

1	클라우드소싱 테스트
문제	클라우드소싱 테스트(Crowdsourcing Test)를 설명하시오.
도메인	소프트웨어 공학(테스트)
정의	다수의 외부 테스터들에 의해 테스트를 진행하고 피드백을 얻는 테스팅 방식
키워드	Pay-per-bug
출제의도분석	모바일 단말의 파편화에 따른 SW 테스트 필요량 증가에 따라 서비스로서 테스트 제공에 대한 관심 증가
답안작성 전략	Cloud 서비스를 이용한 테스트 기반(TaaS) 제공 방식을 언급
참고문헌	Jump Up Day 자료 "완성형 테스트 서비스" TaaS 의 이해 – IDG Summary
풀이 기술사님	강용제 (yongjei.kang@gmail.com)

### ■ 클라우드소싱 테스트의 개념(정의)

- SW 제품을 기업 내부에서 테스트하는 것이 아니라 광범위하게 분포된 다수의 사람들에게 사용하게 하고 피드백을 얻는 테스팅 방식.
- 기존 베타 테스트와 비슷한 방식으로 진행되나 외부 테스터를 보유한 서비스 업체를 통해 테스트 진행과 결과 보고가 이뤄짐.

### ■ 클라우드소싱 테스트의 효과

구분	내용
비용 절감	<ul style="list-style-type: none"> <li>- 수십에서 수백 명의 인력이 테스트에 참여</li> <li>- 업무에 따른 수행 비용만 지불하므로 비용 절감 가능</li> <li>- Pay-per-bug</li> <li>- Cloud Service에서 기대할 수 있는 비용 절감 기대 가능</li> </ul>
높은 테스트 품질	<ul style="list-style-type: none"> <li>- 기존 In-house 테스트보다 넓은 범위를 테스트하는 것이 가능</li> <li>- 실제 사용자만 발견할 수 있는 문제에 접근 가능</li> <li>- 다양한 테스트 경험을 가진 테스터 인력 확보 가능</li> </ul>
실 사용자 참여	<ul style="list-style-type: none"> <li>- 실제 제품을 사용하게 될 실사용자에게 피드백을 얻을 수 있음</li> <li>- 다양한 조건과 환경에서 실제 사용자의 사용 형태로 테스트 가능</li> <li>- 긍정적 관점의 SW 애호가의 참여 유도가 가능</li> </ul>

### ■ 클라우드소싱 테스트의 개념도

- 테스트 의뢰 기업의 제품에 대하여 다양한 환경의 테스터를 중계해 테스트 서비스를 제공하는 기업(TaaS)이 존재함.

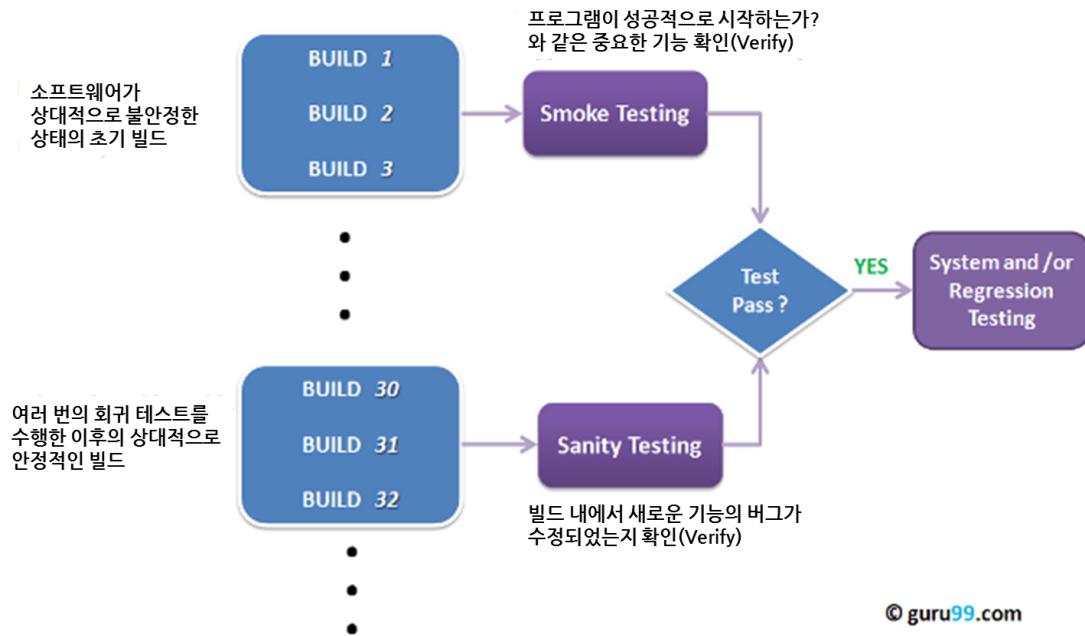


### ■ 클라우드소싱 테스트의 주요 장점

- Cloud Service 가 제공 가능한 다양한 장점을 테스트 서비스에서 기대하는 것이 가능하며 궁극적으로 비용 절감을 목표로 할 수 있음.

구분	내용
집중을 통한 가치 최 대화	<ul style="list-style-type: none"> <li>- SW 개발자와 테스팅 전문가가 각각의 영역에 집중</li> <li>- 핵심 역량에 집중해 가치 최대화</li> </ul>
개발 비용 절감	- 발견된 버그에 대해서만 비용 지불(Pay-per-bug)
Sanity Testing	<ul style="list-style-type: none"> <li>- 신속하게 문제 해결 여부를 확인 가능</li> <li>- 특정 분류의 고객을 대상으로 피드백 수집 가능</li> </ul>
사용성 제고	- 사용자와 개발자 사이의 사용성 사각 지대를 신속히 발견 가능
테스트 커버리지 확대	- 수십에서 수백 명의 테스터 투입으로 다양성에 기반한 테스트 커버리지 확대 가능
빠른 테스트	- In-house 테스트에 비해 참여 인력이 10~20 배이므로 매우 짧은 시간 내에 문제 발견 가능
포괄성	- In-house 테스트 조직이 가질 수 있는 편향성을 제거
유연성	- 짧은 시간 내에 테스트 인력의 확장이 가능
전문성	- 사외의 전문가(SW 애호가)들이 가진 열정을 기반으로 제품 품질의 개선 가능

### ■ 참고: Smoke Testing 과 Sanity Testing 의 차이



- Smoke Testing: SW 빌드 후 프로그램의 핵심 기능들이 잘 수행되는가?
- Sanity Testing: 사소한 변경 후 SW 빌드에서 버그가 수정되었고 변경으로 인해 문제가 유입되지 않았는가? (실패하면 더 엄격한 테스팅은 실행되지 않고 그 빌드는 Reject 됨)

"끝"

2 기계 학습(Machine Learning)	
문제	기계 학습(Machine Learning)을 정의하고 알고리즘 유형을 설명하시오.
도메인	디지털 서비스
정의	새로운 지식을 습득하면서 새로운 상황의 문제를 해결할 수 있는 프로그램
키워드	지도 학습, 자율 학습, 준 지도 학습, 강화 학습
출제의도분석	메가 트랜드로서 Analytics, 자율 주행 기술 등의 대두에 따라 그 기본 원리의 이해를 시험
답안작성 전략	일관된 알고리즘 유형에 따라 알고리즘을 분류하고 설명
참고문헌	위키백과의 관련 항목 <a href="http://www.whydsp.org/237">http://www.whydsp.org/237</a>
풀이 기술사님	강용제 (yongjei.kang@gmail.com)

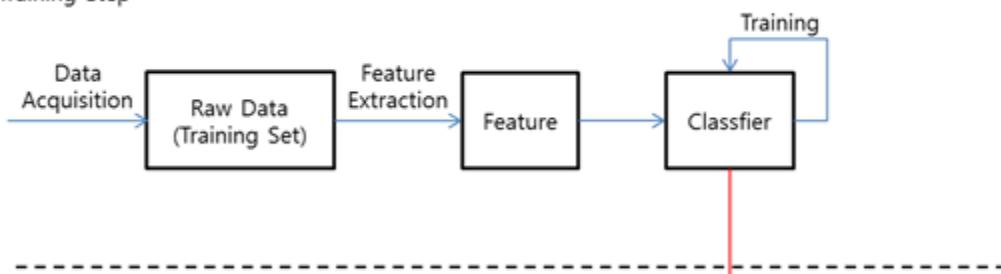
### ■ 기계 학습의 개념(정의)

- 새로운 지식을 습득하면서 새로운 상황의 문제를 해결할 수 있는 프로그램.
- 경험과 컴퓨터 알고리즘을 통해 자신의 동작을 개선할 수 있는 시스템의 능력.
- 컴퓨터에게 배울 수 있는 능력, 즉 코드로 정의하지 않은 동작을 실행하는 능력(아서 사무엘, 1959)
- 훈련 데이터를 통해 이미 알려진 속성을 기반으로 예측하는 능력.

### ■ 기계 학습의 개념도

- 미리 준비된 데이터를 이용하여 Classifier 를 학습시키는 단계와 학습(훈련)된 Classifier 를 이용하여 입력된 데이터가 어떤 분류 결과인지 확인하는 테스트 단계로 나뉨.

<Training Step>



<Test Step>



Notes

### ■ 기계 학습의 주요 알고리즘 유형

유형	내용	알고리즘
지도 학습 (Supervised learning)	<ul style="list-style-type: none"> <li>- 훈련 데이터로부터 하나의 함수를 유추하기 위한 방법</li> <li>- 훈련 데이터로부터 주어진 데이터에 대해 예측하고자 하는 값을 추측해내는 것</li> <li>- 알고리즘을 통해 정해진 답을 찾는 것이 목적</li> <li>- 명확한 input 과 output 이 존재한다는 것이 특징이며 결과물의 형태에 따라 다시 회귀(Regression)과 분류(Classification)으로 나뉨</li> </ul>	<ul style="list-style-type: none"> <li>- 서포트 벡터 머신</li> <li>- 은닉 마르코프 모델</li> <li>- 회귀 분석</li> <li>- 신경망</li> <li>- 나이브 베이즈 분류</li> </ul>
자율 학습 (비 지도 학습, Unsupervised learning)	<ul style="list-style-type: none"> <li>- 입력 값에 대한 목표치가 주어지지 않는 알고리즘</li> <li>- 주로 데이터가 어떻게 구성되어있는지 밝히는 데 사용</li> <li>- 일종의 그루핑 알고리즘</li> </ul>	<ul style="list-style-type: none"> <li>- 군집화(Clustering)</li> <li>- 독립 성분 분석</li> </ul>
준 지도 학습 (Semi-supervised learning)	<ul style="list-style-type: none"> <li>- 목표 값이 표시된 데이터와 표시되지 않은 데이터를 모두 훈련(학습)에 사용하는 방법</li> </ul>	
강화 학습 (Reinforcement learning)	<ul style="list-style-type: none"> <li>- 어떤 환경을 탐색하는 에이전트가 현재의 상태를 인식하여 어떤 행동을 취하면 그 에이전트는 환경으로부터 포상을 받음(포상은 양수와 음수 모두 가능)</li> <li>- 강화 학습은 앞으로 누적될 포상을 최대화하는 일련의 행동으로 정의되는 정책을 찾는 방법</li> </ul>	<ul style="list-style-type: none"> <li>- Q-Learning</li> <li>- Dyna_Q</li> <li>- Policy reuse</li> <li>- Local state feature to bias exploration</li> </ul>

### ■ 기계 학습의 알고리즘 선택 시 고려 사항

- 알고리즘의 견고성: 알고리즘의 밝혀진 이론적 한계. 출력 값의 정확성.
- 알고리즘의 단순성: 정확도 제고를 위한 입력 사례의 수 결정. 사전 지식 및 초기 값 정의.
- 알고리즘의 확장성: 알고리즘의 성능. 새로운 패턴의 이벤트 처리.

"끝"

3 핀테크	
문제	핀테크(FinTech)를 정의하고 보안 측면의 이슈와 해결 방안을 설명하시오.
도메인	디지털 서비스
정의	주로 모바일을 통한 결제, 송금, 자산 관리, 크라우드 펀딩 등 금융과 IT가 융합된 산업
키워드	금융(Financial) + 기술(Technology), FDS(Fraud Detection System)
출제의도분석	각종 금융 규제의 완화로 활성화가 예상되는 핀테크에서 보안 측면의 이슈 조명
답안작성 전략	핀테크에 대한 정확한 이해와 분야별 특징 언급. 기존 뱅킹의 보안과 핀테크 보안의 근본적 차이점에 대한 기술.
참고문헌	Wikipedia의 관련 항목 산업 간 융합 관점에서 본 핀테크의 시사점 – Internet & Security Focus 2014.11.
풀이 기술사님	강용제 (yongjei.kang@gmail.com)

### ■ 핀테크의 개념(정의)

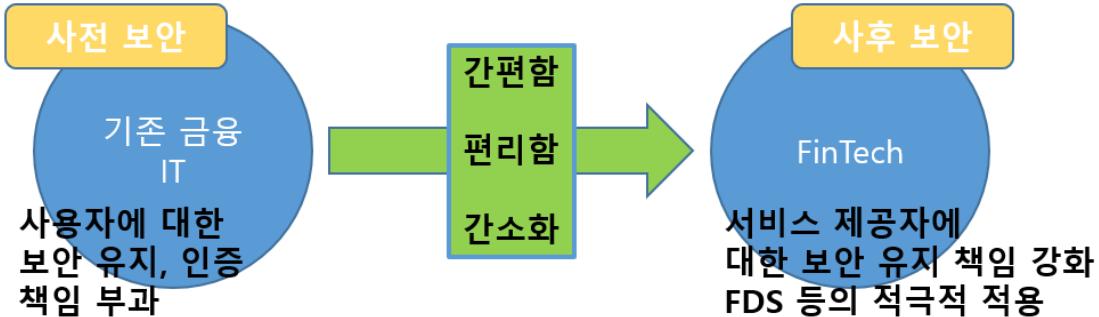
- 주로 모바일을 통한 결제, 송금, 자산 관리, 크라우드 펀딩 등 금융과 IT가 융합된 산업을 의미
- Financial 과 Technique 의 합성어로 금융과 ICT 의 결합을 통해 새롭게 등장한 산업 및 서비스 분야를 통칭하는 용어

### ■ 핀테크 산업의 분야 및 분야별 특징

분야	대표 서비스	특징
송금 분야 – 전자 화폐	비트 코인 M-Pesa 트랜스퍼와이즈	<ul style="list-style-type: none"> <li>- 온라인으로 거래 가능한 가상 화폐.</li> <li>- 비트 코인 등 신규로 등장한 화폐와 M-Pesa 등 기존 화폐의 결제 수단을 보조하는 화폐를 통칭.</li> <li>- 전자 화폐 외 개인, 기업 간 송금 서비스 등도 포함 가능</li> <li>- 비트 코인 사용자 2500 만명 이상(2014.10), M-Pesa의 거래량이 케냐 GDP의 43%(2014.01)에 이를 만큼 전자 화폐의 실물 경제에 대한 영향력이 점차 증가하는 추세.</li> </ul>
결제 분야 – 전자 결제 시스템	페이팔 알리페이 스퀘어	<ul style="list-style-type: none"> <li>- 제품 및 서비스 구매 시 사용자가 미리 등록한 카드나 계좌 정보를 활용하여 간편하게 결제가 가능하도록 한 전자 결제 서비스.</li> <li>- 핀테크 사업자와 은행, 카드사 등 기존 금융 업체가 제휴한 서비스 형태가 많으며 NFC, 전자 지갑 등 사용자에게 친숙한 서비스가 지속적으로 출시 중.</li> <li>- 세계 전자 결제 거래량은 2017년까지 약 7천억 달러(770조원)에 이를 것으로 전망.</li> </ul>
투자 분야 – 금융 투자 플랫폼	엔젤리스트 렌딩 클럽 쿠오보	<ul style="list-style-type: none"> <li>- 대출, 창업 자금 지원 등 투자 관련 금융을 서비스하는 온라인 플랫폼.</li> <li>- 금융 투자 플랫폼은 기존 금융 대비 낮은 수수료, 신속한 이체, 사용자 친화적 UI로 시장 점유율을 높이는 중.</li> <li>- 특히 빅데이터를 기반으로 사용자 신용도를 분석하여 기존 금융 대비 높은 변제율과 낮은 사고율이 특징.</li> <li>- 온라인 플랫폼은 스타트업, 중소 상공인 등 기존 금융의 투자를 받지 못한 기업의 자금처로 인식되며 중소 기업 및 스타트업 활성화의 원동력.</li> </ul>

### ■ 핀테크에서의 보안 이슈 개념

- 기존 인터넷 뱅킹/결제 서비스에서 이뤄지던 사용자에 대한 보안 책임 부과를 탈피해 서비스 제공자의 보안 책임을 강화하는 방향으로 변화.
- 사용자에게 편리함, 간편함, 간소화 서비스를 제공하기 위해 보안 유지의 기준 경계를 넘어야 함.



### ■ 핀테크에서의 보안 이슈 해결 방안

- 편리함, 간편함, 간소화를 제공하기 위해 보안에 대한 새로운 관점을 가져야 함.

이슈	해결 방안
본인 인증	<ul style="list-style-type: none"> <li>- 기존: 서비스 이용 때마다 공인 인증서를 통한 인증</li> <li>- 핀테크: 서비스 가입 시 최초 본인 인증 정보를 이후 거래 시 신뢰하는 체계</li> <li>- ARS 인증, SMS 인증, OTP 인증 등 별도 인증 방안 도입 고려</li> </ul>
부정 사용 방지	<ul style="list-style-type: none"> <li>- 기존: 사전 보안 체계로 사고 발생을 사전에 예방/차단하는 것이 목표</li> <li>- 핀테크: 사후 보안 체계로 거래 시 다양한 거래 패턴을 보유/분석하여 이상 징후 발생 시 거래 차단 및 신속한 사후 처리(2 차 확인 절차)가 목표</li> <li>- 빅데이터 분석, 구매 패턴 분석, 송금 등 패턴 분석 등 FDS 도입 및 적용</li> </ul>
금융 정보 저장	<ul style="list-style-type: none"> <li>- 기존: 사용자의 개별 단말에 공인 인증서 등을 저장하는 체계</li> <li>- 핀테크: 서비스 제공자의 서버에 사용자의 카드 정보, 계좌 정보, 배송 정보 등을 저장하고 거래 시 간단한 암호 입력만으로 거래 가능</li> <li>- 이상 거래 시 2 차 인증/확인 절차 강화</li> <li>- 서버 간 정보 교환 체계의 보안성 강화</li> </ul>

"끝"

4	Bagging, Boosting 비교
문제	빅데이터 분석방법인 Bagging과 Boosting 기법을 비교하여 설명하시오.
도메인	알고리즘
정의	- Bagging: 랜덤 샘플데이터 모델링을 결합해 예측모형을 산출하는 방법 - Boosting: 가중치를 적용해 반복적으로 예측모형의 정확도를 향상시키는 방법
키워드	- Bagging: 샘플링, 분산감소, 평균 - Boosting: 가중치, 반복, 정확도 향상, 선형결합
출제의도분석	데이터 마이닝의 '분류'에 사용되는 알고리즘들에 대한 이해와 특성의 비교
답안작성 전략	알고리즘의 특징, 수행방법 비교와 함께 데이터 특성에 따른 선택적 적용방안 제시
참고문헌	데이터 마이닝에서 배깅, 부스팅, SVM 분류 알고리즘 비교 분석(2005, NDSL 논문)
풀이 기술사님	이홍철 (bluerobe@naver.com)

## 1. 분류모델 생성 알고리즘 Bagging과 Boosting

Bagging	Boosting
주어진 데이터에서 여러 개의 bootstrap 자료를 생성하고, 각 자료를 모델링 한 후 결합(Bootstrap Aggregating)하여 최종 예측 모형을 만드는 알고리즘	잘못 분류된 개체들에 가중치를 적용하여 새로운 분류규칙을 만들고, 이 과정을 반복해 최종 예측 모형을 만드는 Boosting(변형) 알고리즘

- Bagging과 Boosting은 데이터 마이닝에서 분류를 수행하기 위한 분류모델을 생성하는 알고리즘.

## 2. Bagging과 Boosting 알고리즘 비교 설명

### 가. Bagging과 Boosting 알고리즘 수행방법 비교

Bagging	Boosting
<pre> graph LR     RD[Row data] -- Random sampling --&gt; SD1[Sampling data]     RD -- Random sampling --&gt; SD2[Sampling data]     RD -- Random sampling --&gt; SD3[Sampling data]     SD1 --&gt; M1[Modeling]     SD2 --&gt; M2[Modeling]     SD3 --&gt; M3[Modeling]     M1 --&gt; FM[Final Model]     M2 --&gt; FM     M3 --&gt; FM     style RD fill:#ffcc99,stroke:#800000,stroke-width:2px     style SD1 fill:#90EE90,stroke:#008000,stroke-width:1px     style SD2 fill:#90EE90,stroke:#008000,stroke-width:1px     style SD3 fill:#90EE90,stroke:#008000,stroke-width:1px     style M1 fill:#ADD8E6,stroke:#00008B,stroke-width:1px     style M2 fill:#ADD8E6,stroke:#00008B,stroke-width:1px     style M3 fill:#ADD8E6,stroke:#00008B,stroke-width:1px     style FM fill:#DC143C,stroke:#00008B,stroke-width:2px     </pre> <p>1) Row data에서 bootstrap 데이터를 추출 2) 추출을 반복하여 n 개의 데이터를 생성 3) 각 데이터를 각각 모델링하여 모델생성 4) 단일모델을 결합하여 배깅모델 생성</p>	<pre> graph TD     RD[Row data] --&gt; M1[Modeling]     M1 --&gt; WRD1[Weighted Row data]     WRD1 --&gt; FM[Final Model]     RD --&gt; M2[Modeling]     M2 --&gt; WRD2[Weighted Row data]     WRD2 --&gt; FM     RD --&gt; M3[Modeling]     M3 --&gt; WRD3[Weighted Row data]     WRD3 --&gt; FM     style RD fill:#ffcc99,stroke:#800000,stroke-width:2px     style M1 fill:#ADD8E6,stroke:#00008B,stroke-width:1px     style WRD1 fill:#90EE90,stroke:#008000,stroke-width:1px     style M2 fill:#ADD8E6,stroke:#00008B,stroke-width:1px     style WRD2 fill:#90EE90,stroke:#008000,stroke-width:1px     style M3 fill:#ADD8E6,stroke:#00008B,stroke-width:1px     style WRD3 fill:#90EE90,stroke:#008000,stroke-width:1px     style FM fill:#DC143C,stroke:#00008B,stroke-width:2px     </pre> <p>1) Row data에 동일가중치로 모델 생성 2) 생성된 모델로 인한 오분류 데이터 수집 3) 오분류 데이터에 높은 가중치 부여 4) 이 과정을 반복해 모델의 정확도 향상</p>

- Bagging은 여러 번의 sampling을 통해 분산을 줄여 모델의 변동성을 감소시키는 방법

- Boosting은 잘못 분류된 데이터에 집중해 모델의 정확도를 향상시키는 방법

Notes

## 나. Bagging 과 Boosting 알고리즘의 기술적 특징 비교

항목	Bagging	Boosting
수행원리	샘플링에 의한 결합	가중치 재조정에 의한 반복
수행목적	모델의 변동성(분산)을 감소	모델의 정확도 향상
적용연산	평균, 다중투표	가중치 선형결합
초기모델	Bootstrap 모델(개별 모델)	Weak classification 모델
최종모델	Bagging 모델(결합 모델)	Strong classification 모델
분류성능	데이터에 결측치가 존재할 경우 우수	데이터의 수가 많을 경우 우수

- 데이터 마이닝에서 분류문제를 해결하고자 할 때 중요한 문제는 주어진 데이터를 이용해 목표변수를 가장 잘 예측할 수 있는 모델을 생성하는 것이므로, 데이터 특성에 따라 모델을 생성하는 알고리즘의 선택적 적용이 필요함.

"끝"

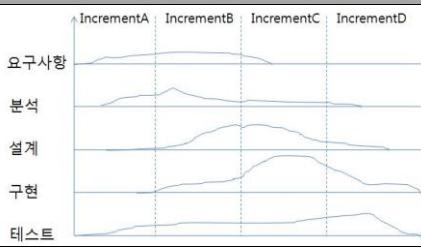
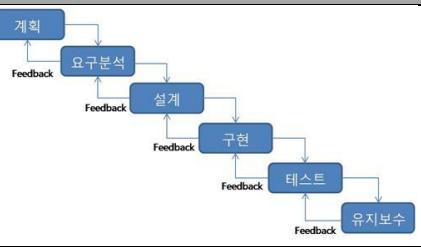
5 반복점증적 개발방법, 폭포수형 개발방법 비교	
문제	반복점증적 개발방법과 폭포수형 개발방법을 비교하여 설명하시오.
도메인	소프트웨어 공학
정의	반복점증적 개발방법: 반복개발을 통해 점증적으로 최종 시스템을 완성하는 방법 폭포수형 개발방법: 하향식으로 순차적으로 개발해 최종 시스템을 완성하는 방법
키워드	반복, 점증, 점진적 구체화, 하향식, 단계적, 상세화
출제의도분석	기본적인 개발방법에 대한 현장통용성 높은 전문적인 지식을 확인
답안작성 전략	개발 실무적 관점으로 다양하고 현장감 있는 비교내용을 제시
참고문헌	소프트웨어 공학(최은만, 정의사)
풀이 기술사님	이홍철 (bluerobe@naver.com)

## 1. 반복점증적 개발방법과 폭포수형 개발방법의 개념

반복점증적 개발방법	폭포수형 개발방법
고객과 함께 요구사항을 지속적으로 찾아가면서 확정된 부분부터 개발을 점증적으로 진행해 최종 시스템을 완성하는 개발방법	고객으로부터 요구사항을 확정한 후 개발단계별로 작업을 분리해 각 단계를 종료한 후 다음단계로 순차적으로 진행해 시스템을 완성하는 개발방법

- 반복점증적 개발방법은 폭포수형 개발방법의 생명주기를 반복적으로 수행함으로써 폭포수형 개발방법의 단점을 개선시킬 수 있음

## 2. 반복점증적 개발방법과 폭포수형 개발방법의 비교

항목	반복점증적 개발방법	폭포수형 개발방법
수행절차	 <p>요구사항 분석 설계 구현 테스트</p>	 <p>계획 요구분석 설계 구현 테스트 유지보수</p>
수행원리	Iterative(반복적, 점증적 접근)	Top-down(하향식, 단계적 접근)
주요산출물	반복수행 계획서, 사용자 피드백	요구명세서, RTM(요구추적매트릭스)
설계	점진적 구체화, 점진적 설계	상세화 설계
개발	반복단위로 반복적 개발	순차적, 단계적인 개발
테스트	회귀테스트의 필요성 높음	빅뱅테스트 주로 이용
배포	반복, 지속적인 배포	종료단계에 배포

- 반복점증적 개발방법은 프로토타입과 폭포수형 개발방법의 장점을 모두 갖고 있으며, RUP 와 Agile 방법론 등을 통해서 그 수행원리가 계승 및 발전되어 왔음

Notes

### 3. 장단점을 고려한 개발방법 적용방향

구분	반복점증적 개발방법	폭포수형 개발방법
장단점	장점) 사용자 요구사항 확인 용이 반복을 통한 리스트의 감소 단점) 개발관리가 복잡함	장점) 정형화된 진행으로 관리용이 사례 많고 문서산출물이 많음 단점) 사용자 피드백 불가능
고려사항	반복을 고려한 관리계획 수립, 형상관리, 버전관리 도구 활용	초기 요구사항의 상세화, 명확화 milestone 등의 점검과정 고려
적용방향	PMO 등을 통해 관리되는 프로젝트	상세 RFP 등이 도입된 프로젝트

- 반복적 개발방법을 모델로한 스크럼, 칸반 등의 개발방법이 널리 확산되었고, 다양한 개발도구와 관리도구 등도 개발되어 방법론의 사용성을 높이고 있음

"끝"

6	회귀시험(Regression Test)
문제	소프트웨어의 회귀시험(Regression Test)에 대하여 설명하시오.
도메인	소프트웨어 공학
정의	회귀시험: 오류제거와 수정에 의한 새로운 오류의 발생에 대한 시험
키워드	riffle effect, side effect, retest all/selective/priority
출제의도분석	개발환경에서 반복적인 release 의 증가로 회귀테스트의 효과적 적용역량 평가
답안작성 전략	회귀시험 전반에 걸친 지식과 실무적인 사항을 순차적으로 제시
참고문헌	개발자도 알아야 할 소프트웨어 테스팅 실무(권원일 등, STA)
풀이 기술사님	이홍철 (bluerobe@naver.com)

## 1. 회귀시험의 개요

### 가. 회귀시험(Regression Test)의 정의

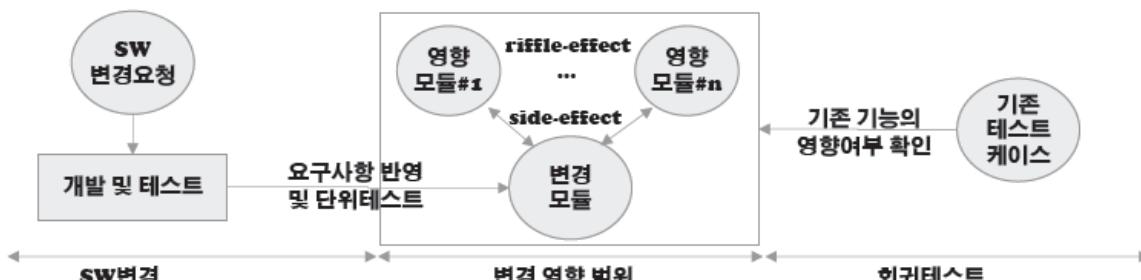
- 오류를 제거하거나 수정한 시스템에서 오류제거와 수정에 의해 새로이 유입된 오류가 없는지 확인하는 일종의 반복시험

### 나. 회귀시험이 검출하는 오류내용

오류내용	설명
Ripple Effect (파급효과)	오류제거나 수정을 위해서 변경된 내용이 이와 연관된 다른 부분으로 영향이 전파되어 변경되는 현상
Side Effect (부작용)	오류제거나 수정작업이 의도한대로 이루어 졌으나 미쳐 고려하지 못한 다른 부분에서 의도치 않은 또 다른 결과가 발생하는 현상

## 2. 회귀시험의 수행방법과 유형

### 가. 회귀시험의 수행방법



- SW 변경작업에 대한 개발을 완료한 후 변경영향 범위에 대해서 기존의 테스트 케이스를 이용해 기존 기능에 대한 영향여부를 확인함

### 나. 회귀시험의 수행유형

항목	Retest All	Selective Test	Priority Test
수행방법	테스트 케이스 전부를 사용해 테스트	영향범위를 결정해 테스트 수행	핵심기능을 위주로 우선순위화 테스트
적용분야	금융권 고위험 시스템	일반 시스템	저위험 시스템
장점	높은 테스트 커버리지	높은 비용 효율성	낮은 테스트 비용
단점	높은 테스트 비용	영향범위 선정 어려움	우선순위 기준 모호함

*Notes*

### 3. 회귀시험 수행 시 고려사항

- 개발프로세스와 통합된 테스트 프로세스 구현을 통한 테스트 관리체계 구축
- 테스트 자동화 도구적용을 통해 테스트 효율성 향상
- Repository 를 통해 테스트 자산 축적 및 재사용성 향상
- 테스트 조직을 통한 수행과 관리를 통해 조직의 테스트 역량 향상

“끝”



Notes

7	분산컴퓨팅 및 분산처리, 투명성
문제	분산컴퓨팅(Distributed Computing) 및 분산처리(Distributed Processing)를 각각 정의하고 투명성(Transparency)에 대하여 설명하시오.
도메인	컴퓨터 구조/운영체제
정의	분산컴퓨팅: 분산된 컴퓨팅 리소스를 연결해 컴퓨팅 파워를 향상시키는 기술 분산처리: 전체처리를 물리적 또는 논리적인 분리처리에 근거해 수행하는 기술
키워드	자원결합, 분리처리, 투명성(위치, 이주, 복제, 병행, 접근, 성능, 규모, 고장 투명성)
출제의도분석	빅데이터, 클라우드 등의 기술확산에 따라 기반기술에 대한 이해의 확인
답안작성 전략	분산처리시스템의 기반기술로써 분산컴퓨팅, 분산처리를 정의하고 투명성을 설명
참고문헌	<a href="http://en.wikipedia.org/wiki/Distributed_computing">http://en.wikipedia.org/wiki/Distributed_computing</a>
풀이 기술사님	이홍철 (bluerobe@naver.com)

## 1. 분산처리 시스템의 기반기술, 분산컴퓨팅, 분산처리

분산컴퓨팅	분산처리
네트워크를 통해서 연결된 여러 컴퓨터들의 처리능력을 이용해 거대한 계산문제를 해결하려는 분산처리 모델	지리적으로 산재해 있는 처리장치나 데이터베이스를 네트워크로 상호 접속해 작업을 처리하는 기술

- 분산처리를 수행할 수 있는 분산컴퓨팅을 통해서 분산처리 시스템을 구축할 수 있으며, 사용자에게는 하나의 시스템으로 인식되도록 투명성을 제공해야 함

## 2. 분산처리 시스템의 투명성

투명성	설명
위치(Location) 투명성	사용자가 하드웨어나 소프트웨어와 같은 자원의 물리적인 위치를 모르더라도 자원에 접근할 수 있음
이주(Migration) 투명성	사용자나 응용프로그램의 동작에 영향을 받지 않고 시스템 내에 있는 자원을 이동할 수 있도록 함
복제(Replication) 투명성	자원의 복제를 사용자에게 통지할 필요 없이 자유롭게 수행할 수 있음
병행(Concurrency) 투명성	자원의 위치를 모르더라도 다중 사용자들이 자원을 병행하여 처리하고 공유할 수 있도록 함
접근(Access) 투명성	각 프로세서의 로그인 등과 같은 동작을 사용하여 지역이나 원격자원에 접근할 수 있음
성능(Performance) 투명성	여러 부하에 대해 성능을 증가시키기 위해 시스템을 재구성할 수 있음
규모(Scaling) 투명성	시스템이나 응용프로그램들이 시스템 구조나 응용 알고리즘에 대한 변경 없이 규모에 맞추어 확장할 수 있음
고장(Failure) 투명성	사용자나 응용프로그램이 하드웨어나 소프트웨어 구성요소의 고장에도 불구하고 그들의 작업을 완료할 수 있음

Notes

### 3. 분산처리 시스템의 활용방향

분야	설명	사례
기능분산	여러 기능을 가진 컴퓨터 시스템을 분산처리 시스템을 통해 구현(한 대의 컴퓨터에 모든 기능을 구비시키는 것은 시스템의 대형화를 초래)	하둡 클러스터 시스템
처리분산	대용량 데이터를 처리하는 작업을 분산된 컴퓨터를 통해서 작업(특정 한 장소에서 처리할 경우 발생하는 과부하나 병목을 초래)	하둡 MapReduce

- 부하분산, 위험분산, 관리분산 등 다양한 목적으로 분산처리 시스템을 구성하여 이용할 수 있음

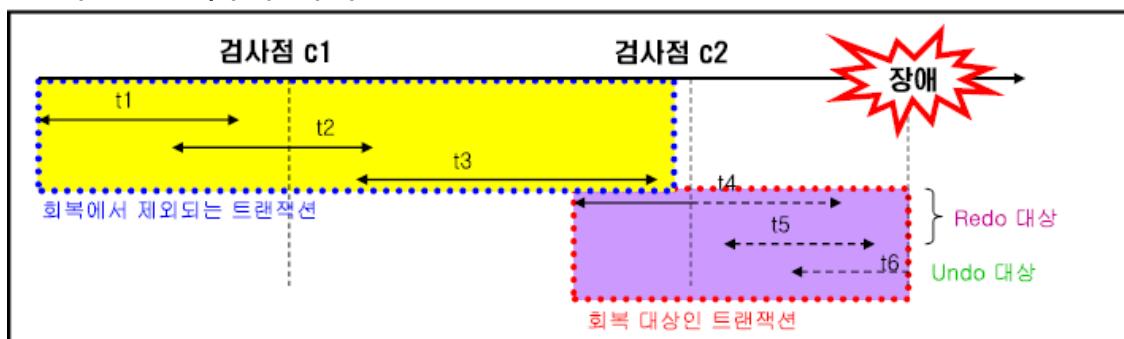
"끝"

8 데이터베이스 체크포인트 회복 기법	
문제	데이터베이스의 체크포인트(Checkpoint) 회복 기법에 대하여 설명하시오.
도메인	데이터베이스
정의	로그 파일에 체크포인트를 기록하고 장애 발생시 이를 이용해 회복 작업을 수행하는 기법
키워드	
출제의도분석	데이터베이스에 저장된 데이터의 품질이 중요해지면서 품질 유지의 기본적인 기법 이해를 시험
답안작성 전략	기본적인 토픽으로 정확한 기법 이해와 여타의 회복 기법과의 비교 필요
참고문헌	
풀이 기술사님	강용제 (yongjei.kang@gmail.com)

### ■ 체크포인트 회복 기법의 개념(정의)

- 트랜잭션 수행 중 발생하는 변경 내용을 로그 파일에 기록하고 일정 기간 단위로 체크포인트를 생성.
- 회복 시 로그 파일의 정보를 모두 검사해야 하며 이미 디스크에 반영된 변경에 대하여 불필요한 REDO 연산을 수행해야 하는 로그 기반 회복 기법의 문제점을 해결하기 위함.
- 장애 발생 시에 체크포인트 이전에 처리된 트랜잭션들은 회복 대상에서 제외되고 체크 포인트 이후에 처리된 트랜잭션에 대해서만 회복 작업을 시행.
- 데이터베이스 회복: 트랜잭션을 실행하는 도중 장애가 발생하여 데이터베이스가 손상되었을 경우 손상되기 이전의 정상 상태로 복구하는 작업.

### ■ 체크포인트 회복 기법의 개념도

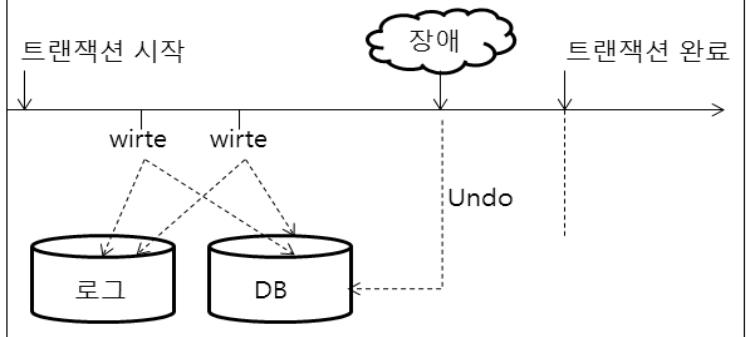
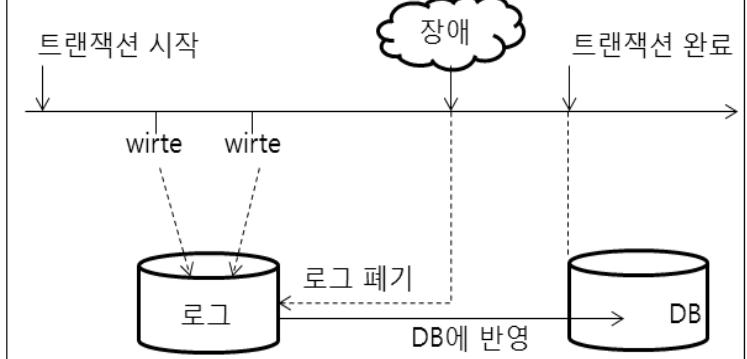
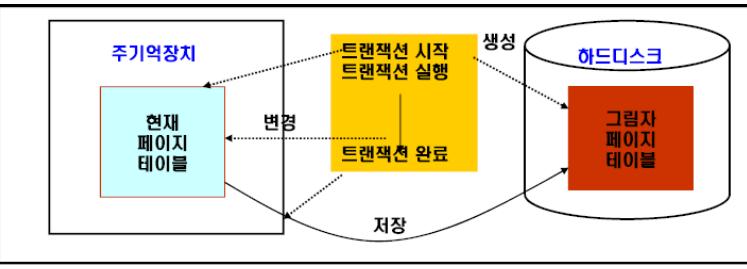


- 체크포인트 이전에 시작하여 체크포인트 이전에 완료된 트랜잭션은 회복 대상이 아님.
- 체크포인트 이전, 이후와 관계 없이 시작하여 장애 발생 시간에 진행 중인 트랜잭션은 Undo 대상임.
- 체크포인트 이전, 이후와 관계 없이 시작하여 장애 발생 시간 이전에 완료된 트랜잭션은 Redo 대상임.

### ■ 회복 기법 간 비교

구분	로그 기반	체크포인트	그림자 페이징
복구 과정	Undo	Undo, Redo	그림자 테이블 교체
복구 속도	느림	로그 기반보다 빠름	빠름
디스크 사용량	적음	적음	많음
복구 데이터	하나의 로그 파일	하나의 로그 파일	분산된 그림자 테이블
확장성	용이	용이	알고리즘 복잡

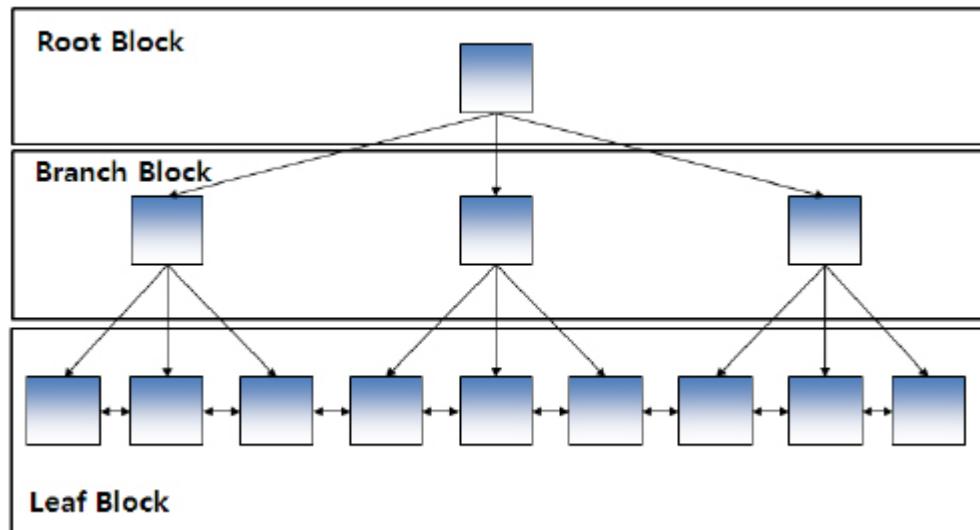
### ■ 데이터베이스 회복 기법 유형

구분	개념
로그 기반 즉시 갱신 기법	 <ul style="list-style-type: none"> <li>- 트랜잭션 수행 중 갱신 결과를 바로 DB에 반영.</li> <li>- 장애 발생 시 로그 파일에 저장된 내용을 참조하여 Undo 수행.</li> <li>- 변경 내역을 즉시 반영하기 때문에 안정성 뛰어남.</li> <li>- 동시성 제어가 중요하며 로그 검색에 부하 발생함.</li> </ul>
로그 기반 지연 갱신 기법	 <ul style="list-style-type: none"> <li>- 트랜잭션 수행 중에는 로그에 보관하며 트랜잭션 종료 시 로그를 이용해 DB에 반영함.</li> <li>- 장애 발생 시 로그를 폐기함.</li> <li>- 트랜잭션 완료까지 부하가 적게 발생.</li> <li>- 로그 검색 부하가 발생하며 DB 반영 시 트랜잭션 전체를 DB에 다시 수행해야 함.</li> </ul>
그림자 페이징 기법	 <ul style="list-style-type: none"> <li>- 트랜잭션 시작 시 현재 페이지 테이블과 동일한 그림자 테이블을 생성.</li> <li>- 트랜잭션 수행 시 현재 페이지 테이블의 내용을 갱신.</li> <li>- 트랜잭션 성공 시 현재 페이지 테이블의 내용을 그림자 테이블에 저장하며 실패 시 그림자 테이블로 복원.</li> </ul>

"끝"

9 B-Tree 와 비트맵 인덱스	
문제	B-Tree 와 비트맵(Bitmap) 인덱스를 비교 설명하시오.
도메인	알고리즘
정의	다수의 외부 테스터들에 의해 테스트를 진행하고 피드백을 얻는 테스팅 방식
키워드	
출제의도분석	데이터베이스에서 테이블에 대한 동작 속도를 높이는 기본적 자료 구조인 인덱스에 대한 이해 수준을 시험
답안작성 전략	B-Tree 인덱스와 비트맵 인덱스를 정확히 정의하고 개념과 효율적인 적용 대상 데이터를 설명
참고문헌	Wikipedia 의 관련 항목 DBGuide.net
풀이 기술사님	강용제 (yongjei.kang@gmail.com)

### ■ B-Tree 인덱스의 개념



- DBMS 에서 가장 일반적인 인덱스로 Leaf Block, Branch Block, Root Block 으로 구성된 자료 구조 중 Leaf Block 에 정렬된 컬럼의 데이터와 해당 데이터를 가진 행의 위치를 가리키는 레코드 식별자로 구성되는 인덱스.

- 리프 블록은 양방향 링크를 가지고 있어 오름 차순, 내림 차순 검색을 용이하게 실행 가능.
- 일치(Exact Match) 검색, BETWEEN, > 등의 범위(Range) 검색에 모두 적합.

### ■ B-Tree 인덱스의 검색

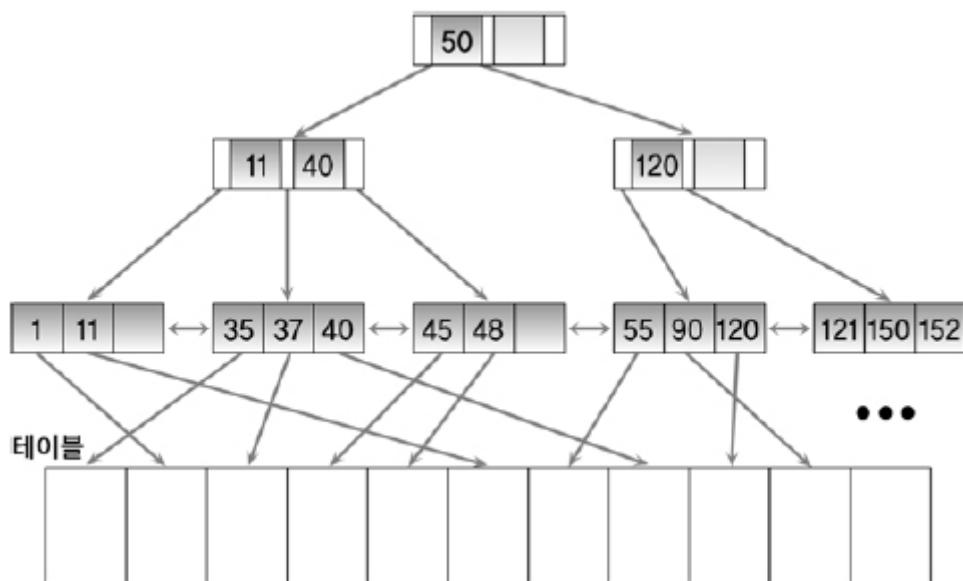
#### Exact Match: 아래의 B Tree 인덱스 구조에서 37 을 찾는 경우

1	루트 블록에서 50 보다 작으므로 왼쪽 포인터로 이동
2	37 은 왼쪽 브랜치 블록의 11 과 40 사이의 값이므로 가운데 포인터로 이동
3	이동 결과 해당 블록이 리프 블록이므로 37 이 블록 내에 존재하는지 검사

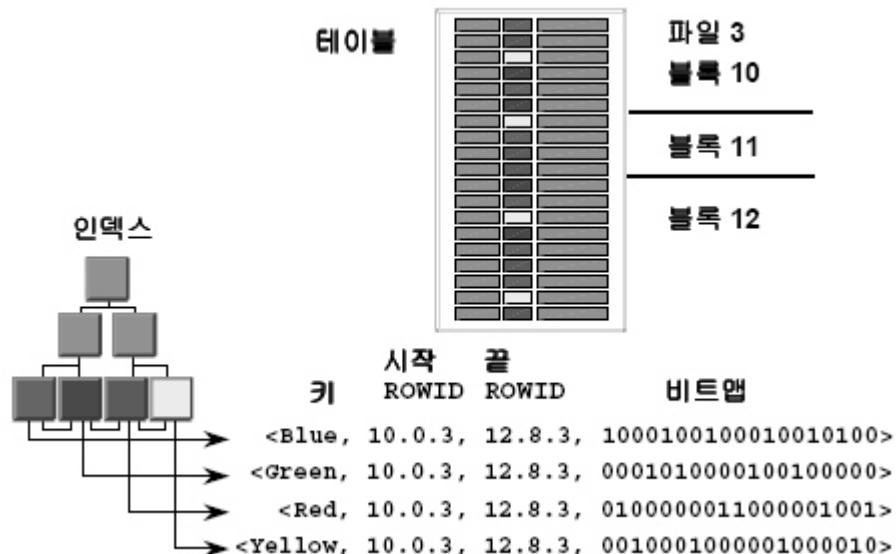
#### Range: 아래의 B Tree 인덱스 구조에서 37 과 50 사이의 모든 값을 찾는 경우

1	위 Exact Match 의 경우와 동일하게 37 을 검색
2	Leaf Block 은 양방향 링크(Double Link)를 가지고 있으므로 50 을 만날 때까지 오른 쪽으로 이동

Notes



### ■ Bitmap 인덱스의 개념



- 컴퓨터에서 사용하는 최소 단위인 비트를 이용하여 컬럼 값을 저장하고 이를 이용하여 ROWID 를 자동으로 생성하는 인덱스
- 비트를 직접 관리하므로 저장 공간이 크게 감소하고 비트 연산의 수행이 가능
- 인덱스 대상 컬럼의 데이터 카디널리티가 낮을수록 유리함

### ■ Bitmap 인덱스의 생성 절차

순서	실행 내용
1	인덱스를 생성하고자 하는 컬럼의 값들을 찾기 위해 테이블 스캔
2	bitmap generator에 의해 컬럼값, start rowid, end rowid, bitmap을 갖는 인덱스 엔트리를 생성
3	2 단계에서 생성된 Bitmap 들을 B-tree 구조에 넣기 쉽도록 key 값과 start rowid 순으로 정렬
4	정렬된 인덱스 엔트리들을 단순히 B-tree 구조로 삽입

Notes

### ■ B Tree 인덱스와 Bitmap 인덱스의 비교

구분	B Tree 인덱스	Bitmap 인덱스
원리	- Balanced Tree 를 이용해 검색을 쉽게 하면서 Leaf Block 간의 Double Link 로 연속된 데이터 조회 효율을 높임	- 적은 수의 카디널리티를 갖는 데이터에 대해 각 경우에 대한 Bitmap 을 구성하여 인덱스 생성
적합한 응용	- OLTP - 저장된 데이터가 수시로 생성, 수정, 삭제되는 경우	- OLAP, DW, Bigdata - 자주 변하지 않으면서 낮은 카디널리티를 갖는 경우
적용 데이터 특성	- 다양성을 유지하며(분포도가 좋으며) - 생성, 수정, 삭제가 빈번	- 소수의 경우의 수를 가지며 - 변경이 자주 일어나지 않음
인덱스 특성	- 실제 컬럼 값을 인덱스에도 보관해야 하므로 대용량 데이터에 부담이 됨 - 다양한 액세스 패턴을 수용하기 위해서는 많은 인덱스가 필요 - NOT 이나 NULL 을 사용하거나 복잡한 OR 조건에서는 인덱스의 성능을 보장받을 수 없음	- DML 작업 시 지속적으로 Bitmap 크기가 커짐 - 따라서 Index rebuild 작업 필요성이 있음(On-line rebuild 작업 미지원)

"끝"

10 DOI(Digital Object Identifier)	
문제	DOI(Digital Object Identifier)에 대하여 설명하시오.
도메인	디지털 서비스
정의	디지털 저작물의 저작권 보호 및 위치 추적 기능을 제공하기 위해 특정 번호를 부여하는 일종의 바코드 시스템
키워드	ISO 26324:2012, UCI(Universal Content Identifier)
출제의도분석	DOI를 대체하는 UCI의 전략적 보급에 따라 DOI 체계의 이해. ISO 표준이 된 DOI의 이해. 콘텐츠 산업의 육성/발전을 위한 저작권 보호 체계에 대한 관심.
답안작성 전략	매우 오래된 기본 토픽으로 정확한 이해와 관련 지식(ISO 26324, UCI 등)에 대한 언급이 필요.
참고문헌	2015년도 예산 사업 설명 자료(1권) – 문화체육관광부 <a href="http://www.doi.org">www.doi.org</a> 학술지 국제화의 지름길 DOI & OA – KOFST, KISTI
풀이 기술사님	강용제 (yongjei.kang@gmail.com)

### ■ DOI(Digital Object Identifier)의 개념(정의)

- 책이나 잡지 등에서 사용되는 ISBN(국제 표준 도서 번호)과 같이 모든 디지털 콘텐츠에 부여되는 고유 식별 번호
- 콘텐츠의 위치 변경에 무관한 디지털 콘텐츠의 영구 식별자
- 디지털 콘텐츠의 저작권 보호, 유료화, 식별, 분배 등을 위한 기본적 관리 체계
- ISO 26324:2012로 국제 표준화

### ■ DOI의 특징

구분	특징
정확한 검색	- 고유 번호를 사용하기 때문에 인터넷 주소인 URL 보다 정확한 검색이 가능
유연하고 광범위한 응용성	- URL로 쉽게 변환 가능 - ISBN과 같은 기존 체계 수용
상호 운영성 제공	- 콘텐츠 간 관계 표현 - 다른 소스에 있는 메타데이터들과 상호 운영성 제공 가능

### ■ DOI의 구조

- Prefix + / + Suffix 형태로 구성됨
- Prefix: 국제 DOI 재단이 부여(한 발행처에서 한 Prefix를 받아서 잡지 별로 다른 Suffix를 부여할 수 있음)
- Suffix: 발생인이 부여
- 마침표를 찍어가면서 계속 확장이 가능하며 알파벳과 숫자 모두를 사용할 수 있음

Prefix + "/" + Suffix  
 IDF에서 부여                              Publisher가 부여

(IDF: International DOI Foundation)

## Journal of Educational Evaluation for Health Professions

J Educ Eval Health Prof 2009; 6: DOI: 10.3352/jeehp.2009.6.1

DOI : 10.3352/jeehp.2009.6.1

Open Access

EDITORIAL

### Faculty Responsible for the Accreditation Process of Medical Education in Korea Should Have Their Workloads Reduced

Sun Huh

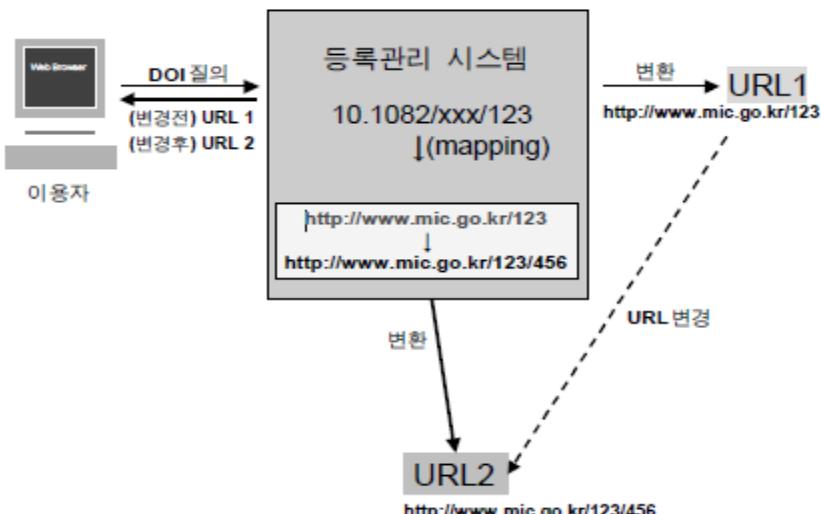
Department of Parasitology, College of Medicine and Institute of Medical Education, Hallym University, Chuncheon, Korea

In Korea, the year 2000 marked the official launch of the accreditation system for medical education. The first accreditation project covered 41 medical schools from 2000 to 2004.

ever, I would like to suggest one more point of improvement during the accreditation process. The present peer review process puts an undue burden on the members of the medi-

▲ 학술지에 사용된 DOI의 예시. Prefix/Suffix 구조를 가지고 있음

#### ■ DOI의 검색(이용) 개념



- DNS 체계와 유사하게 질의와 응답으로 실제 내용 또는 위치에 대한 정보를 반환함

순서	내용
1	이용자가 DOI를 선택(브라우저에서 클릭)
2	DOI system에서 이용자로부터 받은 DOI와 관련된 URL을 찾음
3	이용자의 인터넷 브라우저에 URL을 전송
4	이용자는 콘텐츠 자체를 볼 수 있거나 콘텐츠에 대한 정보나 접근방법에 대한 정보를 보냄

#### ■ DOI 이용 현황

- 국제 또는 국내에서 발행되는 대부분의 학술 논문은 DOI를 표기하고 있음
- 2012년 ISO 26324로 국제 표준으로 수용됨
- 문화체육관광부에서 실행 중인 "국가 디지털 콘텐츠 식별 체계 구축(정보화)" 사업에서 만들고 있는 UCI(Universal Content Identifier, 국가 표준 디지털 식별 체계)로 대체하려고 시도 중임
- 공공 정보 개방 정책에 따른 공개 데이터와 연계해 DOI 또는 UCI의 확산이 전망됨

"끝"

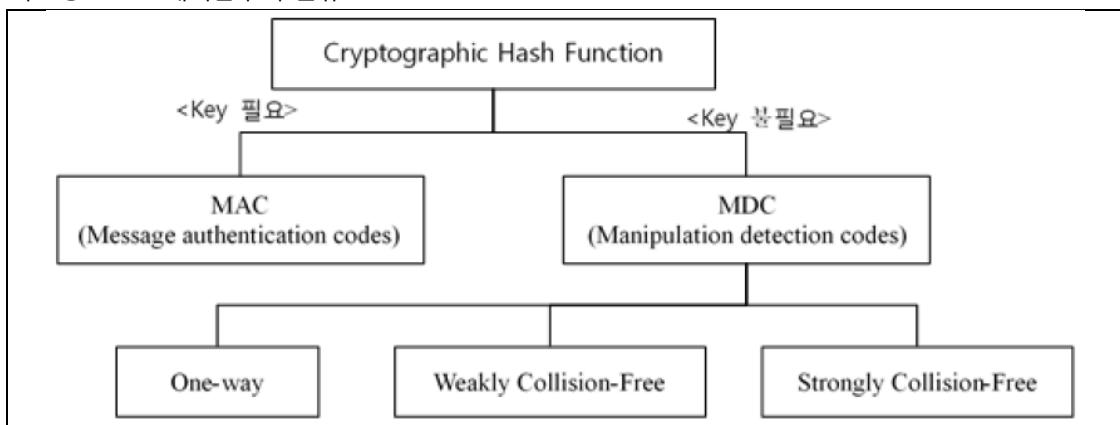
11 정보보호 해시함수(Hash Function)	
문제	정보보호를 위한 해시함수(Hash Function) 종류에 대하여 설명하시오.
도메인	정보 보안
정의	정보보호 해시함수: 해시함수 가운데 암호학적 해시함수에 해당함
키워드	암호학적 해시함수, MAC, MDC
출제의도분석	전자서명 등 정보보호 환경에서 사용되는 해시함수의 특성에 대한 이해
답안작성 전략	암호학적 해시함수의 특징과 함수의 유형별 상세한 설명
참고문헌	정보보안개론(양대일, 한빛아카데미)
풀이 기술사님	이홍철 (bluerobe@naver.com)

## 1. 정보보호 해시함수의 개요

- 역상, 제2역상, 충돌에 대하여 안전성을 갖고 인증에 이용되는 암호학적 해시함수
- 암호학적 해시함수는 임의 길이를 입력 받기는 하지만, 최대 길이에 대한 제한이 존재함

## 2. 정보보호 해시함수의 종류

### 가. 정보보호 해시함수의 분류



### 나. 정보보호 해시함수의 종류와 설명

종류	설명	비고
MAC	비밀키를 이용한 블록암호화 알고리즘에 기초	CBC-MAC, MAA
MDC	비밀키를 사용하지 않는 전용 해시 알고리즘 MAC에 비해 상대적으로 빠른 속도	MD5, SHA-1, HAS-160
일방함수 (One-way)	y 가 주어졌을 때, $h(x)=y$ 에서 x 찾기 불가능	역상저항성 중심
약한 충돌회피 함수 (Weakly Collision-Free)	$h(m1)=h(m2)$ 에서 $m1$ 과 다른 $m2$ 찾기 불가능	2 차 역상 저항성
강한 충돌회피 함수 (Strongly Collision-Free)	$h(m1)=h(m2)$ 에서 $m1, m2$ 를 알아내기 불가능	충돌저항성 중심

Notes

### 3. 정보보호 해시함수의 활용

구분	설명
디지털 포렌식	원본 증거에 대한 해시값과 분석을 완료한 후 제출된 증거의 해시값을 비교해 증거 무결성을 증명하는 데 사용
전자서명	송신측에서 전달한 원문에 대한 해시값과 수신측에서 복호화를 통해 얻은 해시값을 비교해 서명을 검증하는 데 사용

- Birthday paradox 를 이용한 Brute force 공격과 차분연쇄공격 등으로 역상저항성과 출돌저항성을 무효화하는 공격이 존재하므로 메시지블록 길이, 키 길이 등의 조정으로 대응이 필요함

"끝"

Notes

12	데이터베이스 튜닝(Tuning)
문제	데이터베이스 튜닝(Tuning)의 3 단계와 튜닝의 기대효과에 대하여 설명하시오.
도메인	데이터 베이스
정의	데이터베이스 튜닝: 데이터베이스 트랜잭션의 성능을 개선하는 활동
키워드	디자인(설계)튜닝, 애플리케이션(SQL)튜닝, 서버(DBMS)튜닝
출제의도분석	데이터베이스의 성능향상을 위한 기본적인 지식의 확인
답안작성 전략	정확한 3 단계 튜닝절차의 설명과 경험기반의 실무적인 기대효과 제시
참고문헌	데이터베이스 설계와 구축(이춘식, 한빛미디어)
풀이 기술사님	이홍철 (bluerobe@naver.com)

### 1. 데이터베이스 튜닝의 3 단계

단계	튜닝 영역	설명
1단계	설계 튜닝	데이터베이스의 논리적인 구조에 대한 성능개선 작업
2단계	DBMS 튜닝	데이터베이스의 메모리영역, 물리적 구조 등을 개선
3단계	SQL 튜닝	SQL문 분석을 통해서 성능을 발휘할 수 있도록 조율

### 2. 데이터베이스 튜닝의 단계별 방법

튜닝영역	튜닝방법	설명
설계튜닝	테이블 분할/통합	테이블 액세스 패턴에 따라 분할과 통합
	식별자 지정	식별자의 지정으로 관계 재정의
	효율적 인덱스 전략	최소한의 인덱스로 최대한의 효과 지향
	적절한 데이터타입 선택	Join시 데이터타입에 따른 비효율성 개선
DBMS튜닝	I/O최소화	실제 필요한 Data만을 Read
	Buffer Pool튜밍	사용가능성 높은 Data를 오래 머물게 함
	Commit/Check Point	Commit/Check Point주기 조정
	Thread/Reuse	Middleware의 기능과 연동 등
SQL튜닝	옵티마이저 조정	옵티마이저의 실행계획을 점검 및 조정
	힌트사용	힌트를 사용해 원하는 실행계획으로 유도
	부분범위처리 사용	일부만 액세스한 후 결과를 리턴해 빠른응답
	인덱스 활용	인덱스를 무용지물로 만드는 SQL수정

### 3. 데이터베이스 튜닝의 기대효과

구분	기대효과
서비스 품질	시스템 응답속도의 향상으로 서비스 만족도의 향상 잠재적인 시스템의 장애에 대한 사전방지로 가용성 향상
시스템 비용	하드웨어 증설에 대한 투자비용에 대한 절감 업무효율화에 따른 운영인력 등의 비용 절감
시스템 관리	시스템의 개발 및 유지보수의 안정성, 효율성 확보 튜닝결과를 근간으로 교육을 통해 관리인력의 역량 향상

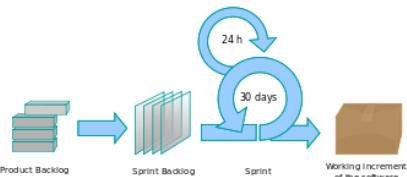
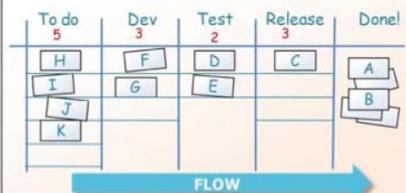
"끝"

13	Scrum, Kanban 방법
문제	Agile 의 Scrum 과 Kanban 방법의 차이점과 보완점을 설명하시오.
도메인	소프트웨어 공학
정의	Scrum: sprint 를 기반으로 반복적으로 진행하는 Agile 개발방법 Kanban: WIP 에 따라 연속적 흐름으로 개발을 진행하는 Lean 개발방법
키워드	Agile, sprint, burn down chart, Lean, WIP, kanban board
출제의도분석	널리 사용되는 Scrum 의 단점을 보완하기 위해 Kanban 과의 비교 분석
답안작성 전략	개발 실무적인 관점에서 차이점을 분석하고 보완가능한 방향을 제시
참고문헌	칸반과 스크럼(헨릭 크니버그/마티아스 스카린, 인사이트)
풀이 기술사님	이홍철 (bluerobe@naver.com)

## 1. Scrum 과 Kanban 의 정의

Scrum	Kanban
product backlog를 sprint단위로 분할하여 sprint를 통해 반복적으로 개발하며 daily meeting, review등을 통해 개선활동을 수행하는 Agile의 대표적인 개발방법	칸반보드를 통해 개발공정을 시각화 하고 WIP 제한을 이용해 workflow 상의 공정을 관리하고 최적화 하는 Lean 생산방식 기반의 개발방법

## 2. Scrum 과 Kanban 의 비교

항목	Scrum	Kanban										
수행절차	 <p>Product Backlog → Sprint Backlog → Sprint (24 h) → Working Increment of the software (30 days)</p>	 <table border="1"> <tr> <th>To do</th> <th>Dev</th> <th>Test</th> <th>Release</th> <th>Done</th> </tr> <tr> <td>H I J K</td> <td>F G</td> <td>D E</td> <td>C</td> <td>A B</td> </tr> </table>	To do	Dev	Test	Release	Done	H I J K	F G	D E	C	A B
To do	Dev	Test	Release	Done								
H I J K	F G	D E	C	A B								
수행원리	sprint 를 통한 반복적 개발	workflow 를 통한 연속적 개발										
수행규칙	1) 3 역할자 - product owner, scrum master, team 2) 3 미팅 - sprint meeting, daily scrum, review 3) 3 산출물 - product backlog, sprint backlog, burn down chart	1) workflow 시각화 2) WIP 제한 3) 소요시간 측정 및 최적화										
수행단위	sprint backlog 를 통한 time boxing	WIP 제한을 통한 작업량 조절										
역할정의	3 개의 역할자로 정의	별도의 역할자 정의하지 않음										
공정관리	burn down chart 와 velocity 측정	workflow 가시화와 WIP 제한										
작업조정	sprint 진행중 작업추가 불가능	workflow 최적화를 통해 조정가능										

- 현재 Scrum 은 Agile 의 확산과 함께 널리 사용되고 있으나, Scrum 이 갖고 있는 상대적인 경직성, 비자율성 등으로 인한 개발팀의 어려움을 해소할 필요가 있음

Notes

### 3. 비교결과에 따른 보완점

항목	설명	
보완방향	1) 한가지 개방방법이 다른 개발방법을 완전히 대체하는 방향은 긍정적이지 않음 2) scrum 기반의 kanban 도입 - 변경 등으로 인해 배포가 빈번하게 요구되는 현 개발상황을 고려하면서 scrum 이 가진 부정적인 이슈를 해소하기 위해 kanban 을 도입함	
보완내용	잦은 미팅의 오버헤드	카이젠(개선)으로 회고를 대체함으로써 뒤돌아 비판하는 대신 진취적인 프로세스 개선으로 긍정적인 에너지 집중
	story point 추정의 불확실성	예상보다 큰 작업은 sprint 에 오버헤드를 초래하므로 WIP 조정, 프로세스 최적화 등으로 불확실성에 대응
	sprint 변경의 경직성	sprint planning, spring meeting 등을 카이젠으로 대체하고 workflow 를 따라 WIP 을 조정함으로써 유연성을 부여
	velocity 에 대한 압박	sprint 로 인한 완료의 압박을 없애고, workflow 시각화와 칸반보드를 통한 공정관리를 통해 더 적은 아이템을 더 빨리 만들기

- 개발조직에 대한 긍정적인 영향으로 생산성의 향상을 기대할 수 있으며, 품질보증활동이 포함된 프로세스를 통해서 100%의 코드리뷰와 95%이상의 코드커버리지를 현실적인 목표로 삼을 수 있음

"끝"

1	Apache Spark, Storm
문제	빅데이터 분산처리시스템인 하둡 MapReduce 의 한계점을 중심으로 Apache Spark 와 Apache Storm 을 비교하여 설명하시오.
도메인	디지털 서비스
정의	Spark: 인메모리 컴퓨팅 기반의 데이터 분산처리 시스템 Storm: 스트리밍 데이터 처리 기반의 실시간 데이터 처리 시스템
키워드	실시간 처리, 스트리밍 데이터, 처리속도, 반복작업
출제의도분석	데이터와 분석목적 등에 따라 선택되는 빅데이터 처리 시스템에 대한 이해
답안작성 전략	MapReduce 의 한계와 연관된 Spark 와 Storm 의 구조, 기능 등을 설명
참고문헌	차세대 실시간 빅데이터 분산 시스템 동향(주간기술동향 2014.9.3)
모범목차	<ol style="list-style-type: none"> <li>1. 하둡 MapReduce 의 한계 및 대안             <ol style="list-style-type: none"> <li>가. 하둡 MapReduce 의 한계</li> <li>나. 하둡 MapReduce 의 대안기술</li> </ol> </li> <li>2. Apache Spark 와 MapReduce             <ol style="list-style-type: none"> <li>가. Apache Spark 의 인메모리 처리 메커니즘</li> <li>나. MapReduce 와 비교한 Spark 의 개선사항</li> </ol> </li> <li>3. Apache Storm 과 MapReduce             <ol style="list-style-type: none"> <li>가. Apache Storm 의 실시간 처리 메커니즘</li> <li>나. MapReduce 와 비교한 Storm 의 개선사항</li> </ol> </li> <li>4. MapReduce, Spark, Storm 의 기능 비교</li> </ol>
풀이 기술사님	이홍철 (bluerobe@naver.com)

## 1. 하둡 MapReduce 의 한계 및 대안

### 가. 하둡 MapReduce 의 한계점

한계	영향
실시간 처리의 한계	일정기간 수집된 자료를 대상으로 하는 일괄처리(Batch)만 처리가능
다양한 데이터 연산의 한계	데이터간의 통신이 필수적인 복잡연산 등의 처리 불가
반복작업의 비효율성	반복작업에 대한 처리비용이 높아 데이터 처리가 비효율적임
Single Point of Failure	별도의 이중화 기술과 솔루션을 통해서 시스템 가용성 확보 필요
Disk I/O 와 NW Traffic 점유	중간데이터 전송단계에서 높은 자원점유로 타 시스템에 영향
높은 기술적 숙련 요구	시스템의 사용 난이도가 높아 별도의 프로그램 개발 등의 필요

- 하둡 MapReduce 의 일괄처리 방식 시스템으로는 RFID, SNS 등의 실시간 데이터 처리에 한계를 나타내, 이를 해결하기 위해 인메모리 데이터 처리기술 기반의 Spark 와 대용량 스트리밍 데이터를 처리가 가능한 Storm 등의 기술들이 대안으로 제시되고 있음

### 나. 하둡 MapReduce 의 대안기술

기술	정의	특징
Apache Spark	UC Berkeley 의 AMP 랩에서 개발한 인메모리 컴퓨팅 기반의 데이터 분산처리 시스템	인메모리 프로세싱, 반복작업 효율성
Apache Storm	데이터의 실시간 처리를 위해 개발된 범용 분산환경 기반 실시간 데이터 처리 시스템	빠른 속도, 내고장성, 낮은 운영난이도

## 2. Apache Spark 와 MapReduce

### 가. 인메모리 기반 Apache Spark 의 동작 메커니즘

메커니즘	설명
<p>A: RDD B: RDD C: RDD D: RDD E: RDD F: RDD</p> <p>Stage 1: groupBy</p> <p>Stage 2: map, filter</p> <p>Stage 3: join</p> <p>= RDD      = cached partition</p>	<p>RDD: 데이터 집합을 추상화한 객체 RDD 연산자: join, union, sort, filter, map, flatMap</p> <ol style="list-style-type: none"> <li>Stage1: 외부의 Local File 과 같은 데이터 소스로부터 RDD 가 생성됨</li> <li>Stage2: RDD 는 map, filter 등의 계획된 연산을 거쳐 변환됨</li> <li>stage3: 변환된 RDD 를 count, collect 등의 action 을 통해 외부 데이터 소스로 변환</li> </ol>

- 데이터를 RDD 단위로 처리하며, 파이프라인을 따라 인메모리에서 대부분의 연산을 수행함

### 나. MapReduce 와 비교한 Spark의 개선사항

개선사항	설명	MapReduce
Disk I/O 효율	MapReduce 의 셔플링 단계와 같이, 연산 실패 시 복구를 위한 경우를 제외한 대부분의 연산이 인메모리에서 동작	I/O 로 인한 병목
반복작업처리	RDD 형태로 데이터를 유지하는 구조는 반복연산이 많은 기계학습 등에 적합하며 하둡에 비해 최대 10 배까지의 성능개선을 보임	반복작업 비효율
다양한 연산처리	MapReduce 와 유사한 map, flatMap 연산은 물론 join, union, sort, filter 등의 RDD 의 변환/관리에 필요한 다양한 연산을 제공	Map, Reduce 만 가능

## 3. Apache Storm 과 MapReduce

### 가. Apache Storm 의 스트리밍 데이터 처리 메커니즘

메커니즘	설명
<p>streams</p> <ul style="list-style-type: none"> <li>tweets</li> <li>favorites, retweets</li> <li>impressions</li> </ul> <p>storm</p> <p>Spout → bolt → bolt → bolt</p> <p>\$\$\$\$ → realtime dashboards → new features</p>	<p>Spout: 스트리밍 데이터를 입력 받는 노드 Bolt: 로직에 따라 Aggregation, Filtering 등 처리</p> <ol style="list-style-type: none"> <li>Spout 는 데이터 소스로부터 스트리밍 데이터를 직접 입력 받음</li> <li>Spout 는 입력된 데이터를 방향성을 지정된 볼트로 전달</li> <li>Bolt 는 사용자가 정의한 지정된 데이터 처리 로직을 수행</li> </ol>

- 하둡과 달리, 사용자가 명시적으로 정지할 때 까지 지속적으로 Spout 와 Bolt 를 통한 처리를 계속함

## 나. MapReduce 와 비교한 Storm 의 개선사항

개선사항	설명	MapReduce
실시간 처리	Storm 은 트위터에서 생산되는 대용량 데이터를 실시간으로 복합분석 할 수 있는 플랫폼으로 개발됨	일괄처리(Batch)
Fault tolerance	작업 노드를 관리하는 Master Node 인 Nimbus 를 통해서 지속적으로 Slave 상태를 모니터링 하여 문제발생시 작업을 재분배함	Single Point of Failure
스트리밍 데이터	Spout, Bolt 로 구성된 토플로지를 통해서 스트리밍 데이터를 직접 입력 받고, 별도의 정지명령이 입력될 때까지 지속적으로 데이터를 처리함	저장된 데이터 처리

## 4. MapReduce, Spark, Storm 의 기능 비교

기능	Spark	Storm	MapReduce
데이터 처리	일괄처리 방식	실시간 스트리밍 처리	일괄처리 방식
업데이트 단위	파일 또는 테이블	스트림(튜플)	레코드
컴퓨팅 환경	인메모리 기반	인메모리 기반	디스크 기반
반복연산	강함	보통	약함
프로그래밍 언어	Scala	Clojure	Java
SQL 지원여부	Spark SQL 에서 지원	관련 없음	Tajo 에서 지원
추천환경	반복 또는 많은 연산, 데이터간 교환이 적은 시스템에 적합	응답시간이 짧고, 동일데이터에 다양한 질의가 존재하는 시스템	복잡도가 높지 않고 데이터 교환이 많은 시스템에 적합

- 빅데이터는 활용하는 응용분야에 따라 데이터의 특성과 사용자의 작업특성이 상이하므로, 특성들을 잘 분석하여 다양한 기술을 적절히 조합하는 형태로 분석하는 것이 효율적임.

"끝"

2	오픈 소스 하드웨어
문제	오픈 소스 하드웨어(Open Source Hardware)의 개념과 구성 요소를 설명하고 아두이노(Arduino)와 라즈베리 파이(Raspberry Pi)를 비교하여 설명하시오.
도메인	디지털 서비스, CA
정의	HW 제작에 필요한 회로도, 설명서, 펌웨어 등을 공개해 누구나 동일한 제품 또는 2차 제품을 개발할 수 있도록 지원하는 하드웨어
키워드	BOM, HW 디자인, 펌웨어
출제의도분석	IOT, Maker 운동 등 오픈 소스 하드웨어를 응용하여 사용하는 분야의 확대와 일반 하드웨어 업체에서의 OSH 활용 추세 확대 대응
답안작성 전략	OSH의 정확한 정의, 개념 설명이 필요하며 아두이노와 라즈베리 파이의 차이를 비교(성능과 소프트웨어 활용 범위의 차이 중심)
참고문헌	Wikipedia의 관련 항목 OSHWA(Opne Source Hardware Association) – <a href="http://www.oshwa.org">www.oshwa.org</a>
모범목차	<ol style="list-style-type: none"> <li>1. 오픈 소스 하드웨어의 개요             <ol style="list-style-type: none"> <li>가. 오픈 소스 하드웨어의 개념(정의)</li> <li>나. 오픈 소스 하드웨어의 특징</li> </ol> </li> <li>2. 오픈 소스 하드웨어의 구성도와 구성 요소             <ol style="list-style-type: none"> <li>가. 오픈 소스 하드웨어의 구성도</li> <li>나. 오픈 소스 하드웨어의 구성 요소</li> </ol> </li> <li>3. 오픈 소스 하드웨어의 주요 플랫폼             <ol style="list-style-type: none"> <li>가. 주요 프랫폼 – 아두이노와 라즈베리 파이의 특징</li> <li>나. 아두이노와 라즈베리 파이의 비교</li> </ol> </li> <li>4. 아두이노와 라즈베리 파이의 대응 영역 차이</li> </ol>
풀이 기술사님	강용제 ( <a href="mailto:yongjeikang@gmail.com">yongjeikang@gmail.com</a> )

### ■ 오픈 소스 하드웨어의 개념(정의)

- 누구나 만들고 수정, 배포하고 사용할 수 있도록 일반에 디자인이 공개되어 있으며 손으로 만져질 수 있는 인공물
- 해당 제품과 똑 같은 모양 및 기능을 가진 제품을 만드는 데 필요한 모든 것(회로도, 자재 명세서, 인쇄회로 기판 도면 등)을 대중에게 공개한 전자 제품
- 하드웨어 기술 언어(VHDL)가 대중에게 공개된 프로그래머블 논리 소자(PLC)

### ■ 오픈 소스 하드웨어의 특징

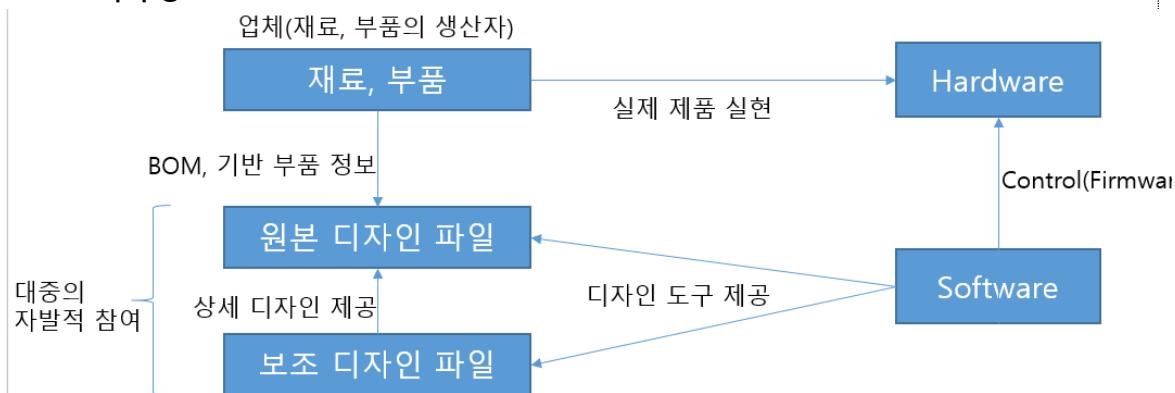
구분	내용
공개성	기술에 대한 특허 라이선스가 없고 제품 개발에 필요한 리소스가 공개되어 있음
대중 제조 가능	누구라도 부품을 직접 구매해 조립하여 완성 가능
2 차 개발 허용	누구라도 이미 공개된 OSH를 이용해 새로운 기기를 만들 수 있음
SW 융합성	HW 운영을 위해 사용되는 펌웨어와 기타 SW를 OSS 형태로 융합 제공

### ■ OSS 와 OSH 의 비교

구분	OSS	OSH
개별 특성	<ul style="list-style-type: none"> <li>- 무형의 SW 제품</li> <li>- 배포 비용이 없거나 저가</li> <li>- SW 만으로 구성됨</li> <li>- 상업적 이용에 대한 심리적 저항 존재</li> </ul>	<ul style="list-style-type: none"> <li>- 유형의 HW 제품</li> <li>- 배포, 재생산을 위해 부품, 생산 설비 비용이 소모됨</li> <li>- SW 와 함께 구성되는 경우가 대부분</li> <li>- 상업적 이용을 권장 또는 촉진하려는 심리(하드웨어 제품, 부품을 생산할 생산자 필요)</li> </ul>
공동 특성	<ul style="list-style-type: none"> <li>- 로열티 없는 오픈 소스 라이선스 채택</li> <li>- 자발적 참여자들에 의한 시간, 노동력 투자</li> <li>- 다수의 능력에 따른 참여, 필요에 따른 소비</li> </ul>	

- OSS 와 OSH 는 서로 다른 개발, 배포 특성을 가지며 이에 따라 지지자들의 태도가 다름

### ■ OSH 의 구성도



- OSS 가 제공하는 다양한 SW 를 활용하여 OSH 의 공개성을 침해하지 않는 범위에서 영리를 목적으로 하는 생산자의 참여를 권장(이익 실현을 부정하지 않음)

### ■ OSH 의 구성 요소

요소	내용	제공 형태
원본 디자인 파일	<ul style="list-style-type: none"> <li>- 핵심 기술 요소</li> <li>- 누구나 접근, 이용 가능한 형식으로 저장된 설계 파일</li> </ul>	CAD, SVG, AI 등
보조 디자인 파일	- 원본 디자인 파일을 더 명확히 이해할 수 있게 하는 파일	Image, 문서 등
재료 명세서	<ul style="list-style-type: none"> <li>- 공개된 HW 를 구성하는 각 부품, 재료에 대한 명세</li> <li>- 누구나 구입이 가능하도록 제조사 부품 번호, 대략적인 가격, 기타 대체품 등에 대한 설명을 포함</li> </ul>	BOM
SW	<ul style="list-style-type: none"> <li>- HW 의 운용, 제어에 필요한 펌웨어 또는 SW</li> <li>- 공개적 이용과 2 차 저작이 가능하도록 OSS 형태를 취함</li> </ul>	소스 코드, 바이너리 코드
설명서	<ul style="list-style-type: none"> <li>- OSH 의 제조, 조립, 이용에 필요한 모든 정보를 포함하는 설명서</li> <li>- 내용에 대한 이해를 높이기 위해 다양한 형태로 제공(동영상, 이미지, 문서 등)</li> </ul>	문서, 이미지, 동영상, 음성 등

## ■ 주요 OSH 플랫폼

구분	특징
아두이노 (Arduino)	<ul style="list-style-type: none"> <li>- 이탈리아에서 만들어졌으며 OSH 의 가장 널리 알려진 구현체</li> <li>- AVR 또는 Cortex 프로세서를 사용하며 응용 하드웨어의 사용을 위한 PIN 내장</li> <li>- 다양한 확장 HW 가 출시되어 광범위한 생태계 구성</li> <li>- HW 와 개발 환경을 동시에 제공하여 초보자의 접근이 쉬움</li> </ul>
라즈베리 파이 (Raspberry Pi)	<ul style="list-style-type: none"> <li>- 영국에서 초중고생의 컴퓨터 교육을 목적으로 탄생</li> <li>- 컨트롤러라기 보다는 작은 컴퓨터의 성격을 가짐</li> <li>- 리눅스 등 범용 OS 의 운영이 가능</li> <li>- HW 확장을 위한 GPIO 를 내장</li> </ul>
갈릴레오 (Galileo)	<ul style="list-style-type: none"> <li>- 인텔이 아두이노와의 협력을 통해 x86 아키텍처를 이용해 제조</li> <li>- 윈도우, 리눅스 등의 다양한 OS 지원</li> <li>- Wearable 등의 응용에 적극적으로 대응 중</li> </ul>
큐비 보드 (Cubie Board)	<ul style="list-style-type: none"> <li>- 중국산 ARM 호환 CPU 를 이용한 마이크로 컴퓨터</li> <li>- 매우 저가에 강력한 HW 성능을 제공</li> <li>- 자체 Linux 배포판 제공으로 다양한 응용 개발이 가능함</li> </ul>

## ■ 아두이노와 라즈베리 파이의 비교

구분	아두이노	라즈베리 파이
개념	 <ul style="list-style-type: none"> <li>- 단순한 회로로 HW 를 구현하여 컨트롤러로 이용이 쉽도록 구성.</li> <li>- AVR 과 간단한 주변 칩, AVR 과 직접 연결되는 PIN 을 제공</li> <li>- 전문 지식이 없더라도 레고 블록을 조립하듯이 이용하여 피지컬 컴퓨팅 구현이 가능</li> </ul>	 <ul style="list-style-type: none"> <li>- 일반 PC 의 대용으로 저가에 다목적 컴퓨팅을 교육하기 위한 범용 싱글 보드 컴퓨터</li> <li>- 다양한 주변 인터페이스(USB, Ethernet, HDMI 등등)를 제공하여 확장성이 높음</li> <li>- 학교에서 기초 컴퓨터 과학 교육에 응용이 가능하도록 저가에 제공</li> </ul>
형태	- 단일 보드 마이크로 컨트롤러	- 단일 보드 마이크로 컴퓨터
프로세서	<ul style="list-style-type: none"> <li>- Atmel AVR(ATMega)</li> <li>- Cortex-M3</li> <li>- Cortex-A8</li> </ul>	- ARM 프로세서(ARM11)
개발환경	<ul style="list-style-type: none"> <li>- C/C++</li> <li>- Processing/Sketch</li> </ul>	<ul style="list-style-type: none"> <li>- Python, C, Perl 등</li> <li>- 범용 OS(Linux)의 직접 운영이 가능</li> </ul>
네트워크	- 별도의 확장 보드를 통해 네트워크 연결	- Ethernet Port 를 내장
활용	<ul style="list-style-type: none"> <li>- 마이크로 컨트롤러로서 간단한 하드웨어의 제어 용도 등에 사용</li> </ul>	<ul style="list-style-type: none"> <li>- 작은 범용 컴퓨터</li> <li>- HTPC, NAS 등</li> </ul>

*Notes*

- 3D 프린터의 활용과 함께 Maker 운동의 핵심 HW로 이용되고 있음
- OSH는 개념상 전자적 제어 없이 기계적 제조 물품(자동차, 카트 등)에도 적용이 가능하나 대부분의 OSH는 아두이노와 같은 전자 제품에 대해 통용되고 있음

"끝"

3	데이터베이스 키추출
문제	학사관리시스템에서 아래 스키마(Schema)를 이용할 때, 키(Key or Candidate Key), 주키(Primary Key), 외부키(Foreign Key), 슈퍼키(Super Key)를 정의하고 추출하는 과정 및 방법을 설명하시오. "학생" 스키마(학번, 주민등록번호, 이름, 전공, 성별, 학과번호) "학과" 스키마(학과번호, 학과명, 설립연도)
도메인	데이터베이스
정의	키: 릴레이션 내에서 튜플을 유일하게 식별할 수 있는 애트리뷰트의 집합
키워드	유일성, 최소성, 불변성, 존재성, 대표성
출제의도분석	데이터베이스의 기본지식인 키에 관한 이해와 모델링에 활용능력 평가
답안작성 전략	키 추출과정을 제시하고, 추출과정에 필요한 근거와 함께 사례에 대한 키를 추출
참고문헌	데이터베이스론(이석호, 정의사)
모범목차	<ol style="list-style-type: none"> <li>1. 데이터베이스 키의 정의</li> <li>2. 데이터베이스 기본키의 특성과 추출과정 설명             <ol style="list-style-type: none"> <li>가. 기본키의 특성</li> <li>나. 데이터베이스 키의 추출과정 설명</li> </ol> </li> <li>3. 주어진 스키마에 대한 키 추출과정 및 방법             <ol style="list-style-type: none"> <li>가. 키 추출과정 및 방법</li> <li>나. 키 추출결과</li> </ol> </li> </ol>
풀이 기술사님	이홍철 (bluerobe@naver.com)

## 1. 데이터베이스 키의 정의

종류	정의
키	릴레이션 내에서 튜플을 유일하게 식별할 수 있는 애트리뷰트의 집합
주키	유일성과 최소성을 만족하는 후보키 중에서 대표성을 지닌 키
외부키	다른 릴레이션의 기본키를 참조하는 키
슈퍼키	튜플에 대한 유일성은 만족하나 최소성을 만족하지 못하는 키

## 2. 데이터베이스 기본키의 특성과 키 추출과정 설명

### 가. 기본키의 특성

특성	설명
유일성	기본키에 의해 릴레이션 내의 모든 튜플을 유일하게 구분함
최소성	기본키를 구성하는 애트리뷰트의 수는 유일성을 만족하는 최소의 수
불변성	기본키가 한 번 특정 튜플에 지정되면 그 기본키의 값은 변하지 않음
존재성	기본키가 지정되면 반드시 그 애트리뷰트의 값이 존재해야 함

### 나. 데이터베이스 키의 추출과정 설명

단계	추출 기준	설명
1) 키 도출	속성 추출	튜플을 식별할 수 있는 후보 애트리뷰트를 추출
2) 슈퍼키 선정	유일성 검증	유일성을 만족하는 애트리뷰트를 선정
3) 후보키 선정	최소성 검증	최소성과 유일성을 만족하는 애트리뷰트를 선정
4) 기본키 선정	대표성 검증	후보키중에서 대표성 만족하는 애트리뷰트를 선정
5) 외래키 선정	관계 설정	릴레이션 간의 관계설정을 통해서 외래키 설정

### 3. 주어진 스키마에 대한 키 추출과정 및 방법

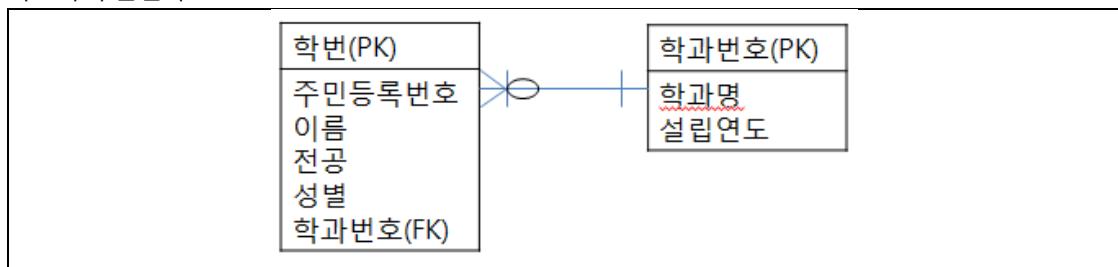
#### 가. 키 추출과정 및 추출방법

추출 과정	추출 기준	추출 방법
1) 슈퍼키 선정	유일성	학생스키마: 학번, 주민등록번호 학과스키마: 학과번호 - 각 스키마에 대해 유일성을 보장
2) 후보키 선정	유일성, 최소성	학생스키마: 학번과 주민등록번호 모두 후보키 학과스키마: 학과번호 - 학과스키마는 단일 후보키만 존재함
3) 기본키 선정	불변성	학생스키마: 학번 학과스키마: 학과번호 - 학생스키마의 주민등록번호는 신변상의 이유로 말소(삭제) 가능성이 있어 불변성을 충족시키지 못함
	존재성	학생스키마: 학번 학과스키마: 학과번호 - 학번 없는 학생, 학과번호 없는 학과는 존재하지 않으므로 존재성을 만족함
4) 외래키 선정	관계 설정	학생스키마: 학과번호 - 학생:학과=M:1 의 관계

- 학생스키마의 기본키는 학번, 학과스키마의 기본키는 학과번호로 선정되었으며,

학생스키마의 학과번호는 외래키로 학과스키마의 학과번호를 참조함

#### 나. 키 추출결과



"끝"

4	ISO 26262
문제	자동차에 탑재되는 소프트웨어의 기능안전 국제규격인 ISO26262에 대한 정의, 배경, 규격 및 기술기준을 설명하시오.
도메인	소프트웨어공학
정의	자동차의 기능안전성에 대해 ISO에서 제정한 자동차 기능안전 국제규격
키워드	안전생명주기, ASIL, 심각도, 노출가능성, 통제가능성, 기능안전 요구사항
출제의도분석	차량의 전자부품 증가와 네트워크화로 인해 이에 대한 기술표준의 이해여부 평가
답안작성 전략	ISO 26262의 규격을 안전생명주기, ASIL 등을 중심으로 이해하고 정확하게 서술
참고문헌	한국산업표준 도로차량-기능안전 제 1 부~10 부(KS R ISO 26262-1 ~ 10)
모범목차	<ol style="list-style-type: none"> <li>1. ISO 26262의 개요             <ol style="list-style-type: none"> <li>가. ISO 26262의 정의</li> <li>나. ISO 26262의 배경</li> </ol> </li> <li>2. ISO 26262의 규격             <ol style="list-style-type: none"> <li>가. ISO 26262 규격의 구조</li> <li>나. ISO 26262 규격의 설명</li> </ol> </li> <li>3. ISO 26262의 기술기준             <ol style="list-style-type: none"> <li>가. 차량 안전성 보전 등급 ASIL</li> <li>나. 기능 안정성을 위한 요구사항</li> </ol> </li> <li>4. ISO 26262에 대한 대응현황과 전략</li> </ol>
풀이 기술사님	이홍철 (bluerobe@naver.com)

## 1. ISO 26262의 개요

### 가. ISO 26262의 정의

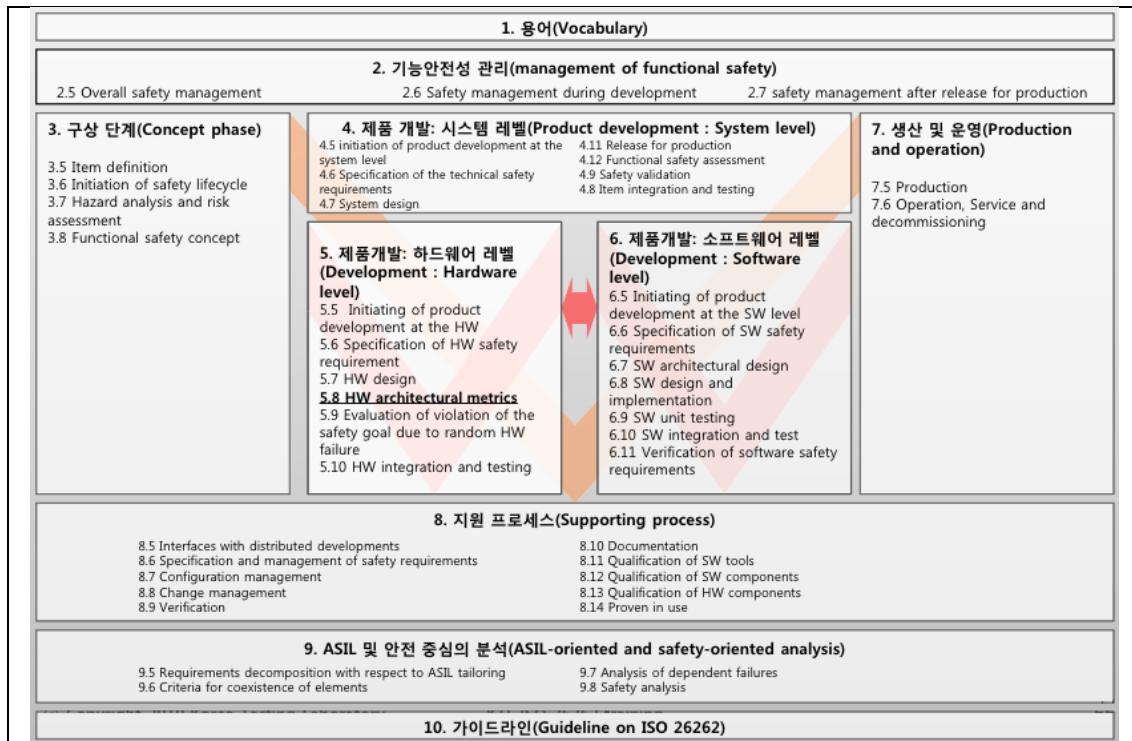
- 자동차에 탑재되는 SW의 오류로 인한 사고방지를 위해 ISO에서 제정한 자동차 기능 안전 국제 규격
- ISO26262는 프로세스 모델과 함께 요구되는 활동, 유무형의 증거물, 그리고 개발과 생산에 사용되는 방식을 정의

### 나. ISO 26262의 배경

- 1) Connected Car, Smart Car 등의 연구와 상용화의 기대에 따라서 차량의 안전성에 대한 요구가 높아짐
- 2) 차량에 탑재되는 전자제어장치(ECU)의 증가와 네트워크화로 인한 오류발생 가능성 증가
- 3) 차량의 제어시스템의 복잡도 증가에 따른 개발프로세스의 효율 및 비용절감 효과 기대
- 4) 일반전기전자장치의 기능안전규격인 IEC 61508의 차량의 특수성 반영에 대한 한계 존재

## 2. ISO 26262 의 규격

### 가. ISO 26262 규격의 구조



- 시스템 레벨, 하드웨어 레벨, 소프트웨어 레벨에 관련된 Part 4, 5, 6은 Embedded V-model을 기반으로 구성되어 있음

### 나. ISO 26262 규격의 설명

Part	구성	내용
1	용어	ISO 26262 내에서 사용되는 용어와 용어의 정의
2	기능안전성 관리	안전수명주기를 기반으로 기능안전관리에 대한 요구사항 정의
3	구상 단계	안전목표와 안전 메커니즘을 정의
4	시스템 레벨	하드웨어와 소프트웨어를 통합하는 시스템 레벨의 규격
5	하드웨어 레벨	하드웨어 수준에서 V 모델에 따른 요구사항 정의
6	소프트웨어 레벨	소프트웨어 수준에서 V 모델에 따른 요구사항 정의
7	생산 및 운영	생산계획, 양산, 서비스 등에 관한 요구사항 정의
8	지원 프로세스	안전 요구사항 관리, 명세 방법 등 적용단계의 지원 프로세스
9	ASIL 및 안전중심의 분석	ASIL에 관한 요구사항 분해와 고장, 안전 분석
10	가이드라인	ISO 26262의 이해와 적용에 도움이 되는 정보

### 3. ISO 26262 의 기술기준

가. 차량 안전성 보전 등급 ASIL

항목	설명																																			
ASIL 등급	<p>Potential Risk</p> <p>Risk reduction measure</p> <p>ASIL A      ASIL B      ASIL C      ASIL D</p> <p>upper risk of system</p> <p>Required risk reduction by the technical solution</p> <p>Standard Development</p>																																			
	<ul style="list-style-type: none"> <li>- 최저등급 ASIL A에서 최고등급 ASIL D까지 4개의 등급으로 구분됨</li> <li>- ASIL 등급이 높다는 것은 개발대상의 오류에 대한 피해가 클 수 있다는 의미</li> <li>- 높은 수준의 안전등급일수록 안전에 대한 요구사항이 더 높아짐</li> </ul>																																			
ASIL 측정방법	<p><b>재난 요인별 심각도 분석</b></p> <p>TABLE 1. 잠재적 재난이나 위험에 대한 심각도 등급</p> <table border="1"> <thead> <tr> <th>등급 (Class)</th> <th>S0</th> <th>S1</th> <th>S2</th> <th>S3</th> </tr> </thead> <tbody> <tr> <td>설명</td> <td>No injuries</td> <td>Light and moderate injuries</td> <td>Severe and life-threatening injuries (survival probable)</td> <td>Life-threatening injuries (survival uncertain), fatal injuries</td> </tr> </tbody> </table> <p><b>재난 요인별 노출 가능성 분석</b></p> <p>위험 및 재난의 노출 가능성 등급</p> <table border="1"> <thead> <tr> <th>등급 (Class)</th> <th>E0</th> <th>E1</th> <th>E2</th> <th>E3</th> <th>E4</th> </tr> </thead> <tbody> <tr> <td>설명</td> <td>Incredible</td> <td>Very low probability</td> <td>Low probability</td> <td>Medium probability</td> <td>High probability</td> </tr> </tbody> </table> <p><b>재난 요인별 통제 가능성 분석</b></p> <p>재난 통제 가능성 등급</p> <table border="1"> <thead> <tr> <th>등급 (Class)</th> <th>C0</th> <th>C1</th> <th>C2</th> <th>C3</th> </tr> </thead> <tbody> <tr> <td>설명</td> <td>Controllable in general</td> <td>Simply controllable</td> <td>Normally controllable</td> <td>Difficult to control to uncontrollable</td> </tr> </tbody> </table> <ul style="list-style-type: none"> <li>- 위험의 심각도(S1~S3), 노출 가능성(E0~E4), 통제 가능성(C0~C3)을 평가한 후, ASIL A~D 등급을 부여함</li> <li>- QM은 상대적으로 위험등급이 낮아 품질관리 차원에서 대응하는 것을 권고</li> </ul>				등급 (Class)	S0	S1	S2	S3	설명	No injuries	Light and moderate injuries	Severe and life-threatening injuries (survival probable)	Life-threatening injuries (survival uncertain), fatal injuries	등급 (Class)	E0	E1	E2	E3	E4	설명	Incredible	Very low probability	Low probability	Medium probability	High probability	등급 (Class)	C0	C1	C2	C3	설명	Controllable in general	Simply controllable	Normally controllable	Difficult to control to uncontrollable
등급 (Class)	S0	S1	S2	S3																																
설명	No injuries	Light and moderate injuries	Severe and life-threatening injuries (survival probable)	Life-threatening injuries (survival uncertain), fatal injuries																																
등급 (Class)	E0	E1	E2	E3	E4																															
설명	Incredible	Very low probability	Low probability	Medium probability	High probability																															
등급 (Class)	C0	C1	C2	C3																																
설명	Controllable in general	Simply controllable	Normally controllable	Difficult to control to uncontrollable																																

ASIL 정의	
ASIL Definition	C1    C2    C3
S1	E1    QM    QM    QM
	E2    QM    QM    QM
	E3    QM    QM    A
	E4    QM    A    B
S2	E1    QM    QM    QM
	E2    QM    QM    A
	E3    QM    A    B
	E4    A    B    C
S3	E1    QM    QM    A
	E2    QM    A    B
	E3    A    B    C
	E4    B    C    D

QM(Quality Management)  
: 기능안전과 무관

ASIL A: 기능안전등급 A  
ASIL B: 기능안전등급 B  
ASIL C: 기능안전등급 C  
ASIL D: 기능안전등급 D

High

## 나. 기능안전성을 위한 요구사항

구분	주요 요구사항
시스템 레벨	HW, SW 의 구조적 결함을 피하기 위한 설계기법(FTA, FMEA 등)
	시스템 설계에 대한 검증기법(Inspection, Simulation, Prototyping 등)
	시스템 검증을 위한 테스트 입력 생성방법(경계값, 동등분할 등)
	시스템 통합을 위한 시험방법(성능 테스팅, 오류주입 등)
하드웨어 레벨	HW 설계 및 안전분석 기법
	고장분류 및 안전기능의 유효성 평가
	HW 우발결함에 대한 제품의 안전성 평가
	Micro Controller에 대한 요구사항
소프트웨어 레벨	SW 모델링 및 코딩기법(SW 구조설계 방법론, MISRA C, 복잡도 산출 등)
	SW 안전기능의 구현방법(오류검사, SW 제어 모니터링 등)
	SW의 안전성 검증방법(Inspection, 정적 분석, 동적 분석 등)
	SW 시험방법(오류 주입, 커버리지 측정시험, 테스트 입력값 생성방법 등)

- 기능안전을 위한 요구사항을 기존의 개발 process에 추가 또는 통합하도록 요구하고 있음

## 4. ISO 26262에 대한 대응현황 및 전략

구분	대응현황
대응 현황	<ul style="list-style-type: none"> <li>- ISO 26262를 국내표준화 한국산업표준(KS)화 하여 국내 현행화에 대응</li> <li>- 현대자동차의 2015년 하반기에 개발을 시작하는 신차에 ISO 26262를 적용</li> <li>- 현대·기아차 주요 전장부품 협력사에 대한 ISO 26262 역량평가 실시</li> </ul>
향후 전략	<ul style="list-style-type: none"> <li>- 자동차의 기능안전 보급 및 확산을 궁극적인 목표로 설정</li> <li>- 기능안전 프로세스를 구축하여 전체 안전관리, 개발단계 안전관리 지원</li> <li>- 엔지니어링 기술지원을 통해 안전성 분석을 지원하고 안전성 설계 지원</li> <li>- 국내 전문인력 양성을 통해 안전관리 전문가를 육성하고 평가체계를 구축</li> </ul>

- 현대·기아차가 협력사를 대상으로 실시한 ISO26262에 대한 역량평가에서는 전체적으로 낙제점에 가까운 평가결과를 나타낸 바, 역량향상을 위한 자동차 산업차원의 대응이 필요함

"끝"

5	정보통신망법
문제	정보통신망법을 기준으로 개인 정보의 기술적 보호 조치 기준과 관리적 보호 조치 기준의 상세 규정 내용을 각각 5 가지 이상 설명하시오.
도메인	경영 전략
정의	정보 통신망의 이용을 촉진하고 정보 통신 서비스를 이용하는 자의 개인 정보를 보호하기 위한 제반 규정을 규정한 법규
키워드	망분리, 마스킹, 일방향 암호화, Opt-in, Opt-out
출제의도분석	2014년 11월 개정된 "정보통신망 이용 촉진 및 정보보호 등에 관한 법률"이 시행됨에 따라 이에 대한 대응 필요
답안작성 전략	정보통신망법에서 규정하고 있는 기술적 보호 조항과 관리적 보호 조항에 대한 언급이 필요하며 법규에 대한 문제이므로 정확성이 보장되어야 함
참고문헌	Wikipedia의 관련 항목 개정 정보통신망법 대응 전략 – 테크앤로 법률 사무소 법률 제 12681 호 – 대한민국 국회 고시 제 2014-28 호 – 방송통신위원회 국가 법령 정보 - law.go.kr
모범목차	1. 정보통신망법의 개요 가. 정보통신망법의 개념 나. 개인정보의 기술적, 관리적 보호조치 기준의 개념 2. 개인정보의 기술적, 관리적 보호조치 기준 가. 기술적 보호조치 기준 나. 관리적 보호조치 기준
풀이 기술사님	강용제 (yongjeikang@gmail.com)

### ■ 정보통신망법의 개념과 목적

- "정보통신망 이용촉진 및 정보보호 등에 관한 법률"은 정보통신망의 이용을 촉진하고 정보통신서비스를 이용하는 자의 개인정보를 보호, 정보통신망을 건전하고 안전하게 이용할 수 있는 환경을 조성하기 위함
- 개인정보의 기술적, 관리적 보호조치 기준: 방송통신위원회 고시 제 2012-50 호. 정보통신서비스 제공자 등이 이용자의 개인정보를 취급함에 있어서 개인정보가 분실, 도난, 누출, 변조, 훼손 등이 되지 아니하도록 안전성을 확보하기 위하여 취하여야 하는 기술적, 관리적 보호 조치의 구체적 기준

### ■ 정보통신망법에서의 개인 정보 정의

- 생존하는 개인에 관한 정보로서 성명, 주민등록번호 등에 의하여 특정한 개인을 알아볼 수 있는 부호, 문자, 음성, 음향 및 영상 등의 정보(해당 정보만으로는 특정 개인을 알아볼 수 없어도 다른 정보와 쉽게 결합하여 알아볼 수 있는 경우에는 그 정보를 포함한다) – 제 2 조 6 항
- 법률에서의 개인 정보는 광범위한 영역에서 사용되는 다양한 정보가 될 수 있음

### ■ 개인정보의 기술적 보호조치 기준

구분	내용	조항
접근 통제	<ul style="list-style-type: none"> <li>- 개인 정보 취급자에 대한 공인 인증서 등 안전한 인증 수단 사용 의무화</li> <li>- 접속 권한을 IP 주소 등으로 제한 의무화</li> <li>- 접속 기록에 대한 분석으로 개인 정보 유출 탐지 의무화</li> <li>- 개인 정보를 취급하는 컴퓨터에 대한 망 분리 의무화</li> <li>- 개인 정보 취급자의 비밀 번호는 영문 대문자, 영문 소문자, 숫자, 특수 문자 등 중 2종 이상으로 조합해 최소 10 자리 이상 또는 3 종 이상으로 조합해 최소 8 자리 이상 사용</li> <li>- 개인 정보 유출 방지를 위해 개인 정보 취급자의 컴퓨터에 대한 기술 조치 요구</li> </ul>	고시 제 2014-28 호 제 4 조
접속 기록	<ul style="list-style-type: none"> <li>- 개인 정보 처리 시스템에 대한 접속 기록의 보존 의무화</li> <li>- 접속 기록의 위/변조 방지를 위한 보관 및 백업 의무화</li> </ul>	고시 제 2014-28 호 제 5 조
암호화	<ul style="list-style-type: none"> <li>- 비밀 번호, 바이오 정보는 일방향 암호화</li> <li>- 보안 서버에 대해 SSL, 암호화 송수신, 저장 정보 암호화 등 사용 규정</li> </ul>	고시 제 2014-28 호 제 6 조
악성 프로그램 방지	<ul style="list-style-type: none"> <li>- 백신 소프트웨어 사용: 월 1 회 이상 주기적 갱신, 점검, 업데이트 사용</li> </ul>	고시 제 2014-28 호 제 7 조
출력, 복사시 보호 조치	<ul style="list-style-type: none"> <li>- 개인 정보 출력 시(인쇄, 화면 표시, 파일 생성 등) 보호 조치 의무화</li> </ul>	고시 제 2014-28 호 제 8 조
개인 정보 표시 제한	<ul style="list-style-type: none"> <li>- 개인 정보 업무 처리 과정에서 개인 정보의 마스킹 규정(성명 중 이름의 첫 번째 글자 이상, 생년월일, 전화번호 또는 휴대폰 전화번호의 국번, 주소의 읍/면/동, IPv4 의 경우 17~24 비트, IPv6 의 경우 113~128 비트 영역)</li> </ul>	고시 제 2014-28 호 제 9 조

### ■ 개인정보의 관리적 보호조치 기준

구분	내용	조항
개인 정보 보호 조직	<ul style="list-style-type: none"> <li>- 개인 정보 관리 책임자 지정</li> <li>- 개인 정보 관리 책임자와 취급자의 역할 및 책임 규정</li> <li>- 개인 정보 내부 관리 계획 수립 요구</li> <li>- 보호 조치의 이행 여부에 대한 내부 점검 요구</li> </ul>	고시 제 2014-28 호 제 3 조
교육	<ul style="list-style-type: none"> <li>- 개인 정보 관리 책임자와 취급자에 대한 년 2회 이상의 교육 의무화</li> </ul>	고시 제 2014-28 호 제 3 조
접근 통제	<ul style="list-style-type: none"> <li>- 개인 정보에 대한 접근 권한을 개인 정보 관리 책임자 또는 취급자에게만 부여하도록 강제</li> <li>- 접근 권한 부여, 변경, 말소에 대한 기록 보관 의무화</li> <li>- 개인 정보 취급자의 비밀 번호 작성 규칙 수립 규정</li> <li>- 비밀 번호 유효 기간 설정(반기별 1회 이상 변경)</li> </ul>	고시 제 2014-28 호 제 4 조
접속 기록	<ul style="list-style-type: none"> <li>- 개인 정보에 대한 접근 기록을 기록, 보존, 관리하도록 의무화</li> </ul>	고시 제 2014-28 호 제 5 조
암호화	<ul style="list-style-type: none"> <li>- 비밀 번호, 바이오 정보, 주민등록번호, 신용카드번호, 계좌번호 등에 대한 암호화 규정</li> </ul>	고시 제 2014-28 호 제 6 조

Notes

<b>출력, 복사시 보호 조치</b>	- 개인 정보 출력시 용도의 특정, 용도에 따른 출력 항목 최소화 규정 - 개인 정보가 포함된 인쇄물, 저장 매체에 대한 보호 조치 의무화	고시 제 2014-28 호 제 8 조
<b>개인 정보 표시 제한</b>	- 개인 정보 업무 처리시 개인 정보의 마스킹하여 표시 제한 조치 의무화	고시 제 2014-28 호 제 9 조

■ 참고: 기타 고려해야 할 정보통신망법 규정

구분	내용	조항
<b>개인 정보 수집 제한</b>	- 개인 정보 수집을 최소화하도록 규정하고 있으며 서비스 제공을 위한 최소한의 개인 정보 이외의 정보를 제공하지 않는다고 서비스 제공을 거부할 수 없도록 함 - 최소 수집 항목이 아닌 경우 수집 항목에서 제외 또는 선택 항목화	정보통신망법 제 23 조 3 항
<b>기준 명확화</b>	- 개인 정보 최소 수집/보관을 위한 온라인 개인 정보 취급 가이드라인을 제시함 - 수집 기준, 수집/이용 등의 시점, 개인 정보 파기 기준, 동의서 작성 기준, 개인 정보 통제권 등을 제시함	방송통신위원회 공표(2014.11)
<b>개인 정보 누출 통지/신고 의무</b>	- 개인 정보의 누출 등 사고 발생시 통지 및 신고 의무를 강화하고 그 기간 역시 단축함 - 신고 항목에 대한 규정	정보통신망법 제 27 조 3 항
<b>개인 정보 파기 의무</b>	- 개인 정보의 파기와 파기 방법을 규정 - 개인 정보를 파기하지 않을 경우 형사 처벌	정보통신망법 제 27 조
<b>법정 손해 배상</b>	- 정보통신서비스 제공자에 대해 이용자의 손해 배상 청구 가능	정보통신망법 제 32 조 2 항
<b>광고 전송 제한</b>	- 영리 목적의 광고 전송에 대해 Opt-out에서 Opt-in 방식으로 변경	정보통신망법 제 50 조

"끝"

6	기능 점수												
문제	<p>기능 점수(Function Point) 산출 방법에 대하여 설명하고 간이법을 적용하여 아래의 이벤트(Event) 리스트와 "ISBSG(International Software Benchmark Standard Group) 평균 복잡도"를 근거로 하여 기능 점수를 산출하시오.</p> <p>&lt;이벤트(Event) 리스트&gt;</p> <ul style="list-style-type: none"> <li>가. 담당자는 고객 주문을 입력, 수정 삭제한다. (고객 DB)</li> <li>나. 인사 담당자는 사원 목록을 부서 단위로 조회한다. (사원 DB, 부서 DB)</li> <li>다. 인사 담당자는 사원 목록을 단순 출력한다. (사원 DB)</li> <li>라. 인사 담당자는 일정 금액 이상의 급여 수령자 사원 목록을 검색한다. (사원 DB, 급여 DB)</li> <li>마. 월화에 대한 미국 달러(USD) 가치를 찾기 위해 A 은행 외환 DB에서 환율을 검색한다. (외환 DB)</li> <li>바. 인사 담당자는 5년 경력 이상이고, 해당 직무 수행 경험이 있는 사원 목록을 추출한다. (사원 DB)</li> <li>사. 인사 담당자는 신입/경력 사원 입사 시 사원 파일을 갱신한다. (사원 DB)</li> <li>아. 인사 담당자는 외국 사원 입사 시 사원의 급여를 결정하기 위해 H 연합회 통화 정보를 참조한다. (사원 DB, 급여 DB, 통화 DB)</li> <li>자. 회계 시스템은 전표 번호(부서 번호 중 앞자리 2 + 년도 + 일련 번호)를 자동 채변한다. (전표 DB)</li> <li>차. 인사 담당자는 사원 현황을 엑셀 파일로 업로드 시킨다. (사원 DB)</li> </ul> <p>&lt;ISBSG(International Software Benchmark Standard Group) 평균 복잡도&gt;</p> <table border="1"> <thead> <tr> <th>기능 유형</th> <th>평균 복잡도</th> </tr> </thead> <tbody> <tr> <td>EI</td> <td>4.3</td> </tr> <tr> <td>EO</td> <td>5.4</td> </tr> <tr> <td>EQ</td> <td>3.8</td> </tr> <tr> <td>ILF</td> <td>7.4</td> </tr> <tr> <td>EIF</td> <td>5.5</td> </tr> </tbody> </table>	기능 유형	평균 복잡도	EI	4.3	EO	5.4	EQ	3.8	ILF	7.4	EIF	5.5
기능 유형	평균 복잡도												
EI	4.3												
EO	5.4												
EQ	3.8												
ILF	7.4												
EIF	5.5												
도메인	소프트웨어 공학												
정의	사용자 관점에서의 요구 기능을 정량적으로 산정하여 소프트웨어 규모를 측정하고 이를 바탕으로 소프트웨어 개발과 유지보수를 위한 비용과 자원 소요를 산정하는 방법												
키워드	기능 복잡도, 평균 복잡도, 내부 논리 파일, 외부 연계 파일, 외부 입력, 외부 출력, 외부 조회, 트랜잭션 기능 점수												
출제의도분석	다양한 프로젝트에서 정당하고 공정한 대가 산정을 위해 FP의 적용이 요구됨												
답안작성 전략	기능 점수에 대하여 명확히 설명하고 주어진 조건에 따라 간이법으로 FP를 계산해 제시												
참고문헌	기능점수 산정 실전 가이드: SW 개발사업 중심 – 소프트웨어공학센터 SW 공학 연구팀 SW 사업 대가 산정 가이드(2014년 개정판) - KOSA												
모범목차	<ol style="list-style-type: none"> <li>1. 기능 점수의 개요</li> <li>가. 기능 점수의 개념</li> <li>나. 기능 점수의 특징</li> </ol>												

	2. 기능 점수 산출 방법 가. 기능 점수 측정 프로세스 나. 간이법과 정규법의 비교 3. 문제에서 제시된 이벤트 리스트에 의한 기능 점수 산정 가. 데이터 기능과 트랜잭션 기능의 구분 나. 기능 점수 합계(ISBSG 평균 복잡도의 적용)
풀이 기술사님	강용제 (yongjei.kang@gmail.com)

## ■ 기능 점수의 개념

- 논리적 설계에 기초하여 사용자에게 제공되는 소프트웨어의 기능 규모를 정량적으로 표현하는 단위
- 소프트웨어가 사용자에게 제공하는 기능의 수를 수치로 정량화하고 소프트웨어 규모를 산정



용어	개념
데이터 기능	<ul style="list-style-type: none"> <li>- 사용자가 요구한 정보를 의미</li> <li>- 사용자가 필요로 하는 데이터 단위를 하나의 기능으로 인지</li> </ul>
내부 논리 파일 (ILF)	<ul style="list-style-type: none"> <li>- Internal Logical File</li> <li>- 사용자 관점에서 식별 가능한 논리적으로 연관된 하나의 데이터 그룹</li> <li>- 제어 정보로서 측정 대상 어플리케이션 내부에서 유지되는 파일</li> <li>- 기능 점수 측정 대상 시스템 내부에서 관리되는 데이터</li> <li>- 시스템 내부에서 다루는 파일, 클래스, 외부 인터페이스를 위한 Structure 등</li> <li>- 외부 인터페이스에서 인터페이스의 주체가 내부 시스템인 경우 송수신 데이터는 ILF</li> </ul>
외부 연계 파일 (EIF)	<ul style="list-style-type: none"> <li>- External Interface File</li> <li>- 측정 대상 어플리케이션 내부에서는 오직 참조만 되고 다른 어플리케이션에서 유지되는 파일</li> <li>- 외부 연계 시스템에 존재하는 데이터</li> </ul>
트랜잭션 기능	<ul style="list-style-type: none"> <li>- 일반적으로 생각하는 업무 기능</li> <li>- 기능 점수에서 트랜잭션 기능의 복잡도는 단지 기능이 입력이냐, 조회냐에 의하여만 결정됨</li> <li>- 외부 입력, 외부 출력, 외부 조회 등 세가지 유형을 가지며 기능 점수 산출 시 각 기능이 이 세가지 유형을 몇 개나 가지고 있느냐에 따라 복잡도가 결정됨</li> </ul>
외부 입력 (EI)	<ul style="list-style-type: none"> <li>- External Input</li> <li>- 어플리케이션 경계 밖에서 들어오는 데이터나 제어 정보를 처리하는 단위 프로세스</li> </ul>

	<ul style="list-style-type: none"> <li>- 하나 이상의 논리 파일을 변경하거나 시스템 동작을 변경함</li> <li>- 특정 데이터를 특정 논리/연계 파일에 저장하는 기능</li> <li>- DB 관점에서는 Create, Update, Delete 가 해당</li> <li>- 외부 연계 입장에서 수신은 외부 시스템으로부터 데이터가 입력되므로 EIF 에 대한 EI 에 해당</li> </ul>
외부 출력 (EO)	<ul style="list-style-type: none"> <li>- External Output</li> <li>- 데이터나 제어 정보를 사용자에게 보여주는 프로세스</li> <li>- 처리 로직은 하나 이상의 수학적 계산, 공식 또는 파생 데이터를 포함하거나, 하나 이상의 내부 논리 파일(ILF)의 유지 관리 작업 발생하거나, 시스템의 동작 변경이 수반됨</li> <li>- 통계와 같이 데이터를 수학적인 계산에 의하여 표시하는 경우에 해당</li> </ul>
외부 조회 (EQ)	<ul style="list-style-type: none"> <li>- External Inquiry</li> <li>- 데이터나 제어 정보를 단순히 사용자에게 제공</li> <li>- 검색이나 정보 조회와 같이 단순히 데이터를 조회하여 그대로 표시하는 경우에 해당</li> <li>- 외부 연계의 경우 송신 프로세스는 외부 시스템에 정보를 출력하는 경우에 해당하므로 ILF 에 대한 EQ 에 해당</li> </ul>

- 시스템 입장에서 외부에서 입력이 일어나거나 시스템의 정보가 시스템의 외부로 출력되어야만 기능으로 봄. 즉, 시스템이 외부와 통신할 때만 기능 유형으로 기능 점수의 대상이 됨

- ISBSG 표준 복잡도: IFPUG(International Function Point Users Group) 산하의 ISBSG(International Software Benchmarking Standards Group)에서 유지 관리하는 복잡도 정보

### ■ 기능 점수 측정 프로세스



항목	내용
기능 점수 측정	<ul style="list-style-type: none"> <li>- 기능 점수 측정에 필요한 문서란 소프트웨어에 의해 인도되는 기능량을 기술하거나 측정되는 소프트웨어 프로젝트에 의해 영향을 받는 기능량을 기술한 문서</li> <li>- 요구 사항, 데이터/객체 모델, 클래스 다이어그램, 데이터 흐름도, 유스케이스, 절차 기술서, 보고서 및 화면 레이아웃, 사용자 매뉴얼, 기타 소프트웨어 개발 산출물 등</li> </ul>
측정 목적과 측정 유형에 따른 결정	<ul style="list-style-type: none"> <li>- 측정 목적: 기능 점수 측정이란 비즈니스 문제에 답을 하기 위해 수행하는 것으로 목적을 결정하는 것은 비즈니스 문제임</li> <li>- 측정 유형: 개발 프로젝트의 기능 점수, 어플리케이션의 기능 점수, 개선 프로젝트의 기능 점수</li> <li>- 측정 범위: 기능 점수 측정에 포함되는 사용자의 기능적 요구 사항의 집합</li> </ul>
측정 범위 결정	<ul style="list-style-type: none"> <li>- 측정 범위: 기능 점수 측정에 포함되는 사용자의 기능적 요구 사항의 집합</li> </ul>

경계의 정의와 기능적 사용자 요구 사항의 식별	<ul style="list-style-type: none"> <li>- 경계: 소프트웨어와 사용자 간의 개념적인 인터페이스로 사용자의 외부 비즈니스 관점에 의존적이고 기술 또는 구현상의 고려 사항과는 독립적임</li> <li>- 대부분의 국내 환경에서는 개발되는 시스템을 경계로 함</li> <li>- 사용자 요구 사항: 기능 요구 사항과 비기능 요구 사항을 식별하고 기능 점수 측정에는 기능 요구 사항을 대상으로 함</li> </ul>
데이터 기능 측정	<ul style="list-style-type: none"> <li>- 데이터를 저장하고 참조하는 기능적 사용자 요구 사항으로 논리적으로 연관되고 사용자가 식별 가능한 모든 데이터나 제어 정보</li> </ul>
트랜잭션 기능 측정	<ul style="list-style-type: none"> <li>- 데이터를 처리하는 기능적 사용자 요구 사항으로 사용자에게 의미 있고 비즈니스를 일관된 상태로 유지하는 처리 로직을 의미</li> </ul>

### ■ 기능 점수 산정 정규법과 간이법의 비교

구분	정규법	간이법
개념	<ul style="list-style-type: none"> <li>- 소프트웨어의 기능을 도출하고 각 기능의 유형별 복잡도를 고려하여 정확한 기능 점수 산정을 필요로 할 경우 사용되는 방법</li> </ul>	<ul style="list-style-type: none"> <li>- 기능의 복잡도를 판단하기 어려운 경우 적용하는 방법</li> <li>- 계산 방법은 정규법과 동일하나 기능 점수 산정 시 기능 유형별 평균 복잡도를 적용하여 기능 점수 산출</li> </ul>
장점/단점	<ul style="list-style-type: none"> <li>- 규모 측정 정확도가 간이법 대비 상대적으로 높음</li> <li>- 기능 점수 측정 소요 시간이나 측정 지식 습득 시간이 간이법 대비 상대적으로 깊음</li> </ul>	<ul style="list-style-type: none"> <li>- 기능 점수 측정 소요 시간이 정규법 대비 상대적으로 짧음</li> <li>- 기능 점수 측정 지식 습득 시간이 정규법 대비 상대적으로 짧음</li> <li>- 측정 결과의 정확도가 떨어짐</li> </ul>
목적	<ul style="list-style-type: none"> <li>- SW 분석/설계 시 개발 범위/일정/원가 산정</li> <li>- SW 개발 완료 시 Historical DB 구축</li> </ul>	<ul style="list-style-type: none"> <li>- 예산 수립</li> <li>- 제안 견적 산정</li> <li>- 계약 시 SW 사업 대가 산정</li> </ul>
적용 시점	<ul style="list-style-type: none"> <li>- 통상적으로 SW 개발 공정 상 설계 공정 후 사용</li> </ul>	<ul style="list-style-type: none"> <li>- 통상적으로 기획 및 발주 단계에서의 기능 점수 측정에 사용</li> </ul>
측정 항목	<ul style="list-style-type: none"> <li>- 데이터 기능 및 DET, RET</li> <li>- 트랜잭션 기능 및 DET, FTR</li> </ul>	<ul style="list-style-type: none"> <li>- 데이터 기능</li> <li>- 트랜잭션 기능</li> </ul>
사용 복잡도	<ul style="list-style-type: none"> <li>- 기능별 복잡도 매트릭(Low, Avg, High)</li> </ul>	- 평균 복잡도

DET: Data Element Type, 데이터 요소 유형

RET: Record Element Type, 레코드 요소 유형

FTR: File Type Referenced, 참조 파일 유형

■ 문제에서 제시된 이벤트 리스트에 의한 기능 점수 산정(간이법)

- 이벤트 리스트

순번	이벤트
가	담당자는 고객 주문을 입력, 수정 삭제한다. (고객 DB)
나	인사 담당자는 사원 목록을 부서 단위로 조회한다. (사원 DB, 부서 DB)
다	인사 담당자는 사원 목록을 단순 출력한다. (사원 DB)
라	인사 담당자는 일정 금액 이상의 급여 수령자 사원 목록을 검색한다. (사원 DB, 급여 DB)
마	원화에 대한 미국 달러(USD) 가치를 찾기 위해 A 은행 외환 DB 에서 환율을 검색한다. (외환 DB)
바	인사 담당자는 5 년 경력 이상이고, 해당 직무 수행 경험이 있는 사원 목록을 추출한다. (사원 DB)
사	인사 담당자는 신입/경력 사원 입사 시 사원 파일을 갱신한다. (사원 DB)
아	인사 담당자는 외국 사원 입사 시 사원의 급여를 결정하기 위해 H 연합회 통화 정보를 참조한다. (사원 DB, 급여 DB, 통화 DB)
자	회계 시스템은 전표 번호(부서 번호 중 앞자리 2 + 년도 + 일련 번호)를 자동 채번한다. (전표 DB)
차	인사 담당자는 사원 현황을 엑셀 파일로 업로드 시킨다. (사원 DB)

- 기능 점수 산정: 이벤트 리스트에서 EI, EO, EQ, ILF, EIF 의 수를 추출한 후 ISBSG 평균 복잡도를 적용

순번	EI (4.3)			EO (5.4)	EQ (3.8)	ILF (7.4)	EIF (5.5)	FP
	입력	수정	삭제					
가	1	1	1			3		$4.3 * 3 + 7.4 * 3 = 35.1$
나				1		2		$5.4 + 7.4 * 2 = 20.2$
다					1	1		$3.8 + 7.4 = 11.2$
라				1		2		$5.4 + 7.4 * 2 = 20.2$
마				1			1	$5.4 + 5.5 = 10.9$
바				1		1		$5.4 + 7.4 = 12.8$
사		1				1		$4.3 + 7.4 = 11.7$
아				1		2	1	$5.4 + 7.4 * 2 + 5.5 = 25.7$
자	1					1		$4.3 + 7.4 = 11.7$
차	1					1	1	$4.3 + 7.4 + 5.5 = 17.2$
합계								<b>176.7</b>

가: 입력/수정/삭제를 각각 산정

자: 자동 채번은 입력 시 발생하는 것으로 가정. 만약 사용자에게 보이지 않는 것이라면 삭제

차: 엑셀 파일을 시스템에서 제외하는 시각도 있을 수 있음

전체: 시스템 전체를 하나로 보고 ILF를 중복 처리하면 ILF의 수가 줄어들 수 있음

"끝"

1	데이터베이스 정규화
문제	<p>아래의 스키마(Schema)와 함수 종속성(FD: Functional Dependency)을 이용하여 함수 종속 도표(Functional Dependency Diagram)를 작성한 뒤, 키(Key)를 찾아내는 과정을 설명하고, BCNF(Boyce-Codd Normal Form)의 정의를 기술하고 조건을 만족시키는 테이블을 설계하시오.</p> <div style="border: 1px solid black; padding: 5px;"> <p>대출 스키마(지점명, 자산, 장소, 대출번호, 고객명, 금액)</p> <p>FD: 지점명 <math>\rightarrow</math> 자산      지점명 <math>\rightarrow</math> 장소      대출번호 <math>\rightarrow</math> 지점명      대출번호 <math>\rightarrow</math> 금액</p> </div>
도메인	데이터베이스
정의	정보 시스템에서 데이터의 품질(일관성, 정확성) 확보, 중복성 제거 등 이상 현상을 방지하기 위해 무손실 분해하는 과정
키워드	함수 종속성
출제의도분석	공공 데이터 등에 있어 요구되는 품질을 보장하기 위한 기반 조건으로 데이터베이스의 정규화 이해 수준 검증
답안작성 전략	<p>함수 종속성에 대한 명확한 이해를 기반으로 개념을 설명하고 키의 선정, BCNF 정의.</p> <p>주어진 문제의 테이블 설계를 명확히 제시</p>
참고문헌	
모범목차	<ol style="list-style-type: none"> <li>1. 데이터베이스 정규화의 개요</li> <li>    가. 데이터베이스 정규화의 정의</li> <li>    나. 함수 종속성의 개념</li> <li>2. 데이터베이스 키 선정 과정과 정규화의 개념</li> <li>    가. 데이터베이스 설계에서 키 선정 과정</li> <li>    나. 데이터베이스 정규화의 단계</li> <li>3. BCNF의 개요 및 주어진 조건에의 적용</li> <li>    가. BCNF의 개요</li> <li>    나. 주어진 조건을 만족하는 테이블의 설계</li> </ol>
풀이 기술사님	강용제 (yongjeikang@gmail.com)

### ■ 데이터베이스 정규화의 정의와 원칙

- 관계형 데이터 모델에서 데이터의 중복성을 제거하여 이상 현상 발생을 방지하고 데이터의 일관성과 정확성을 유지하기 위해 무손실 분해를 적용하는 과정

구분	설명
정보의 무손실	분해된 Relation 이 표현하는 정보는 분해되기 전의 정보를 모두 포함하고 있어야 함
데이터 중복성의 감소	중복으로 인한 이상 현상의 제거
관계성 분리	하나의 독립된 관계성은 하나의 독립된 Relation 으로 분리하여 표현

### ■ 함수 종속성의 개념과 유형

- 데이터들이 어떤 기준 값에 의해 종속되는 현상
- 결정자(Determinant): 기준 값

- 종속자/의존자(Dependent): 종속되는 값
- 어떤 릴레이션 R에서 X와 Y가 각각 R의 속성 집합의 부분 집합인 경우 속성 X의 값 각각에 대해 시간에 관계 없이 항상 속성 Y의 값이 오직 하나만 연관되어 있을 때 Y는 X에 함수 종속이라 하고  $X \rightarrow Y$ 라고 표기

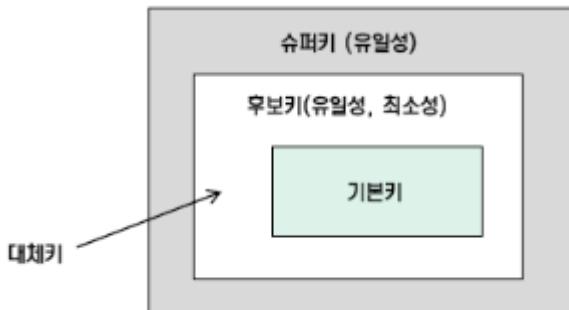
유형	내용
완전 함수 종속성	<p>- 릴레이션 R의 어떤 속성 Y가 다른 복합 속성 X에 함수 종속이면서 X의 어떤 진부분집합에도 함수 종속이 아닐 때 Y는 X에 완전 함수 종속  - {학번, 과목번호} <math>\rightarrow</math> 성적  (진부분집합: 집합 A가 집합 B의 부분 집합이지만 집합 A와 집합 B가 같지 않은 경우)</p>
부분 함수 종속성	<p>- 학년은 {학번, 과목번호}에 부분 함수 종속  - 학번 <math>\rightarrow</math> 학년  - 어떤 속성 Y가 다른 복합 속성 X에 함수 종속이면서 속성 X의 진부분집합에도 함수 종속인 경우</p>
이행 함수 종속성	<p>- 릴레이션 R에서 속성 A가 <math>A \rightarrow X</math>이고 <math>X \rightarrow Y</math>이면 <math>A \rightarrow Y</math>  - 학번 <math>\rightarrow</math> 지도교수, 지도교수 <math>\rightarrow</math> 학과. 학번 <math>\rightarrow</math> 학과</p>
결정자 함수 종속성	<p>- 함수적 종속이 되는 결정자가 후보기가 아닌 경우  - <math>X \rightarrow Y</math>에서 X가 후보기가 아님  - 교수 <math>\rightarrow</math> 과목</p>
다중 값 종속성	<p>- X, Y, Z 3개의 속성을 가진 릴레이션 R에서 속성 쌍 [X, Y]에 대응하는 Y 값의 집합이 X 값에만 종속되고 Z 값에는 독립이면 Y는 X에 다중 값 종속  - <math>X -&gt;&gt; Y</math>로 표기</p>
조인 종속성	<p>- 관계 중에서 둘로 나눌 때는 원래의 관계로 회복할 수 없으나 셋 또는 그 이상으로 분리시킬 때 원래의 관계를 복원할 수 있는 특수한 경우</p>

### ■ 데이터베이스에서 키의 개념

- 한 릴레이션 내의 튜플을 유일하게 식별할 수 있는 속성의 집합
- 유일성, 최소성, 대표성의 특성을 가져야 함

특성	내용
유일성	- Key 값으로 릴레이션 내의 튜플들을 구분 가능(Unique/Not null)
최소성	- 유일성을 지니는 최소한의 속성만을 포함
대표성	- 해당 릴레이션을 대표할 수 있는 속성

### ■ 데이터베이스에서 키의 종류



특성	내용
기본키	<ul style="list-style-type: none"> <li>- Primary Key</li> <li>- 여러 개의 후보키 중에서 하나를 선정하여 테이블을 대표하는 키</li> </ul>
후보키	<ul style="list-style-type: none"> <li>- Candidate Key</li> <li>- 키의 특성인 유일성과 최소성을 만족하는 키</li> <li>- 슈퍼키에서 튜플을 유일하게 식별하는데 필요 없는 속성을 제거한 슈퍼키의 부분 집합</li> </ul>
슈퍼키	<ul style="list-style-type: none"> <li>- Super Key</li> <li>- 유일성은 만족하나 최소성을 만족하지 않는 키</li> <li>- 한 개의 테이블은 여러 개의 슈퍼 키를 가질 수 있음</li> </ul>
대체키	<ul style="list-style-type: none"> <li>- Alternate Key</li> <li>- 여러 개의 후보키 중에서 기본키로 선정되고 남은 나머지 키</li> <li>- 기본키를 대체할 수 있는 키</li> </ul>
외래키	<ul style="list-style-type: none"> <li>- Foreign Key</li> <li>- 어느 한 릴레이션 속성의 집합이 다른 릴레이션에서 기본키로 이용되는 키</li> </ul>

### ■ 데이터베이스에서 키 선정 과정

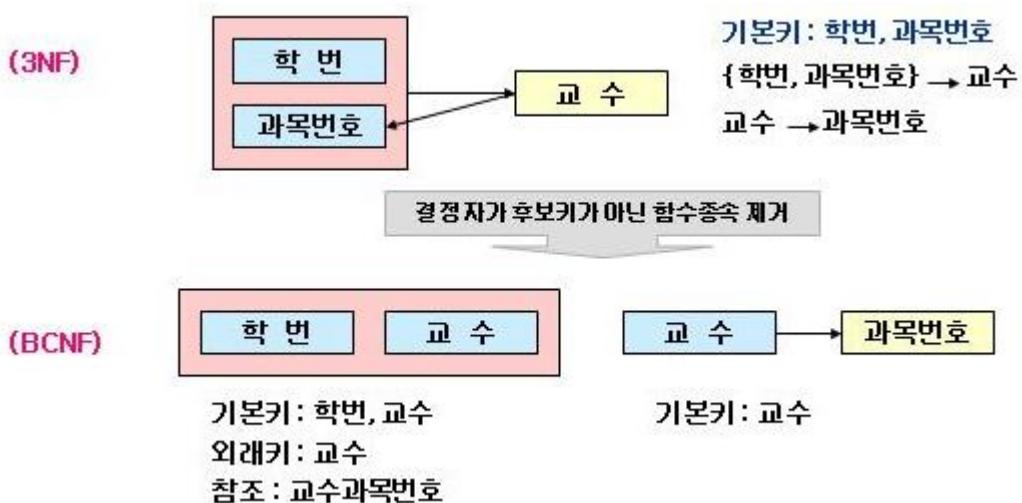
단계	내용
결정자 도출	- 릴레이션 내의 결정자들을 식별
슈퍼키 선정	- 릴레이션 내의 각 튜플을 구분할 수 있는 속성 또는 속성 집합을 식별
후보키 선정	- 식별된 슈퍼키들 중 최소성을 만족하는 키들을 식별
기본키 선정	- 식별된 후보키들 중 릴레이션의 성격을 대표할 수 있는 키를 기본키로 선정

### ■ 데이터베이스 정규화의 단계



### ■ BCNF(Boyce-Codd Normal Form)의 개요

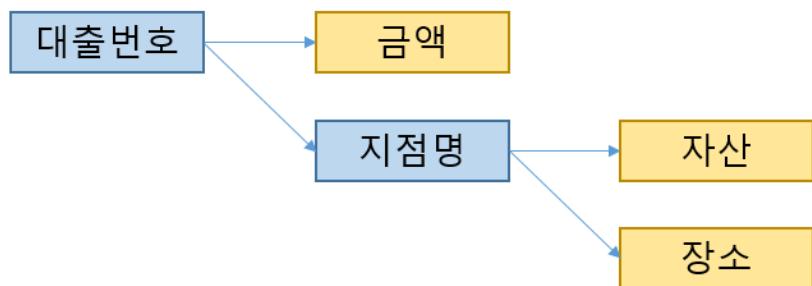
- 릴레이션 R의 모든 결정자가 후보키이면 릴레이션 R은 BCNF 임
- 어떤 릴레이션 R이 결정자 함수 종속성을 갖는 경우 이를 제거하면 BCNF 임
- 제 3 정규형을 확장한 것으로 이 정규형에 가까워질수록 데이터의 중복성이 배제됨
- 따라서 갱신 시에 부정합 또는 불일치가 잘 생기지 않음



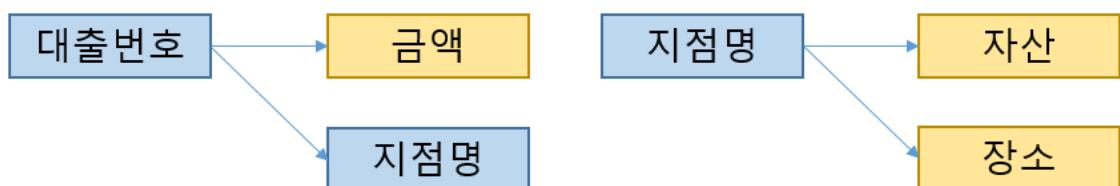
Notes

### ■ 주어진 문제 풀이

- 함수 종속 도표



- BCNF

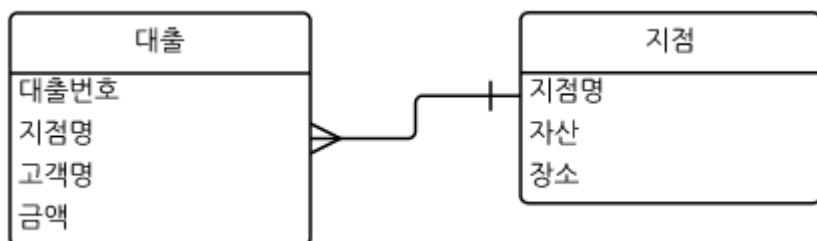


기본키: 대출번호

외래키: 지점명

참조: 자산, 장소

- 결과 테이블: 3 차 정규화 후 BCNF 적용



"끝"

2	SAD(Software Architecture Document)
문제	귀하는 차세대시스템 구축팀의 아키텍트(Architect)로서 상세화단계(Elaboration Phase)에서 작성해야 하는 SAD(Software Architecture Document) 문서의 목차를 작성하시오.
도메인	소프트웨어공학
정의	SAD: Software Architecture 를 이해당사자 별 다양한 view 로 명세한 문서
키워드	IEEE 1471, 아키텍처 뷰(SEI 뷰, UML 뷰 등)
출제의도분석	Platform 중심 개발의 확산으로 소프트웨어 아키텍처의 중요성이 높아지고 있음
답안작성 전략	SAD 목차를 잘 작성하는 것은 물론, SAD 의 작성절차, 핵심구성요소 등을 설명
참고문헌	소프트웨어 아키텍처 설계가이드(강승준, 프리렉)
모범목차	<ol style="list-style-type: none"> <li>1. SAD 의 개요           <ol style="list-style-type: none"> <li>가. SAD(Software Architecture Document)의 정의</li> <li>나. IEEE 1471 에 따른 SAD 작성 가이드</li> </ol> </li> <li>2. SAD 작성방법과 핵심요소           <ol style="list-style-type: none"> <li>가. SAD 의 작성방법</li> <li>나. SAD 의 핵심요소 뷰</li> </ol> </li> <li>3. SAD 의 목차 작성           <ol style="list-style-type: none"> <li>가. SAD 목차의 주요항목</li> <li>나. SAD 목차의 작성</li> </ol> </li> </ol>
풀이 기술사님	이홍철 (bluerobe@naver.com)

## 1. SAD 의 개요

가. SAD(Software Architecture Document)의 정의

- 이해관계자들의 관심사를 파악해 관점을 정의하고, 관점에 따라 다양한 뷰를 통해 소프트웨어 아키텍처를 기술한 문서

- IEEE 1471 은 소프트웨어 아키텍처가 표현해야 하는 내용 및 이들간의 관계를 정의한 표준

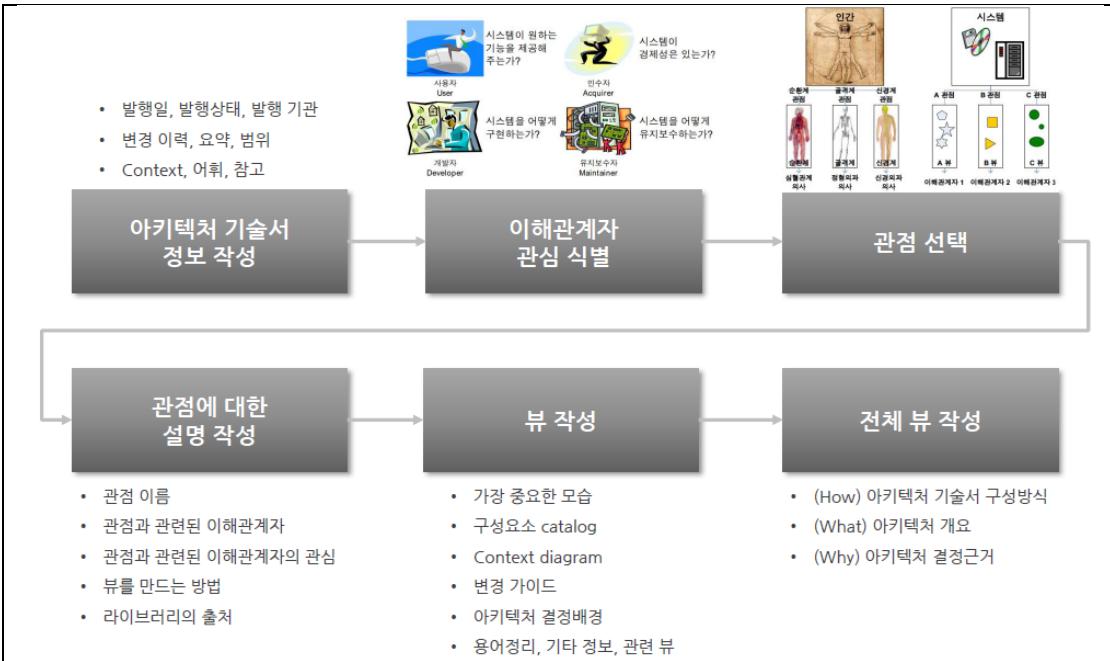
나. IEEE 1471 에 따른 SAD 작성 가이드

권고사항	가이드 내용
권고활동	<ul style="list-style-type: none"> <li>- 이해관계자, 그들의 역할 및 아키텍처상의 관심사항의 파악</li> <li>- 이해관계자의 관심에 따른 뷰 포인트의 선택 및 명세</li> <li>- 선택되어 설계된 아키텍처에 대한 논리적 근거 작성</li> </ul>
명세내용	<ul style="list-style-type: none"> <li>- 뷰 포인트에 의해 다루어지는 이해관계자</li> <li>- 뷰 포인트에 의해 다루어지는 관심사항</li> <li>- 뷰 포인트에 근거한 뷰를 만들기 위한 방법론</li> </ul>

- SAD 의 핵심구성요소인 아키텍처 뷰에 대한 작성방법을 권고활동과 명세내용으로 가이드 함

## 2. SAD의 개요

### 가. SAD의 작성방법



### 나. SAD의 핵심요소 뷰

- SAD의 뷰: 다양한 이해관계자가 시스템을 바라보는 관점에 따라 시스템을 묘사한 결과물로 다양한 Diagram의 형태로 주로 표현됨.

SEI view	UML 4+1 view	Siemens's 4view
- Module View	- Usecase View	- Module View
- Component & Connector View	- Logical View	- Conceptual View
- Allocation View	- Development View - Process View - Physical View	- Execution View - Code View

- 뷰는 시스템의 모든 구성요소가 아니라 일부를 표현한 것으로, 아키텍처는 뷰모델을 사용해 뷰를 표현함

## 3. SAD의 목차 작성

### 가. SAD 목차의 주요 항목

항목	서술내용
Introduction	시스템의 목적, 개발범위에 대한 설명과 함께 아키텍처 명세에 대한 참고자료와 아키텍처의 Overview를 제시
Goals and Constraints	시스템 아키텍처가 지향하는 목표와 시스템이 가지는 제약사항을 다양한 품질속성을 통해서 설명
Architecture Views	뷰 모델을 이용해 다양한 이해관계자의 관점에 따라서 다양한 Diagram의 형태로 시스템의 뷰를 표현

Notes

## 나. SA 목차의 작성

1. INTRODUCTION.....	3.
1.1. PURPOSE.....	3.
1.2. SCOPE.....	4.
1.3. DEFINITIONS, ACRONYMS AND ABBREVIATIONS.....	4.
1.4. REFERENCES.....	4.
1.5. OVERVIEW.....	5.
2. ARCHITECTURAL REPRESENTATION.....	5.
3. ARCHITECTURAL GOALS AND CONSTRAINTS.....	6.
3.1. TECHNICAL PLATFORM .....	6.
3.2. TRANSACTION .....	7.
3.3. SECURITY.....	7.
3.4. PERSISTENCE.....	7.
3.5. RELIABILITY/AVAILABILITY (FAILOVER).....	7.
3.6. PERFORMANCE.....	8.
3.7. INTERNATIONALIZATION (I18N).....	8.
4. USE-CASE VIEW.....	8.
4.1. ORDERING MENUS .....	8.
4.2. USE-CASE REALIZATIONS.....	9.
5. LOGICAL VIEW.....	9.
5.1. OVERVIEW.....	9.
5.2. ARCHITECTURALLY SIGNIFICANT DESIGN PACKAGES.....	11.
6. PROCESS VIEW.....	13.
7. DEPLOYMENT VIEW.....	13.
8. IMPLEMENTATION VIEW.....	15.
8.1. OVERVIEW.....	15.
8.2. LAYERS.....	15.
9. DATA VIEW.....	15.
10. SIZE AND PERFORMANCE.....	16.
11. QUALITY.....	16.

"끝"



Notes

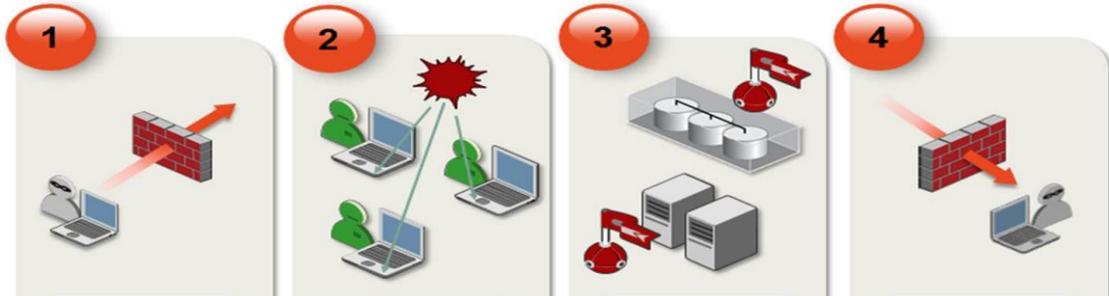
4	APT 공격, 악성 코드
문제	최근 APT(Advanced Persistent Threats) 공격과 변종 악성 코드 공격이 늘어나고 있다. APT 공격 기법과 악성 코드(Malicious codes)에 대하여 설명하시오.
도메인	디지털 보안
정의	APT: 다양한 IT 기술과 방식들을 이용해 조직적으로 특정 목적을 위해 다양한 보안 침해 방법들을 생산해 지속적으로 특정 대상에게 가하는 일련의 보안 위협 악성 코드: 사용자 정보 탈취, 분산 서비스 거부 공격 등의 악의적인 목적을 실행하는 모든 프로그램 또는 파일(자기 복제, 정상 파일에의 삽입 등 가능)
키워드	우회 공격, 사회 공학, 제로 데이 취약점, 다중 벡터, 스톡스넷
출제의도분석	대규모 정보 유출 사고의 원인이 악성 코드를 이용한 APT 공격인 경우가 많은데 이에 대한 깊이 있는 이해 필요
답안작성 전략	APT 의 명확한 개념, 공격 단계, 공격 기법 등을 설명하고 APT 에서 악성 코드의 이용에 대해 설명
참고문헌	APT 공격 방어 전략 - <a href="http://forensic-proof.com/archives/5341">http://forensic-proof.com/archives/5341</a> 악성 코드 포렌식 - <a href="http://forensic-proof.com/archives/3806">http://forensic-proof.com/archives/3806</a> 알려지지 않은 악성 코드 탐지 기법 - 안랩 전문가 칼럼
모범목차	<ol style="list-style-type: none"> <li>1. APT 공격의 개요             <ol style="list-style-type: none"> <li>가. APT 공격의 개념</li> <li>나. APT 공격의 프로세스</li> </ol> </li> <li>2. APT 공격 기법</li> <li>3. APT 공격의 주요 도구, 악성 코드의 개요             <ol style="list-style-type: none"> <li>가. 악성 코드의 개념</li> <li>나. 악성 코드의 유형</li> </ol> </li> <li>4. APT 공격과 악성 코드에 대한 대응 체계             <ol style="list-style-type: none"> <li>가. APT 공격에 대한 대응 체계</li> <li>나. 악성 코드에 대한 대응 체계</li> </ol> </li> </ol>
풀이 기술사님	강용제 (yongjei.kang@gmail.com)

### ■ APT(Advanced Persistent Threats) 공격의 개념

- 다양한 IT 기술과 방식들을 이용해 조직적으로 특정 목적을 위해 다양한 보안 침해 방법들을 생산해 지속적으로 특정 대상에게 가하는 일련의 보안 위협
- 구체적인 특정 데이터 혹은 특정 피해를 목적으로 하며 목적이 구체적이다 보니 보다 많은 시간이 소요되고 목적을 이루는 동안 발각되지 않기 위해 보다 고급 기법이 사용됨
- 어떤 특정한 침해 기법, 기술을 말하는 것이 아님 → 완벽한 차단이 불가능함

### ■ APT 공격의 프로세스

## APT 공격 프로세스



침투	검색	수집	유출
- 공격자가 취약한 시스템이나 직원들을 악성 코드로 감염시켜 네트워크 내부로 침투	- 침투한 내부 시스템 및 인프라 구조에 대한 정보를 수집한 후 다음 단계를 계획	- 보호되지 않은 시스템 상의 데이터 수집 또는 시스템 운영 방해	- 공격자의 근거 자료 데이터 전송 - 시스템 운영 방해 또는 장비 파괴

### ■ APT 공격에 사용되는 주요 공격 기법

단계	공격 기법	내용
침투 (Incursion)	관찰	- APT 공격자들은 표적 대상을 파악하기 위해 수 개월에 걸쳐 철저히 분석
	사회 공학	- 내부 임직원의 실수 또는 부주의로 첨부 파일을 열도록 유도하거나 링크를 클릭하도록 하는 사회 공학적 기법을 사용
	제로 데이 취약점	- 개발자들이 패치를 제공하기 전 침투할 수 있는 보안상의 허점
	수동 공격	- 자동화 대신 각각의 개별 시스템과 사람을 표적으로 삼고 고도의 정교한 공격을 감행
검색 (Discovery)	다중 벡터	- APT 공격 시 일단 악성 코드가 호스트 시스템 내에 구축이 되면 소프트웨어, 하드웨어 및 네트워크의 취약점을 탐색하기 위해 추가적인 공격 툴들을 다운로드
	은밀한 활동	- APT의 목표는 표적의 내부에 잠복하면서 장시간 정보를 확보하는 것으로 모든 잠복 프로세스는 보안 탐지를 회피하도록 설계됨
	연구 및 분석	- 정보 검색은 네트워크 구성, 사용자 아이디 및 비밀 번호 등을 포함하여 확보된 시스템과 데이터에 대한 연구 및 분석을 수반함
수집 (Capture)	은닉	- 1 차 공격 성공 후 정상 이용자로 가장하여 정보 수집 및 모니터링 활동을 진행
	권한 상승	- 시스템 접근을 위한 시스템 접근 권한 보유 직원에 대한 계정 정보를 수집하기 위한 각종 접근 행위
제어 (Control)	유출	- 기밀 데이터가 웹 메일 또는 암호화된 패킷, 압축 파일의 형태로 공격자에게 전송
	중단	- APT 공격자는 원격 시동이나 SW, HW 시스템의 종료, 파괴 등이 가능함

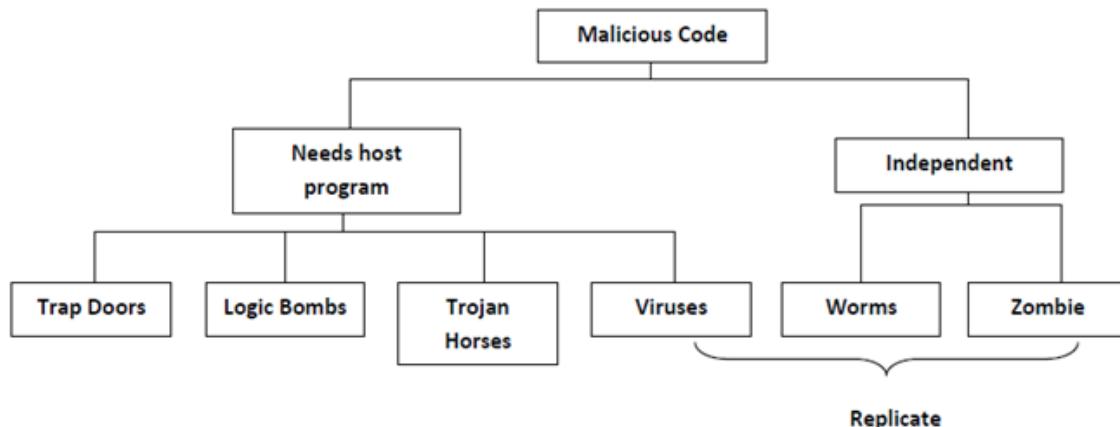
### ■ APT 공격에 사용되는 주요 공격 기법

기능	내용	특징
시스템운영	서버, 스토리지, 네트워크, 보안장비 등 정보자원을 통합운영/관리	- ITIL 기반의 표준화/자동화된 시스템 통합운영 툴 개발/활용 - ISO20000 인증 획득
정보보호	외부의 물리적/사이버적 위협으로부터 정보시스템을 보호	ISMS 인증 획득
IT 자원관리	필요한 HW, 시스템 SW 등 IT 자원의 효과적 활용	클라우드 컴퓨팅 서비스 환경으로 전환
통합전산망	전용통신망 구성을 통한 각 지점 서비스 제공	SLA 에 기반한 서비스품질(QoS) 보장
IT 기반환경제공	시스템의 안정적인 가동을 보장하는 최적의 인프라 제공	지능형빌딩시스템, 출입보안시스템, 전력관리 시스템 등
업무연속성관리	재난 상황에 대비한 대비 체계 구축	BS25999 인증 획득
공통플랫폼관리	각 지점에서 공통으로 활용할 수 있는 모듈화된 공유 플랫폼 제공	검증된 표준환경 제공으로 효과적 서비스 개발을 지원
모바일지원	모든 지점에 일관된 모바일 업무 서비스 환경 제공	Mobile Service Life Cycle 관리

### ■ 악성 코드(Malicious Code)의 개념

- 사용자 정보 탈취, 분산 서비스 거부 공격 등의 악의적인 목적을 실행하는 모든 프로그램 또는 파일(자기 복제, 정상 파일에의 삽입 등 가능)

### ■ 악성 코드의 분류



분류	유형	내용
호스트 프로그램 의존형 (Needs host program)	트랩 도어 (백 도어)	- 관리자나 개발자가 시스템 접근을 용이하게 하기 위해 숨겨놓은 코드 - 공격자가 복잡한 공격 없이 신속하고 용이하게 시스템 접근 가능
	논리 폭탄	- 정상적인 프로그램의 일부를 조작하여 특정 조건을 충족하는 경우 실행 - 자료나 소프트웨어 파괴를 목적으로 실행
	트로이 목마	- 정상적인 소프트웨어 내부에 악의적인 행위를 하는 코드 삽입 - 정상 사용자 권한을 이용하여 파일 권한 변경, 파일 삭제, 계정 생성 등의 행위를 사용자가 인식하지 못하게 수행
	바이러스	- 감염시키고자 하는 파일 검색, 자신의 코드를 삽입하는 악성 코드
	웜	- 네트워크 연결을 이용하여 자신의 실행 코드를 복제하는 악성 코드 - 바이러스와 차이점은 다른 프로그램에 자신을 삽입하지 않음
독립형 (Independent)	줄비	- 인터넷에 연결된 컴퓨터에 대한 제어권을 획득하기 위해 은닉 - 웹 사이트 분산 서비스 거부 공격에 사용됨

### ■ APT 공격에 대한 대응

- 대응 시기에 따른 대응 방안

시기	활동	내용
위험 발생 이전 (시큐리티 대응 센터)	위험 예방 전략 수립	- 보안 관제 수행 - 위험 분석 수행 - 보안 전략 유효성 분석
	악성 코드 유입 최소화	- 보안 인식 교육 수행 - 지속적 보안 업데이트 관리 - 보안 소프트웨어 설치 및 운영
위험 발생 이후 (침해 사고 대응 센터)	악성 코드 감염 예방	- 어플리케이션 화이트 리스트 - 접근 권한 최소화 - 네트워크 접근 제한 및 분리 - 신원 확인 및 접근 권한 관리
	데이터 유출 예방	- 중요 데이터 보호 - 중요 데이터 유출 예방
	위험 탐지와 대응	- 호스트 및 네트워크 이상 징후 탐지 - 침해 사고 대응 프로세스 수행 - 침해 사고 포렌식 프로세스 수행

\* 다수의 항목에서 악성 코드에 대한 활동을 볼 수 있음

\* 악성 코드는 APT 공격의 다양한 단계에서 적극적으로 사용되는 침입 수단임

## - 대응 기법

대응 기법	보안 강화 내용	대상 조직
보안 관리 및 운영	- 조직의 종합적인 보안 위험 분석 후 보안 체계 재정비	보안 정책 관리 조직
보안 교육 강화	- 조직 구성원들의 적극적인 참여 유도 - 효과적인 보안 교육 실시	임직원 대상
엔드포인트 보안	- APT 공격의 1 차 대상인 엔드포인트 보안 강화 - 인터넷, 이메일, 메신저, P2P 통신 서비스 등 제한	엔드포인트 단말
접근 권한 관리	- 조직의 중요 정보 보호를 위한 접근 권한 관리, 권한 관리자 최소화, 접근 권한 세분화, 사람/기기 인증 추가	권한 관리 인가자
중요 정보 암호화 및 DLP 운영	- 조직 내 중요 정보는 암호화 저장, 정보 유출 방지를 위한 DLP(Data Loss Prevention) 솔루션 이미지	암호화 솔루션

\* 외부에서 들어오는 공격과 내부에서 유출되는 것을 동시에 감시 및 대응하는 것이 필요

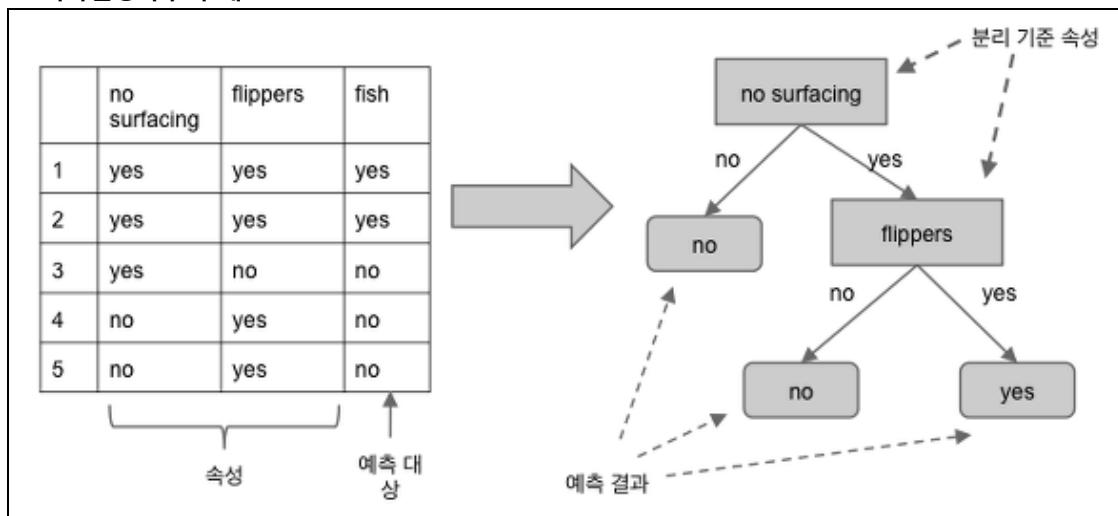
## ■ 악성 코드에 대한 대응

탐지 기법	내용
시그니처 기반 진단	- Entry point/프로그램의 전체 크기 정보 기반 진단 위치 확인 - 특정 영역 또는 특정 코드에서 시그니처 비교
전용 진단 함수	- 선행 조건 및 사용된 명령어의 패턴 기반 진단
행위 기반	- 악성 코드의 특성을 보이는 행위를 판단하여 진단 - 파일 이름, 인젝션, 파일 Drop, 통신 포트 open 의 특징 이용
Emulation	- 다형성 기법을 사용하는 악성 코드 탐지를 위한 실행 상태에서 탐지 - Emulation 시작 위치 확인, 진단 속도 및 예외 처리, Anti-Emulation 기법 대비 필요 - 시스템에 가상의 환경, 가상의 하드웨어 이미지 등을 생성해 가상 환경에서 간접적으로 악성 코드를 실행해 탐지

"끝"

4	의사결정나무(Decision Tree)
문제	데이터 마이닝에서 데이터를 분류(Classification)하는 의사결정나무(Decision Tree)를 설명하고 의사결정나무의 형성과정과 장단점에 대하여 설명하시오.
도메인	디지털서비스
정의	의사결정규칙을 나무(Tree)구조로 표현하여 분류, 예측에 활용하는 방법
키워드	성장, 가지치기, 타당성 평가, 해석 및 예측, 분할속성
출제의도분석	데이터 마이닝 결과에 영향을 미치는 의사결정나무의 형성과정에 대한 이해
답안작성 전략	주요질문인 형성과정에 대해서 사례를 통한 상세한 설명
참고문헌	데이터마이닝(이안위튼 등, 에이콘출판사)
모범목차	<ol style="list-style-type: none"><li>1. 의사결정나무의 개요</li><li>2. 의사결정나무의 형성과정<ol style="list-style-type: none"><li>가. 의사결정나무의 형성단계</li><li>나. 의사결정나무의 형성사례</li></ol></li><li>3. 의사결정나무의 장단점<ol style="list-style-type: none"><li>가. 의사결정나무의 장점</li><li>나. 의사결정나무의 단점</li></ol></li><li>4. 의사결정나무의 활용분야</li></ol>
풀이 기술사님	이홍철 (bluerobe@naver.com)

## 1. 의사결정나무의 개요



- 관찰된 데이터로부터 분할기준 속성을 판별하고, 분할기준 속성에 따라 트리 형태로 모델링한 분류, 예측 모델
  - root node, child node, parent node, terminal node, internal node, branch, depth 등의 요소를 사용해 의사결정나무로 표현함

## 2. 의사결정나무의 형성과정

### 가. 의사결정나무의 형성단계

단계	수행 내용
1) 의사결정나무 형성	분석의 목적과 자료구조에 따라서 적절한 분리기준(split criterion)과 정지규칙(stopping rule)을 지정하여 의사결정 나무를 획득
2) 가지치기	분류오류(classification error)를 크게 할 위험(risk)이 높거나 부적절한 규칙을 가지고 있는 가지(branch)를 제거
3) 타당성평가	이익도표(gain chart)나 위험도표(risk chart) 또는 검정용 자료(test data)에 의한 교차타당성(cross validation) 등을 이용해 평가
4) 해석 및 예측	구축된 의사결정나무를 해석하고 분류 및 예측모형을 설정하여 데이터의 분류 및 예측에 선정된 의사결정나무를 이용

- 정지기준, 분리기준, 평가기준 등을 어떻게 지정하느냐에 따라서 서로 다른 의사결정나무가 형성됨

### 나. 의사결정나무의 형성사례

1)최적 분할속성 선정	2)분할속성에 따른 분류	3)하위 분할속성 선정	4)동일한 분류결과 정리																								
<p>no surfacing이 최적 분할 속성임</p> <table border="1"> <tr><td></td><td>no surfacing</td><td>flippers</td><td>fish</td></tr> <tr><td>1</td><td>yes</td><td>yes</td><td>yes</td></tr> <tr><td>2</td><td>yes</td><td>yes</td><td>yes</td></tr> <tr><td>3</td><td>yes</td><td>no</td><td>no</td></tr> <tr><td>4</td><td>no</td><td>yes</td><td>no</td></tr> <tr><td>5</td><td>no</td><td>yes</td><td>no</td></tr> </table>		no surfacing	flippers	fish	1	yes	yes	yes	2	yes	yes	yes	3	yes	no	no	4	no	yes	no	5	no	yes	no	<p>분할함 칭향이 모두 같은 분류 분할 속성</p>	<p>같은 분류</p>	<p>같은 분류</p>
	no surfacing	flippers	fish																								
1	yes	yes	yes																								
2	yes	yes	yes																								
3	yes	no	no																								
4	no	yes	no																								
5	no	yes	no																								
분류를 가장 잘 분할하는 속성으로 최적 분할속성을 선택	선택된 분할속성에 따라 데이터를 분할, 트리 형성	트리의 가지에서 분할속성을 선택하여 하위 가지를 형성	동일한 결과값을 가진 분류결과를 대표값으로 정리																								

- 데이터로부터 트리 형성과정을 완료한 후 타당성 평가과정을 거쳐 분류 및 예측모형으로 사용함

## 3. 의사결정나무의 장단점

### 가. 의사결정나무의 장점

장점	설명
해석의 용이성	모형의 이해가 쉽고 새로운 자료의 모형에 적합하며, 어떤 입력변수가 목표변수를 설명하기에 좋은지 쉽게 파악
교호효과의 해석	두 개 이상의 변수가 결합하여 목표변수에 어떻게 영향을 주는지 쉽게 파악
비모수적 모형	선형성, 정규성, 등분산성 등의 가정이 불필요
분류결과 직관성	결과의 이해도가 높으며 의사결정에 있어 직접적으로 사용할 수 있어 활용도가 높음

## 나. 의사결정나무의 단점

단점	설명
불안정성	레코드 개수의 작은 차이에도 트리모양이 크게 달라짐
비연속성	Greedy 알고리즘의 사용으로 최적의 해를 보장하지 못함
비안정성	연속형 변수를 비연속적인 값으로 취급하기 때문에 분리의 경계점 부근에서 예측오류가 클 수 있음
낮은 정확도	분류율의 정확도 측면에서 신경망, 로지스틱 회귀분석 등의 분류방법보다 정확도가 낮음

## 4. 의사결정나무의 활용분야

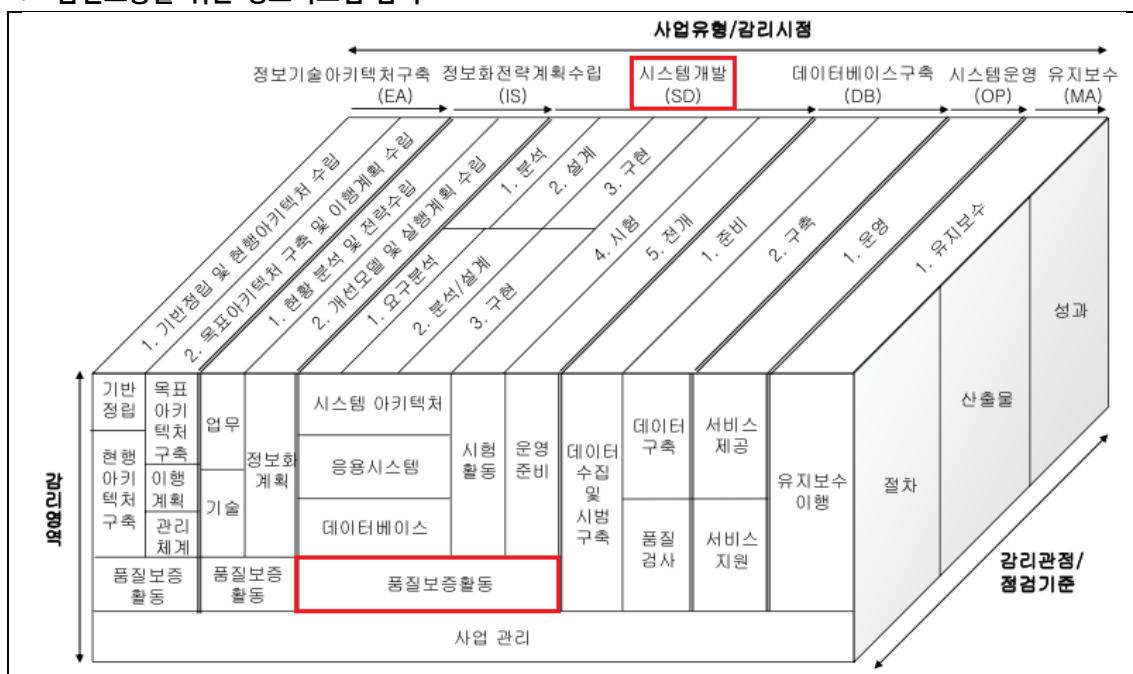
분야	설명
세분화	관측개체를 유사특성 그룹으로 분할하여 그룹별 특성을 발견
분류 및 예측	예측변수에 근거한 목표변수 분류, 규칙 발견을 통한 미래 예측
차원축소 및 변수선택	다수의 예측변수 중에서 목표변수에 큰 영향을 미치는 변수 추출
교호작용 효과 파악	다수의 예측변수 결합으로 목표변수에 작용하는 교호작용 파악
범주의 병합	범주형 목표변수의 범주를 소수의 몇 개로 병합

- 의사결정나무는 분류, 예측 등 다양한 목적으로 사용될 수 있으나 분석의 정확도보다는 분석과정의 설명이 필요한 경우에 더 유용하게 사용됨

"끝"

5 정보시스템 감리 절차	
문제	소프트웨어 개발 프로젝트 품질보증(Quality Assurance)을 위한 정보시스템 감리 절차에 대하여 설명하시오.
도메인	소프트웨어공학
정의	정보시스템 감리: 제 3 자 입장에서 요구사항 구현 일치 여부를 검토하는 활동
키워드	시스템개발(SD), 품질보증 활동,
출제의도분석	감리프레임워크에 대한 이해를 바탕으로 한 품질보증 감리에 대한 이해 확인
답안작성 전략	정확한 감리절차의 설명과 감리에서 사용되는 정확한 용어를 사용
참고문헌	정보시스템 감리점검 해설서 V3.0(한국정보사회진흥원) 정보시스템 감리수행 가이드 V2.1(한국정보사회진흥원)
모범목차	<ol style="list-style-type: none"> <li>1. 품질보증을 위한 정보시스템 감리</li> <li>2. 소프트웨어 개발 프로젝트 품질보증 감리절차           <ol style="list-style-type: none"> <li>가. 정보시스템 감리절차</li> <li>나. 소프트웨어 개발 프로젝트의 품질보증 감리 점검사항</li> </ol> </li> <li>3. 감리절차에 따른 SW 개발 단계별 상세 감리활동</li> </ol>
풀이 기술사님	이홍철 (bluerobe@naver.com)

## 1. 품질보증을 위한 정보시스템 감리



- 시스템개발 유형의 사업에서 SDLC 전단계(요구분석~전개)에 걸친 품질보증 활동에 대한 감리를 실시함
- 감리영역 중 품질보증 활동은 사업관리 중 중요한 요소이며, 사업 수행 시에도 별도의 품질보증활동 조직에 의해서 관리되고 있어 이를 독립된 관리영역으로 구분함

## 2. 소프트웨어 개발 프로젝트 품질보증 감리절차

### 가. 정보시스템 감리절차

단계	감리활동
감리계약체결	<ul style="list-style-type: none"> <li>- 감리기관은 감리비, 감리일정, 장소 등 감리계약에 필요한 기본적인 사항 등 감리계약서에 포함될 내용을 협의</li> <li>- 감리기관과 감리의뢰기관 사이의 법적인 충돌을 방지하는 목적으로 상호합의에 따라 감리계약서를 작성</li> </ul>
예비조사 실시 및 감리계획 수립	<ul style="list-style-type: none"> <li>- 감리준비: 감리일정, 감리대상 사업 개략적 조사 및 이해</li> <li>- 개별감리계획서 작성: 개별 감리 실시계획 작성</li> <li>- 예비조사: 감리대상 사업의 특성, 중점검토사항 도출</li> </ul>
감리 착수회의 실시	<ul style="list-style-type: none"> <li>- 공식적인 감리시작을 알리는 행위</li> <li>- 감리기관, 감리의뢰기관, 피 감리기관 등 관계기관의 참여 하에 공식적인 감리의 시작을 위한 착수회의를 실시</li> </ul>
감리 시행 및 감리보고서의 작성	<ul style="list-style-type: none"> <li>- 감리인이 현장 실사, 피 감리인 및 감리의뢰기관 관계자와의 면담 등을 통하여 중점검토항목별 문제점 및 개선사항을 발견하여 문서화</li> <li>- 각 감리인들이 현장감리기간 동안 발견된 주요 문제점 및 개선사항을 보고서화하고 감리의뢰인 및 피 감리인에게 확인</li> </ul>
감리 종료회의 실시	<ul style="list-style-type: none"> <li>- 감리기간중 발견된 중대 사항을 확인하기 위해 종료회의를 실시하며 회의는 감리총괄이 진행</li> <li>- 주관감리인이 종합적인 감리의견(총평)과 개선권고사항을 설명하고 피 감리인의 질의, 응답 과정을 통해 발견된 사항을 조정 및 확정</li> </ul>
감리보고서의 통보	<ul style="list-style-type: none"> <li>- 감리종료회의의 실시 후 각 감리인들은 상세검토사항의 내용을 조정</li> <li>- 종료회의가 끝난 날로부터 10 일 이내에 감리의뢰기관 및 피 감리기관의 장에게 통보</li> </ul>
감리에 따른 시정조치 결과의 확인 및 통보	<ul style="list-style-type: none"> <li>- 피 감리기관은 조치계획을 감리의뢰기관에 즉시 제출하고 감리의뢰기관은 감리결과 조치계획 및 조치결과를 감리인에게 제출</li> <li>- 조치사항에 대한 검토는 산출물 위주로 이루어지며, 조치결과 검토공문을 통보함으로서 조치사항 검토는 종료됨</li> </ul>

- 감리절차는 전자정부법 시행령 제 72 조 2 항에 규정된 바를 따름

### 나. 소프트웨어 개발 프로젝트 품질보증 감리 점검사항

감리 시점	개요
분석 (요구분석)	사업추진을 위한 방법론, 절차, 표준, 품질보증계획을 수립하고, 이에 따라 관련 산출물을 적정하게 작성하였는지 점검
설계 (분석/설계)	기 수립된 방법론, 절차, 표준, 품질보증 계획에 의거하여 각 활동을 수행하고 있으며, 관련 산출물을 적정하게 작성하였는지 점검
구현	기 수립된 방법론, 절차, 표준, 품질보증 계획에 의거하여 각 활동을 수행하고 있으며, 관련 산출물을 적정하게 작성하였는지 점검
시험	기 수립된 방법론, 절차, 표준, 품질보증 계획에 의거하여 각 활동을 수행하고 있으며, 관련 산출물을 적정하게 작성하였는지 확인하고, 초기 사업의 목표달성을 여부, 교육계획 등의 적정성을 점검
전개	사업을 마감하고, 구축된 시스템을 운영환경으로 이관하기 위한 준비와 각종 절차 및 계획을 적정하게 수행하였는지 점검

## 3. 감리절차에 따른 SW 개발 단계별 상세감리활동

SW 개발단계	감리활동(점검항목)
요구분석	<ul style="list-style-type: none"> <li>1. 사업 목표의 수립 여부</li> <li>2. 방법론 및 절차/표준의 수립 여부</li> <li>3. 반복계획을 적정하게 수립하였는지 여부</li> <li>4. 품질보증활동 계획을 적정하게 수립하였는지 여부</li> <li>5. 총괄시험 계획을 적정하게 수립하였는지 여부</li> <li>6. 방법론 및 절차/표준의 준수 여부</li> <li>7. 품질보증활동을 적정하게 수행하였는지 여부</li> <li>8. 사용자 요구사항 및 관련 산출물 간의 추적성, 일관성</li> </ul>
분석/설계	<ul style="list-style-type: none"> <li>1. 방법론 및 절차/표준의 준수 여부</li> <li>2. 이전단계 반복에 대한 평가 및 다음단계 반복계획 수립이 적정한지 여부</li> <li>3. 품질보증활동을 적정하게 수행하였는지 여부</li> <li>4. 사용자 요구사항 및 관련 산출물 간의 추적성, 일관성</li> <li>5. 시스템 전환 전략을 적정하게 수립하였는지 여부</li> </ul>
구현	<ul style="list-style-type: none"> <li>1. 방법론 및 절차/표준의 준수 여부</li> <li>2. 이전단계 반복에 대한 평가 및 다음단계 반복계획 수립이 적정한지 여부</li> <li>3. 품질보증활동을 적정하게 수행하였는지 여부</li> <li>4. 사용자 요구사항 및 관련 산출물 간의 추적성, 일관성</li> </ul>
시험	<ul style="list-style-type: none"> <li>1. 방법론 및 절차/표준의 준수 여부</li> <li>2. 이전단계 반복에 대한 평가 및 다음단계 반복계획 수립이 적정한지 여부</li> <li>3. 품질보증활동을 적정하게 수행하였는지 여부</li> <li>4. 사용자 요구사항 및 관련 산출물 간의 추적성, 일관성</li> <li>5. 사업목표의 달성을 여부</li> <li>6. 교육계획을 적정하게 수립하였는지 여부</li> </ul>
전개	<ul style="list-style-type: none"> <li>1. 방법론 및 절차/표준의 준수 여부</li> <li>2. 사용자 교육을 적정하게 실시하였는지 여부</li> <li>3. 인수 운영조직을 적정하게 구성하였는지 여부</li> </ul>

"끝"

6	NAT(Network Address Translation)
문제	NAT(Network Address Translation)의 IP Masquerading, Port Forwarding, Load Balancing에 대하여 설명하시오.
도메인	디지털 네트워크
정의	IP 패킷의 TCP/UDP 포트 숫자와 소스 및 목적지의 IP 주소 등을 재기록하면서 라우터를 통해 네트워크 트래픽을 주고 받는 기술
키워드	Local Address, Global Address
출제의도분석	공공기관의 지방이전 등과 맞물려 효과적인 전산통합센터 구축/운영과 관련된 이슈에 대한 문제해결 역량 요구
답안작성 전략	정확한 정의와 이해에 근거한 서술이 필요하며 IPv4에서의 IP 자원 고갈과 IPv4와 IPv6의 공존 문제에서의 관계성을 언급
참고문헌	Wikipedia의 관련 항목 제 95 회 기출 문제 해설 및 풀이 – 김영옥 PE Linux IP Masquerade HOWTO – KLD 포트 포워딩 – 리눅스랩 배철수
모범답안	1. NAT의 개요 가. NAT의 개념 나. NAT의 구성 유형 2. NAT의 주요 기능 가. IP Masquerading의 개요 나. Port Forwarding의 개요 다. Load Balancing의 개요 3. NAT의 활용 현황 및 전망
풀이 기술사님	강용제 (yongjeikang@gmail.com)

### ■ NAT(Network Address Translation, 네트워크 주소 변환)의 개념

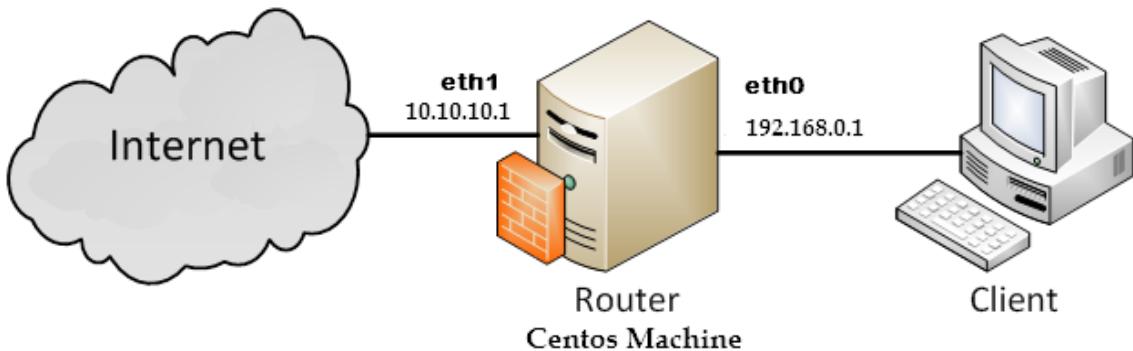
- IP 패킷의 TCP/UDP 포트 숫자와 소스 및 목적지의 IP 주소 등을 재기록하면서 라우터를 통해 네트워크 트래픽을 주고 받는 기술
- 패킷에 변화가 생기기 때문에 IP나 TCP/UDP의 체크섬도 다시 계산하여 재기록해야 함
- 사설 네트워크에 속한 여러 개의 호스트가 하나의 공인 IP를 사용하여 인터넷에 접속하기 위해 사용
- 호스트 간 통신에 있어 복잡성을 증가시킬 수 있으므로 네트워크 성능에 영향을 줄 수 있음

### ■ 구성 유형

유형	내용	매핑 테이블 예시	
Static NAT	- 사설 IP 주소와 공인 IP 주소가 1:1로 연결되는 구성	Private IP	Public IP
		192.168.0.1	211.210.10.201
		192.168.0.2	211.210.10.202
		192.168.0.3	211.210.10.203
		192.168.0.4	211.210.10.204
Dynamic NAT	- 사설 IP와 공인 IP 주소가 N:1 또는 N:M으로 연결되는 구성 - 사설 IP를 가진 장치의 요청시 동적으로 공인 IP 주소 Table에서 할당하여 인터넷과 연결하는 방식	Private IP	Public IP
		192.168.0.1	211.210.10.203
		192.168.0.2	
		192.168.0.3	
		192.168.0.4	211.210.10.204

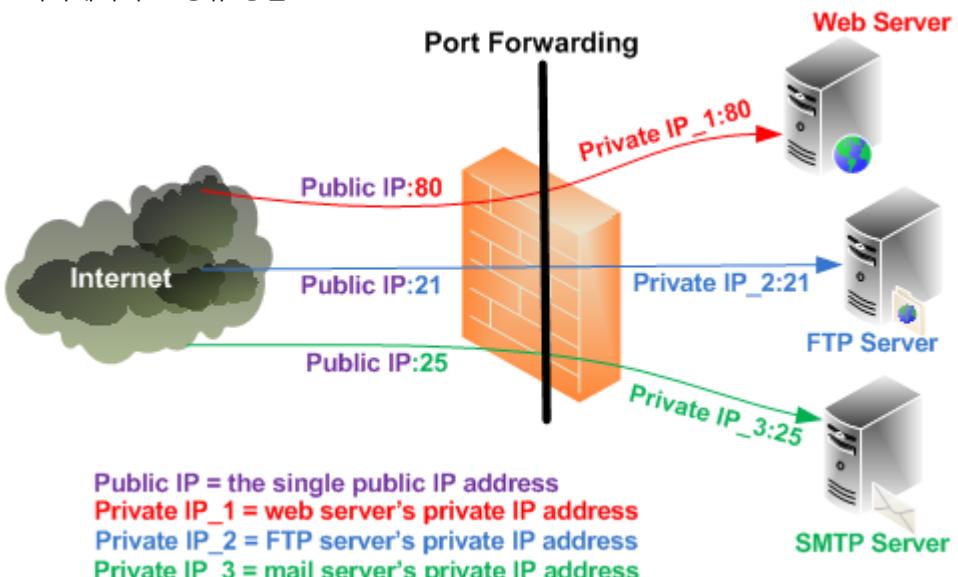
### ■ IP Masquerading 의 개요

- 리눅스의 네트워킹 기능으로 상용 방화벽 또는 네트워크 라우터에서 흔히 지원하는 1 대 다 방식의 NAT 와 유사
- 개인 IP 를 갖지 못한 터미널들에게 1 대의 서버(리눅스)를 통해 인터넷으로의 연결을 제공할 수 있는 네트워킹 기능으로 MASQ 게이트웨이에서 NAT 역할을 수행
- 클라이언트 쪽에서의 IP 공유 방법



### ■ Port Forwarding 의 개요

- 내부 네트워크에 대하여 Gateway 역할을 하는 서버의 특정 포트로 네트워크 연결이 들어오면 해당 연결을 내부 네트워크의 특정 서버의 포트로 연결(Forward) 시켜주는 네트워킹 기능
- 서버에서의 IP 공유 방법



### ■ Load Balancing 의 개요

- 하나의 인터넷 서비스에서 발생하는 트래픽이 많을 때 여러 대의 서버가 분산 처리하여 서버의 로드(부하) 증가, 부하량, 속도 저하 등을 해결하는 기술
- NAT 에서의 Load Balancing 은 Gateway 역할을 하는 서버에서 외부에서 인입되는 요청을 내부 네트워크의 여러 서버에게 분산해 연결하는 네트워킹 기술



### ■ NAT 의 활용 현황 및 전망

- IPv4 의 주소 부족 문제를 완전히 해결하는 IPv6 의 완전한 도입 이전에 "임시" 해결책으로 각광
- IPv4 와 IPv6 네트워크 간의 연결을 위해서 비슷한 기술이 적용됨
- 내부 네트워크의 보안 유지를 위해 NAT 를 활용(잘 설치된 IP MASQ 는 내부 네트워크의 서버들을 거의 완벽하게 보호함)
- 인터넷 서비스의 부하를 분산 처리하기 위하여 서비스 단위의 분산(Load balancing) 방법으로 사용되며 이는 이종 서비스의 분산 처리와 동종 서비스의 분산 처리 모두에 사용됨
- Appliance 형태의 NAT 를 활용해 IP Masquerading, Port Forwarding, Load Balancing 등을 간단히 이용하는 것이 가능하며 이를 통해 부족한 IPv4 의 IP 고갈 문제도 완화하는 상태임

"끝"

Notes

1	Cross Validation
문제	충분한 데이터가 확보되지 않은 상황에서 데이터 분석을 수행할 경우 통계적 신뢰도를 높이기 위하여 사용하는 Cross Validation 방법을 3 가지 이상 예를 들어 설명하시오.
도메인	알고리즘
정의	모델에 대한 타당성을 모델링에 사용하지 않은 다른 데이터로 평가하는 방법
키워드	training set, test set, random/leave one out/k-fold cross validation
출제의도분석	빅데이터 분석에 대한 깊이 있는 지식으로써 모델평가에 대한 이해여부 평가
답안작성 전략	Cross Validation 의 각 유형에 대한 상세한 설명에 집중
참고문헌	<a href="http://en.wikipedia.org/wiki/Cross-validation_(statistics)">http://en.wikipedia.org/wiki/Cross-validation_(statistics)</a>
모범목차	<ol style="list-style-type: none"> <li>1. Cross Validation 의 개요</li> <li>2. Cross Validation 의 유형과 평가방법             <ol style="list-style-type: none"> <li>가. Cross Validation 의 유형</li> <li>나. Cross Validation 의 평가방법</li> </ol> </li> <li>3. Cross Validation 방법 설명             <ol style="list-style-type: none"> <li>가. leave one out cross validation</li> <li>나. k-fold cross validation</li> <li>다. repeated random sub-sampling validation</li> </ol> </li> <li>4. Cross Validation 방법의 비교</li> </ol>
풀이 기술사님	이홍철 (bluerobe@naver.com)

## 1. Cross Validation 의 개요

항목	설명
정의	전체 Data Set 을 Training Set 과 Test Set 으로 분리하고, Training Set 을 사용한 학습 등을 통해서 모델을 생성한 후, 학습에 사용되지 않은 Test Set 을 통해서 생성된 모델의 성능을 측정하고 평가하는 기법
목적	<ul style="list-style-type: none"> <li>- 확보된 데이터의 수가 적은 데이터 분석 신뢰도에 대한 평가</li> <li>- 학습결과로 생성된 모델에 대한 성능을 측정하고 평가</li> <li>- 과적합(overfitting)과 같은 학습단계에서의 문제를 해결</li> </ul>

- 교차검증(Cross Validation)은 데이터의 수가 부족한 경우 유용하게 사용되고 있으며, 역사에 대한 검증 등 다양한 분야에서도 사용되고 있음

## 2. Cross Validation 의 유형과 평가방법

### 가. Cross Validation 의 유형

Exhaustive cross validation	Non-exhaustive cross validation
<ul style="list-style-type: none"> <li>- leave p out cross validation(LpOCV)</li> <li>- leave o out cross validation(LOOCV)</li> </ul>	<ul style="list-style-type: none"> <li>- k-fold cross validation</li> <li>- 2-fold cross validation</li> <li>- repeated random sub-sampling Validation</li> </ul>

- Data Set 을 Training Set 과 Test Set 으로 분할하는 모든 경우의 수를 고려하는지 여부에 따라서 Exhaustive/Non-exhaustive 로 구분함

## 나. Cross Validation 의 평가방법

		실제결과/분류	
추론결과 /분류	참	거짓	
	참	tp	fp
거짓	fn	tn	

- tp(true positive), fp(false positive)  
fn(false negative), tn(true negative)

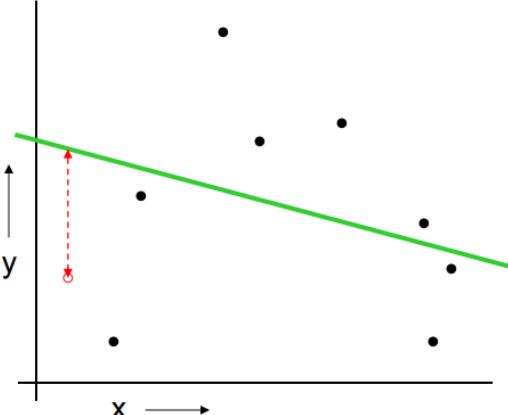
$$\text{정확률} = \frac{tp}{tp+fp}$$

$$\text{재현률} = \frac{tp}{tp+fn}$$

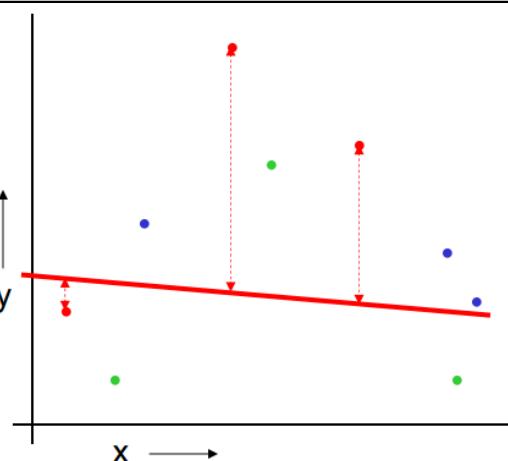
- Training Set 으로 모델링된 모델의 정확률과 재현률을 측정해 Data Set 에 대한 최적의 모델을 검증

## 3. Cross Validation 방법 설명

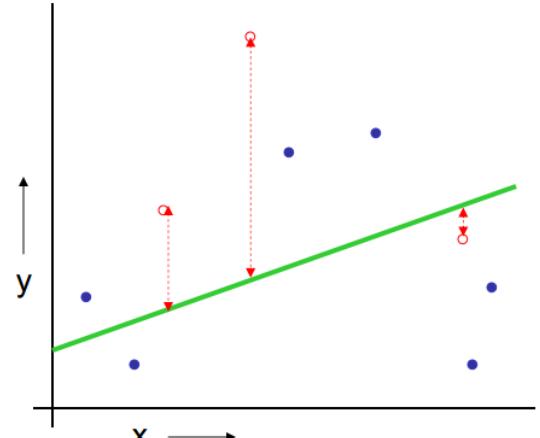
## 가. LOOCV(leave one out cross validation)

개념도	수행방법
	<ol style="list-style-type: none"> <li>임의의 Data (X, Y)를 선택</li> <li>Data Set 에서 선택된 Data 를 제외한 나머지 Data 로 Training Set 을 구성</li> <li>Training Set 으로 학습과정을 통해 모델생성</li> <li>Training Set 에서 제외된 Data 를 Test Set 으로 모델과의 오류정도를 측정</li> <li>모든 Data 에 대해서 1)~4)를 수행하여 평균오류를 측정</li> </ol>

## 나. K-fold cross validation

개념도	수행방법
	<ol style="list-style-type: none"> <li>Data Set 을 분리할 임의수 k 를 선택</li> <li>Data Set 을 k 개의 Set 으로 분리 (본 사례에서는 k=3, 즉, 3 개 Set 으로 분리)</li> <li>1 개의 Set 을 제외한 나머지 Data 로 Training Set 을 구성</li> <li>Training Set 으로 학습과정을 통해 모델생성</li> <li>Training Set 에서 제외된 Set 을 Test Set 으로 모델과의 오류정도를 측정</li> <li>모든 Data Set 에 대해서 1)~5)를 수행하여 평균오류를 측정</li> </ol>

다. Repeated random sub sampling validation

개념도	수행방법
	1) Data Set에서 임의의 30% Data를 선택 2) 선택된 30%의 Data로 Test set으로 사용 3) 나머지 70%의 Data를 Training Set으로 사용 4) Training Set으로 학습과정을 통해 모델생성 5) Test Set으로 모델과의 오류정도를 측정 6) 임의의 횟수로 1)~5)를 반복해 평균오류를 측정

#### 4. Cross Validation 방법의 비교

항목	LOOCV	k-fold	random sampling
장점	Data Set에서 낭비되는 Data가 없음	LOOCV에 비해 측정, 평가비용이 적음	측정, 평가비용이 가장 적음
단점	측정, 평가 비용이 가장 비쌈	10-fold를 사용한다면 10%의 Data가 낭비됨	미래 예측 시 신뢰성을 추정할 수 없음

"끝"

Notes

2	데이터 이관
문제	차세대 시스템 구축 프로젝트에서 기존 시스템(AS-IS) 데이터를 차세대 시스템(TO-BE)으로 이관 작업하는 절차 및 유의 사항에 대하여 설명하시오.
도메인	데이터베이스
정의	기 사용 중인 시스템의 저장 데이터를 추출하고 변형해 신규 시스템에서 사용할 수 있도록 신규 시스템의 데이터 공간으로 입력하는 일련의 과정
키워드	Data Mapping, ETL, 시스템 표준화
출제의도분석	시스템 전환 과정에서 발생하는 대량 데이터 이관 업무에 대한 이해 측정 및 실무적 절차와 유의 사항 지식 평가
답안작성 전략	시스템의 전환 과정에서 데이터 이관의 발생 위치, 실제 업무 수행 과정에서의 문제점, 유의 사항에 대해 서술
참고문헌	데이터 마이그레이션 – 끊임없이 계속해서 되풀이되는 주제 – QLOGIC 데이터베이스 마이그레이션 방안 – Microsoft TechNet
모범목차	<ol style="list-style-type: none"> <li>1. 데이터 이관 작업의 개요</li> <li>2. 데이터 이관 작업 실행 방식</li> <li>3. 데이터 이관 작업의 유의 사항 및 대응 방안</li> </ol>
풀이 기술사님	강용제 (yongjei.kang@gmail.com)

### ■ 데이터 이관 작업의 개념

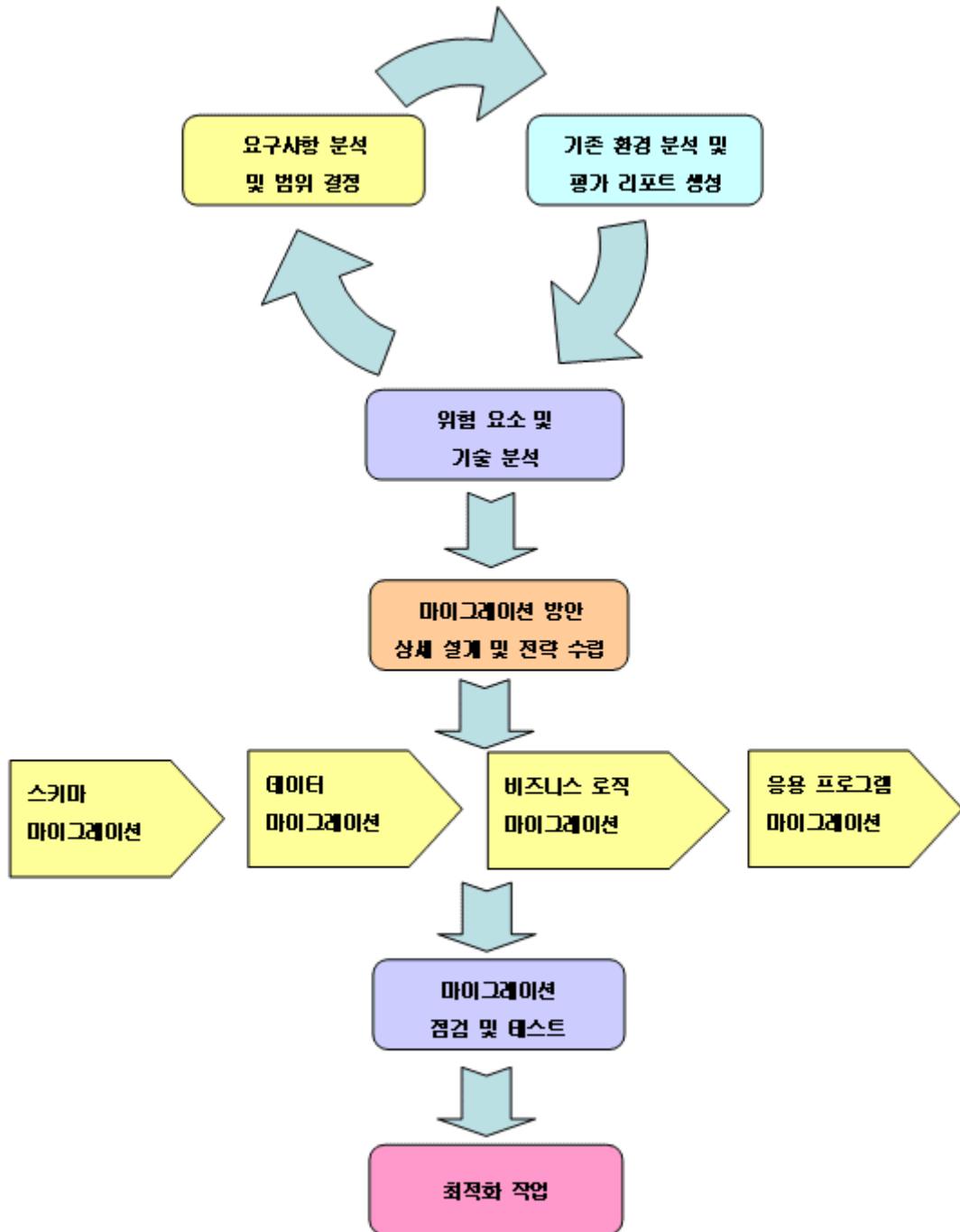
- 기 사용 중인 시스템의 저장 데이터를 추출하고 변형해 신규 시스템에서 사용할 수 있도록 신규 시스템의 데이터 공간으로 입력하는 일련의 과정
- 일반적으로 데이터 이관 작업이라고 하면 DBMS를 통해 관리되는 데이터의 이관을 의미

### ■ 데이터 이관 작업 실행 방식

실행 방식	내용
빅뱅 방식	<ul style="list-style-type: none"> <li>- 전환 프로그램을 단순하게 유지하는 것이 가능</li> <li>- 검증 작업이 상대적으로 단순해짐</li> <li>- 전환 작업의 수행 시간이 매우 긴</li> <li>- High Risk</li> <li>- 주로 대규모의 차세대 시스템 Open 시 사용</li> </ul>
단계적 수행 방식	<ul style="list-style-type: none"> <li>- 전환 Risk 감소</li> <li>- 데이터 검증 작업을 단계적으로 수행해 전체 정확성을 높일 수 있음</li> <li>- 복잡하고 여러 단계로 분리된 데이터 검증 작업 수행 필요</li> <li>- 전환 작업 전체 소요 시간이 빅뱅 방식보다 긴</li> <li>- 업무 시스템별 업그레이드 형태의 작업에 주로 적용</li> </ul>

### ■ 데이터 이관 작업의 절차

- 일반적으로 데이터 이관 작업은 이관의 범위와 대상 시스템의 분석 작업 이후 수행되며 이관 받을 시스템의 개발 작업이 종료되는 시점에 시작되는 경우가 많음



절차	내용
マイ그레이션 요구 사항 분석	- 마이그레이션의 범위를 결정하고 이전 과정의 요구 사항을 분석
기존 환경 진단 및 분석	- 기 축적된 데이터에 대한 분석 - 마이그레이션 평가 및 평가 리포트 생성
マイ그레이션 위험 분석	- 이전 과정의 위험 요소 식별과 기술에 대한 분석 수행
マイ그레이션 방안 설계	- 이전 작업 수행 전략 수립 - 이전 작업 수행 방안 설계
스키마 마이그레이션	- 동종 DBMS 간의 데이터 이전의 경우 용이한 마이그레이션 가능

Notes

	- 이종 DBMS 간의 이전인 경우 각 DBMS 의 기능과 한계를 명확히 분석해야 함
데이터 마이그레이션	<ul style="list-style-type: none"> <li>- 실제 운영해 축적된 데이터 자체의 이전 과정</li> <li>- 테이블의 개수, 데이터의 양 등을 고려해 작업 방법 결정</li> <li>- 오류 발생의 위험을 줄이기 위해 가능한 동일한 방법으로 모든 테이블의 이전을 수행</li> <li>- ETL 도구를 이용한 이전은 마이그레이션의 작업 수행 절차를 간소화 시켜 줌</li> </ul>
비즈니스 로직 마이그레이션	<ul style="list-style-type: none"> <li>- 이기종 DBMS 간의 내장 프로그램(PL/SQL, T-SQL 등)의 전환/이전</li> <li>- 구 시스템과 신 시스템 양쪽에 대해 충분한 지식을 가진 인력 필요</li> <li>- 자동 변환 도구 SW 를 이용하는 것이 효율적이나 도구의 한계를 인식해 전문 개발자의 수동 변환 작업도 필요</li> </ul>
응용 프로그램 마이그레이션	<ul style="list-style-type: none"> <li>- 기존 응용 프로그램의 데이터베이스 연결 부분만 변경</li> <li>- 연결 계층 구조에 따른 변환 작업 필요</li> </ul>
마이그레이션 점검 및 테스트	<ul style="list-style-type: none"> <li>- 이전된 데이터의 행수, 정합성을 점검</li> <li>- 프로시저, 함수, 트리거 등의 수행 결과 비교</li> <li>- 응용 프로그램의 오류 점검</li> <li>- 백업 및 보안 정책 정합성 점검</li> <li>- 신규 시스템 환경 설정 점검</li> </ul>
최적화 작업	<ul style="list-style-type: none"> <li>- 신규 시스템의 기능과 한계에 적합한 시스템 최적화 작업 수행</li> <li>- 전문가에 의해 진행</li> </ul>

### ■ 데이터 이관 작업의 유의 사항 및 대응 방안

구분	유의 사항	대응 방안
시스템 측면	시스템 성능 및 용량 고려	<ul style="list-style-type: none"> <li>- 정확한 DISK/CPU 소요량 산정</li> <li>- 대규모 IO(Network) 발생에 대한 대비 필요</li> <li>- Network 장애 발생에 대비한 비상 계획, 복구 계획을 사전에 수립</li> <li>- 병렬 Extract 및 Load 도구를 충분히 이용</li> </ul>
	언어 코드 전환	<ul style="list-style-type: none"> <li>- DB 의 문자 셋에 따라 코드 변환 작업 수행 필요</li> <li>- 한글 문자 코드의 경우 변환 시 오류 검증 필요</li> </ul>
데이터 측면	전환 결과 검증	<ul style="list-style-type: none"> <li>- 정합성 검증 방법과 대상을 사전에 식별</li> <li>- 여러 번의 리허설을 통해 검증 방법에 대한 오류 수정</li> <li>- 검증 결과에 따라 무결성 유지를 위한 복구 시나리오 준비</li> </ul>
관리적 측면	조직, 업무 일정 지연 대비	<ul style="list-style-type: none"> <li>- 데이터 전환에 따른 업무 시스템의 중단 등에 대비</li> <li>- 필수 업무 수행을 위한 임시 지원 방안 수립 및 실행</li> <li>- 중단 서비스 또는 지연 가능성 등에 대한 사전 고지</li> </ul>
	사용자(현업) 참여 문제	<ul style="list-style-type: none"> <li>- 개별 데이터에 대한 Mapping 오류 처리 방안 준비</li> <li>- 데이터에 대한 Ownership 을 가진 현업 담당자 지정 및 대기</li> </ul>
	이관 실행	<ul style="list-style-type: none"> <li>- 데이터 전환/이관 우선 순위 사전 지정</li> <li>- 데이터 전환/이전의 Critical Path 에 대한 성능 튜닝</li> <li>- 병렬(분산) 처리 시스템의 적극적 활용</li> </ul>

"끝"

Notes

3 반복 수행 계획서, 반복 수행 평가서	
문제	반복 수행 계획서(Iteration Software Development Plan)와 회차 종료 전에 작성되어져야 하는 반복 수행 평가서(Iteration Assessment)의 목차를 작성하시오.
도메인	경영전략
정의	반복 수행 계획서: 해당 반복 주기에 해야 할 일을 제반 환경을 고려하여 세분화한 계획서 반복 수행 평가서: 반복 주기의 결과물을 평가하기 위해 필요한 평가 기준, 이슈 사항, 변경 내용 등 정리하는 문서
키워드	SDP(SW Development Plan), RUP(Rational Unified Process)
출제의도분석	RUP 개발 방법론에서 사용되는 주요 문서에 대한 지식 확인 (비슷한 반복 개발 방법인 Agile의 유행에 따라 양자간의 지식 구분 여부 필요)
답안작성 전략	반복 수행 계획서와 반복 수행 평가서의 역할 정의 및 RUP 전체 프로세스에서 이 두 문서의 위치 등을 언급
참고문헌	Wikipedia의 관련 항목 RUP(Themis OO/Themis Web) Project Management: 반복 주기 계획 및 반복 주기 평가 – NEXTREE 김상일 The Art of Agile Development - <a href="http://www.jamesshore.com/Agile-Book/">http://www.jamesshore.com/Agile-Book/</a> ( <a href="http://www.jamesshore.com/Agile-Book/iteration_planning.html">http://www.jamesshore.com/Agile-Book/iteration_planning.html</a> )
모범목차	1. 반복 수행 계획서와 반복 수행 평가서의 개요 가. 반복 수행 계획서의 개념 나. 반복 수행 평가서의 개념 2. 반복 수행 계획서와 반복 수행 평가서의 목차 가. 반복 수행 계획서의 목차 나. 반복 수행 평가서의 목차 3. RUP에서 반복 수행 계획서와 반복 수행 평가서의 위치
풀이 기술사님	강용제 (yongjei.kang@gmail.com)

### ■ 반복 수행 계획서의 개념

- 해당 반복 주기에 해야 할 일을 제반 환경을 고려하여 세분화(태스크와 활동)한 계획서
- 무엇을 언제 해야 하는지와 다른 활동에 따라 결정되는 것을 계획하는 문서
- 세밀한 방법으로 완료된 것을 자세하게 기술해야 원래의 역할과 책임에 대해 명확하지 않게 될 소지가 줄어듬
- 하나의 반복 주기를 위한 세분화된 계획 – 간혹 2개의 계획이 작성되는데 현재 반복 주기에 대한 것과 다음 반복 주기에 대한 것임

### ■ 반복 수행 계획서의 목차

목차	내용
1. 목적 및 범위	- 반복 계획서의 목적 및 범위 기술 - 이전 반복의 결과를 반영한 최신의 위험 평가
2. 반복 회차의 일정	- 해당 반복 회차의 수행 기간 정의
3. 구축 범위 및 개발 계획	- 해당 반복 회차의 구축 대상 범위 정의 - 상세한 반복 계획의 준비 - 진행 상황을 감시하기 위한 중간 이정표 포함
4. 테스트 및 평가 방안	- 단위 반복 개발 시 수행할 테스트 활동 계획 - 이전 개발 완료 기능과의 연계 및 통합 테스트 수행 방안 - 반복을 위한 평가 기준의 결정
5. 전개 방안	- 반복 개발 완료 후 사용자 Delivery 방안 수립

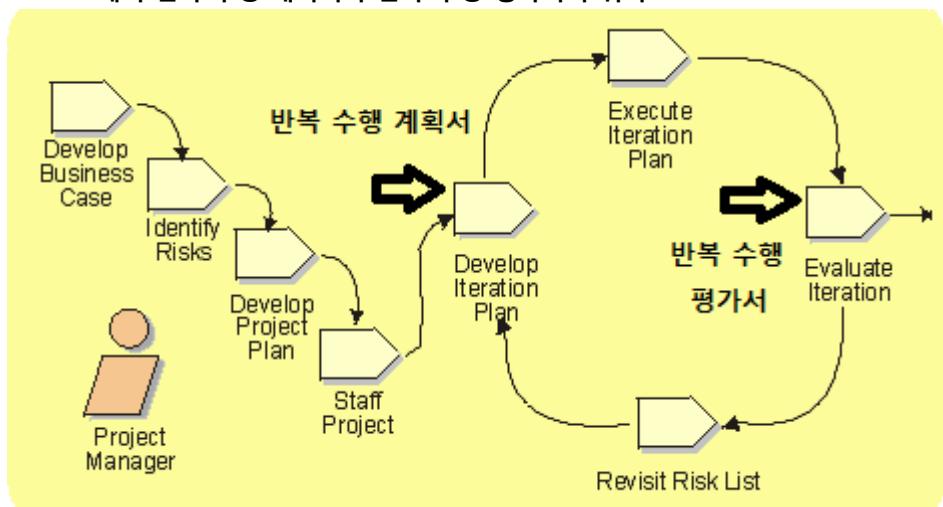
### ■ 반복 수행 평가서의 개념

- 반복 주기의 결과를 획득하고 평가 기준에 대응하는 문제점과 완료된 변경 내용의 정도를 획득하는 문서
- 각 반복 주기의 마지막에 작성됨
- 매우 중요한 수행 단계이며 프로젝트의 범위, 위험 요소, 반복 주기의 특징 등을 기록
- 반복 수행 평가는 매우 중요한 활동이며 그 산출물인 반복 수행 평가서 역시 매우 중요하므로 정확하게 평가하고 기록해야함
- 반복 수행의 진행 상황과 평가 기준(Metric)을 모니터링

### ■ 반복 수행 평가서의 목차

목차	내용
1. 평가 일정	- 결과 산출물에 대한 평가 수행 일정
2. 평가 대상	- 각 반복 개발 결과에 평가 대상 정의
3. 평가 항목	- 기능성, 사용성, 상호 연계성 등의 개발 품질 평가 항목 정의
4. 평가 기준	- 해당 평가 항목에 대한 평가 단위 및 기준값 정의 - 반복 수행 계획에서 수립한 평가 방안 반영
5. 평가 방법	- 평가 항목별 평가 기준 부합 여부 판별을 위한 방법 제시 - Walkthrough, Review 포함
6. 평가 결과	- 평가 방법에 따른 평가 결과 도출 - 요구 사항, 경쟁사 계획 등의 변경 - 필요하다면 재작업 필요성 결정 후 이후 반복에 배정

■ RUP에서 반복 수행 계획서와 반복 수행 평가서의 위치



- RUP는 객체 지향 SW 개발 프로세스를 제시함
- 점진적, 반복적인 개발 과정을 가짐 → Iteration 의 연속으로 개발 수행: 워크 플로우의 반복적 프로세스
- 각 반복마다 실행 가능한 릴리즈 산출
- 각 반복에서 반복의 시작점에 반복 수행 계획서를 작성하고 해당 반복의 릴리즈 시기에 반복 수행 평가서를 작성하게 됨

"끝"

4	데이터 무결성 제약
문제	데이터베이스에 저장된 값과 현실 세계를 표현하는 실제값이 일치하는가에 대해 정확성을 의미하는 데이터 무결성 제약(Integrity Constraint)의 유형으로 도메인 무결성 제약과 릴레이션 무결성 제약이 있다. 이 두가지에 대하여 설명하시오.
도메인	데이터베이스
정의	데이터무결성: 데이터의 중복이나 누락 없이 정확성, 일관성이 보장된 특성
키워드	NULL/NOT NULL, DEFAULT, CHECK, 입력 참조무결성, 수정/삭제 참조무결성
출제의도분석	데이터 무결성에 대한 기본적인 지식의 확인
답안작성 전략	질문인 두 가지 무결성에 대해서 사례를 기반으로 상세하게 설명
참고문헌	데이터베이스론(이석호, 정의사)
모범목차	<ol style="list-style-type: none"> <li>1. 데이터 무결성의 개요</li> <li>2. 도메인무결성의 설명             <ol style="list-style-type: none"> <li>가. 도메인무결성 위반사례와 영향</li> <li>나. 도메인무결성 보장방법</li> </ol> </li> <li>3. 릴레이션무결성의 설명             <ol style="list-style-type: none"> <li>가. 릴레이션무결성 위반사례와 영향</li> <li>나. 릴레이션무결성 보장방법</li> </ol> </li> </ol>
풀이 기술사님	이홍철 (bluerobe@naver.com)

## 1. 데이터 무결성의 개요

항목	설명	
개념	데이터가 중복되거나 누락되지 않도록 정확성과 일관성이 보장된 특성	
종류	개체 무결성	한 개체는 중복되거나 누락될 수 없는 특성
	도메인 무결성	도메인의 값은 정의된 규칙을 준수해야 하는 특성
	릴레이션 무결성	외래키가 참조하는 값은 다른 개체의 기본키 또는 NULL
	사용자 무결성	사용자가 요구하는 의미적 요구사항을 준수하는 특성

- 데이터무결성의 훼손은 점진적으로 이뤄지며, 장기적으로 지속될 경우 데이터베이스에 무의미한 데이터가 누적되어 비즈니스의 무결성을 훼손하게 되므로, 위반사례에 대한 분석과 대응이 필요함

## 2. 도메인 무결성의 설명

### 가. 도메인무결성 위반사례와 영향

위반사례	영향												
<b>member_table</b> <table border="1"> <tr> <td>이메일</td> <td>이름</td> <td>나이</td> <td>성별</td> </tr> <tr> <td>a@b.c</td> <td>홍길동</td> <td>20</td> <td>남</td> </tr> <tr> <td>b@c.d</td> <td>김철수</td> <td>25</td> <td>남</td> </tr> </table> * '성별'에 별도의 제약조건을 설정하지 않음	이메일	이름	나이	성별	a@b.c	홍길동	20	남	b@c.d	김철수	25	남	<ul style="list-style-type: none"> <li>- 도메인의 속성이 취할 수 있는 값이 이론적으로 무한대로 늘어날 수 있음</li> <li>- '남' 또는 '여'로 데이터를 검색할 경우 의미상으로는 데이터가 존재함에도 불구하고 누락되는 현상이 발생함</li> </ul>
이메일	이름	나이	성별										
a@b.c	홍길동	20	남										
b@c.d	김철수	25	남										
INSERT INTO member_table VALUES('c@d.e','이영희','22','여자'); - 위의 SQL 을 실행하여 새로운 데이터를 입력시 '성별'도메인에 '여자'라는 값이 입력됨	<ul style="list-style-type: none"> <li>- 조회, 통계 등의 데이터 활용단계에서 누락이 발생함에 따라 정확성이 보장되지 않는 데이터를 출력할 수 있으므로 데이터무결성이 훼손됨</li> </ul>												

## 나. 도메인무결성 보장방법

보장방법	설명
NULL/NOT NULL	상품테이블에 상품명이 반드시 존재해야 할 경우 등, NOT NULL 제약조건 사용
DEFAULT	계산이 필요한 속성에 NULL 이 존재하면 안되므로 DEFAULT로 '0'을 설정
CHECK	비즈니스 규칙으로 성별을 '남', '여'로만 구분할 경우 CHECK로 유효값 검증

## 3. 릴레이션 무결성의 설명

## 가. 릴레이션무결성 위반사례와 영향

위반사례	영향															
<p>parent_table</p> <table border="1"> <tr> <td>ID</td> <td>이름</td> </tr> <tr> <td>A</td> <td></td> </tr> <tr> <td>B</td> <td></td> </tr> </table> <p>child_table</p> <table border="1"> <tr> <td>ID</td> <td>이름</td> <td>Parent</td> </tr> <tr> <td>가</td> <td>홍길동</td> <td>A</td> </tr> <tr> <td>나</td> <td>김철수</td> <td>B</td> </tr> </table> <p>* child_table 은 parent_table 의 ID 를 참조함</p> <p>DELETE FROM parent_table WHERE ID='B';      INSERT INTO child_table VALUES('다','이영희','C');</p> <p>- 위의 SQL 을 통해 입력과 삭제를 할 경우      참조무결성이 훼손됨</p>	ID	이름	A		B		ID	이름	Parent	가	홍길동	A	나	김철수	B	<ul style="list-style-type: none"> <li>- Parent 가 삭제 될 경우, Parent 를 참조하는 Child 는 유효하지 않은 속성값을 갖게 되어 데이터의 무결성이 훼손됨</li> <li>- Parent 를 유효하지 않은 속성값으로 갖고 있는 Child 가 삽입될 경우 데이터의 무결성이 훼손됨</li> </ul>
ID	이름															
A																
B																
ID	이름	Parent														
가	홍길동	A														
나	김철수	B														

## 나. 릴레이션무결성 보장방법

구분	보장방법	설명
입력 참조무결성	DEPENDENT	참조되는(부모) 테이블에 PK 값이 존재할 때만 입력 허용
	AUTOMATIC	참조되는(부모) 테이블에 PK 값이 없을 때는 PK 생성후 입력
	DEFAULT	참조되는(부모) 테이블에 PK 값이 없을 때는 기본값 입력
	CUSTOMIZED	특정한 조건이 만족할 때만 입력을 허용
	NULL	참조되는(부모) 테이블에 PK 값이 없을 때는 외부키를 NULL 처리
	NO EFFECT	조건 없이 입력을 허용
수정/삭제 참조무결성	RESTRICT	참조하는(자식) 테이블에 PK 값이 없는 경우 삭제/수정 허용
	CASCADE	참조되는(부모) 테이블과 참조하는 테이블 외부키 연쇄 수정/삭제
	DEFAULT	참조되는(부모) 테이블의 수정을 항상 허용하고, 참조하는(자식) 테이블의 외부키를 지정된 기본값으로 변경
	CUSTOMIZED	특정한 조건이 만족할 때만 수정/삭제 허용
	NULL	참조되는(부모) 테이블의 수정을 항상 허용하고, 참조하는(자식) 테이블의 외부키를 NULL 값으로 수정
	NO EFFECT	조건없이 수정/삭제 허용

"끝"

5	디지털 포렌식
문제	최근 IT 와 법률의 융합 서비스 분야의 디지털 포렌식 기술이 수사 기관에서 기업의 감사 영역으로 확대되고 있다. 기업에서 활용 가능한 디지털 포렌식 기술을 설명하고 기업에서 적용 가능한 분야를 설명하시오.
도메인	디지털 보안
정의	법정에서의 증거로 사용하기 위한 디지털 포렌식 기술을 기업 활동에서의 사실 관계 규명을 위해 사용하는 경향
키워드	침해 사고 포렌식, 악성 코드 포렌식, 저작권 포렌식, 감사 포렌식, 회계 포렌식
출제의도분석	기업 활동의 전반에 있어 디지털 기기가 전방위적으로 사용되는 경향에 맞춰 기업 활동에서의 사실 관계 규명에 디지털 포렌식 기술을 적용할 필요
답안작성 전략	기업 활동에 있어 디지털 포렌식을 적용할 수 있는 분야와 사용 기술을 나열하고 디지털 포렌식 기술 사용시의 주의 사항에 대해 언급
참고문헌	Wikipedia 의 관련 항목 데일리시큐( <a href="http://www.dailysecu.com">www.dailysecu.com</a> )
모범답안	<ol style="list-style-type: none"> <li>1. 기업에서의 디지털 포렌식 개요             <ol style="list-style-type: none"> <li>가. 기업에서의 디지털 포렌식 개념</li> <li>나. 기업에서의 디지털 포렌식 특성/필요성</li> </ol> </li> <li>2. 기업에서의 디지털 포렌식 적용             <ol style="list-style-type: none"> <li>가. 기업 내에서 디지털 포렌식 적용 분야</li> <li>나. 기업 내에서 디지털 포렌식 적용시 사용 기술</li> </ol> </li> <li>3. 기업에서 디지털 포렌식 기술 적용시 유의 사항</li> </ol>
풀이 기술사님	강용제 ( <a href="mailto:yongjei.kang@gmail.com">yongjei.kang@gmail.com</a> )

### ■ 기업에서의 디지털 포렌식 개념

- 법정에서 사용될 증거 수집을 위한 디지털 포렌식 기술 적용이 아니라 기업 내의 특정 행위 결과에 대한 사실 관계를 규명하기 위해 디지털 포렌식 기술을 사용하는 경향
- 기업 내의 거의 모든 업무 수행 과정에서 디지털 기기가 사용되고 있고 업무의 진행, 결과 등의 기록 역시 디지털 기기에 디지털화해 저장되므로 기업 내 업무에서 사실 관계 규명이 요구될 경우 디지털 포렌식 기술의 적용은 필수적임

### ■ 기업 내에서의 디지털 포렌식 적용 특성

구분	특성
규제 준수	- 기업 내부의 업무에 대한 적용이므로 기업 내외부의 업무와 정보에 대한 법적, 행정적 규제를 준수해야 함
적용 경계 인식	- 해킹 사고 등에 대한 적용 시 기업 외부의 네트워크와 정보에 대한 접근에 유의해야 함
경영 목표에 일치	- 기업의 경영 목표와 발전에 초점을 두고 기업 정보 유출 예방과 감사 등에 집중

### ■ 기업 내 디지털 포렌식 적용 분야

구분	적용 내용
침해 사고 포렌식	<ul style="list-style-type: none"> <li>- 침해 사고의 원인과 영향을 분석</li> <li>- 침해 사고 발생 여부의 검증</li> <li>- 침해 유입 경로의 규명, 침해 방법의 규명</li> <li>- 침해 예방을 위한 방안 수립</li> </ul>
악성 코드 포렌식	<ul style="list-style-type: none"> <li>- 유입된 악성 코드의 흔적을 분석</li> <li>- 악성 코드 자체에 대한 분석</li> <li>- 악성 코드의 존재, 유포 경로 등에 대한 분석/규명</li> <li>- 악성 코드 예방을 위한 방안 수립</li> </ul>
저작권 포렌식	<ul style="list-style-type: none"> <li>- 기업의 지적 재산에 대한 저작권 등의 보호를 위한 분석</li> <li>- 저작권 침해 여부에 대한 검증</li> <li>- 저작권 침해자 규명</li> <li>- 저작권 침해 예방을 위한 방안 수립</li> </ul>
감사 포렌식	<ul style="list-style-type: none"> <li>- 기업 업무의 수행 과정, 제반 규정의 준수 등에 대한 감사를 위한 분석</li> <li>- 업무 수행 과정과 결과에서의 규정 준수 여부 분석</li> <li>- 배임 등의 예방을 위한 감사 방안 수립</li> </ul>
회계 포렌식	<ul style="list-style-type: none"> <li>- 회계 데이터의 은닉과 조작 여부를 판단하기 위한 분석</li> <li>- 분식 회계 등의 예방, 사실 관계 분석</li> <li>- 회계 자료에 대한 조작 여부 규명 및 원천 데이터 정합성 보증</li> </ul>

- 정보 유출 등의 보안 사고 발생 시 민/형사 상의 책임 소재를 가리기 위한 증거 자료 확보도 중요
- e-Discovery 대응을 위해 대량 자료의 분석, 추출 역시 디지털 포렌식의 일부로 볼 수 있음

### ■ 기업 내 디지털 포렌식 사용 기술

- 기업 내의 디지털 포렌식에서 사용되는 기술 역시 법정 증거를 위한 디지털 포렌식에 사용되는 기술에서 큰 차이는 없으나 조사 범위에 대한 법률적 권한에서 제약이 많음

분류	기술	내용
수집 기술	디스크 이미징	<ul style="list-style-type: none"> <li>- 원본 디스크에 손상이나 변경 없이 원본 디스크를 물리적으로 동일하게 복제하거나 미러 이미지 파일 생성</li> </ul>
	메모리 덤프	<ul style="list-style-type: none"> <li>- 프로세스가 사용 중인 가상 메모리의 덤프를 획득하여 정복 추출</li> </ul>
	무결성 입증	<ul style="list-style-type: none"> <li>- 증거가 변경되지 않았음을 입증하기 위한 해쉬, 메시지 디아제스트</li> </ul>
분석 기술	Time line 분석	<ul style="list-style-type: none"> <li>- 파일의 생성 시간, 최근 접근 시간, 수정 시간 등 분석</li> <li>- 파일 시스템에 대한 사용 로그 분석</li> <li>- 업무 시스템 등에 대한 사용 로그 시간 분석</li> </ul>
	파일 복구	<ul style="list-style-type: none"> <li>- 삭제 파일의 복원</li> <li>- 변경된 파일의 이전 내용 복구</li> <li>- 복구가 불가능한 경우 파일의 존재 여부 증명</li> </ul>
	비정상 파일 검색	<ul style="list-style-type: none"> <li>- 숨김 속성의 파일 및 확장자 변환 파일의 검색</li> <li>- 파일의 고유 헤더 정보를 이용한 확장자 변경 확인</li> </ul>
	이메일 분석	<ul style="list-style-type: none"> <li>- 삭제된 이메일의 복구</li> <li>- 이메일의 전송 과정에서의 기록 복구</li> </ul>
	슬랙 공간 분석	<ul style="list-style-type: none"> <li>- 디스크의 남는 공간에서의 데이터 은닉 등을 분석</li> </ul>

Notes

	<b>암호 복구</b>	- 시스템 또는 파일에 설정된 암호 크랙
	<b>덤프 메모리 분석</b>	- 코드 영역, 데이터 영역, 스택 영역에 대한 분석 - 침해 사고, 악성 코드 등의 발생 시 실행
	<b>수치 자료 정합성 분석</b>	- 회계 자료에 대한 조작, 삭제 등의 분석 - 회계 원칙과 규제에 대한 고려 필요

### ■ 기업 내 디지털 포렌식 적용 시 유의 사항

- 법원의 영장 또는 수사권에 근거한 포렌식이 아니므로 분석/검증 과정에서 대상 선정의 한계를 명확히 해야 함
- 사용하는 포렌식 SW의 기능과 한계를 파악하고 과도한 신뢰나 불신이 발생하는 것을 피해야 함
- 개인 정보 침해, 기업 비밀 유출 등의 사안은 법률 기관의 규제 대상이므로 규제에 근거해 법적 조치 의무를 다 하여야 함

"끝"



Notes

6	Quick Sort 알고리즘
문제	<p>Quick Sort 알고리즘에 대하여 설명하고 아래의 C 언어 소스코드에서 필요시 함수등을 추가하여 완성하시오. (단, Sort 순서는 오름차순)</p> <pre>#include &lt;stdio.h&gt; #include &lt;string.h&gt; void Quicksort(int *data, int n) { } void main() {     char data[8] = {'B','T','D','O','Z','L','H'};     puts(data);     QuichSort(data, 7);     puts(data); }</pre>
도메인	알고리즘
정의	Pivot 을 이용해 리스트를 분할하여 재귀적으로 정렬과정을 반복하는 알고리즘
키워드	Pivot, 분할과 정복, 재귀적 수행
출제의도분석	알고리즘의 수행절차에 대한 이해와 프로그래밍으로의 구현능력 확인
답안작성 전략	사례를 통해 알고리즘 수행과정을 설명하고, 알고리즘을 간결한 코드로 구현
참고문헌	뇌를 자극하는 알고리즘(박상현, 한빛미디어)
모범목차	<ol style="list-style-type: none"> <li>1. Quick Sort 알고리즘의 개요             <ol style="list-style-type: none"> <li>가. Quick Sort 알고리즘의 정의</li> <li>나. Quick Sort 알고리즘의 수행원리</li> </ol> </li> <li>2. Quick Sort 알고리즘의 수행             <ol style="list-style-type: none"> <li>가. 알고리즘 수행절차</li> <li>나. 사례에 대한 알고리즘 수행과정</li> </ol> </li> <li>3. Quick Sort 의 소스코드 완성</li> </ol>
풀이 기술사님	이홍철 (bluerobe@naver.com)

## 1. Quick Sort 알고리즘의 개요

### 가. Quick Sort 알고리즘의 정의

- 기준이 되는 Pivot 을 정해서 그것보다 큰 원소와 작은 원소들로 분할하여 분할된 각 배열을 같은 방법으로 다시 정렬해 나가는 알고리즘
- 하나의 커다란 입력데이터의 집합을 정렬하는 것보다, 두 개의 작은 입력데이터들을 정렬하는 것이 더 빠르다는 일반적인 사실에 바탕을 둔 정렬 알고리즘

### 나. Quick Sort 알고리즘의 수행원리

원리	설명
비교 정렬	다른 원소와의 비교과정만을 거쳐서 정렬을 수행하는 비교정렬
분할과 정복	Pivot 을 기준으로 리스트를 분할하여 분할된 리스트를 각각 정렬
재귀적 수행	분할된 리스트에 대하여 재귀적으로 분할과 정렬과정을 반복해 정렬

## 2. Quick Sort 알고리즘의 수행

### 가. 알고리즘 수행절차

절차	수행방법
1) Pivot 설정	리스트에서 하나의 원소를 선택해 Pivot 으로 설정
2) 원소교환, 분할	Pivot 앞에는 Pivot 보다 작은 값, Pivot 뒤에는 Pivot 보다 큰 값이 오도록 원소를 교환하며 Pivot 을 기준으로 리스트를 둘로 분할
3) 재귀적 반복	분할된 두 개의 리스트에 대해 재귀적으로 리스트의 크기가 0 또는 1 이 될 때까지 반복 수행

### 나. 사례에 대한 알고리즘 수행과정

Step	수행 과정								Pivot
1	B	I	D	O	Z	L	H		B
2	B(sorted)	I	D	H	Z	L	O		I
3	B(sorted)	H	D	I	Z	L	O		I
4	B(sorted)	H	D	I(sorted)	Z	L	O		I
5	B(sorted)	H	D	I(sorted)	Z	L	O		H
6	B(sorted)	D(sorted)	H(sorted)	I(sorted)	Z	L	O		H
7	B(sorted)	D(sorted)	H(sorted)	I(sorted)	Z	L	O		Z
8	B(sorted)	D(sorted)	H(sorted)	I(sorted)	L(sorted)	Z	O		Z
9	B(sorted)	D(sorted)	H(sorted)	I(sorted)	L(sorted)	O(sorted)	Z(sorted)		Z

- Step4 에서 Pivot(I)을 기준으로 리스트를 분할하고, Step5~6 은 왼쪽의 리스트 정렬, Step7~8 은 오른쪽의 리스트를 정렬함

Notes

### 3. Quick Sort의 소스코드 완성

```
void Quicksort(int *data, int n)
{
    q_sort(data, 0, n - 1);
}

void q_sort(int *data, int left, int right)
{
    if(left == right) return;
    int pivot, l_hold, r_hold;
    l_hold = left;
    r_hold = right;
    pivot = data[left];
    while (left < right)
    {
        while ((data[right] >= pivot) && (left < right)) right--;
        if (left != right)
        {
            data[left] = data[right];
            left++;
        }
        while ((data[left] <= pivot) && (left < right)) left++;
        if (left != right)
        {
            data[right] = data[left];
            right--;
        }
    }

    data[left] = pivot;
    pivot = left;
    left = l_hold;
    right = r_hold;
    if (left < pivot) q_sort(data, left, pivot-1);
    if (right > pivot) q_sort(data, pivot+1, right);
}
```

"끝"