예측	- 스케줄 예측이 잘못된 경우 장애 발생 예측 어려움	- 부하 측정 통한 장애 발생 예측 가능
적용 사례	- 세일 기간 사전 자원 확보	- 동시 접속자 증가 시 자원 확보

- 이벤트 추이가 예측 가능한 경우 스케줄 기반, 변동이 큰 경우는 부하 기반 Auto Scale 적용

4. Auto Scale 서비스 적용 방안

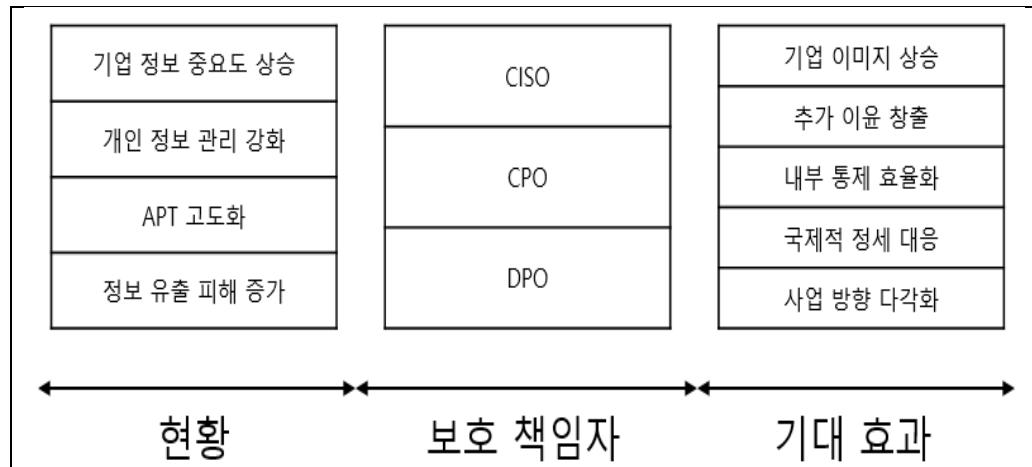
단계	작업	설명
사전 검토	- 적용 대상 검토	- Auto Scale 기능 필요 여부 확인
	- 일정, 비용 확인	- 일정을 고려한 투자 비용 분석
구축	- 스케줄, 부하 기반 결정	- 업무에 적합한 방식 결정
	- Auto Scale 환경 구축	- 필요 자원 확보 및 조직 구성
운영	- 성능 모니터링	- 기능 정상 수행 여부 및 오류 확인
	- 자원 최적화	- 성능 모니터링 결과 반영 자원 효율화

- 효율적 Auto Scale 기능 적용 통한 추가적 기업 이윤 발생 기대

"끝"

6 CISO(Chief Information Security Officer)	
문제	6. CISO(Chief Information Security Officer)의 자격요건과 겸직 여부 등을 개정한 「정보통신망 이용촉진 및 정보보호 등에 관한 법률」이 2020년 1월 1일부터 적용되고 있다. 다음에 대하여 설명하시오. 가. CISO의 직무 나. CISO의 지정·신고 의무대상자 다. CISO의 겸직 제한 대상 라. CISO의 일반 자격요건
도메인	보안
정의	기업의 정보통신시스템 등에 대한 보안 및 정보의 안전한 관리 등 정보보호 업무를 총괄하는 최고책임자
키워드	CISO, CPO, DPO, CIO, CTO
출제의도분석	최근 개정 발표된 정보 보호 관련 법률 연관 지식 확인
답안작성 전략	물어 본 항목이 정확하게 나와 있으므로 개념적으로 작성 하지 않도록 유의
참고문헌	정보보호 최고책임자 지정·신고 제도 안내서 (https://www.kisa.or.kr/public/laws/laws3_View.jsp?cPage=6&mode=view&p_No=259&b_No=259&d_No=116&ST=T&SV=)
풀이 기술사님	서경석 기술사 (제 119 회 정보관리 기술사 / akslemf@naver.com)

1. 정보화 시대 기업 정보 보호의 총 책임자, CISO의 개요



- CISO란 기업의 정보 통신 시스템 등에 대한 보안 및 정보의 안전한 관리등 정보 보호 업무를 총괄 하는 최고 책임자

2. CISO의 직무 및 지정·신고 의무 대상자

가. CISO의 직무

구분	직무	설명

사전 예방	사전 정보 보호 대책 마련 및 보안 조치 설계 · 구현	- 위험 처리하기 위한 정보보호 대책 선정 - 처리 우선순위 결정, 이행계획에 대한 구현
	정보 보호 사전 보안성 검토	- 새로운 정보통신망을 구축, 정보통신서비스를 제공하고자 하는 경우 수행 - 정보보호 사전점검(법 제45조의2)
	중요 정보 암호화 및 보안서버 적합성 검토	- 개인 및 중요 정보 보호 위한 암호화 진행 - 암호화 대상, 강도, 보안서버 기능·관리 적합성 검토
운영 대응	정보보호관리체계의 수립 및 관리 · 운영	- 정보통신망 안정성·신뢰성 확보 위한 종합적 관리체계(법 제47조제1항)의 수립 및 관리·운영
	정보보호 취약점 분석 · 평가 및 개선	- H/W, S/W 결함, 설계상의 허점 및 취약점 분석 - 발생한 취약점에 대한 분석·평가·개선
	침해사고의 예방 및 대응	- 침해사고 정의 및 범위, 대응체계, 방법 절차 수립 - 침해사고 대응 계획 마련·시행 등
	이 법 또는 관계 법령에 따라 정보보호 필요 조치 이행	- 정보통신망법 및 관계 법령 등에 규정 조치의 이행 - 정보 수집, 가공, 저장 중 발생 가능한 변조, 유출 방지·복구 - 암호·인증등의 보안 기술을 활용, 재난·범죄 등에 대응, 관련 장비 시설을 안전하게 운영

- CISO는 기업의 정보 통신 시스템 서비스 전반에 걸쳐 정보 보호 업무에 대한 책임 소재

나. CISO의 지정·신고 의무 대상자

구분	기준	비고
개념	아래의 지정·신고 제외 대상자를 제외한 모든 정보통신서비스 제공자는 임원급의 정보보호 최고책임자를 지정하고 과학기술정보통신부장관 (중앙전파관리소장에게 위임)에게 신고하여야 함	
제외 대상	자본금 1 억원 이하 부가통신사업자	「전기통신사업법」에 따라 부가통신사업을 신고한 것으로 보는 자(자본금 1 억원 이하의 부가통신사업자)
	소상공인	소기업 중 상시 근로자 수*가 10 명 미만이고, 사업에 종사 상시 근로자 수가 제조, 건설업 10 명 미만, 그 밖의 업종 5 명 미만
	소기업	「중소기업기본법」상 소기업(전기통신사업자와 집적정보통신시설사업자 제외)으로 전년도 정보통신서비스 부문 매출액이 100 억원 미만이고, 전년도 말 기준 직전 3 개월의 일일 평균 이용자수가 100 만명 미만인 자

- 상호출자 제한기업집단 소속 회사는 소기업에 해당하지 않으므로 CISO의 지정·신고 필요

3. CISO의 겸직 제한 대상 및 일반 자격 요건

가. CISO의 겸직 제한 대상

구분	설명
대상	<ul style="list-style-type: none"> - 직전 사업연도 말 기준 자산총액*이 5조원 이상 - 정보보호 관리체계 인증 의무대상자 중 자산총액 5천억원 이상 정보통신서비스 제공자
정보보호 관리체계 인증 의무대상 자	<ul style="list-style-type: none"> - 기간 통신 사업자 중 특별시, 광역시에서 정보통신망서비스를 제공하는 자 - 타인의 정보통신서비스 제공 위해 집적 정보통신시설 운영/관리 사업자 - 연간 매출액, 세입이 1,500 억원 이상인 상급 종합 병원 - 연간 매출액, 세입 1,500억원 이상, 전년도 재학생 수 1만명 이상대학 - 정보통신서비스 부문 전년도 매출액이 100 억원 이상인 자 - 전년도 말 기준 직전 3 개월의 일일평균 이용자 수 100 만명 이상인 자
사례	<ul style="list-style-type: none"> - 정보통신서비스 제공자가 사업장별로 정보보호 책임자를 두고 법인의 정보 보호 최고 책임자가 이를 총괄 하거나, 특정 사업장과 법인의 정보 보호 업무를 총괄 하는 경우는 겸직에 해당 하지 않음 - 업무 특성 상 정보 보호, 개인 정보 보호 업무의 명확한 분리가 곤란하거나, 혼재 되어 있는 경우는 정보보호 최고책임자가 관련 업무 수행 가능
관련 법령	<ul style="list-style-type: none"> <정보통신망 이용촉진 및 정보보호 등에 관한 법률> 제 45 조의 3 <정보통신망 이용촉진 및 정보보호 등에 관한 법률 시행령> <정보통신망 이용촉진 및 정보보호 등에 관한 법률> 제 47 조 <정보통신망 이용촉진 및 정보보호 등에 관한 법률 시행령> 제 49 조

- 정보통신망법 상 겸직 제한은 직위 겸직 제한이 아니라, 업무에 대한 겸직 제한에 해당

나. CISO의 일반 자격 요건

구분	일반 자격 요건	비고
개념	<ul style="list-style-type: none"> - 정보통신서비스 제공자가 정보보호의 전문성을 갖춘 정보보호 최고 책임자를 임명할 수 있도록 자격요건 규정 	
일반자격 요건	정보보호, 정보기술 분야 석사 이상 학위 취득자	<ul style="list-style-type: none"> - 전자 관련 학과, 정보통신 관련 학과, 정보보호 또는 정보 처리 기술 관련 학과의 과정일 이수, 졸업한 학력
	정보보호, 정보기술 분야 학사 학위 취득 및 해당 분야 실무 수행 경력 3 년 이상인 자	<ul style="list-style-type: none"> - 학위 취득 시기와 경력의 선/후는 자격 요건 부합 여부 판단 관계 없음 - 정보 보호를 위한 공통 기반 기술, 시스템, 네트워크, 응용 서비스 보호 업무 등
	정보보호, 정보기술 분야의 전문학사학위, 해당 분야 실무 수행 경력 5 년 이상인 자	<ul style="list-style-type: none"> - 정보통신서비스, 정보통신기기, 소프트웨어 및 컴퓨터 관련 서비스 업무 등

Notes

정보보호, 정보기술 분야의 업무 수행 경력 10년 이상인 자	
정보보호 관리체계 인증 심사원 자격 취득자	- 정보보호 관리체계(ISMS) 인증심사원 또는 정보 보호 및 개인 정보 보호 관리 체계(ISMS-P) 인증 심사원 자격 보유자
해당 정보통신서비스 소속 정보보호 관련 업무 담당 부서의 장으로 1년 이상 근무한 경력이 있는 자	- 해당 조직이 정보 보호 업무를 주된 업무로 하고, 다른 업무를 함께 소관한 경우에도 정보 보호 부서에 해당

- 겸직 제한 정보 보호 최고 책임자는 일반 자격 요건에 추가적 특별 자격 요건 필요

4. 보호 책임자 유사 개념 비교

구분	CISO	CPO	DPO
명칭	정보보호최고책임자	개인정보보호책임자	데이터 보호 책임자
관련 근거	정보통신망법 제45조의 3	개인정보보호법 제31조	GDPR 제37~39조
자격 요건	학위 + 경력	직위, 직급	특별히 없음
주요 역할	정보보호관리체계 수립, 운영 등	개인정보보호 계획 수립 등	GDPR 준수 확인, 자문
책임성	개인적 책임 있음	개인적 책임 있음	개인적 책임 없음

- 상황별 적합한 보호 책임자 교육 및 지정에 따른 기업 정보 보호 필요

"끝"

Notes