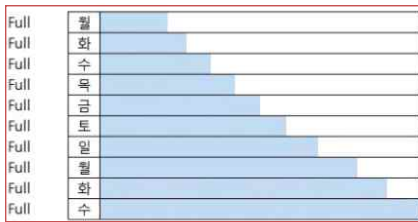


1. 오류-부재의 궤변 : SW 결함을 모두 제거해도 사용자의 요구사항을 만족시키지 못하면 해당 SW는 품질 ↓
결함 집중 : 애플리케이션 결함의 대부분은 소수의 특정한 모듈에 집중되어 존재함 (파레토 법칙)
파레토 법칙 : 전체 결함의 80%는 SW 제품의 전체 기능 중 20%에 집중되어 있음
살충제 패러독스 : 동일한 테스트케이스로 반복하면 결함을 발견하지 못하므로 주기적으로 테스트케이스를 리뷰 및 개선
2. 스토리보드 : 디자이너와 개발자가 최종적으로 참고하는, 서비스 구축을 위한 대부분의 정보가 수록돼있는 설계 산출물
3. 웹 서비스의 3요소 : UDDI, SOAP, WSDL
 - ① UDDI : **WSDL**을 등록하여 서비스와 서비스 제공자를 검색하고 접근하는데 사용되는 XML 기반의 웹 서비스 요소
 - ② SOAP : 네트워크상에서 **HTTP/HTTPS, SMTP 등을 이용하여 XML을 교환**하기 위한 통신 규약 (Header, Body)
 - ③ WSDL : **웹 서비스에 대한 상세 정보**를 표준적인 방법으로 기술하고 게시하기 위한 기술 (XML 형식으로 구현)
4. 연계 메커니즘의 연계 방식 : 직접 연계 방식, 간접 연계 방식
 - ① 직접 연계 방식 : DB Link, DB Connection Pool, JDBC, API/Open API, 하이퍼링크
 - ② 간접 연계 방식 : EAI, ESB, 웹 서비스, Socket
5. 회귀 테스트 : 모듈 또는 컴포넌트의 변화 시에도 새로운 오류가 발생하지 않았음을 보증하기 위한 반복 테스트
6. 구조 다이어그램 : CCOD CPP (Class, Component, Object, Deployment, Composite, Package, Profile)
행위 다이어그램 : SSAC UIT (Sequence, State, Activity, **Communication**, Usecase, Interaction, Timing)
7. SSL : 개인정보를 보호하기 위해 필요한 개인정보 유지 프로토콜로, 데이터 통신 보안을 제공
S-HTTP : HTTP에 보안기능을 부가하기 위한 통신 규약(S-HTTP는 제공되는 페이지만, HTTPS는 전체내용 암호화)
8. 이상 현상 : 일부 속성들의 종속으로 인해 데이터의 중복이 발생하여 테이블 조작 시 불일치가 발생하는 현상
 - ① 갱신 이상 : 반복된 데이터 중 일부만 수정하면 데이터의 불일치가 발생
 - ② 삽입 이상 : 불필요한 정보를 함께 저장하지 않고는 어떤 정보를 저장하는 것이 불가능 [갱 삽 삭]
 - ③ 삭제 이상 : 유용한 정보를 함께 삭제하지 않고는 어떤 정보를 삭제하는 것이 불가능 [일부 불필 유용]
9. 프로세스 스케줄링 기법 : 비선점 스케줄링(FIFO, SJF, HRN) / 선점 스케줄링(Round Robin, SRT)
10. SW 개발보안 3요소 : 기밀성, 무결성, 가용성(기무가)
 - ① 기밀성 : 인가된 사용자만 정보자산에 접근이 가능한 성질
 - ② 무결성 : 적절한 권한을 가진 사용자만 정보를 변경할 수 있는 성질
 - ③ 가용성 : 적절한 시간에 정보자산에 접근이 가능한 성질
11. DRM : 웹을 통해 유통되는 각종 디지털 콘텐츠의 안전 분배와 불법 복제 방지를 위한 보호 방식을 의미 (제분하소~)
콘텐츠 **제공자** : 저작권자 [콘텐츠, 메타데이터, 패키지(콘텐츠를 메타 데이터와 함께 배포 가능한 단위로 묶는 기능)]
콘텐츠 **분배자** : 암호화된 콘텐츠를 콘텐츠 소비자에게 제공 [콘텐츠 상거래]
클리어링 **하우스** : 자유로운 상거래 제공을 위해 콘텐츠 키 관리 및 라이선스 발급을 관리 [권한정책, 라이선스]
콘텐츠 **소비자** : [DRM 컨트롤러(배포된 콘텐츠의 이용권한 통제), 보안 컨테이너(안전유통을 위한 전자적 보안장치)]
12. SQL 처리단계 : PARSE → EXECUTE → FETCH (PEF)
13. 테스트 검증 기준의 테스트 종류 : 명세기반 테스트, 구조기반 테스트, 경험기반 테스트
 - ① 명세기반 테스트 : 주어진 명세를 빠짐없이 테스트 케이스로 구현하고 있는지 확인하는 테스트
 - ② 구조기반 테스트 : SW 내부 논리 흐름에 따라 테스트 케이스를 작성하고 확인하는 테스트
 - ③ 경험기반 테스트 : 테스터의 경험을 토대로 수행하는 테스트

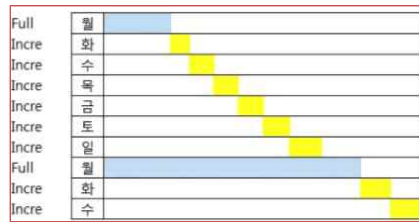
14. 테스트 계획서 : 테스트 목적과 범위, 수행절차, 일정, 조직의 역할과 책임 등 테스트 수행을 계획한 문서
 테스트 케이스 : 입력값, 출력값, 결과 등 테스트할 구체적인 조건을 문서화한 테스트 항목 명세서
 테스트 시나리오 : 여러 테스트 케이스의 집합으로, 테스트 케이스의 동작 순서를 기술한 문서
 테스트 결과서 : 테스트 결과를 정리한 문서(테스트 프로세스를 리뷰하고, 테스트 결과를 평가)
 테스트 커버리지 : 테스트 케이스들이 테스트에 얼마나 적절한지를 판단하는 품질 측정 기준 (기능기반, 라인, 소스코드)
 테스트 오라클 : 테스트의 결과가 참인지 거짓인지(올바른지)를 판단하기 위해서 사전에 정의된 참 값을 대입하여 비교
 테스트 하네스 : 시스템 및 컴포넌트를 테스트하는 환경의 일부분으로, 테스트 지원 목적으로 생성된 코드와 데이터
 테스트 데이터 : 컴퓨터의 동작이나 시스템의 적합성을 시험하기 위해 특별히 개발된, 조건을 갖춘 데이터

15. 백업 방식 : 전체 백업(Full), 증분 백업(Incremental), 차등 백업(Differential)

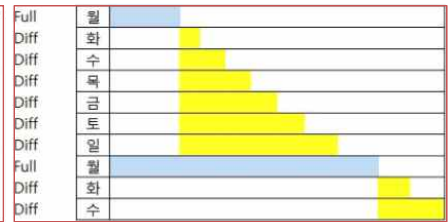
- ① 전체 백업(Full) : 데이터 전체를 백업받는 방식
 ② 증분 백업(Incremental) : 백업 데이터 영역 중 변경되거나 증가된 데이터만을 백업받는 방식
 ③ 차등 백업(Differential) : Full 백업 이후 변경사항을 모두 백업받는 방식



① 전체 백업



② 증분 백업



③ 차등 백업

16. 시큐어코딩 : 안전한 SW를 개발하기 위해 지켜야 할 코딩 규칙과 소스코드 취약목록이 포함(기무가, 인증, 부인방지)

17. 정규화 : 테이블을 무손실 분해하여, 데이터의 중복을 방지하고 이상 발생 가능성을 줄이는 작업 (원부이결 다조~)

- ① 제1정규형 : 모든 속성이 원자값을 가짐 (원자 도메인)
 ② 제2정규형 : 부분 함수적 종속 제거 (함수적 종속 표기법 : $X \rightarrow Y$)
 ③ 제3정규형 : 이행 함수적 종속 제거
 ④ BCNF : 모든 결정자가 후보키
 ⑤ 제4정규형 : 다치 종속(다중값 종속) 제거
 ⑥ 제5정규형 : 조인 종속성 이용

18. JAVA 접근 제한자

- ① Public : + 모든 클래스에서 접근이 가능함
 ② Protected : # 자신을 포함하거나 해당 클래스를 상속받은 클래스에서 접근이 가능함
 ③ Private : - 자신을 포함한 클래스에서만 접근이 가능함
 ④ Package : ~

19. 라우팅 알고리즘 : 거리 벡터 알고리즘(DV), 링크 상태 알고리즘(LS)

- ① 거리 벡터 알고리즘(DV) : 라우터와 라우터 간 거리, 방향 정보를 이용하여 최단경로를 찾는 방식
 ① RIP : 최초의 라우팅 프로토콜. 30초 주기로 전체 라우팅 정보 갱신하므로 많은 시간 소요 (소규모)
 ② IGRP : RIP의 문제점 개선. 네트워크 상태를 고려하여 라우팅 (중규모 네트워크에 적합)
 ② 링크 상태 알고리즘(LS) : 라우터와 라우터 간 모든 경로를 파악한 뒤 대체 경로를 사전에 마련하는 방식
 네트워크를 일관성 있게 파악할 순 있으나 DV에 비해 계산이 복잡함
 ① OSPF : 라우팅 정보에 변화가 생길 경우, 변화된 정보만 네트워크 내의 모든 라우터에게 알림
 RIP의 난점을 해결하였고, 경로 수에 제한이 없으므로 대규모 네트워크에 사용됨
 ② BGP : 자율 시스템 간, 인터넷 서비스 업체 간의 상호 라우팅 (규모가 큰 네트워크의 상호 연결)

20. 암호화 알고리즘 : 단방향 알고리즘, 양방향 알고리즘(공개키 알고리즘, 개인키 알고리즘)

① 단방향 : 암호화가 가능하면 복호화 불가능, 복호화가 가능하면 암호화 불가능 (SHA-256, MD5 ...)

SHA-256 : SHA 알고리즘의 한 종류로서, 256비트의 축약된 메시지를 생성하는 해시 알고리즘

② 양방향 : 암호화와 복호화 둘 다 가능

③ 공개키 : 암호화키 ≠ 복호화키(암호화키 공개) / 보안수준 ↑ / 복잡 · 느림 · 크기 큼 / RSA(비대칭형)

RSA : 공개 키 암호 체계의 사실상의 세계 표준

④ 개인키 : 암호화키 = 복호화키(키 모두 비공개) / 보안수준 ↓ / 단순 · 빠름 · 크기 작음 / DES · AES

SEED : 1999년 한국인터넷진흥원에서 개발된 128 및 256비트의 국산 대칭형 블록 암호화 알고리즘

ARIA : 128비트의 블록 크기와 키 길이에 따라 128, 192, 256으로 분류되는 블록 암호화 알고리즘

AES : NIS에 의해 개발된 128비트의 대칭형 블록 암호화 알고리즘

DES : 64비트 암호화 단위와 56비트 암호화 키를 사용하는 비대칭형 블록 암호화 알고리즘

21. SW 테스트 유형 : 정적 테스트, 동적 테스트 (화구조루기데, 불기동경원비상오)

① 정적 테스트 : 프로그램 실행 없이 소스코드의 구조를 분석하여 논리적으로 검증하는 테스트

② 워크스루 : 오류 조기검출에 목적을 두고 해결책은 나중에 미루는 비공식적 검토방법

③ 인스펙션 : 요구사항 명세서 작성자를 제외한 다른 전문가 또는 팀이 결함 발견하는 공식적 검토방법

④ 동적 테스트 : 프로그램의 실행을 요구하는 테스트

⑤ 화이트박스 테스트 : 프로그램 내부 로직(구조, 루프 등)을 보면서 테스트 수행 (논리적 경로를 제어)

키워드) 소스코드 · 논리 · 구조 · 경로 · 조건 · 루프 · 흐름

· 조건 검사 : 논리적 조건 테스트

· 루프 검사 : 반복 구조 중심 테스트

· 기초 경로 검사 : 논리적 복잡성을 측정할 수 있게 해주며, 측정 결과는 실행경로 기초 정의에 사용

· 데이터 흐름 검사 : 변수 정의, 변수 사용 위치에 초점을 맞춰 테스트

⑥ 블랙박스 테스트 : 사용자 요구사항 명세서를 보며 테스트. 주로 구현된 기능을 테스트

키워드) 성능 · 기능 · 인터페이스 오류 (논리 구조상의 오류는 화이트박스!)

· 동치 분할 검사 : 입력 자료에 초점을 맞춰 테스트 케이스를 만들고 검사하는 기법

· 경계값 분석 : 입력 조건의 경계값에서 오류 발생 확률이 높으므로 입력 조건의 경계값으로 테스트

· 원인-효과 그래프 검사 : 입력 자료 간의 관계와 출력에 영향을 미치는 상황을 체계적으로 분석

· 비교 분석 검사 : 여러 버전의 프로그램에 동일한 자료를 제공해 동일한 결과가 출력되는지 테스트

· 상태 전이 테스트

· 오류 예측 검사 : 과거 경험이나 테스터의 감각으로 테스트

22. UI : 사용자와 컴퓨터가 정보를 주고받기 위해 사용자와 프로그램이 상호작용을 매개하는 것

사용자와 컴퓨터 상호 간의 소통을 원활하게 도와주는 연계작업

23. UML : SW 개발에 사용되는 다이어그램을 정의하는 활동으로, 모델링 언어 표기법의 표준화를 목적(객체지향 모델링)

24. 휴먼 에러 : 부적절하거나 원치 않는 인간의 결정이나 행동, 즉 사람의 판단 · 조작 실수 등으로 인해 발생하는 에러

25. 프로토콜의 기본 구성 요소 : 구문, 타이핑, 의미 (구타의미)

26. redo 연산 : 로그에 기록된 변경 연산 후의 값을 이용하여 변경 연산을 재실행

undo 연산 : 로그에 기록된 변경 연산 이전의 값을 이용하여 변경 연산을 취소

27. 반입 전략 : 데이터를 언제 주기억장치로 적재할 것인지를 결정 (요구 반입, 예상 반입)
배치 전략 : 데이터를 주기억장치의 어디에 위치시킬 것인지를 결정 (최초 적합, 최적 적합, 최악 적합)
교체 전략 : 데이터의 어느 영역을 교체할 것인지를 결정 (FIFO, OPT, LRU, LFU, NUR)
① FIFO : 가장 먼저 들어와서 가장 오래 있었던 페이지를 교체 (First In First Out)
② OPT : 가장 오랫동안 사용하지 **않을** 페이지를 교체 (OPTimal replacement = 최적 적합)
③ LRU : 가장 오랫동안 사용하지 **않은** 페이지를 교체 (Least Recently Used)
④ LFU : 사용 빈도가 가장 적은 페이지를 교체 (Least Frequently Used)
⑤ NUR : 최근에 사용하지 않은 페이지를 교체 (Not Used Recently)
28. 애플리케이션 모니터링 도구(APM) : 앱의 흐름 모니터링과 성능예측을 통해 최적의 앱 상태를 보장하고 관리하는 도구
29. ① 경계 : 시스템과 외부 액터와의 상호작용을 담당하는 클래스
② 엔티티 : 시스템이 유지해야 하는 정보를 관리하는 기능을 전담하는 클래스
③ 제어 : 시스템이 제공하는 기능의 로직 및 제어를 담당하는 클래스
30. MVC 패턴 : 구현하려는 전체 애플리케이션을 모델, 뷰, 컨트롤러로 구분하여 UI와 비즈니스로직을 서로 분리하여 개발
31. 이름 LIKE 김%(김으로 시작) / 이름 LIKE %김%(김을 포함) / 이름 LIKE 김_(김으로 시작하는 3글자)
32. OSI 7계층? = 국제 표준화 기구(ISO)에서 제안한 7단계 표준화 프로토콜이며, 각 계층은 서로 독립적임
① 물리 계층 : 전기적·기능적·절차적, 즉 물리적인 연결 방식 (랜 케이블)
② 데이터링크 계층 : 두 컴퓨터 간 데이터 통신 규정 (오류제어, 흐름제어) → 이더넷, PPP
③ 네트워크 계층 : 여러 컴퓨터 간 데이터 통신 규정 (경로설정, 교환기술) → IP, ARP, ICMP, IGMP
④ 전송 계층 : 실제 전송을 하기 위한 규정 (주소설정, 다중화, 종단 간 제어) → TCP, UDP
⑤ 세션 계층 : 데이터 교환을 관리하기 위한 규정 (대화제어, 동기제어) → SSH, TLS, RPC
⑥ 표현 계층 : 데이터 표현 규정 (암호화, 코드변환) → MIME, SSL, JPEG, MPEG
⑦ 응용 계층 : 사용자 프로그램 (전자 사서함, 카카오톡, 메신저) → HTTP, FTP, SMTP, DNS

33.

응집도	내용
기능적 응집도	모듈 내부의 모든 기능이 단일한 목적을 위해 수행되는 경우 (가장 좋은 품질)
순차적 응집도	한 활동으로부터 나온 출력 값을 다른 활동이 입력 값으로 사용할 경우
통신적 응집도	동일한 입력과 출력을 사용하여 서로 다른 기능을 수행하는 구성요소들이 모여 있을 경우
절차적 응집도	모듈이 다수의 관련 기능을 가질 때, 모듈 안 구성요소들이 기능을 순차적으로 수행할 경우
시간적 응집도	특정 시간에 처리되어야 하는 활동들을 한 모듈에서 처리할 경우
논리적 응집도	유사한 성격을 갖거나 특정 형태로 분류되는 처리 요소들이 한 모듈에서 처리되는 경우
우연적 응집도	모듈 내부의 각 구성요소들이 연관이 없을 경우 (가장 나쁜 품질)

결합도	내용
자료 결합도	모듈 간의 인터페이스로 전달되는 파라미터를 통해서만 상호작용 발생 (가장 좋은 품질)
스탬프 결합도	모듈 간의 인터페이스로 배열이나 오브젝트, 스트럭처 등이 전달되는 경우
제어 결합도	단순 처리할 대상인 값만 전달되는 게 아니라 제어요소까지 전달되는 경우
외부 결합도	외부로 선언한 데이터(변수)를 다른 모듈에서 참조할 때의 결합도 / 모듈 간 결합이 심각
공통 결합도	파라미터가 아닌 모듈 밖에 선언되어 있는 전역변수(공통 데이터 영역)를 참조하는 경우
내용 결합도	다른 모듈 내부에 있는 변수나 기능을 다른 모듈에서 사용하는 경우 (가장 나쁜 품질)

34. 요구공학 : 무엇을 개발해야 하는지 요구사항을 정의하고, 문서화하고, 관리하는 프로세스를 연구하는 학문
요구사항 도출 → 요구사항 분석 → 요구사항 명세 → 요구사항 확인 (요구공학 프로세스 or 요구사항 개발 프로세스)

35. 소스코드 품질분석 도구 : 정적 분석도구, 동적 분석도구
- ① 정적 분석도구 : 코드 자체만으로 코딩표준 준수여부, 소스코드 복잡도 계산, 결함 발견여부 확인 (실행X)
 - ② 동적 분석도구 : 메모리 누수 현황을 발견하고, 발생한 스레드의 결함 등을 분석하기 위한 도구 (실행O)
36. 운영체제 : 컴퓨터 시스템의 자원들을 효율적으로 관리하며, 사용자 편의성을 제공하기 위해 환경을 제공하는 SW
- ① 처리능력 : 일정시간 내에 시스템이 처리하는 일의 양
 - ② 신뢰도 : 시스템이 주어진 문제를 정확하게 해결하는 정도
 - ③ 사용가능도 : 시스템을 사용할 필요가 있을 때 즉시 사용 가능한 정도
 - ④ 응답시간, 반환시간 : 시스템에 작업을 의뢰한 시간부터 처리가 완료될 때까지 걸린 시간
37. 반드시 : <<include>> 선택적 : <<extend>>
38. 릴리즈노트 : 최종 사용자인 고객과 SW 제품정보(릴리즈 정보)를 공유하는 문서. SW 제품과 함께 배포되는 문서들
39. 단위 테스트 → 통합 테스트 → 시스템 테스트 → 인수 테스트 (단통시인~)
- ① 알파 테스트 : 개발자의 장소에서 사용자(개발인력)가 시험하고, 개발자는 뒤에서 결과를 지켜보는 검사(인수 테스트)
 - ② 베타 테스트 : 실업무를 가지고 사용자가 직접 시험하는 검사 (인수 테스트)
40. UML 구성요소 : 사물, 관계, 다이어그램 (그 중 관계에 대한 내용)
- ① 연관 관계 : 두 클래스가 서로 어떠한 연관을 가지고 있다는 의미 [Association]
 - ② 직접 연관 관계 : 한 쪽으로 연관 관계를 가질 때 [Directed Association]
 - ③ 집합 연관 관계 : 독립적으로 존재하는 클래스가 다른 클래스에 포함되는 경우 (부서-직원) [Aggregation]
 - ④ 복합 연관 관계 : 특정 클래스가 존재하기 위해서 다른 클래스를 내포 (손-손가락) [Composition]
- 포함하는 사물의 변화가 포함되는 사물에게 어떤 영향을 미치는지? (③의 특수한 형태)
- ⑤ 의존 관계 : 한 클래스의 변화가 다른 클래스에 영향을 미치는 관계 (짧은 시간동안만 연관) [Dependency]
 - ⑥ 상속 관계 : 하나의 사물이 다른 사물에 비해 더 일반적인지 구체적인지를 표현 [Generalization]
 - ⑦ 구현 관계 : 클래스가 특정 인터페이스를 구현하는 관계 (의존 관계 + 상속 관계) [Realization, 실체화]
41. IPSec : OSI 3계층인 인터넷 프로토콜에서 보안성을 제공해주는 표준화된 기술. IPv4는 선택, IPv6는 필수로 제공
- IKE : 인터넷 표준 암호키 교환 프로토콜이며, IPSec을 암호화하는 데 사용됨
- AH : 출발지 인증, 데이터 무결성은 제공하지만 기밀성은 제공하지 않음
- ESP : 출발지 인증, 데이터 무결성, 기밀성 등 지원 (IPSec에 있어서 데이터 무결성과 프라이버시를 제공하는 기능)
42. 데이터 모델의 구성요소 3가지 : 구조(Structure), 연산(Operation), 제약조건(Constraint) [SOC 쏙!]
43. REST : 월드 와이드 웹(WWW)과 같은 분산 하이퍼미디어 시스템을 위한 SW 아키텍처의 한 형식
- RESTful : HTTP와 REST의 원칙을 사용하여 구현되는 웹 서비스
44. 프레임워크(재사용 가능하도록 뼈대를 제공하는 프로그램)의 4가지 특징 : 모듈화, 재사용성, 확장성, 제어의 역흐름
- ① 모듈화 : 캡슐화를 통해 모듈화를 강화하고 변경에 따르는 영향을 극소화하여 SW 품질을 향상시킴
 - ② 재사용성 : 반복적으로 사용할 수 있는 컴포넌트를 재사용하여 SW 품질 및 개발자의 생산성을 향상시킴
 - ③ 확장성 : 다형성을 통해 애플리케이션이 프레임워크의 인터페이스를 확장할 수 있게 함
 - ④ 제어의 역흐름 : 개발자가 관리하고 통제해야 하는 객체들의 제어 권한을 프레임워크에 넘겨 생산성을 향상시킴
45. 디버그 or 디버깅 : 컴퓨터 프로그램의 논리적인 오류를 찾아내는 과정

46. 회선(서킷) 교환 방식 VS 패킷 교환 방식

- ① 회선(서킷) 교환 방식 : 물리적 전용선을 활용하여 데이터 전달 경로가 정해진 후, 동일 경로로만 전달됨
 - ㉠ 대역폭이 고정되고 안정적인 전송률을 확보할 수 있음
 - ㉡ 접속에 긴 시간이 소요되나, 일단 접속이 되고 나면 전송속도가 유지되며 전송지연이 거의 없음
 - ㉢ 전송 중 동일한 경로를 가지며, 연속적인 전송에 적합
 - ㉣ 통신을 원하는 두 지점을 교환기를 이용하여 물리적으로 접속시키는 방식
 - ㉤ 속도나 코드의 변환이 불가능
 - ㉦ 오류제어나 흐름제어는 사용자에게 의해 수행되어야 함
- ② 패킷 교환 방식 : 고정된 경로를 설정하지 않고 데이터를 전송하는 동안만 작은 블록의 패킷으로 분할하여 전송
 - ㉠ 안정성이 낮고, 패킷 분해와 결합 때 지연시간이 발생 (대량의 데이터 전송 시 전송지연 발생)
 - ㉡ 메시지 교환방식의 단점 보완(응답시간 개선)
 - ㉢ 회선을 공유하므로 회선 이용률이 높고, 데이터 전송에 적합

47. DBMS 특징 : 데이터 일관성, 데이터 회복성, 데이터 보안성, 데이터 효율성, 데이터 무결성

데이터 무결성 : 부적절한 자료 입력으로 동일한 내용에 서로 다른 데이터의 저장을 허용하지 않는 성질

- ① 참조 무결성 : 외래키 값은 Null이거나 참조 릴레이션의 기본키 값과 동일해야 함
자식 테이블의 외래키는 반드시 부모 테이블의 기본키여야 함을 의미
- ② 개체 무결성 : 기본키를 구성하는 어떠한 속성 값도 Null 값이나 중복 값을 가질 수 없음
- ③ 도메인 무결성 : 각 속성 값은 반드시 미리 정의된 도메인 범위에 속한 값이어야 함
- ④ Null 무결성 : 릴레이션의 특정 속성 값이 Null이 될 수 없음
- ⑤ Unique 무결성 : 릴레이션의 특정 속성 값이 중복되어서는 안됨

48. ① GUI : 명령어를 직접 입력하지 않고, 마우스로 화면을 클릭하여 컴퓨터를 제어하는 방식 → 그래픽, 마우스 기반
- ② WUI : 웹 페이지를 열람하고 조작하는 인터페이스
- ③ CLI : 사용자가 컴퓨터 자판 등을 이용해 컴퓨터에게 직접 명령을 내리는 방식 → 텍스트, 키보드 기반 (MS-DOS)
- ④ TUI : CLI와 구분하기 위해 GUI 발명 이후 만들어진 용어
- ⑤ NUI : 오브젝트(신체) 기반 → 인간의 말과 행동 등 감각으로 기기 조작
- ⑥ OUI : 오브젝트(책, 캔, 모든 물질) 기반 → 입출력이 동시에 이루어짐

49. AJAX : HTML만으로 어려운 작업을 웹 페이지에서 구현하여 이용자가 웹 페이지와 자유롭게 상호작용할 수 있도록!
비슷한 기능의 Active X나 Flash 등에 비해 가볍고 속도가 빨라 차세대 웹 기술로 각광받고 있음
자바스크립트를 사용한 비동기 통신기술로, 클라이언트와 서버 간에 XML 데이터를 주고받는 기술

50. SGML : 텍스트 내에 표현하고자 하는 모양이나 구조, 의미 부여를 태그 형식으로 표기한 마크업 언어
표현의 우수성에도 불구하고 너무 복잡하고 어려워 일반화되진 못함

51. OLAP : 이용자가 다양한 각도에서 직접 대화식으로 정보를 분석해서 문제점 및 해결책을 찾는 분석형 애플리케이션
OLAP 툴 : 온라인 검색을 지원하는 데이터 웨어하우스 지원 도구. **대규모 연산이 필요한 질의를 고속으로 지원함**
데이터 큐브 : 데이터를 추상화하고 차원별로 우리가 상상하지 못하는 데이터를 효과적으로 운용토록 지원
데이터 웨어하우스 : DB로부터 정제되어 추출된 데이터를 다차원 분석 도구로 분석하여 의사결정에 필요한 자료 얻음

52. 형상 관리 : SW 개발과정에서 SW의 변경사항을 관리하기 위해 개발된 일련의 활동 (SW 개발의 전 단계에 적용)
- ① 형상식별 : 형상관리 대상을 식별하여 이름과 관리번호를 부여하고, 베이스라인의 기준을 정하는 활동
 - ② 변경제어 : 식별된 형상항목의 변경요구를 검토/승인 → 현재 베이스라인에 잘 반영될 수 있도록 조정
 - ③ 형상상태 보고 : 베이스라인의 현재 상태 및 변경 항목들이 제대로 반영되는지 여부를 보고하는 절차
 - ④ 형상검사 : 베이스라인의 무결성을 평가하기 위해 확인/검증 과정을 통해 공식적으로 승인하는 작업

53. 애플리케이션 통합 테스트 수행 방법 : 비점증적 통합방식(빅뱅 방식), 점증적 통합방식 [하스 상클드]
- ① 비점증적 통합 방식 : 모든 프로그램을 대상으로 한꺼번에 테스트 → 오류발견 및 장애위치 파악 어려움 (소규모)
 - ② 점증적 통합 방식 : 오류 수정이 용이하며, 인터페이스와 연관된 오류를 완전히 테스트할 가능성이 높음
 - ㉠ 하향식 통합 : 메인제어 모듈(컴포넌트)로부터 아래 방향으로 이동하면서 하향식으로 통합하며 테스트 더미모듈인 스텝이 필요하며, 깊이-우선 방식(수직) or 너비-우선 방식(수평)으로 통합됨
 - ㉡ 상향식 통합 : 최하위 레벨의 모듈 또는 컴포넌트로부터 위쪽 방향으로 이동하면서 상향식으로 통합하며 테스트 클러스터와 더미모듈인 드라이버가 필요
 - ㉢ 혼합식 통합 : 상위 레벨은 하향식 통합, 하위 레벨은 상향식 통합으로 진행하는 최적의 테스트

54. 개념 데이터 모델 : 현실세계에 존재하는 개체를 사람이 이해할 수 있는 정보구조로 표현
- 개체-관계(E-R) 모델 : 현실세계에 존재하는 데이터와 그들 간의 관계를 사람이 이해할 수 있는 형태로 표현하는 모델
- ① 개체(Entity) : DB에 표현하려고 하는 현실세계의 대상체
 - ② 속성(Attribute) : 개체의 성질, 분류, 식별, 수량, 상태 등 데이터의 가장 작은 논리적 단위
 - ③ 관계(Relationship) : 개체 간의 관계 또는 속성 간의 논리적인 연결

논리 데이터 모델 : 목표 DBMS에 맞추어 논리적 모델로 설계, 정규화(Normalization) 과정 수행

물리 데이터 모델 : 논리 데이터모델을 사용하고자 하는 각 DBMS의 특성을 고려하여 물리 데이터모델로 변환

- ① 엔티티 → 테이블
- ② 속성 → 컬럼
- ③ 주식별자 → 기본키
- ④ 보조 식별자 → 대체키
- ⑤ 관계 → 외래키

55. 배치 프로그램 : 사용자와의 상호작용 없이 일련의 작업들을 정기적으로 반복 수행하거나 정해진 규칙에 따라 일괄처리
- ①대용량 데이터, ②자동화, ③견고함, ④안정성, ⑤성능

56. HDLC : ISO에서 표준화한 대표적인 데이터 통신 제어 절차로, 비트 프레임 동기 방식의 프로토콜(FCFS 있음)
- FCFS : 프레임 오류 검사 부호

구분	개념
스토리보드	디자이너와 개발자가 최종적으로 참고하는, 서비스 구축을 위한 대부분의 정보가 수록돼있는 설계 산출물
와이어프레임	페이지에 대한 개략적인 레이아웃이나 서비스의 간략한 흐름에 대한 뼈대를 설계하기 위한 설계 작업
프로토타입	위와 같은 정적 화면에 동적효과를 적용하여 실제 구현된 것처럼 테스트 (동적 형태의 모형)
목업	와이어프레임보다 좀 더 실제화면과 유사하게 만든 정적 형태의 모형
유스케이스	사용자가 원하는 목표를 달성하기 위해 수행할 내용 기술

58. Windows의 4가지 특징 : GUI, 선점형 멀티태스킹, 플러그 앤 플레이(PnP), OLE
- ① GUI : 마우스의 아이콘을 이용하여 SW 실행
 - ② 선점형 멀티태스킹 : 우선순위가 높은 다른 프로세스가 할당된 CPU를 강제로 빼앗음 (ex. 시분할처리)
 - ③ 플러그 앤 플레이(PnP) : 컴퓨터에 주변기기를 추가할 때, 설치만 하면 그대로 사용할 수 있도록 도움
 - ④ OLE : 개체를 다른 프로그램에 연결하거나 삽입하여 자료를 공유하는 방식

59. CI(지속적인 통합) : 통합 시 발생하는 여러 가지 문제점을 조기에 발견하고, 피드백 사이클을 짧게 하여 SW 개발의 품질과 생산성을 향상시키는 것

60. OTP : 로그인 할 때마다 그 세션에서만 사용할 수 있는 일회성 패스워드(해시)를 생성하는 보안시스템(SHA와 혼동 X)

61. ROLLUP(지점, 부서) : 지점을 기준으로 부서를 결합하여 집계값 산출 (2개의 지점과 3개의 부서가 있을 경우, 총 9행)
62. 시퀀스 다이어그램 : 시스템이나 객체들이 메시지를 주고받으며 시간의 흐름에 따라 상호작용하는 과정을 표현
63. 모듈 : 전체 프로그램의 기능 중 특정 기능을 처리할 수 있는 실행 코드
 모듈화 : SW성능 향상, 시스템 수정 및 재사용, 유지관리 등이 용이하도록 시스템 기능을 모듈 단위로 분해하는 활동
 공통 모듈 : 전체 프로그램의 기능 중 공통적으로 사용할 수 있는 모듈 (정확성, 명확성, 완전성, 일관성, 추적성)
64. 인덱스 : 자주 사용되는 컬럼 값을 빠르게 검색할 수 있도록 표의 형식으로 구성된 보조적인 데이터 구조
 ① B 트리 인덱스 : 루트노드에서 하위노드로 키 값의 크기를 비교하며 단말노드에서 찾으려는 데이터 검색
 ② 비트맵 인덱스 : 인덱스 컬럼의 데이터를 비트 값인 0, 1로 변환하여 인덱스 키로 사용 (데이터가 비트로 구성)
 ③ 함수기반 인덱스 : 컬럼의 값 대신 컬럼에 특정 함수나 수식을 적용하여 산출된 값을 사용
 ④ 해시 인덱스 : 해시함수로 직접 데이터에 접근 (특정키 값에 대한 검색방법 중 가장 빠름)
 ⑤ 클러스터드 인덱스 : 인덱스 키의 순서에 따라 데이터가 물리적으로 정렬되어 저장 (한 테이블에 하나만 생성 O)
 ⑥ Non-클러스터드 인덱스 : 인덱스 키 값만 정렬하는 방식이며, 한 테이블에 여러 개의 인덱스를 생성할 수 있음
65. QoS(Quality of Service) : 서비스의 질을 의미하며, 지연 시간이나 데이터 손실률 등을 보증하기 위한 서비스 규격
 실시간 프로그램은 QoS를 사용함으로써 네트워크 대역폭을 가장 효율적으로 사용 가능
66. 절차형 SQL : 프로그래밍 언어처럼 연속적인 실행이나 조건에 따른 분기, 반복 등의 제어가 가능한 SQL
 ① 프로시저 : 절차형 SQL을 활용하여 특정 기능을 수행(DB조작, 트랜잭션 처리)할 수 있는 트랜잭션 언어
 ② 사용자 정의 함수 : 일련의 작업을 연속적으로 처리하며, Return을 사용하여 결과를 단일 값으로 반환
 ③ 트리거 : 삽입·수정·삭제 등 데이터 변경 이벤트가 발생할 때마다 DBMS에서 자동적으로 작업 실행되는 프로그램
 → 프 사 트 / 특 연 단 자
67. 시큐어 코딩 : 안전한 SW를 개발하여 각종 보안위협으로부터 예방하고 보안취약점을 사전에 제거하는 활동
 ① SQL 삽입 : 사용자의 입력값 등 외부 입력값이 SQL 쿼리에 삽입되어 공격(유효성 검증 하지 않은 경우)
 ② 크로스사이트스크립트(XSS) : 웹에 악의적인 스크립트를 포함시켜, 클라이언트 웹에서 악성스크립트 실행
 ③ 크로스사이트 요청위조(CSRF) : 쿼리를 임의로 조작하여, 사용자가 자신의 의지와는 무관하게 공격자 의도를 행함
 ④ 위험한 형식 파일 업로드 : 서버 측에서 실행될 수 있는 스크립트 파일이 업로드 가능한 보안약점
 ⑤ 디렉토리 경로조작 : 데이터 입출력 경로를 조작하여 서버 자원을 수정/삭제할 수 있는 보안약점
 ⑥ 운영체제 명령어 삽입 : 외부 입력값을 통해 시스템 명령어의 실행을 유도함으로써 시스템 장애를 유발
 ⑦ 신뢰되지 않는 URL 주소로 자동 접속연결 : 외부 입력값이 링크 표현에 사용되어 악의적인 사이트로 리다이렉트
 ⑧ 적절한 인증 없는 중요기능 허용 : 적절한 인증 없이 중요정보를 열람 또는 변경 가능한 보안 취약점
 ⑨ 부적절한 인가 : 부적절한 접근제어로 외부 입력값이 포함된 문자열이 서버 자원에 접근되는 보안약점
 ⑩ 중요한 자원에 대한 잘못된 권한 설정 : 중요한 자원에 부적절한 접근권한이 부여되어 중요정보가 노출되는 약점
 ⑪ 취약한 암호화 알고리즘 사용 : 기밀성이 취약한 암호화 알고리즘을 사용하여 정보가 노출되는 보안약점
 ⑫ 사용자 중요정보 평문 저장 및 전송 : 중요 정보를 평문으로 송수신할 경우 스니핑을 통해 중요 정보가 노출
 ⑬ 하드코딩된 비밀번호 : 프로그램 코드내부에 하드코딩(데이터를 코드내부에 직접 입력)된 패스워드 포함
 ⑭ TOCTOU 경쟁조건 : 하나의 자원에 대하여 동시에 검사시점과 사용시점이 달라 생기는 보안약점
 ⑮ 제대로 제어되지 않은 재귀 : 적절한 제어문 처리가 되어있지 않은 재귀함수에서 무한루프가 발생
 ⑯ 널 포인터 역참조 : Null로 설정된 변수의 주소 값을 참조했을 때 발생하는 보안약점
 ⑰ 부적절한 자원 해제 : 사용되는 리소스 자원을 적절하게 해제하지 않아서 리소스 자원의 누수가 발생하는 약점
 ⑱ 해제된 자원 사용 : 해제한 메모리를 참조하여 예상치 못한 값 또는 코드를 실행하게 되는 보안약점
 ⑲ 초기화되지 않은 변수 사용 : 초기화되지 않은 변수를 사용하여 의도치 않은 결과가 발생하는 보안약점
 ⑳ 취약한 API 사용 : gets(), strcat(), strcpy(), strncpy(), sprintf() 등 취약한 API 사용 시 발생하는 보안약점

68. 결함관리 활동 프로세스 : 에러발견 → 에러등록 → 에러분석 → 결함**확정** → 결함**할당** → 결함조치 → 검토 및 승인
69. SQL : 관계형 DB의 조작과 관리에 사용되는 DB 프로그래밍 언어로, 관계대수(절차)와 관계해석(비절차)을 기초로 함
70. API : 응용 프로그램 개발 시 라이브러리(함수들을 모아놓은 것)를 이용할 수 있도록 규칙들을 정의해놓은 인터페이스
71. 애플리케이션 성능측정 지표 4가지 : 처리량, 응답시간, 경과시간, 자원사용률 (처응경자)
72. 웹 서비스 : UDDI나 SOAP, WSDL 등의 표준 기술을 사용하여 네트워크에 연결된 다른 분산 컴퓨팅을 지원하는 SW
- ① UDDI : **WSDL**을 등록하여 서비스와 서비스 제공자를 검색하고 접근하는데 사용되는 XML 기반의 웹 서비스 요소
 - ② SOAP : 네트워크상에서 **HTTP/HTTPS, SMTP 등을 이용하여 XML을 교환**하기 위한 통신 규약 (Header, Body)
 - ③ WSDL : **웹 서비스에 대한 상세 정보**를 표준적인 방법으로 기술하고 게시하기 위한 기술 (XML 형식으로 구현)
73. ① SELECT 함수명 FROM 테이블명;
 ② DELETE FROM 테이블명 WHERE 속성명 = 함수명;
 ③ INSERT **INTO** 테이블명 **VALUES**(함수명); ex) INSERT INTO 회원 VALUES(004, '장석현', '남', '사무');
 ④ UPDATE 테이블명 **SET** 속성명 = 함수명; ex) UPDATE 상품목록 SET 생산지='Korea', 제조가=100000;
74. 운영체제가 자원들을 관리하는 과정 : 자원상태 파악 → 분배정책 수립 → 자원 배당 → 자원 회수
75. 방식에 따른 버전 관리 도구
- ① 공유폴더 방식 : 개발완료 파일은 약속된 위치의 공유 폴더에 복사하고, 담당자가 에러와 정상동작 유무를 확인
 - ⓐ RCS : CVS와 달리 수정을 한 사람만으로 제한하여 동시에 수정할 수 없도록 파일을 잠그는 방식
 - ② 클라이언트/서버 방식 : 중앙 집중형 서버 저장소를 두고 클라이언트가 접속해서 버전 관리를 실행
 - ⓐ CVS : 서버와 클라이언트로 구성되어 다수의 인원이 동시에 접근 가능하고, 클라이언트가 이클립스에 내장
 - ⓑ SVN : GNU의 버전관리 시스템으로, CVS의 장점은 이어받고 단점은 개선한 방식. 사실상 업계표준으로 사용
 - ③ 분산 저장소 방식 : 로컬 저장소와 원격 저장소로 분산된 구조로, 원격 저장소와 로컬 저장소에 함께 저장
 - ⓐ Bitkeeper : SVN과 비슷한 중앙통제 방식의 분산 저장소 방식이며, 대규모 프로젝트에서 빠른 속도를 냄
 - ⓑ Git : Bitkeeper를 대체하며, 속도에 중점을 둔 분산형 버전 관리 시스템(DVCS). 대형 프로젝트에서 효과적인 커밋은 로컬저장소에서 이루어지고 push로 원격저장소에 반영. 중앙서버에 의존 않는 완전한 저장소 형태
76. 데이터 분석 함수
- ① 집계 함수 : COUNT, SUM, AVG, MAX, MIN, VARIAN, STDDEV
 - ② 그룹 함수 : ROLLUP, CUBE, GROUPING SETS
 - ③ 윈도우 함수 : PARTITION BY, ORDER BY, RANK, DENSE_RANK, ROW_NUMBER
77. 관계대수 中 순수 관계 연산자(①~④)와 일반 집합 연산자(⑤~⑧)
- ① SELECT : 조건을 만족하는 **튜플**들을 검색. 시그마를 이용 (**행**, 수평적 연산)
 - ② PROJECT : 조건을 만족하는 **속성**들을 검색. 파이를 이용 (**열**, 수직적 연산) / 중복 발생하면 중복 제거
 - ③ JOIN : 2개의 릴레이션 A와 B에서 공통된 속성을 연결하는 것
 - ④ DIVISION : R의 속성이 S의 속성 값을 모두 가진 튜플 중 S가 가진 속성을 제외한 속성만을 구함
 - ⑤ 합집합 : 두 릴레이션의 합집합을 구하되, 중복되는 튜플은 제거 (차수와 도메인이 같아야 연산 수행 가능)
 - ⑥ 교집합 : 두 릴레이션의 교집합을 구하는 연산 (차수와 도메인이 같아야 연산 수행 가능)
 - ⑦ 차집합 : 두 릴레이션의 차집합을 구하는 연산 (차수와 도메인이 같아야 연산 수행 가능)
 - ⑧ 카티션 프로덕트 : 두 릴레이션의 교차곱을 구하는 연산 (각각 3x2의 값을 가지고 있다면 행=3x3=9 / 열=2x2=4)

78. 파티션 : 대용량의 테이블이나 인덱스를 관리하기 쉬운 **논리적** 단위로 분리하는 것 (범위분할[열], 해시분할, 조합분할)
79. JDBC : SQL문을 수행할 때 다양한 종류의 DB에 접속할 수 있도록 사용되는 자바 표준 API
ODBC : DB에 접속하기 위한 표준 개방형 API로, 마이크로소프트사에서 출시함
80. 스프링 프레임워크 : 자바 플랫폼을 위한 오픈소스 애플리케이션 프레임워크. 전자정부 표준 프레임워크의 기반 기술
Django : Python을 기반으로 만들어진 프레임워크
Node.JS : 자바스크립트를 기반으로 만들어진 프레임워크
Rails : Ruby를 기반으로 만들어진 프레임워크
81. 뷰 : 허용된 자료만을 제한적으로 보여주기 위해 테이블로부터 유도된 가상테이블 → 논리적(O) 물리적(X)
82. 커널 : UNIX에서 가장 핵심적인 부분이며, 자원을 관리하고 시스템을 제어함(프로세스, 기억장치, 파일, 입출력 관리)
셸 : 시스템과 사용자 간 인터페이스를 담당하며, 사용자의 명령을 해석하는 명령어 해석기
83. 미들웨어 : 프로그램들 사이에서 매개 역할을 하거나 프레임워크 역할을 하는 중간 계층 프로그램 (중간에 위치함)
운영체제가 제공하는 서비스 외에 추가적인 서비스를 제공하고, 표준화된 인터페이스를 통해 일관성 보장
84. 페이징 기법 : 주기억장치를 동일한 크기로 나눈 후, 나뉜 페이지를 동일하게 주기억장치의 영역에 적재시켜 실행
세그먼테이션 기법 : 다양한 크기의 논리적인 단위로 나눈 후, 주기억장치에 적재시켜 실행 [외부단편화O 내부단편화X]
워킹셋 : 프로세스가 일정 시간동안 자주 참조하는 페이지들의 집합으로, 자주 참조되는 워킹셋을 주기억장치에 상주
스래싱 : 프로세스의 처리시간보다 페이지 교체에 소요되는 시간이 더 많아지는 현상 (CPU 이용률 저하)
85. 테스트 커버리지의 테스트 목적
- ① 회복 테스트 : 시스템에 고의로 실패를 유도하고 시스템이 정상적으로 복구하는지 테스트
 - ② 안전 테스트 : 불법적인 SW가 접근하지 못하도록 소스코드 내의 보안적인 결함을 미리 점검하는 테스트
 - ③ 강도 테스트 : 시스템에 과다 정보량을 부과하여 과부하 시에도 시스템이 정상적으로 작동하는지 테스트
 - ④ 성능 테스트 : 사용자의 이벤트에 시스템이 응답하는 시간, 반응 속도, 처리하는 업무량 등을 테스트
 - ⑤ 구조 테스트 : 시스템의 내부 논리 경로, 소스코드의 복잡도를 평가하는 테스트
 - ⑥ 회귀 테스트 : 변경 또는 수정된 코드에 대하여 새로운 결함 발견 여부를 평가하는 테스트
 - ⑦ 병행 테스트 : 변경된 시스템과 기존 시스템에 동일한 데이터를 입력 후 결과를 비교하는 테스트
86. 사용자 인터페이스(UI)의 기본 원칙 4가지 : 직관성, 유효성, 유연성, 학습성
87. 프로세스 상태 전이도
- ① 준비 상태(Ready) : 프로세스가 프로세서를 할당받기 위해 기다리는 상태
 - ② 실행 상태(Run) : 프로세스가 프로세서를 할당받아 실행중인 상태
 - ③ 대기 상태(Wait) : 입출력처리가 완료될 때까지 대기하고 있는 상태
 - ④ 타이머 런 아웃 : 할당시간을 초과하면 CPU를 다른 프로세스에게 양도
 - ⑤ 디스패치 : 프로세서를 할당받아 실행 상태로 전이
 - ⑥ 블록(Block) : 다른 작업을 위해 스스로 프로세스를 양도
88. TCP/IP 계층
- | | |
|--------------------|---------------|
| ① 물데 : 네트워크 액세스 계층 | ② 네 : 인터넷 계층 |
| ③ 전 : 전송 계층 | ④ 세표응 : 응용 계층 |

89. DBMS의 필수 기능 3가지 + 트랜잭션 제어어 → 정조제 DMC CADrop SIDU

- ① 데이터 **정의어**(DDL) : DB 객체를 생성(CREATE), 수정(ALTER), 삭제(DROP) / DB 관리자나 설계자가 사용
- ② 데이터 **조작어**(DML) : 검색(SELECT), 삽입(INSERT), 삭제(DELETE), 수정(UPDATE)
- ③ 데이터 **제어어**(DCL) : 역할·권한(ROLE), 권한 및 롤 부여(GRANT), 권한 및 롤 회수(REVOKE)
- ④ 트랜잭션 제어어(TCL) : 트랜잭션의 DML 작업단위 제어하는 명령어(COMMIT, ROLLBACK, SAVEPOINT)

90. DB 정의 (필기 내용) → 운공통저

- ① **운영 데이터** : 존재 가치가 확실하고 없어서는 안 될 반드시 필요한 데이터
- ② **공용 데이터** : 시스템들이 공동으로 소유하고 유지하는 데이터
- ③ **통합된 데이터** : 검색의 효율성을 위해 중복이 최소화된 데이터
- ④ **저장된 데이터** : 저장 매체에 저장된 데이터

91. EAI : 기업 내 각종 애플리케이션 및 플랫폼 간의 정보 전달, 연계, 통합 등 상호연동이 가능케하는 솔루션 (포허메하)

ESB : EAI와 유사하지만 애플리케이션보다는 서비스 중심의 통합을 지향하는 아키텍처

92. 페이퍼 프로토타이핑 : 아날로그적인 방법 (키노트, 스케치, 그림 등을 이용하여 손으로 직접 작성)

디지털 프로토타이핑 : 디지털적인 방법 (파워포인트, 아크로벳, 비지오 등과 같은 프로그램을 사용)

93. 교착상태의 4가지 필요조건 (필기 내용) : ①상호배제, ②점유 및 대기, ③비선점, ④환형대기

94. 기본키 : PRIMARY KEY

외래키 : FOREIGN KEY (속성명) REFERENCES

95. 클러스터 : 액세스 효율의 향상을 위해 동일한 성격의 데이터를 동일한 데이터 블록에 저장하는 **물리적** 저장 방법

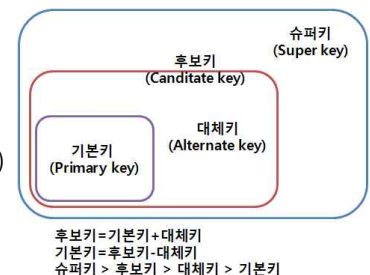
96. 분산 데이터베이스의 목표(투명성) → 필기 내용

- ① 위치 투명성 : 접근하려는 DB의 실제 위치를 알 필요 없이 단지 DB의 논리적인 명칭만으로 접근 가능
- ② 중복 투명성 : 동일한 데이터가 여러 곳에 중복되어 있더라도 사용자는 마치 하나의 데이터만 존재하는 것처럼
- ③ 장애 투명성 : 트랜잭션, DBMS, 네트워크, 컴퓨터 **장애**에도 불구하고 트랜잭션은 정확하게 수행
- ④ 병행 투명성 : 다수의 트랜잭션들이 **동시에 실행**되더라도 트랜잭션들의 수행 결과는 서로 영향 X (**일관성** 유지)

97. 정형 분석 : 정형화된 언어를 이용해 요구사항을 수학적 기호로 표현한 후 이를 분석 (요구사항 분석의 마지막 단계)

98. 키의 종류

- ① 슈퍼키 : 속성들의 집합으로 구성된 키(유일성 만족, 최소성 불만족)
- ② 후보키 : 기본키로 사용할 수 있는 키(유일성 만족, 최소성 만족)
- ③ 기본키 : 후보키 중 선택한 키(중복되어서는 안되며, Null값 가질 수 없음)
- ④ 대체키 : 기본키를 제외한 나머지 후보키
- ⑤ 외래키 : 다른 릴레이션의 기본키를 참조하는 속성



99. UI 흐름 설계에서 사용하는 컨트롤

- ① 텍스트 박스 : 사용자가 텍스트 데이터를 입력·수정할 수 있는 상자
- ② 체크 박스 : 여러 항목 중 동시에 여러 개 선택 가능
- ③ 라디오 박스 : 여러 항목 중 하나만 선택 가능
- ④ 콤보 박스 : 이미 지정된 목록 상자에서 선택 가능 (새로운 내용 입력 가능)
- ⑤ 리스트 박스 : 이미 지정된 목록 상자에서 선택 가능 (새로운 내용 입력 불가능)

100. 에이징 : 프로세스가 자원을 기다리고 있는 시간에 비례하여 우선순위를 부여함으로써 무기한 대기하는 문제를 방지

101. 4+1 View : 고객의 요구사항을 정리해놓은 시나리오를 4개의 관점에서 바라보는 SW적인 관점을 의미
- ① **사용사례 뷰**(유스케이스 뷰) : 외부사용자 관점에서 **사용사례들** 간의 관계 정의
 - ② **논리 뷰** : 시스템의 **논리적인** 구조 및 행위를 클래스 인터페이스·협력관계로 정의
 - ③ **구현 뷰** : 독립적으로 실행되는 **컴포넌트**와 이들 간 관계를 정의
 - ④ **프로세스 뷰** : 시스템의 병렬처리 및 동기화 처리를 위한 **스레드**와 **프로세스**를 정의
 - ⑤ **배치 뷰** : 실행되는 시스템 **HW**와 **SW** 관계를 정의
- 사 논 구 프 배 / 사 논 컴 프 HS
102. 데이터베이스 암호화 기법
- ① 플러그인 방식 : **DB 서버**에서 플러그인을 장착하여 암호화 수행
 - ② API 호출방식 : **애플리케이션**에서 암호화를 수행
 - ③ 하이브리드 방식 : 플러그인 방식과 API 방식을 혼합
103. DNS : 도메인 이름을 컴퓨터가 이해할 수 있는 IP 주소로 변환하는 역할을 하는 시스템
DHCP : 네트워크상에서 동적으로 IP 주소 및 기타 구성정보를 부여 및 관리하는 프로토콜. IP를 자동으로 할당
FTP : 원격 파일 전송 프로토콜
104. JVM : 자바 바이트코드를 실행할 수 있는 주체로, 운영체제 종류와 무관하게 동일하게 동작하도록 보장하는 가상머신
105. 트랜잭션의 4가지 특성 (ACID 특성)
- ① 원자성(Atomicity) : 트랜잭션 연산은 DB 모두 반영되든지 혹은 전혀 반영되지 않아야 함
 - ② 일관성(Consistency) : 트랜잭션이 실행을 성공적으로 완료하면 언제나 일관성 있는 DB 상태로 변환
 - ③ 독립성(Isolation) : 어느 하나의 트랜잭션 실행 도중 다른 트랜잭션의 연산이 끼어들 수 없음
 - ④ 영속성(Durability) : 성공적으로 완료된 트랜잭션 결과는 시스템이 고장이 나도 영구적으로 반영
106. 웹 서버 : HTML 파일·이미지·그림·자바스크립트 등 정적인 콘텐츠 제공
웹 애플리케이션 서버(WAS) : DB접속·외부 시스템 연동 등 동적인 콘텐츠 제공
107. 가상화 : 물리적인 리소스들을 하나로 보이게 하거나, 반대로 하나의 물리적인 리소스를 여러 개로 보이게 하는 기술
클라우드 : 인터넷 기반에서 구동되는 컴퓨팅 기술
- ① SaaS : SW를 웹에서 쓸 수 있게 함 → SaaS의 S : 소프트웨어
 - ② IaaS : 서버, 스토리지 등의 IT 인프라 장비를 빌려줌 → IaaS의 I : 인프라
 - ③ PaaS : SW 개발에 필요한 플랫폼(개발환경)을 빌려줌 → PaaS의 P : 플랫폼
108. RAID : 여러 개의 하드디스크에 일부 중복된 데이터를 나눠서 저장하는 기술
109. 전문가 판단 기법 : 2명 이상(소수)의 전문가에게 의뢰하여 비용을 산정하는 방식
델파이 기법 : 1명의 조정자와 다수의 전문가 의견을 종합하여 비용을 산정하는 방식 (전문가 판단기법 보완)
110. 반정규화 : 전체 성능향상을 위해 테이블을 통합하고 중복을 허용하며 테이블을 합치는 의도적 정규화 원칙 위배 행위

111.

SQL	SQL*Plus
DB 조작 및 DB와 통신하는 표준 언어	SQL 명령어를 서버에 전송하는 도구 (데이터 조작 X)
ANSI 표준	Oracle사 제공 도구
데이터와 테이블에 대한 정의 O	데이터에 대한 어떠한 정의 X (데이터 조작 X)
SQL Buffer 사용 O (마지막 명령문 저장)	SQL Buffer 사용 X
여러 행 입력 O	여러 행 입력 X (한 줄 실행)
명령어 실행 시 종료문자 ';' 사용 O	명령어 실행 시 종료문자 ';' 사용 X
키워드 축약 X	키워드 축약 O
연결문자 '-' 불필요	연결문자 '-' 필요

112. 옵티마이저 : 사용자가 질의한 SQL문에 대해 가장 빠르고 효율적으로 수행할 최적의 실행방법을 결정하는 역할 수행

113. EAI 방식

- ① Point-to-Point(PPP) : 두 대의 컴퓨터가 중간에 미들웨어 없이 직렬 인터페이스를 이용하는 프로토콜(기본방식)
- ② Hub & Spoke : 단일 접점인 허브 시스템을 통해 데이터를 전송하는 중앙 집중형 방식
- ③ Message Bus : 중간에 미들웨어를 두어 처리하는 방식이며, 확장성과 대용량 처리가 우수함
- ④ Hybrid : 그룹 내에는 Hub & Spoke 방식, 그룹 간에는 Message Bus 방식이며, 데이터 병목현상을 최소화함

114. 웹 프로그래밍 언어

- ① HTML : 기본적인 프로그래밍 언어로, 하이퍼텍스트를 작성하기 위해 개발
- ② DHTML : 동적 HTML이며, 애니메이션 강화
- ③ XML : HTML 문법이 호환되지 않는 문제 해결을 위해 획기적으로 개선하여 개발된 다목적 마크업 언어
- ④ WML : 무선 접속을 통해 PDA나 휴대전화 같은 이동 단말기에 표시될 수 있도록 해 주는 언어
- ⑤ VRML : 웹상에서 3차원 가상공간을 표현하기 위한 언어
- ⑥ JSP : HTML 내에 자바 코드를 삽입하여 웹 서버에서 동적으로 웹 페이지를 생성하는 서버 측 스크립트
- ⑦ 자바스크립트 : JSP와는 반대로 클라이언트 측에서 실행되는 스크립트
- ⑧ 애플릿 : HTML 문서 내에 포함될 수 있는 자바 프로그램
- ⑨ ASP : 서버 측에서 동적으로 처리되는 Microsoft사 언어 (리눅스 환경 = PHP)

115. 백그라운드 프로그램 : 우선순위가 낮은 프로그램. 순위가 높은 프로그램이 우선 처리된 중간 중간의 여유시간에 수행

116. ① 멀티캐스트 : 전자메일 · 화상회의를 위해, 둘 이상의 다른 수신자들에게 동시에 정보를 전송하는 방식 (1대다)

② 유니캐스트 : 특정 한 사람에게만 정보를 전송하는 방식 (1대1)

③ 브로드캐스트 : 불특정 다수에게 정보를 전송하는 방식

④ 애니캐스트 : IPv6에게 브로드캐스트가 없어지고 생김. 그룹 내에서 가장 가까운 호스트에게만 전송 (1대1)

117. IPv6 : IPv4의 주소공간을 4배 확장한 128비트 인터넷 주소체계 (유니캐스트, 멀티캐스트, 애니캐스트)

IPv4의 주소고갈 문제를 해결하기 위해 등장했으며, 128비트를 콜론(:)으로 8부분으로 구분함

각 구분은 16진수로 표현되며, 보안 기능을 강화하였고, 서비스에 따라 각기 다른 대역폭을 지원함

118. 계층 별 장비 및 특징 (허리 브스 라 계)

- ① 허브(1계층) : 여러 대의 컴퓨터를 연결하여 하나의 네트워크로 수신된 정보를 여러 대의 컴퓨터로 송신
- ② 리피터(1계층) : 장거리 데이터 전송에서 신호를 증폭하는 장치
- ③ 브리지(2계층) : 두 시스템 혹은 LAN을 연결하는 네트워킹 장치. SW 기반이므로 속도가 느림
- ④ 스위치(2계층) : 두 시스템 혹은 LAN을 연결하는 네트워킹 장치. HW 기반이므로 속도가 빠름
- ⑤ 라우터(3계층) : 네트워크 계층에서 연동하여 경로를 설정하고 가장 빠르고 신뢰성 있는 경로로 데이터를 전달
- ⑥ 게이트웨이(4계층) : 각 계층에서 프로토콜 구조가 전혀 다른 외부 네트워크와 접속하기 위한 프로토콜 변환 장비

119. TCP : 연결형(신뢰성 ↑, 속도 ↓) / 다중화, 순서제어, 오류제어, 흐름제어 / 유니캐스트형 / 양방향성

UDP : 비연결형(속도 ↑, 신뢰성 ↓) / 실시간 전송 유리 / 멀티캐스트형

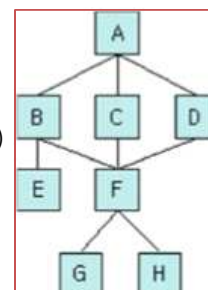
120. ① 공유도(Fan-In) : 어떤 모듈을 제어하는 상위 모듈의 개수

② 제어도(Fan-Out) : 어떤 모듈에 의해 제어되는 하위 모듈의 개수
시스템 복잡도 최적화 : Fan-In은 높게, Fan-Out은 낮게 (공높제낮)

기출) 다음은 프로그램 구조를 나타낸다.

모듈 F에서의 Fan-In과 Fan-Out의 수는 얼마인가?

정답) Fan-In : 3 / Fan-Out : 2



121. 병행 제어 : 여러 트랜잭션 간의 간섭으로 문제가 발생하지 않도록 트랜잭션의 실행 순서를 제어하는 기법
- ① 연기 갱신 : 성공적으로 종료될 때까지 갱신을 연기 (DB에 즉시 반영하지 않고 로그 파일에만 기록)
부분완료 시점에 저장된 로그를 사용하여, 변경내용을 실제 DB에 반영(장애 시 로그만 버림)
 - ② 즉시 갱신 : 트랜잭션 수행 도중 데이터에 변경이 생기면 즉시 DB에 해당 변경 사항을 반영하는 기법
변경된 모든 내용은 로그 형태로 보관 / redo와 undo 모두 수행 (장애 시 undo 수행)
 - ③ 그림자 페이지징 : DB를 일정 크기의 페이지로 나누고 각 페이지마다 복사하여 그림자페이지로 별도 보관
변경되는 내용은 원본 페이지에만 적용하고, 장애 발생 시 그림자페이지를 이용해 회복
 - ④ 검사점 : 실행 도중 주기적으로 변경 내용이나 시스템 상황 등에 관한 정보와 함께 검사점을 로그에 보관
로그를 이용한 회복 기법(redo와 undo를 수행하기 위해 로그 전체를 조사)을 피하기 위한 기법
→ redo나 undo를 적용해야 하므로 회복에 많은 시간이 소요되는 즉시 갱신을 보완하는 기법
장애 발생 시 가장 최근의 검사점으로부터 회복 작업을 수행하므로 회복시간 단축 가능
 - ⑤ 미디어 회복 : 전체 DB 내용을 일정 주기마다 다른 안전한 저장장치에 복사해두는 덤프를 이용
장애 발생 시 가장 최근에 복사해둔 덤프를 이용해 복구

122. CSS : 웹 문서의 전반적인 스타일을 미리 저장해둔 스타일시트(색깔, 글씨체와 같은 디자인요소를 관리)

-
- 방화벽(침입차단) : 외부로부터 허가받지 않은 불법적인 접근이나 해커의 공격으로부터 내부를 효과적으로 보호
외부로 나가는 패킷은 그대로 통과시키고, 내부로 들어오는 패킷은 인증된 패킷만 통과시킴
 - IDS(침입탐지) : 정보시스템의 보안을 위협하는 침입행위가 발생할 경우 이를 탐지, 적극 대응하기 위한 시스템
 - IPS(침입방지) : 방화벽과 IDS를 혼합시킨 시스템으로, 둘과 달리 침입경고 이전에 공격을 중단시키는 데 중점을 둠
 - DMZ(비무장지대) : 내부 네트워크와 외부 사이에 위치하며, 서버에 대한 부정접속을 방지하기 위한 침입차단 기능
내부/외부 공격으로부터 중요 데이터를 보호하거나, 서버의 서비스 중단을 방지할 수 O
 - DLP(데이터유출 방지) : 기업 내부자의 고의나 실수로 정보가 밖으로 새어나가는 것을 방지하는 시스템
 - 웹 방화벽 : 일반 방화벽이 탐지하지 못하는 SQL 삽입공격, XSS 등의 웹 기반 공격을 방어
 - Anti-DDoS : DDoS 차단 보안 솔루션
 - VPN(가상 사설망) : 공중망을 사설망처럼 구성하여 안전한 통신을 할 수 있도록 보장하는 네트워크 기술
 - NAC(네트워크 접근제어) : 내부 PC의 MAC 주소를 IP 관리 시스템에 등록한 후, 일관된 보안관리 기능 제공
 - UTM(통합 위협관리) : 침입차단 시스템, VPN 등 다양한 보안 솔루션 기능을 하나로 통합한 보안 솔루션
 - ESM(기업 보안관리) : 방화벽, 침입탐지 시스템, VPN 등의 보안 솔루션을 하나로 모은 통합 보안관리 시스템
이 기존 보안 시스템을 통합 + 관리하는 시스템으로, 보안관련 장비가 복잡화됨에 따라
이 기존 보안 솔루션을 중앙에서 하나의 콘솔로 관리하는 보안솔루션
 - 죽음의 핑 : 인터넷 프로토콜 허용범위 이상의 큰 패킷을 고의로 전송하여 발생한 서비스 거부 공격(DoS)
 - SYN Flooding : 대량의 SYN 패킷을 이용해서 타겟 서버의 서비스를 더 이상 사용할 수 없도록 만드는 기법
SYN을 넘치도록 공격하는 방식이며, TCP 3-way handshaking 문제점을 이용한 공격방식
서버에 다시 ACK 패킷을 보내야 연결이 되는데, 보내지 않으면 대기 상태가 됨
 - LAND 공격 : 출발지 주소나 포트를 임의로 변경하여, 출발지와 목적지 주소를 동일하게 하여 무한응답 발생

- **스머핑** : 엄청난 양의 데이터를 한 사이트에 집중적으로 보냄으로써 시스템 상태를 불능으로 만드는 공격기법
패킷을 홍수처럼 상대컴퓨터 시스템에 퍼붓는 방식 (ICMP : 서버와 게이트웨이 사이에서 메시지 제어)
- **TearDrop 공격** : 패킷 제어로직을 악용하여 시스템의 자원을 고갈시키는 공격
데이터의 송수신 과정에서 분할 순서를 알 수 있도록 Fragment Offset을 함께 전송하는데,
이 값을 변경시켜 수신측에서 패킷 재조립 시 과부하가 발생하도록 공격하는 기법
- **DDoS** : 감염된 대량의 숙주 컴퓨터를 이용해 특정 시스템을 마비시키는 사이버 공격(=분산 서비스 거부 공격)
일반 컴퓨터를 감염(좀비 PC)시켜, 공격 대상의 시스템에 대량의 패킷이 무차별로 보내지도록 조정
일반적으로 트로이 목마에 감염되어 여러 손상된 시스템(분산)이 서비스 거부 공격(DoS)을 일으킴
- **트로이목마** : 마치 유용한 프로그램인 것처럼 위장하여 사용자들로 하여금 설치를 유도하는 프로그램
- **피싱** : 개인 정보와 낚시의 합성어로, 개인 정보를 몰래 빼내는 것
- **큐싱** : QR코드와 피싱의 합성어로, QR코드를 이용한 해킹
- **스미싱** : SMS와 피싱의 합성어로, 문자메시지를 이용하여 피싱
- **스피어 피싱** : 조직 내에 신뢰할만한 발신인으로 위장하여, ID 및 PW 정보를 요구하는 피싱 공격 (사람들의 심리 이용)
- **APT** : 특정 기업이나 조직 NW에 침투해 활동 거점을 마련한 뒤, 때를 기다리면서 보안을 무력화시키고 정보를 수집
 - ① 내부자에게 악성코드가 포함된 이메일을 오랜 기간 동안 꾸준히 발송해, 1번이라도 클릭되길 기다림
 - ② 스텝스넷과 같이, 악성코드가 담긴 이동식 디스크(USB) 등으로 전파하는 형태
 - ③ 악성코드에 감염된 P2P 사이트에 접속하면 악성코드에 감염되는 형태
- **스텝스넷** : 원자력 발전소 같은 산업기반 시설에 사용되는 제어 시스템에 침투하여 오동작 유도 (독일 지멘스사)
- **SQL 주입 공격** : 웹 응용 프로그램에 강제로 SQL 쿼리를 변조, 삽입하여 내부 DB 서버의 데이터를 유출 및 변조
- **무작위 대입 공격(브루트 포스 공격)** : 조합 가능한 모든 경우의 수를 다 대입해보는 공격
- **제로데이 공격** : 보안패치가 발표되기 전에 해당 취약점을 이용해 이뤄지는 해킹이나 악성코드 공격
- **랜섬웨어** : 내부분서나 그림파일 등을 암호화해 열지 못하도록 만든 후, 복호화의 대가로 금전을 요구하는 악성 프로그램
- **트랩도어(백도어)** : 액세스 편의를 위해 시스템 설계자가 고의로 시스템 보안을 제거하여 만들어놓은 비밀통로
- **웜 바이러스** : 다른 프로그램을 감염시키지 않고, 자기 자신을 복제하면서 통신망 등을 통해서 널리 퍼짐
- **키로거 공격** : 컴퓨터 사용자의 키보드 움직임을 탐지해 중요한 정보를 몰래 빼 가는 해킹공격
- **스니핑** : 네트워크의 중간에서 남의 패킷 정보를 도청
- **스누핑** : 남의 정보를 엿탐하여 불법으로 가로채는 행위
- **스푸핑** : 승인받은 사용자인 것처럼 속이는 것
- **스위치 제밍** : 위조된 MAC주소를 지속적으로 네트워크로 흘려보내, 스위치 저장기능을 혼란시켜 더미허브처럼 작동
- **RFIC** : 극소형칩에 상품정보를 저장하고, 태그를 식별하여 정보를 처리하는 시스템. 바코드를 대체할 차세대 인식 기술
- **빅데이터** : 기존의 관리 방법이나 분석 체계로는 처리하기 어려운 막대한 양의 정형 또는 비정형 데이터 집합

공통 모듈 특징 5가지 : ①정확성, ②명확성, ③완전성, ④일관성, ⑤추적성

프레임워크 특성 4가지 : ①모듈화, ②재사용성, ③확장성, ④제어의 역흐름

배치프로그램 필수요소 5가지 : ①대용량데이터, ②자동화, ③견고성, ④안정성, ⑤성능

배치프로그램 배치방법 3가지 : ①정기배치, ②이벤트성배치, ③On-Demand 배치

사용자 인터페이스(UI)의 기본원칙 4가지 : ①직관성, ②유효성, ③유연성, ④학습성

애플리케이션 테스트 기반에 따른 테스트 종류 3가지 : ①명세기반 테스트, ②구조기반 테스트, ③경험기반 테스트

테스트 검증기준 종류 3가지 : ①기능기반 커버리지, ②라인 커버리지, ③소스코드 커버리지

Windows의 특징 4가지 : ①GUI, ②선점형 멀티태스킹, ③플러그 앤 플레이(PnP), ④OLE

DB의 상호 관련된 데이터들의 모임 4가지 : ①운영 데이터, ②공통 데이터, ③통합된 데이터, ④저장된 데이터 (운공통저)

프로토콜의 기본 구성요소 3가지 : ①구문, ②타이밍, ③의미 (구타의미)

운영체제의 4가지 목적 : ①처리능력, ②신뢰도, ③사용가능도, ④응답시간, 반환시간

애플리케이션 성능측정 지표 4가지 : ①처리량, ②응답시간, ③경과시간, ④자원사용률 (처응경자)

교착상태의 필요조건 4가지 (필기 내용) : ①상호배제, ②점유 및 대기, ③비선점, ④환형대기

DBMS의 필수 기능 3가지 : ①정의 기능, ②조작 기능, ③제어 기능

데이터모델의 구성요소 3가지 : ①구조, ②연산, ③제약조건 (SOC) 또는 ①엔티티, ②속성, ③관계

UML을 구성하는 기본요소 사물 4가지 : ①구조사물, ②행동사물, ③그룹사물, ④주해사물

인덱스 종류 2가지 : ①클러스터드 인덱스, ②Non-클러스터드 인덱스

파티션 종류 3가지 : ①범위분할, ②해시분할, ③조합분할

분산 DB의 목표 4가지 : ①위치 투명성, ②중복 투명성, ③장애 투명성, ④병행 투명성

SW 개발보안 요소 3가지 : ①기밀성, ②무결성, ③가용성 (기무가~)

ESB: EAI와 유사하지만 애플리케이션보다는 서비스 중심의 통합을 지향하는 아키텍처 (연계 매커니즘 中 간접연계방식)

ESP : 출발지 인증, 데이터 무결성, **기밀성** 등 지원 (IPSec에 있어서 데이터 무결성과 프라이버시를 제공하는 기능)

ESM : 방화벽, 침입탐지 시스템, VPN 등의 보안 솔루션을 하나로 모은 통합 보안관리 시스템 (기업 보안 관리)

스프링 배치 : Sprint사와 Accenture사가 2007년에 공동 개발한 오픈소스 프레임워크

쿼츠 : 스프링 프레임워크로 개발되는 응용 프로그램들의 일괄 처리를 위한 다양한 기능을 제공하는 오픈소스 라이브러리

크론 : 리눅스의 스케줄러 도구로, cronlab 명령어를 통해 작업을 예약할 수 있음

하둡 : 오픈소스를 기반으로 한 분산 컴퓨팅 플랫폼이며, 가상화된 대형 스토리지를 형성하는 자바 SW 프레임워크(구글)

OLTP : 온라인 업무처리 형태의 하나로 네트워크상의 여러 이용자가 실시간으로 DB 데이터를 갱신하거나 검색하는 방식

애플리케이션 테스트 : 애플리케이션에 잠재되어 있는 결함을 찾아내는 절차. SW가 기능을 정확히 수행하는지를 검증함