|  |  |
| --- | --- |
| **Alien Code** | **베드 코드**의 유형으로 **아주 오래**되거나 참고문서 또는 **개발자가 없어**서 유지보수 작업이 아주 어려운 응용프로그램의 소스코드 |
| **DBA** | DBMS에서 최고 수준의 권한을 갖는 자 |
| **DBMS** | 데이터베이스를 관리해주는 소프트웨어  실시간으로 발생하는 대용량의 연속적인 데이터의 흐름을 관리하는 시스템 |
| **DFD (데이터 흐름도)** | 데이터가 각 프로세스를 따라 흐르면서 변환되는 모습을 나타낸 그림 |
| **DNS  (Domain Name System)** | 도메인 네임을 IP 주소로 변환하는 시스템 |
| **EA** | 기업의 목표와 요구사항을 효과적으로 지원하기 위해 IT인프라 각 부분의 구성과 구현 기술을 체계적으로 기술 |
| **EAI** | 기업에서 운영되는 서로 다른 플랫폼 및 애플리케이션 간의 정보연결, 연계, 통합을 가능하게 해주는 솔루션 |
| **ESB** | 기업에서 운영되는 서로 다른 플랫폼 및 애플리케이션 간 하나의 시스템으로 관리 운영할 수 있도록 **서비스 중심**의 통합을 지향하는 아키텍처 또는 기술 |
| **ERD (E-R다이어그램)** | 업무분석 결과로 도출된 **엔티티와 엔티티 간의 관계**를 도식화한 다이어그램  현실 세계에 존재하는 데이터와 그들 간의 관계를 사람이 이해할 수 있는 형태로 명확히 표현 |
| **GRANT OPTION FOR** | 다른 사람에게 주는 권한을 뺏자 |
| **Hipo** | 하향식 소프트웨어 개발 문서화 도구. 기능과 자료의 의존 관계를 동시에 표현할 수 있어 이해하기 쉬움. 차트 종류로 가시적 도표, 총체적 도표, 세부적 도표가 존재 |
| **HRN** | 대기중인 프로세스 중 현재 응답률이 가장 높은 것을 선택하는 비선점형 스케줄링 우선순위 = (대기시간 서비스 시간) / 서비스 시간 |
| **ITIL** | **IT** 서비스의 운영 및 관리를 돕기 위한 **문서들의 집합**. 표준 참고 문헌 |
| **JVM** | **자바**로 개발한 프로그램을 컴파일하여 만들어지는 **바이트코드**를 실행시키기 위한 **가상 머신**  서로 다른 플랫폼에서도 JAVA 프로그램이 동작할 수 있도록 함 |
| **Locality (지역성)** | 프로세스가 실행되는 동안 주기억장치를 참조할 때 일부 페이지만 집중적으로 참조하는 성질 |
| **MPEG-21** | 멀티미디어 관련요소 기술들이 통일된 형태로 상호운용성을 보장하는 멀티미디어 표준규격 |
| **MVC패턴** | 사용자 인터페이스로부터 비즈니스 로직을 분리하려 애플리케이션의 시각적 요소나 그 이면에서 실행되는 **비즈니스 로직을 서로 영향없이 쉽게 고칠 수 있는 패턴**  애플리케이션을 모델, 뷰, 컨트롤러 3가지 역할로 구분한 패턴 |
| **Node.js** | V8으로 빌드 된 이벤트 기반 자바스크립트 런타임으로, 웹 서버와 같이 확장성 있는 네트워크 애플리케이션 개발에 사용되는 SW플랫폼. 대부분 JS가 웹 브라우저에서 실행되는 것과 달리, 서버 측에서 실행되는 특징을 가진 기술 |
| **NoSQL (Not only SQL)** | 데이터 저장에서 고정된 테이블 스키마가 필요하지 않고 조인 연산을 사용할 수 없으며, **수평적으로 확장이 가능한 DBMS** |
| **OWASP** | 웹 정보 노출이나 악성 코드, 스크립트, 보안이 취약한 부분을 연구하는 **비영리 단체** |
| **PFF (페이지 부재 빈도)** | 페이지 부재율의 상한과 하한을 정해 직접적으로 페이지 부재율을 예측하고 조절해서 페이지 교체 현상을 줄이는 기법 |
| **PHP** | 동적 웹페이지 제작에 사용되는 언어. 인터페이스 방식으로 자체 인터프리터를 제공하며, 대중적인 언어의 문법과 유사하기 때문에 배우기 쉬움 |
| **PL/SQL** | 표준 SQL을 기본으로 Oracle에서 개발한 데이터 조작언어 |
| **PnP (Plug and Play)** | 운영체제에 하드웨어를 새로 설치했을 때 해당 하드웨어를 사용하는데 필요한 시스템 환경을 운영체제가 자동으로 구성해주는 기능. Window의 특성 중 하나 |
| **QoS (Quality of Service)** | 네트워크에서 일정 기준 이하의 지연 시간이나 데이터 손실률 등을 보증하기 위한 서비스 규격 |
| **REDO** | 트랜잭션의 비정상적 종료 시, 트랜잭션 시작과 완료의 기록이 있는 트랜잭션을 재작업 |
| **UNDO** | 트랜잭션의 비정상적 종료 시, 트랜잭션의 시작은 있지만 완료 기록은 없는 트랜잭션을 재작업 |
| **REST** | HTTP URL을 통해 자원을 명시하고 HTTP 메소드를 통해 해당 자원에 대한 생성.조회.갱신.삭제 등의 명령을 적용할 수 있는 분산 하이퍼 미디어 시스템을 위한 소프트웨어 아키텍처 |
| **rollback** | 트랜잭션 수행 시 오류가 발생하면, 오류 이전의 특정 지점으로 되돌려주는 명령어 |
| **Ruby on Rails** | Ruby 언어를 기반으로 한 서버 개발 프레임워크 |
| **SDLC** | 소프트웨어 생명 주기. 소프트웨어 개발의 전 과정을 각 단계별로 나눈 것 |
| **SDN (SW Defined NW)** | 네트워크를 컴퓨터처럼 모델링하여 여러 사용자가 각각의 소프트웨어 프로그램들로 네트워킹을 가상화하여 제어하고 관리하는 네트워크 |
| **SRT** | 선점 스케줄링 기법 중 하나로 비선점 스케줄링 기법인 SJF를 선점방식으로 변형한 것 |
| **SSL (보안 소켓 계층)** | 데이터를 송,수신하는 두 컴퓨터 사이에 위치하여 인증, 암호화, 무결성을 보장하는 업계 표준 프로토콜. 웹 브라우저와 서버 간의 통신에서 정보를 암호화하여 도중에 정보가 유출되더라도 정보의 내용을 보호할 수 있는 보안 솔루션 |
| **SVN(Subversion)** | **CVS**를 대체하는 **버전관리 도구**. **클라이언트/서버 방식**으로 하나의 저장소를 만들어 그곳에 소스를 저장해 소스 중복이나 여러 문제를 해결하기 위한 도구 |
| **TRUNCATE** | DDL 중 하나. 데이터가 하나도 없이 테이블 구조만 남은 최초의 테이블이 만들어진 상태로 돌아가도록 하는 명령어 |
| **UI** | **사용자와 시스템 간** **상호작용**을 원활하게 하기 위한 인터페이스 |
| **UX** | 제품과 시스템, 서비스 등을 사용자가 직/간접적으로 경험하면서 느끼고 생각하는 총체적 경험 |
| **URL** | 인터넷이 있는 자원을 나타내는 유일한 주소 |
| **V-모델** | 소프트웨어 생명주기 단계 별 개발자 관점에서 명세서에 맞게 소프트웨어가 개발되었는지 검증, 사용자 입장에서 요구사항이 잘 반영됐는지 확인을 지원하기 위한 테스트 모델 |
| **WAN** | 광대역 통신망. 멀리 떨어져 있는 기기들 간의 통신을 지원하는 네트워크 구조 |
| **WAS (웹 애플리케이션 서버)** | 사용자의 요구에 따라 변하는 동적인 콘텐츠를 처리하기 위해 사용되는 미들웨어 |
| **WITH GRANT OPTION** | 다른 사람에게 권한 나눠주기도 가능 |
| **xUnit** | 다양한 언어를 지원하는 **단위테스트**를 하기 위한 프레임워크 |
| **가비지 콜렉션 (GC)** | 메모리 관리 기법의 하나  프로그램이 동적으로 할당했던 메모리 영역 중에서 필요 없게 된 영역을 해제하는 기능 |
| **가상화** | 물리적인 리소스들을 사용자에게 하나로 보이게 하거나 하나의 물리적인 리소스를 여러 개 로 보이게 하는 기술 |
| **개념모델링** | 요구사항 분석 기법 중 하나로 쉬운 이해를 위해 현실세계 상황을 단순화하여 개념적으로 표현하는 것 |
| **결함 분포** | 결함 추이 분석 중 모듈 또는 컴포넌트에 포함되는 특정 속성에 해당하는 결함 수 |
| **결함 에이징** | 결함 추이 분석 중 특정한 결함 상태의 지속시간을 측정하여 분석 |
| **결함추세** | 결함 추이 분석 중 테스트 진행시간에 따른 결함 상태의 지속시간 |
| **경쟁상태** | 메모리가 공유되는 하나의 객체에 두개 이상의 스레드가 동시에 변경하려고 할 때 발생하는 성능저하 문제점 |
| **공통 평가 기준 (CC)** | 정보화 순기능 역할을 보장하기 위해 채택된 정보 보호 제품 평가 기준 |
| **관계대수** | 관계형 DB에서 원하는 정보와 그 정보를 어떻게 유도하는가를 기술한 절차적 정형 언어 |
| **관계해석** | 관계데이터의 연산을 표현하는 방법 비절차적이며 질의어로 표현함 |
| **고가용성 (HA)** | 서버와 네트워크, 프로그램 등의 정보시스템이 시스템의 장애에 대응하여 상당히 **오랜 기간 동안 지속적으로 정상 운영이 가능한 성질** |
| **교착상태** | 둘 이상의 프로세스가 서로가 가진 한정된 자원을 요청하는 경우, 프로세스가 진전되지 못하고 모든 프로세스가 대기 상태로 돌아가는 현상 |
| **다크 데이터** | 수집된 후 저장은 되어 있지만 분석에 활용되지는 않는 다량의 데이터 |
| **데이터 모델** | 현실세계의 정보들을 컴퓨터에 표현하기 위해서 단순화, 추상화하여 체계적으로 표현한 개념적 모형 |
| **동치 분할 테스트** | 입력 자료에 초점을 맞춰 테스트 케이스를 만들고 검사하는 방법 프로그램의 입력 조건에 타당한 입력자료와 타당하지 않은 입력자료의 개수를 균등하게 하여 테스트 케이스를 정하고, 해당 입력 자료에 맞는 결과가 출력되는지 확인하는 테스트 기법 |
| **디스패치 (Dispatch)** | 준비상태에 있는 여러 프로세스를 선정하여 CPU에 할당하는 것. 문맥교환이 발생하고 프로세스를 준비상태에서 실행상태로 전이하는 프로세스 상태 전이 |
| **라우팅 (routing)** | 어떤 **네트워크** 안에서 통신 **데이터를 보낼 경로**를 선택하는 과정  패킷 스위칭 네트워크에서 패킷 전달, 패킷의 변환을 총괄하는 기능 |
| **루트킷** | 불법적인 해킹에 사용되는 악의적인 기능들을 제공하는 도구. 시스템의 취약점을 통해 설치됨 |
| **마리아DB** | 리눅스 환경에서 사용하는 MySQL DB를 기반으로 제작한 GPL 오픈 소스 DB |
| **메소드** | 프로그램의 구성요소인 클래스의 기능으로 작업의 최소단위. 반드시 호출되어야만 기능을 사용할 수 있는 요소 |
| **명령형 언어** | 문제를 해결하기 위한 방법을 기술하는 언어. 알고리즘을 명시함 |
| **문맥교환** | CPU가 현재 실행하고 있는 프로세스의 문맥 상태를 프로세스 제어블록(PCB)에 저장하고 다음 프로세스의 PCB로부터 문맥을 복원하는 작업 |
| **미들웨어** | 분산 컴퓨팅 환경에서 응용 프로그램과 프로그램이 운영되는 환경 간에 원만한 통신이 이루어질 수 있도록 제어해주는 소프트웨어 |
| **백도어** | 시스템 설계자가 기술자의 액세스 편의를 위해 시스템 보안을 제거하여 만들어 놓은 비밀통로 |
| **백본망** | 네트워크를 상호 연결하는 컴퓨터 네트워크의 일부. 각기 다른 LAN선이나 부분 망 간에 정보를 교환하도록 제공하는 망 |
| **베이스라인** | 개발과정에서 각 단계별 산출물들의 검토.변경.처리 등 변화하는 시점에서 통제하는 기준 |
| **벤치마킹** | 특정 분야에서 우수한 상대를 모델로 하여 자기 기업과 비교 분석 단점 극복을 위해 상대를 배우면서 꾸준히 노력하는 자기혁신 경영 기법 |
| **브룩스의 법칙** | 소프트웨어 개발 일정이 지연된다고 해서 새로운 개발 인력을 진행 중인 프로젝트에 투입할 경우 작업 적응 기간과 부작용으로 인해 일정이 더욱 지연된다는 법칙 |
| **빅데이터** | 시스템, 서비스, 조직 등에서 주어진 비용, 시간 내에 처리 가능한 데이터의 범위를 넘어서는 방대한 양의 비정형 데이터 |
| **빅뱅 통합 테스트** | 모든 컴포넌트를 사전에 통합하여 전체 프로그램을 한꺼번에 테스트하는 방식 |
| **사용성 테스트** | 사용자가 제품을 사용하면서 **시나리오에 맞춰** 과제를 수행한 후 질문에 답하도록 하는 테스트 |
| **사이트 맵** | 화면의 정보를 한 눈에 파악하기 위한 시각적인 콘텐츠 모형 |
| **사회공학** | 컴퓨터 보안에 있어 인간 상호 작용의 깊은 신뢰를 바탕으로 사람들을 속여 정상 보안 절차를 깨뜨리기 위한 비기술적 시스템 침입 수단 |
| **살라미** | 많은 대상으로부터 눈치 채지 못할 만큼의 적은 금액이나 양을 빼내는 컴퓨터 사기 기법  장기간 보관하여 두고 조금씩 썰어 먹는 이탈리아 소시지에서 따옴 |
| **살충제 패러독스** | 동일한 테스트 케이스로 반복적인 테스트를 감행할 경우 더 이상 오류를 찾을 수 없는 소프트웨어 테스트 원리 |
| **서브넷 마스크** | IP주소 중 네트워크 주소와 호스트 주소를 구분하기 위한 비트 |
| **선언형 언어** | 무엇을 수행할 것인가 문제를 기술하는 언어. 목표를 명시함 |
| **선점형 스케쥴링** | 현재 프로세스가 CPU를 점유하고 있는데 다른 프로세스가 더 우선시 된다면 현 프로세스가 중단되고 새로이 점유되는 스케쥴링 |
| **스래싱(Thrashing)** | 가상 기억 장치를 구현 시 시스템 성능에 영향을 미치는 요소 중 하나 프로세스 수행에 소요되는 시간보다 페이지 교체에 소요되는 시간이 더 커지는 현상 |
| **스레드(Thread)** | 프로세스 내 작업단위 |
| **스턱스넷 (Stuxnet)** | **원자력 발전소**와 같은 산업 기반 시설에 사용되는 제어 시스템에 침투하여 오작동을 유도하는 명령코드를 입력해 시스템을 마비시키는 **악성코드** |
| **시간 복잡도** | 알고리즘의 소요 시간 정령화를 위해 알고리즘이 처리해야 할 연산의 수가 증가할 때 대략적인 패턴을 수치화한 것 |
| **시멘틱 웹** | 인터넷과 같은 분산환경에서 리소스에 대한 정보와 자원 사이의 관계-의미 정보를 컴퓨터가 처리할 수 있는 **온톨로지 형태로 표현**하고, 이를 자동화된 컴퓨터가 처리하도록 하는 **지능형 웹** |
| **시스템 카탈로그** | DB에 저장되어 있는 **모든 데이터 개체들에 관한 정보나 명세에 대한 정보**가 수록되어 있는, **DBMS가 생성하고 유지**하는 DB내의 **테이블들의 집합체** |
| **시큐어 코딩** | SW를 개발하는 과정에서 보안상의 취약점을 제거하고자 하는 코딩기법 |
| **신뢰도** | 시스템이 주어진 문제를 정확하게 해결하는 정도 |
| **에이징** | 프로세스가 자원을 기다리고 있는 시간에 비례하여 우선순위를 부여함으로써 무기한 대기하는 문제를 방지하는 기법 |
| **오류-부재의 궤변** | 테스트의 원칙 중 하나로, 모든 결함을 제거하여도 사용자의 요구사항을 만족하지 못한다면 품질이 높다고 말할 수 없음 |
| **오픈 스택** | Apache 라이선스 형태로 배포되고 있는 공개소스 기반의 클라우드 운영체제  클라우드 컴퓨팅 환경을 개발하고 관리할 수 있는 플랫폼  SaaS 형태를 가지고 리눅스 기반의 운영이 이루어짐 |
| **온톨로지 (Ontology)** | 실세계의 존재하는 모든 개념들과 개념들의 속성, 그리고 개념들 간의 관계 정보를 컴퓨터가 이해하기 쉽게 서술해 놓은 개념화 명세서 |
| **옵저버 (observer)** | 한 객체의 상태가 바뀌면 그 객체에 의존하는 다른 객체들에게 연락이 가서 자동으로 내용이 갱신되는 방식으로, 일대다의 의존성을 정의하는 패턴 |
| **옵테인 메모리** | RAM과 플래시 메모리의 중간 형태에 해당하는 비휘발성 메모리 |
| **옵티마이저** | SQL을 가장 빠르고 효율적으로 수행할 최적의 처리 경호를 생성해주는 DBMS 핵심엔진. 모듈 |
| **요구공학** | 시스템의 요구사항을 개발(도출 분석 명세 검증 관리)하기 위해 수행되는 구조화된 활동의 집합 소프트웨어를 설계하기에 앞서 무엇을 개발해야 하는지 요구사항을 정의하고 분석 및 관리하는 프로세스를 연구하는 것. 소프트웨어 프로젝트 실패 최소화를 목표로 함 |
| **운영체제 (OS)** | 사용자와 하드웨어 간의 인터페이스로서 동작하는 시스템 소프트웨어 |
| **워킹셋 (Working Set)** | 프로세스가 일정 시간 동안 자주 참조하는 페이지 집합을 메모리 공간에 계속 상주시켜 빈번한 페이지 교체 현상을 줄이는 기법 |
| **이상(Anomaly)** | 테이블에서 일부 속성들의 종속으로 인해 데이터의 중복이 발생하고, 이 중족으로 인해 테이블 조작 시 문제가 발생하는 현상 |
| **인스턴스** | 객체 지향 프로그래밍에서 해당 클래스의 구조로 컴퓨터 저장공간에 할당된 실체 |
| **인터프리터** | 프로그램 문장을 하나씩 변역하고 실행할 수 있도록 하는 프로그램 컴파일 과정이 없기 때문에 개발하는 과정에서 사용하면 유용하다는 장점이 있지만, 실행 속도가 느리고 메모리 사용이 효율적이라는 단점이 있음 |
| **정규화 (Normalization)** | 테이블의 속성들이 상호 종속적인 관계를 갖는 특성을 이용해 테이블을 무손실 분해하는 **데이터 모델링 기법**. 중복을 최소화하고 이상현상이 생기지 않도록 함 |
| **반정규화** | 시스템의 성능 강화. 개선을 위해 **정규화한 데이터들을 통합. 분리**하는 **모델링 기법** |
| **모듈화** | 소프트웨어의 성능을 향상시키거나 시스템의 수정 및 재사용, 유지 관리를 용이하게 하여 프로그램을 효율적으로 관리할 수 있도록 **시스템을 분해하고 추상화하는 기법**  **모듈**을 사용해 성능을 향상시키고 복잡성을 줄이고 시스템 디버깅 시험 수정이 용이하도록 설계하는 기법 |
| **정황 시나리오** | 사용자가 목표를 달성하기 위해 수행하는 방법을 순차적으로 묘사하는 것이며 이를 토대로 UI 요구사항을 작성한다 |
| **제어의 역흐름** | 프레임워크의 특성(재확모역) 중 하나 개발자가 관리하고 통제해야 하는 객체들의 제어 권환을 프레임워크에 넘겨 생산성을 향상시킴 |
| **참조 무결성의 CASCADE 법칙** | 연관 데이터에서 오류나는 일을 방지. 참조하는 테이블이 삭제될 경우 참조받는 테이블도 연쇄로 함께 삭제된다. |
| **컴포넌트** | 특정한 기능을 수행하기 위해 독립적으로 개발되어 보급되는 잘 정의된 인터페이스룰 가지며 다른 부품과 조립되어 응용시스템을 구축하기 위해 사용하는 소프트웨어 프로그램 |
| **크라임웨어** | 온라인에서 범죄와 같은 불법적인 행위를 수행하기 위해 만든 컴퓨터 프로그램. **공격용 툴킷** |
| **클러스터** | 데이터 저장시 데이터 액세스 효율을 향상시키기 위해 동일한 성격의 데이터를 동일한 데이터 블록에 저장하는 물리적 저장방법 |
| **테일러링** | **프로젝트의 특성과 필요에 따라** 소프트웨어 개발 프로세스, 기법, 산출물 등을 비즈니스적으로 또는 기술적인 **요구에 맞도록 최적화**하는 과정 및 방법론 |
| **텍스트 마이닝** | 대규모의 텍스트 데이터를 가지고 체계적인 규칙 패턴분석을 통해 원하는 정보를 찾아내는 기술 |
| **톰캣(tomcat)** | WAS 중 하나. 웹서버에서 자바 코드를 위해하기 위해 사용하는 자바를 이해하는 엔진  자바 기반 웹서버를 구성할 때 리눅스, 아파치와 함께 조합으로 많이 사용 |
| **트랜잭션** | DB시스템에서 **하나의 논리적인 기능을 수행**하기 위한 작업의 **기본 단위**이자 **한꺼번에 모두 수행**되어야 할 **일련의 연산** |
| **트러스트 존** | 하나의 프로세서 내에 일반구역과 보안구역으로 분할해 관리하는 보안 기술 |
| **트리거** | DB시스템에서 이벤트가 발생할 때 마다 관련 작업이 자동으로 수행되는 절차형 SQL |
| **파레토 법칙** | 테스트의 원리 중 결함집중과 관련된 단어로 20%의 모듈에서 80%의 결함이 발견된다는 의미로 사용 |
| **파밍 (Pharming)** | 사용자의 도메인을 탈취하거나 DNS 이름을 속여 사용자들이 진짜 사이트로 오인하도록 유도. 개인정보를 훔치는 신종 인터넷 사기 수법 |
| **파티션** | 데이터베이스에서 대용량의 테이블이나 인덱스를 작은 논리적 단위인 ( )으로 나누는 것 |
| **페이징 기법** | 가상 기억장치에 보관되어 있는 프로그램과 주기억장치의 영역을 동일한 크기로 나눈 후 나눠진 페이지를 동일하게 나눠진 주기억장치의 영역에 적재시켜 실행하는 기법 외부 단편화는 발생하지 않으나 내부 단편화는 발생할 수 있음 |
| **세그먼테이션 기법** | 가상 기억장치에 보관되어 있는 프로그램을 다양한 크기의 단위로 나눈 후 주기억장치에 적재시켜 실행하는 기법 내부 단편화는 발생하지 않으나 외부 단편화는 발생할 수 있음 |
| **포렌식** | 범죄 사실을 사법기관에 제출하기 위해 디지털 증거자료를 수집/복사/분석/제출하는 일련의 과정을 지칭하는 용어 |
| **프로시저** | 복잡한 트랜잭션을 처리하기 위해 SQL을 이용해 생성된 데이터를 조작하는 프로그램. 일정 조건을 달성하면 자동으로 수행함 |
| **함수적 종속** | X의 값 각각에 대해서 항상 Y의 값이 하나만 연관. X->Y. Y는 X에 종속. X=결정자 Y=종속자 |
| **해시(HASH) 암호화** | 임의의 길이 입력데이터나 메세지를 고정된 길이의 값이나 키로 변환하는 단방향 암호 알고리즘 |
| **형상 관리 (SCM)** | 소프트웨어의 개발 과정에서 소프트웨어의 변경 사항을 관리하기 위해 개발된 일련의 활동 |
| **회선 교환 방식** | 라우팅 방식 중에서 교환기를 이용하여 물리적으로 접속하는 방식으로 기존의 음성 전화망이 이에 해당한다. 공간 분할 교환 방식과 시분할 교환 방식이 있다 |
| **캡슐화** | **정보은닉**. 객체의 상세한 내용을 객체 외부에 감추고 메시지를 통하여 다른 객체와 상호작용함 |

|  |  |
| --- | --- |
| 요구사항 개발 프로세스 (4단계) | [**도분명확**] (타당성조사) - 도출 – 분석 – 명세 - 확인 |
| 논리 데이터 모델링의 특성 (4가지) | [**정포완독**] 정규화 - 포용성 - 완전성 - 독립성 |
| 물리 데이터 모델링 변환 절차 (6단계) | 개체→테이블 / 속성→컬럼 / UID→기본키 / 관계→외래키 / 컬럼 유형과 길이 정의 / 반정규화 수행 |
| 이상의 종류 (3가지) | 삽입 이상 / 삭제 이상 / 갱신 이상 |
| 분산 데이터베이스 저장 방식의 목표 (4가지) | 병행 투명성 / 장애 투명성 / 중복 투명성 / 위치 투명성 |
| 검증 기준 (3가지) | 기능 기반 커버리지 / 라인 커버리지 / 코드 커버리지 |
| 와이어프레임 도구 (4종류) | 파워포인트, 키노트, 스케치, 일러스트 |
| 인터페이스 설계 지침 (9가지) | 사용자 중심 / 가시성 / 접근성 / 일관성 / 단순성 / 결과 예측 가능 / 표준화 / 명확성 / 오류 발생 해결 |
| 프레임워크의 특성 (4가지) | [**재확모역**] 재사용 / 확장성 / 모듈화 / 제어의 역흐름 |
| 결함 추적 순서 (6단계) | 결함 등록 - 검토 - 할당 - 수정 - 종료 - 해제 |
| 릴리즈 노트 작성 프로세스 (6단계) | [**모정개영정추**] 모듈 식별 - 릴리즈 정보 확인 - 릴리즈 노트 개요 작성  - 영향도 체크 - 정식 릴리즈 노트 작성 - 추가 개선 항목 식별 |
| 애플리케이션 성능 측정 지표 (4가지) | [**처응경자**] 처리량 / 응답시간 / 경과시간 / 자원사용률 |
| 운영체제의 성능 측정 지표 (4가지) | [**처반사신**] 처리능력 / 반환시간 / 사용가능도 / 신뢰도" |
| 공통모듈 구현 시 준수 할 명세기법 (5가지) | [**정명완일추**] 정확성 / 명확성 / 완전성 / 일관성 / 추적성 |
| 제품 소프트웨어 패키징 작업 순서 (7단계) | [**기모빌환시변배**] 기능식별 - 모듈화 - 빌드 진행 - 사용자 환경 분석  - 패키징 및 적용 시험 - 패키징 변경 개선 – 배포 |
| 데이터베이스 구축 과정 | 요구 분석 - 개념적 설계 - 논리적 설계 – 물리적 설계 – 구현 |
| 데이터베이스 내 데이터 (4종류) | 운영 데이터 / 통합된 데이터 / 저장된 데이터 / 공용 데이터 |
| 결함 관리 프로세스 (6단계) | [**발등분확할조**] 에러 발견 - 에러 등록 - 에러 분석  - 결함 확정 - 결함 할당 - 결함 조치 - (결함 조치 검토 및 승인) |
| 보안 점검 항목 (8가지) | [**세입보시에코캡에**] 세션통제 / 입력 데이터 검증 및 표현 / 보안 기능  / 시간 및 상태 / 에러처리 / 코드 오류 / 캡슐화 / API오용 |
| Windows의 특징 | 선점형 멀티태스킹 / Single-User / OLE / PnP |
| 서버 프로그램 구현 단계 (5단계) | [**디스다써클**] DTO/VO - SQL - DAO - Service - Controller 구현 |
| 배치 스케줄러 종류 (3종류) | 스프링배치 / Quartz / Cron |
| **응집도** (높은 순) (독립성 높은 순) | [**기순교절시논우**]  기능적 – 순차적 –교환적(통신적) – 절차적 – 시간적 – 논리적 – 우연적 |
| **결합도** (낮은 순) (독립성 높은 순) | [**자스제외공내**] 자료 – 스탬프 – 제어 – 외부 – 공통 – 내용 |
| 트랜잭션의 특징 (4가지) | [**ACID**] 원자성 / 일관성 / 독립성 / 지속성 |
| UI의 기본 원칙 | [**직유학유**] 직관성 / 유효성 / 학습성 / 유연성 |
| 객체지향프로그래밍의 특성 (3가지) | 상속 / 추상화 / 다형성 / 캡슐화 / 정보은닉 |
| UML의 구성요소 (3가지) | 사물 / 관계 / 다이어그램 |

테스트

|  |  |
| --- | --- |
| **테스트 케이스** | **사용자의 요구사항을 준수**하는지 확인하는 테스트 항목에 대한 **명세서**  입력값, 실행 조건, 예상된 결과의 집합 |
| **테스트 하네스** | 테스트를 지원하기 위한 코드, 데이터 등 환경의 일부분 |
| 테스트 하네스의 구성요소 | [**스드슈스케목**] 테스트 스텁 / 드라이버 / 슈트 / 스크립트 / 케이스 / 목 오브젝트 |
| 테스트 스텁 | 상향식 통합 테스트에서 사용. 타 모듈의 기능을 단순히 수행하는 도구  다른 프로그래밍 기능을 대리하는 코드. 기존 코드를 흉내내거나 아직 개발되지 않은 코드를 임시로 대체하는 역할 수행 |
| 테스트 드라이버 | 하향식 통합 테스트에서 사용. 하위 모듈을 호출하는 상위 더미 모듈 |
| **테스트 슈트** | **단순히** 테스트 케이스를 묶은 **집합**. |
| **테스트 시나리오** | 테스트 케이스를 적용하는 **순서에 따라** 여러 개의 테스트 케이스를 묶은 **집합** |
| **테스트 오라클** | 테스트 케이스에 테스트의 **예상 결과(참 값)를 대입**해 비교하는 기법 |
| 테스트 오라클의 종류 | 참 오라클 / 샘플링 오라클 / 추정 오라클 / 일관성 검사 오라클 |
| 목 오브젝트 | 사전에 조건부를 입력해두고 조건이 만족할 시에 입력한 조건부가 실행 |
| 테스트의 분류 | 프로그램 실행 여부 : **정적 테스트**(X) vs **동적 테스트**(O)  테스트의 기반 : **명세** 기반 테스트 vs **구조** 기반 테스트 vs **경험** 기반 테스트  시점 : **검증(Verification)(개발자)** vs **확인(Validation)(사용자)**  목적 : **회복** / **안전** / **성능** / **구조** / **회귀** / **병행** / **강도** 테스트 |
| 정적 테스트 종류 (3가지) | 동료검토 / 워크스루 / 인스펙션 |
| **동료검토** | **2~3명**이서 진행하는 리뷰. **요구사항 명세서 작성자**가 요구사항 명세서를 설명하고, **이해관계자**들이 설명을 들으면서 코드에 대한 결함을 발견하는 기법 |
| **워크스루** | 검토 자료를 **회의 전에 배포**해서 사전에 검토를 진행한 후 **짧은 시간 회의**를 통해 코드의 오류를 검출하고 문서화하는 기법 |
| **인스펙션** | 소프트웨어 요구, 설계, **원시 코드** 등의 **저작자 외의 다른 전문가** 또는 팀이 검사하여 오류를 찾아내는 **공식적 검토 기법** |
| 동적 테스트 종류 (2가지) | 화이트 박스 테스트 / 블랙 박스 테스트 |
| **화이트박스 테스트** | 모듈의 **원시 코드를 오픈** 시킨 상태에서 원시 코드의 논리적인 모든 **경로**를 확인하는 테스트 |
| **블랙박스 테스트** | SW가 수행할 특정 기능을 알기 위해서 각 **기능**이 완벽히 작동되는 것을 입증하는 테스트  기법 – 경계값분석 / 동등분할 / 결정테이블 / 상태전이 / 유스케이스 / 분류트리 / 페어와이즈 |
| **경계값 분석 테스트** | 등가분할 수 **경계값**을 포함해 테스트 케이스 설계하는 기법 |
| **동등 분할 테스트** | 입력 데이터의 영역을 유사한 도메인별로 유효 값/무효 값을 **그룹핑**하여 **대표 값**을 테스트 케이스로 도출하여 테스트 하는 기법 |
| **회복 테스트** | **고의로 실패** 유도. 시스템의 **정상 복귀 여부**를 테스트 |
| **안전 테스트** | 소스코드 내 보안적인 **결함을 미리 점검**하는 테스트 |
| **강도 테스트** | 시스템에 과다 정보량을 부과. **과부하시에도 정상 작동** 검증 테스트 |
| **성능 테스트** | 이벤트에 응답시간, 특정시간 내 처리 업무량, 사용자 요구 반응속도를 측정하는 테스트 |
| **회귀 테스트** | 오류를 제거하거나 수정한 시스템에서 **새로이 유입된 오류가 없는지 확인**하는 **반복** 테스트 |
| **병행 테스트** | 변경된 시스템과 기존 시스템에 **동일 데이터를 입력**해 결과를 **비교**하는 테스트 |

테스트 커버리지

|  |  |
| --- | --- |
| 테스트 커버리지 | 주어진 테스트 케이스에 의해 수행되는 소프트웨어의 **테스트 범위를 측정하는 테스트 품질 측정 기준**. 테스트의 정확성과 신뢰성을 향상시키는 역할 |
| 테스트 커버리지 유형(3가지) | [**기라코**] 기능 기반 커버리지 / 라인 커버리지 / 코드 커버리지 |
| **코드 커버리지** | 소스 코드의 구문, 조건, 결정 등의 구조 코드 자체가 얼마나 테스트되었는지를 측정하는 방법 |
| 코드 커버리지 유형 (6가지) | 구문 커버리지 / 결정 커버리지 / 조건 커버리지 외 3가지. (구조 테스트 유형과 동일) |
| **구문 커버리지** | 프로그램 내의 **모든 명령문**을 적어도 한 번 수행하는 커버리지 |
| **결정 커버리지** | 프로그램 내의 **전체 결정문**이 적어도 한 번은 **참과 거짓**을 결과를 수행하는 커버리지 |
| **조건 커버리지** | **결정 명령문 내의 각 조건**이 적어도 한 번은 **참과 거짓**을 결과가 되도록 수행하는 커버리지 |

재해 재난 관련

|  |  |
| --- | --- |
| **RPO (Recovery Point Object)** | 재해복구 프로세스 중 하나.  재해로 인해 서비스가 중단되었을 때 유실을 감내할 수 있는 손실 허용 시점. 복구시점 |
| **RTO (Recovery Time Object)** | 재해복구 프로세스 중 하나. 재해로 인해 서비스가 중단되었을 때 서비스를 복구하는데 걸리는 시간 |
| **DRS (Disaster Recovery System)** | 재해 복구 시스템. 정보시스템 기반 구조의 전체 또는 일부를 다른 위치해 구축, 재해가 발생했을 경우 신속하게 복구해 비즈니스에 대한 영향을 최소화 하기 위한 제반 계획 시스템 |
| **애드혹 네트워크** | **재난 현장**과 같이 별도의 고정된 유선망을 구축할 수 없는 장소에서 **모바일 호스트만**을 이용해 구성한 네트워크. 단 기간 사용하거나 유선망 구성이 어려울 때 적합 |
| **BCP (업무 연속성 계획)** | 재난 발생 시 비즈니스의 연속성을 유지하기 위한 계획 |

네트워크 장비

|  |  |
| --- | --- |
| **라우터** | 데이터 패킷을 발신지에서 목적지까지 전달하기 위해 최적의 경로를 지정하고 이 경로를 따라 데이터 패킷을 다음 장치로 전달하는 네트워크 장비 |
| **스위치** | 동일 네트워크 내에서 출발지에 들어온 데이터 프레임을 목적지 MAC 주소 기반으로 빠르게 전달하는 네트워크 장비 |

OSI 7계층

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **OSI 7계층** | | ISO에서 개발한 컴퓨터 네트워크 프로토콜  디자인과 통신을 계층으로 나누어 설명한 개방형 시스템 상호 연결 모델 | | |
| 7 | **응**용 | 사용자가 접근할 수 있게 |  |  |
| 6 | **표**현 | 데이터 변환 |  |  |
| 5 | **세**션 | 대화 제어 |  |  |
| 4 | **전**송 | 단말기 간. 종단. End-to-End | 세그먼트 | 게이트웨이 |
| 3 | **네**트워크 | 경로 배정 | 패킷 | 라우터 |
| 2 | **데**이터**링**크 | 두 개의 인접 개방 시스템 간 | 프레임 | 랜카드, 브리지, 스위치 |
| 1 | **물**리 | 실제 접속. 절단 | 비트 | 리피터, 허브 |

프로토콜

|  |  |
| --- | --- |
| 프로토콜 | 시스템끼리 원활하게 정보를 주고받기 위해 표준화된 통신규약 |
| 프로토콜의 기본요소 | 구문 / 의미 / 타이밍 |
| **BGP** | 라우팅 프로토콜 중 하나. EGP의 단점을 보완한 것 |
| **RIP** | 라우팅 프로토콜 중 하나. 자율 시스템 내에서 사용하는 거리벡터 알고리즘에 기초하여 개발된 통신규약. 최대 홉(Hop) 수를 15개로 제한 |
| **OSPF (Open Shortest Path First)** | 라우트 프로토콜 중 링크 상태 알고리즘을 사용하여, 토폴로지에 대한 정보가 전체 라우터에 동일하게 유지되는 프로토콜 |
| **DHCP** | RFC1541에서 규정. TCP/IP 통신을 실행하기 위해 필요한 설정 정보를 자동적으로 할당.관리 하기위한 통신 규약 |
| **VoIP (Voice Over Internet Protocol)** | 전통적인 PSIN 회선 전송망이 아니라 IP를 사용해 **음성 데이터**를 패킷 형태로 송신하는 것을 통해 사용자가 인터넷을 전화 통신용 송신 매체로 사용할 수 있도록 하는 HW. SW 분류 |
| **RedSec Protocol (레드섹)** | ‘RADIUS over TLS’의 준말. RADIUS 데이터를 전송 제어 프로토콜(TCP)이나 전송 계층 보안(TLS)을 이용해 전송하기 위한 프로토콜. |
| **SET (안전한 전자 거래)** | 온라인 상의 안전한 거래를 위해 Visa와 Master Card에서 개발한 프로토콜 |

TCP/IP 4계층

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TCP/IP 4계층** | | 인터넷에 연결된 서로 다른 기종의 컴퓨터들이 데이터를 주고 받을 수 있도록 하는 표준 프로토콜 | | |
| 4 | 응용 | 세표응 | **FTP** | 서버와 클라이언트 사이의 **파일**을 전송하기 위한 프로토콜 |
| SMTP | **전자 우편** 전송을 위한 프로토콜 |
| TELNET | **원격 컴퓨터 접속을 위한 프로토콜** |
| SNMP | **네트워크 관리** |
| **DNS** | **도메인 네임을 IP 주소로** |
| HTTP | **HTML 문서 송수신** |
| 3 | 전송 (TCP계층) | 전 | **TCP** | 근거리 통신망이나 인트라넷 인터넷에 연결된 컴퓨터에서 실행되는 프로그램 간에 일련의 옥텟을 안정적으로 순서대로 에러없이 교환하게 해주는 프로토콜 |
| **UDP** | 비연결성이고 신뢰성이 없음. 순서화되지 않은 데이터그램 서비스를 제공. |
| RTP | UDP 기반 실시간 프로토콜 |
| **RTCP** | RTP 패킷 전송 품질 제어, 세션에 참여한 참여자에게 제어 정보 전송, 하위 프로토콜은 데이터 패킷과 제어 패킷의 다중화 제공의 역할을 하는 프로토콜 |
| 2 | 인터넷  (IP계층) | 네 | **IP** | 전송할 데이터 주소 지정 (IPv4, IPv6) |
| ICMP | 오류처리 메시지 |
| IGMP | 멀티캐스트 |
| ARP | IP주소 -> MAC주소 |
| RARP | MAC주소 -> IP주소 |
| **IPSec** | IP 패킷 단위의 데어터 변조 방지 및 은닉 기능을 제공 |
| 1 | 네트워크 액세스 | 물데 |  |  |

IPv4 vs IPv6

|  |  |
| --- | --- |
| **IPv4** | 패킷 교환 네트워크 상에서 데이터를 교환하기 위한 32bit(8비트씩 4부분) 주소체계를 갖는 프로토콜 |
|  | 전송방식(주소체계) – 유니캐스트 / 멀티캐스트 / 브로드캐스트 |
| **IPv6** | IPv6 주소 부족 문제 해결을 위해 개발된 128bit(16비트씩 8부분) 주소체계를 갖는 프로토콜. 전송속도가 더 빠름 |
|  | 전송방식(주소체계) – 유니캐스트 / 멀티캐스트 / 애니캐스트 |

보안 공격 기법

|  |  |
| --- | --- |
| 서비스 거부(DoS) 공격 : PoD. 스머핑. SNY flooding. TearDrop. Land. DDoS  네트워크 침해 공격 : 스미싱. 스피어피싱. APT. 무작위 대입 공격. 큐싱. XSS  정보보안 침해 공격 : SQL 인젭션. 좀비PC. 웜. 제로데이 공격. 키로거 공격. 랜섬웨어. 트로이목마 | |
| **Ping of Death (죽음의핑, PoD)** | 인터넷 프로토콜 범위 이상의 패킷을 전송하여 네트워크를 마비시키는 공격 |
| **스머핑(SMURFING)** | IP나 ICMP 특성을 이용해 한 사이트에 대량의 데이터를 집중 전송하여 네트워크 또는 시스템의 상태를 불능하게 만드는 공격 기법 출발지 주소를 공격 대상의 IP로 설정하여 네트워크 전체에게 ICMP echo 패킷을 직접 브로드캐스팅하여 마비시키는 공격 |
| **SNY flooding** | 공격자가 가상의 클라이언트로 위장하여 의도적으로 **3-way-handshake**를 중단하여 서버가 대기 상태에 놓여 정상적인 서비스를 하지 못하게 하는 서비스 공격 유형 |
| **TearDrop Attack** | **IP 패킷의 재조합 과정**에서 **잘못된 Fragment Offset** 정보로 인해 수신시스템이 문제를 발생하도록 만드는 DoS 공격 |
| **랜드어택** | 공격자가 출발지 주소와 송신 IP를 임의로 변경해 동일하게 만듦  **자신이 자신에게 무한한 응답**을 수행하게 해 마비시키는 공격 |
| **스미싱** | **문자메세지**를 이용해 개인 신용 정보를 빼돌리는 공격 |
| **스피어 피싱** | **메일의 링크나 파일**의 클릭을 유도한 뒤 개인정보를 탈취하는 공격기법 |
| **APT** (지능형 지속 공격) | 특정 기업, 조직에 **거점을 마련 후 때를 기다리다가** 보안을 무력화 시키는 공격 |
| **CSRF** | **악성 스크립트** 구문을 삽입하여 **자신의 의도와 무관하게** 공격자가 의도한 행위를 하게 되는 공격 |
| **XSS** | 특정 게시물 또는 이메일 링크 클릭 시 **악성 스크립트**가 실행되는 공격 기법 |
| CSRF는 공격자가 사용자를 이용해 시스템을 공격. XSS는 공격자가 사용자를 공격 | |
| **SQL 인젝션** | 사용자의 입력 값 등 외부 입력 값이 **SQL 쿼리에 삽입**되어 공격 |
| **키로거 공격** | 컴퓨터 사용자의 **키보드 움직임을 탐지**하여 ID, PW등의 개인정보를 빼내는 공격 |
| **제로데이 공격** | 보안 취약점이 발견된 후 **널리 공표되기 전**에 취약점을 공략해 공격하는 기술 |
| **트로이 목마** | 정상 프로그램으로 **위장**하고 있다가 해당 프로그램이 실행되었을때 활성화 되는 공격 |
| **스니핑** | 네트워크 상 떠돌아다니는 패킷을 **몰래 들여다 보는(도청) 수동적** 공격기법 |
| **스누핑** | 네트워크 상의 정보를 **염탐**하여 불법적으로 **얻는** 공격기법 |
| **스푸핑** | 공격 대상 컴퓨터와 서버(스위치) 사이의 MAC 테이블을 속여 권한을 획득하고 정보를 가로채는 공격기법 |
| **IP 스푸핑** | 공격자가 자신의 IP주소를 변조하여 공격에 대한 추적을 어렵게 만드는 공격 기술 |
| **Slowloris** | HTTP 프로토콜의 취약점을 이용한 공격. 헤더와 바디를 구분하기 위해 헤더 끝에 /r/n/r/n을 전송해야 하는데 /r/n만 보내 나머지 /r/n이 올 때까지 기다리게 하는 공격 |
| **RUDY** | 요청 헤더의 Content-Length를 비정상적으로 크게 설정하여 메시지 바디 부분을 매우 소량으로 보내 계속 연결 상태를 유지. 자원을 소진시키는 기법 |
| **세션 하이재킹** | TCP 신뢰성 기반의 연결을 이용한 공격 방법. 통신 내용을 엿보거나 세션을 가로채어 정상적인 인증 과정을 무시하고, 불법으로 시스템에 접슨할 수 있는 공격 |
| **타이포스쿼팅**  **(= URL 하이재킹)** | 네티즌이 유명 사이트들의 주소를 잘못 입력했을 때 광고 사이트로 몰래 이동시키기 위해 유사 도메인을 미리 선점해 등록하는 것 |
| **허니팟** | 가상의 정보시스템을 만들어 놓고 실제로 공격을 당하는 것처럼 보이게 해 해커나 스팸, 바이러스를 유인해 침입자들의 정보를 수집. 추적하는 기법 |
| **버퍼 오버플로우** | 정해진 메모리의 범위를 넘치게 해서 원래의 리턴 주소를 임의로 변경. 임의의 프로그램이나 함수를 실행시키는 해킹 기법 |

보안 솔루션

|  |  |
| --- | --- |
| **VPN (Virtual Private Network)** | 공중망에서 터널링, 암호화기법 등을 사용해 **사설 통신망과 같은** 서비스를 제공하는 **가상의 네트워크**. 기업에서 내용을 바깥 사람에게 드러내지 않고 통신할 목적으로 쓰임 |
| **방화벽 (Firewall)** | 외부로부터 내부망을 보호하기 위한 네트워크 구성요소 외부 불법 침입으로부터 내부의 정보자산을 보호하기 위한 보안 솔루션  보안규칙에 기반하여 들어오고 나가는 네트워크 트래픽을 모니터링하고 제어하는 보안 시스템 |
| **IDS (침입 탐지 시스템)** | 컴퓨터 시스템의 비정상적 사용.오용.남용 실시간 탐지 |
| **IPS (침입 방지 시스템)** | 방화벽 + IDS. 비정상적 트래픽을 능동적으로 차단하고 격리하는 시스템 |
| **NAC** | 내부 PC의 MAC주소를 IP 관리시스템에 등록. 일관된 보안 관리 기능을 제공 |
| **DLP (데이터 유출 방지)** | 내부 **정보 유출 방**지를 의미하며 기업 내에서 이용하는 다양한 주요 정보들을 보호하고 외부 유출 방지를 위해서 사용하는 보안기술 |
| **코드 난독화** | **역공학**을 통한 공격을 막기 위해 프로그램 **소스 코드를 알아보기 힘든 형태로** 바꾸는 기술 |
| **MAC** | 객체에 포함된 정보의 비밀성과 이러한 비밀성의 접근정보에 대하여 주체가 갖는 권한에 근거하여 객체에 대한 접근을 제한하는 접근제어(AC) 정책 |

보안 요소 3가지

|  |  |
| --- | --- |
| **기밀성** | 인가된 사용자만 접근.조회 할 수 있도록 하는 보안 특성 |
| **무결성** | 인가된 사용자만 수정할 수 있도록 하도록 하는 보안 특성 |
| **가용성** | 필요한 경우 인가된 사용자에게 언제든지 정보 제공 할 수 있도록 하는 보안 특성 |

다이어그램

|  |  |
| --- | --- |
| **클래스 다이어그램** | 시스템 내의 클래스의 정적 구조를 표현하고 속성과 동작으로 구성되어 있는 다이어그램 |
| **시퀀스 다이어그램** | 시스템의 동적 행위를 표현하고 객체 간 상호작용을 메시지 흐름으로 표현하는 다이어그램 |

소프트웨어 아키텍처

|  |  |
| --- | --- |
| **소프트웨어 아키텍처** | 소프트웨어 구성요소의 특성. 관계를 표현하는 시스템 구조 |
| **소프트웨어 아키텍처 프레임워크** | 소프트웨어 집약적인 시스템에서 아키텍처가 표현해야 하는 내용 및 이들 간의 관계를 제공하는 아키텍처 기술 표준 |
| 소프트웨어 아키텍처 프레임워크의 구성요소 (6가지) | 명세서 / 이해관계자 / 뷰 / 근거 / 관점 / 관심사 |
| **소프트웨어 아키텍처 4+1뷰** | 고객의 요구사항을 정리해 놓은 시나리오를 4개 관점으로 접근하는 방법 [**논구배프유**] |
| 논리 뷰 | 설계 모델의 추상화. 기능적 요구사항 지원 |
| 프로세스 뷰 | 런타임 시 태스크.스레드.프로세스의 관계 표현. 비기능적 요구사항 고려 |
| 구현 뷰 | 소프트웨어 구현.관리 측면 표현. 개발자 관점 |
| 배포 뷰 | 물리적인 구성. 상호 연결 관계 표현 |
| 유스케이스 뷰 | 아키텍처를 도출.설계하는 작업 주도. 다른 뷰 검증에 이용 |

화면 설계 도구

|  |  |
| --- | --- |
| **와이어프레임** | 페이지에 대한 개략적인 레이아웃이나 UI 요소 등에 대한 뼈대를 설계하기 위해 제작하는 UI 설계 도구 이해관계자들과의 화면구성을 협의하거나 서비스의 간략한 흐름을 공유하기 위해 화면 단위의 레이아웃을 설계하는 작업 |
| **목업** |  |
| **스토리보드** | UI 화면 설계를 위해서 구축하는 서비스를 위한 모든 정보가 담겨져 있는 문서. 설계 산출물 |
| **프로토타입** | 와이어프레임 또는 스토리보드에 동적 효과를 적용함으로써 실제 구현된 것처럼 시뮬레이션 할 수 있는 모형 |
| **유스케이스** | 시스템이 액터에게 제공해야 하는 기능으로 시스템 요구사항이자 사용자 입장에서 바라본 시스템의 기능 |

차수 & 카디널리티

|  |  |
| --- | --- |
| **차수 (Degree)** | **속성**(attribute. DB를 구성하는 가장 작은 단위. **열**)의 개수 |
| **카디널리티 (Cardinality)** | **튜플**(tuple. 릴레이션을 구성하는 각각의 **행)**의 개수 |

JAVA 웅앵웅

|  |  |
| --- | --- |
| **Ajax** | 자바스크립트를 사용한 **비동기 통신기술**. 클라이언트와 서버 간에 XML 데이터를 주고받는 기술 |
| **JSON** | **속성-값** **쌍**으로 이루어진 데이터객체를 전달하기위해 사람이 읽을 수 있는 텍스트를 사용하는 개방형 표준 포맷. 비동기처리를 하는 ajax에서는 XML을 대신하여 사용됨 |
| **JDBC** | DBMS 접속 기술 중 하나. 직접 연계 방식 중 하나. **JAVA에서 데이터베이스에 접속할 수 있도록 하는 자바 API**. 수신 시스템의 프로그램에서 드라이버를 이용하여 송신 시스템 DB와 연결하는 연계 기술 |
| **MyBatis** | DBMS 접속 기술 중 하나. **JDBC 코드를 단순화한 SQL Mapping 기반 오픈 소스 접속 프레임워크**. SQL을 거의 그대로 사용 가능하여 국내 환경에 적합 |
| **Spring** | JAVA 플랫폼을 위한 오픈소스 애플리케이션 프레임워크. **전자정부 프레임워크** 기반 기술로 사용되는 서버 개발 프레임워크 |

DBMS

|  |  |
| --- | --- |
| **DBMS** | 데이터베이스를 관리해주는 소프트웨어 |
| DBMS의 종류 | Oracle / IBM DB2 / MySQL / Microsoft SQL Server / SQ Lite |
| DBMS 접속기술 (API 또는 프레임워크) (3가지) | JDBC. ODBC. MyBatis |
| **JDBC** | 직접 연계 방식 중 하나. JAVA에서 데이터베이스에 접속할 수 있도록 하는 자바 API. 수신 시스템의 프로그램에서 드라이버를 이용하여 송신 시스템 DB와 연결하는 연계 기술 |
| **ODBC** | 데이터베이스에 접근하기 위한 표준 개방형 API |
| **MyBatis** | JDBC 코드를 단순화한 SQL Mapping 기반 오픈 소스 접속 프레임워크 SQL을 거의 그대로 사용 가능하여 국내 환경에 적합 |
| DBMS 필수 기능 (3가지) | 정의기능 / 조작기능 / 제어기능 |

데이터 웨어하우스

|  |  |
| --- | --- |
| **OLAP** (OnLine Analytical Processing) | 사용자가 정보에 대해 직접 접근해 **대화식**으로 정보를 분석하고 **의사결정에 활용**하는 과정 **다차원**으로 이루어진 데이터로부터 통계적인 요약 정보를 분석하여 의사결정에 활용하는 방식 |
| OLAP 연산 | Roll-up / Roll-down(Drill-down) / Drill Across / Drill Through / Pivoting / Slicing / Dicing |
| **데이터 웨어하우스** | 급증하는 다량의 데이터를 분석하고 정보화하고 이를 **여러 계층의 사용자들**이 효율적으로 사용할 수 있도록 한 데이터베이스  조직의 중심이 되는 업무 시스템에서 모아진 정보를 일관된 스키마로 저장한 **저장소**  **의사결정용 데이터베이스** |
| 데이터 마이닝을 위해 OLAP 연산을 지원도구로 사용한 저장소가 데이터 웨어하우스 | |
| 데이터 마이닝 | 대규모로 저장된 데이터 안에서 체계적이고 자동적으로 통계적 규칙이나 패턴을 찾아내는 기술  데이터 웨어하우스에 저장된 데이터 집합에서 사용자의 요구에 따라 정보를 발견하기 위한 기법 |
| 군집화 | 데이터 마이닝의 기법 중 하나. 상호간 유사한 특성을 갖는 데이터들을 집단화 하는 방법 |
| **OLTP**  (OnLine Transaction Processing) | **온라인 업무 처리 형태**의 하나로 네트워크 상의 여러 이용자가 **실시간**으로 DB를 갱신하거나 검색하는 등 단위 작업을 처리하는 방식  **트랜잭션**을 수집하고 분류, 저장, 유지, 갱신, 검색하는 기능을 수행하는 **실시간 거래 처리 시스템** |

프레임워크 / 라이브러리 / API

|  |  |
| --- | --- |
| **프레임워크** | 소프트웨어의 구체적 설계 구현을 클래스로 제공해 도움을 줌 (어떻게 개발할지 결정 틀) |
| 프레임워크의 특성 | [**재확모역**] 재사용 / 확장성 / 모듈화 / 제어의 역흐름 |
| **라이브러리** | 미리 함수들을 만들어 두고 편리하게 사용하게 제공함 (필요한 것을 빌려 쓰는 대상) |
| **API** | 프로그램 또는 애플리케이션이 처리를 위해 호출하는 함수의 집합 (라이브러리 + 명세)  데이터를 송신 시스템의 DB에서 읽어와 제공하는 애플리케이션 프로그래밍 인터페이스 프로그램 |

웹 서비스 구성요소

|  |  |
| --- | --- |
| **SOAP** | HTTP, HTTPS, SMTP 등을 활용해 XML 기반의 메세지를 네트워크 상에서 교환하는 프로토콜 |
| **WSDL** | 웹 서비스 명, 서비스 제공 위치, 프로토콜 등 웹 서비스에 대한 상세 정보를 XML 형식으로 구현 |
| **UDDI** | WSDL을 등록하여 서비스와 서비스 제공자를 검색하고 접근하는데 사용하는 레지스트리 |

\_\_ML

|  |  |
| --- | --- |
| **HTML** | 하이퍼텍스트 문서를 작성하기 위해 사용하는 마크업 언어 |
| **UML** | 객체지향 모델링 언어 |
| **XML** | W3C에서 개발한 다른 특수한 목적의 마크업 언어를 만들기 위해서 사용하는 다목적 마크업 언어. HTML의 한계를 극복할 목적으로 만들어진 언어. |
| **XrML** | 디지털 콘텐츠. 웹서비스 권리 조건을 표현한 XML 기반의 마크업 언어 |

클라우드 컴퓨팅 유형

|  |  |
| --- | --- |
| 클라우드 컴퓨팅 | 인터넷을 통해 가상화된 컴퓨터 시스템 리소스를 제공 정보를 자신의 컴퓨터가 아닌 클라우드에 연결된 다른 컴퓨터로 처리하는 기술 |
| **IaaS (인프라형 서비스)** | 서버, 스토리지 같은 **시스템 자원**(인프라)을 클라우드로 제공하는 서비스  서버에 부하가 많아 성능 저하 우려가 있을 때 하드웨어 자원들을 빌려와 부하를 감소시킴 |
| **PaaS (플랫폼형 서비스)** | 애플리케이션을 개발, 실행, 관리 할 수 있게 하는 **플랫폼**을 제공하는 서비스  인프라를 생성, 관리하는 복잡함 없음 |
| **SaaS (소프트웨어형 서비스)** | 주문형 소프트웨어. 제품이 아닌 빌려 쓰는 모델  SW 및 관련 데이터는 중앙에 호스팅되고 사용자는 클라이언트를 통해 접속하여 이용하는 서비스  SW의 여러 기능 중 사용자가 필요로 하는 서비스만 이용 가능 |

UNIX

|  |  |
| --- | --- |
| 유닉스 구성요소 (5단계) | **하**드웨어 - **커**널 - **쉘** - 유틸리티 - 사용자 |
| **커널(Kernel)** | 유닉스의 구성요소 중 가장 **핵심**이며 프로그램-하드웨어 사이 **인터페이스** 역할을 담당하는 하드웨어 보호기 |
| **쉘(Shell)** | 유닉스의 구성요소 중 시스템-사용자 간 인터페이스를 담당하는 사용자의 **명령어 해석기** |

키

|  |  |
| --- | --- |
| **기본키 (주키)** | 유일성, 최소성, 대표성을 만족하는 키 |
| **후보키** | 기본키로 사용할 수 있는 속성들 |
| **대체키** | 후보키 – 기본키 |
| **슈퍼키** | 유일성O 최소성X. 속성들의 집합 |
| **외래키** | 릴레이션 간 참조 무결성을 보장하기 위해 사용 |

무결성

|  |  |
| --- | --- |
| 무결성 | DB에 저장된 데이터 값과 그것이 표현하는 현실 세계의 실제 값이 일치하는 정확성 |
| **개체 무결성** | 기본키를 구성하는 어떤 속성도 NULL값이나 중복된 값을 가질 수 없다 |
| **도메인 무결성** | 주어진 속성값이 정의된 도메인에 속한 값이여야 한다 |
| **참조 무결성** | 릴레이션은 참조할 수 없는 외래키 값을 가질 수 없다 |

스키마

|  |  |
| --- | --- |
| 스키마 | DB 구조와 제약 조건에 관한 전반적인 명세 |
| **외부 스키마 (서브 스키마)** | 사용자나 응용 프로그래머가 개인의 입장에서 필요로 하는 **DB의 논리적인 구조**를 정의 |
| **개념 스키마** | 단순한 스키마. **개체 간의 관계와 제약 조건**을 나타내고 명세를 정의. 전체적인 논리적 구조 |
| **참조 무결성** | **DB의 물리적 구조**를 정의 |

옵티마이저

|  |  |
| --- | --- |
| **옵티마이저** | SQL을 가장 빠르고 효율적으로 수행할 최적의 처리 경호를 생성해주는 DBMS 핵심엔진. 모듈 |
| **실행 계획** | 옵티마이저가 SQL을 가장 빠르고 효율적으로 수행할 수 있도록 생성한 최적의 처리 경로 |
| **힌트(Hint)** | 옵티마이저가 비정상적으로 수립한 계획에 엑세스 경로 및 조인순서를 제어할 수 있도록 하기 위한 기법. 항상 최적으로 실행계획을 수립하기 위해 명시적인 이것을 사용. |
| 옵티마이저 종류 (2가지) | RBO(Rule-Based Optimization) / CBO(Cost-Based Optimiation) |
| **RBO** | 통계정보가 없는 상태에서 사전에 등록된 규칙에 따라 질의 실행 계획을 선택하는 옵티마이저 |
| **CBO** | 통계정보로부터 모든 접근 경로를 고려한 질의 실행 계획을 선택하는 옵티마이저 |

네트워크 연결 방식

|  |  |
| --- | --- |
| **DAS (Direct Area Network)** | 서버와 저장장치를 전용케이블로 직접 연결하는 방식 |
| **NAS (Network Attached Storage)** | 서버와 저장장치를 네트워크로 연결하는 방식 |
| **SAN (Storage Attached Network)** | 서버와 스토리지를 광케이블 및 광채널스위치를 통해 근거리 네트워크 환경을 구성. 빠른 속도로 데이터를 처리할 수 있는 특수 목적용 고속 네트워크 |

DTO / VO / DAO

|  |  |
| --- | --- |
| **DTO** | 프로세스 사이에서 데이터를 전송하는 객체. 데이터 저장 회수 외 다른 기능 없음 |
| **VO** | 간단한 엔티티를 의미하는 작은 객체가 클래스인 DTO와 달리 고정 클래스를 가짐 |
| **DAO** | 특정 타입의 DB에 추상 인터페이스를 제공하는 객체. 세부내용 노출없이 데이터를 조작 |

UNIX 명령어

|  |  |
| --- | --- |
| **cat** | 파일 내용을 화면에 표시하는 명령어 |
| **chmod** | 특정 파일 또는 디렉토리 퍼미션 수정 명령어 |
| **chown** | 파일이나 디렉토리의 소유자, 소유 그룹 수정 명령어 |
| **fsck** | 파일 시스템을 검사 및 보수하여 무결성을 검사하는 명령어 |

암호 알고리즘

|  |  |  |
| --- | --- | --- |
| **DES** | 대칭키 (블록 방식) | **미국 표준**(NIST) 암호화 알고리즘  **56bit**의 키를 이용하여 64bit의 평문 블록을 64bit의 암호문 블록으로 만듦 |
| AES | 대칭키 (블록 방식) | DES를 대체하는 고급 암호화 알고리즘 |
| **SEED** | 대칭키 (블록 방식) | 한국(**정보보호진흥원**)에서 개발한 128bit 대칭키 암호화 알고리즘  **전자상거래**, 금융, 무선통신 등에서 전송되는 개인정보를 보호하기 위해 **국내**서 개발 |
| ARIA | 대칭키 (블록 방식) | 한국(국가보안기술연구소)에서 개발한 128, 192, 256 3종류의 양방향 알고리즘 |
| RC4 | 대칭키 (스트림 방식) |  |
| LFSR | 대칭키 (스트림 방식) |  |
| **RSA** | 비대칭키 | **소인수분해**의 어려움을 이용한 방식. **가장 널리 쓰이는** 비대칭 키 암호 방식  오늘날 양자 컴퓨터의 등장으로 보안 위협에 직면해있음 |
| SHA | 해시 | 미국(국가안보국.NSA)에서 개발한 국가 표준 해시 암호화 알고리즘 |
| **MD5** | 해시 | MD4를 대체하는 128bit 해시 암호화 알고리즘 |
| QKD | | 양자 통신을 위해 비밀키를 분배하고 관리하는 기술. 비밀키를 안전하게 공유 |
| PKI (공개키 기반 구조) | | 공개키 암호 방식으로 디지털 인증서를 활용하는 SW, HW, 사용자, 정책 제도를 총칭 |

페이지 교체 알고리즘

|  |  |
| --- | --- |
| **OPT** | 앞으로 가장 오랫동안 사용하지 않을 페이지를 교체 |
| **FIFO** | First In First Out. **가장 오래 존재**한 페이지를 교체. FCFS(First Come First Served) |
| **LRU** | Least Recently Used. **가장 오래 사용하지 않은** 페이지를 교체 |
| **LFU** | Least Frequncy Used. **가장 자주 사용하지 않은** 페이지를 교체 |
| **NUR** | Not Used Recently. **최근에 사용하지 않은** 페이지를 교체 |
| **SCR** | Second Chance. **2번째 기회**를 주어서 페이지를 교체 (FIFO의 단점 보완) |

DRM

|  |  |
| --- | --- |
| **DRM** | **디지털 저작권 관리**. 저작권자가 의도한 용도로만 사용 되도록 통제함 |
| **클리어링 하우스** | 저작권에 대한 **사용 권한과 라이선스를 발급**, 사용량에 따른 결제 관리 등을 수행 |
| **패키저 (Packager)** | 콘텐츠를 메타 데이터와 함께 **배포 가능한 단위로 묶**는 기능 (+암호화) |
| **메타데이터** | **데이터에 관한 데이터**. 속성 정보를 설명 |
| 메타데이터 레지스트리 (MDR) | 메타 데이터의 등록과 인증을 통해 유지.관리.명세를 공유하는 레지스트리 |
| **DRM 컨트롤러** | 배포된 콘텐츠에 대한 **이용권한을 통제** |
| **보안 컨테이너** | 안전 **유통**을 위한 보안장치 |

비용 산정 기법

|  |  |  |
| --- | --- | --- |
| **전문가 감정 기법** | 하향식 | 조직 내 경험이 많은 전문가가 참여하는 비용 산정 기법  편리하고 신속하지만 유사경험이 없는 경우에는 난감. 개인적이고 주관적 |
| **델파이 기법** | 하향식 | **전문가** 감정 기법을 보완한 방법  다수의 전문가와 한 명의 중재자를 통해 비용을 산정하는 **하향식 비용 산정 기법** |
| **LOC 기법** | 상향식 | 원시 코드 라인 수 |
| **Man Month 기법** | 상향식 | 개발 단계별 인월 수 기법. LOC기법의 단점을 보완한 방법 |
| **COCOMO 모형** | 상향식 | 과거를 기반으로 자동화하는 수학적 산정 기법 중 하나. **보헴**이 제안  라인 수에 따라 조직형(~5만). 반분리형(5~30만). 내장형(30만~) 세 모드로 구분 |
| **Putumn 모형** | 상향식 | 수학적 산정 기법 중 하나. 노력의 분포를 가정 |
| **FP 모형** | 상향식 | 수학적 산정 기법 중 하나. 기능 점수(Function Point)를 이용 |

병행 제어

|  |  |
| --- | --- |
| **병행 제어** | 다중 프로그램의 이즘을 활용하여 동시에 여러 개의 트랜잭션을 병행 수행 시 트랜잭션 간의 상호작용을 제어하는 기술. 기법으로 로킹과 타임스탬프가 있음 |
| **로킹(Locking)기법** | 하나의 트랜잭션이 데이터를 액세스하는 동안, **다른 트랜잭션이 그 데이터 항목을 엑세스할 수 없도록 제어**하는 기법 |
| **타임 스탬프 기법** | 트랜잭션이 실행하기 전에 **시간표**를 부여해 부여된 시간에 따라 트랜잭션 작업을 수행하는 기법 |

신기술 (기출)

|  |  |
| --- | --- |
| **SSO (Single Sign-On)** | 한 번의 로그인으로 개인이 가입한 모든 사이트를 이용할 수 있게 해주는 시스템 |
| **RAID** | 여러 개의 **하드디스크**에 일부 중복된 데이터를 서로 다른 디스크에 **분산 저장**하는 기술 |
| **M-DISC** | 한번의 기록만으로 **자료를 영구적으로 저장**이 가능한 **광디스크** 기술 |
| **MEMS** | **초정밀 반도체** 제조기술을 바탕으로 한 초 미세 장치 |
| **지그비** | 저속 전송 속도를 갖는 **홈 오토메이션** 및 데이터 네트워크를 위한 표준 기술 |
| **커넥티드 카** | 자동차와 정보통신기술을 연결. 양방향 인터넷 및 모바일 서비스 이용이 가능한 차량 |
| **킬 스위치** | 스마트폰 이용자가 도난당한 스마트폰의 작동을 웹사이트를 통해 정지시킬 수 있는 자폭 기능 |
| **아이핀** | 인터넷에서 주민등록번호 대신 쓸 수 있도록 만든 사이버 주민등록번호 |
| **BLE** | 저전력 블루투스 기술. 연결되지 않은 대기 상태에서는 절전 모드로 유지됨 |
| **그레이웨어** | 제공하는 입장과 달리 사용자 입장에서는 유용할 수도 있고 악의적일 수도 있는 악성 소프트웨어  정상 SW의 이미지인 백색과 악성 SW의 이미지인 흑색의 중간 |
| **컴패니언 스크린** | N screen의 한 종류. TV방송 시청 시 추가 기능을 수행할 수 있는 스마트폰, 태플릿PC, 세컨드 스크린 |
| **블록체인** | P2P 네트워크를 이용하여 **온라인 금융 거래 정보**를 온라인 네트워크 참여자의 디지털 **장비에 분산 저장**하는 기술. 예시로 비트코인 |
| **노모포비아** | 스마트폰이 곁에 없을 때 불안감이나 두려움을 느끼는 증상 |
| **인포러스트** | 정보 열광자. 필요한 정보를 기다리지 않고 적극적으로 찾아내고 그것을 자신의 홈페이지나 SNS를 통해 대중화시킴 |
| **크리슈머** | 단순히 제품 사용에 그치지 않고 제품의 기능이나 가치에 스토리를 담아 기존의 콘텐츠를 발전시켜 새로운 문화와 소비 유행을 만들어가는 사람들. 프로슈머보다 발전한 개념 |
| **클라우드 소싱** | 군중+아웃소싱. 클라우드 컴퓨팅이 실용화되면서 가능하게 된 정보기술 아웃소싱 전략 |

시스템 (기출)

|  |  |
| --- | --- |
| **DSS (Decision Support System)** | 컴퓨터를 사용하여 정형화 되지 않는 문제에 관해 의사결정자가 효과적인 **의사결정**을 할 수 있도록 **지원**하는 **시스템** |
| **SIS (Strategic Information System)** | 기업이 경쟁 우위 확보를 위해 **전략적**으로 구축하는 **정보 시스템** |
| **CMS (Contents Management System)** | 다양한 미디어 포맷에 따라 각종 **콘텐츠**를 작성.수집.관리.배급(생산~활용~폐기) 전 공급과정을 **관리**하는 **기술** |
| **GLONASS (글로나스)** | **러시아의 전파 위성 항법 시스템** |
| **DOI (Digital Object Identifer)** | 디지털저작물의 저작권보호 및 정확한 위치추적이 가능한 시스템 |
| **TPS (Transaction Processing System)** | 기업에서 업무 처리 중 발생하는 **거래** 자료를 전산화해서 신속하고 정확하게 **처리**하는 정보화 **시스템** |
| **CALS (광속 상거래)** | 1985년 미국 국방부에서 군수품 납품 체계를 개선할 목적으로 도입 모든 기술정보 등을 통합 DB로 관리. 오늘 날 전자 상거래로 개념 확대 |
| **GIS (Geographic Information System)** | **지리적인 자료**를 수집.저장.분석.출력할 수 있는 컴퓨터 응용 **시스템** |
| **LBS (Location Based Service)** | 기능을 제어하기 위해 **위치 데이터**를 사용하는 프로그램 **단계 서비스** |
| **VoC (Voice of Customer)** | 시스템 수준 향상 목적으로 **고객의 의견**이나 불만사항을 실시간으로 관리.평가하는 고객 관리 시스템 |

<학생> 테이블에 최대 3문자로 구성되는 학년 속성을 추가  
**ALTER** **TABLE** 학생 **ADD** 학년 VARCHAR(3);

<학생> 테이블에 학번이 98170, 성명이 ‘한국산’, 학년이 3, 과목이 ‘경영학개론’, 연락처가 ‘?-1234-1234’인 학생의 정보 입력  
**INSERT INTO** 학생 **VALUES(**98170, ‘한국산’, 3, ‘경영학개론’, ‘?-1234-1234’**)**;

<학생> 테이블에서 3학년과 4학년 학생의 학번과 이름을 검색  
**SELECT** 학번, 이름 **FROM** 학생 **WHERE** 학년 **IN(**3,4**)**;

<학생> 테이블에서 이름이 ‘이’로 시작하는 학생들의 학번을 검색하되, 학년을 기준으로 내림차순 출력  
**SELECT** 학번 **FROM** 학생 **WHERE** 이름 LIKE ‘이%’ **ORDER BY** 학년 DESC;

<학생> 테이블에서 4학년인 학생의 수를 출력  
**SELECT** COUNT(\*) **FROM** 학생 **WHERE** 학년=4;

<학생> 테이블에서 학과가 전산과인 튜플을 골라 학과 중복없이 출력  
**SELECT** COUNT(**DISTINCT** 학과) **FROM** 학생 **WHERE** 학과=‘전산과’;

<학생> 테이블에서 이름이 Scott인 튜플을 삭제  
**DELETE** **FROM** 학생 **WHERE** 이름=‘Scott’;

길동에게 <학생> 테이블에 대한 모든 권한과 다름 사람에게도 권한을 부여할 수 있는 권한까지 부여  
**GRANT** ALL **ON** 학생 **TO** 길동 WITH GRANT OPTION;

길동에게 부여된 <학생> 테이블에 대한 권한은 유지하고, 다른 사용자에게 권한을 부여할 수 있는 권한만 취소  
**REVOKE** GRANT OPTION FOR ALL **ON** 학생 **FROM** 길동;