1과목 : 소프트웨어 설계

Chapter 1: 요구사항 확인

Section 1: 소프트웨어 생명주기

Q. 애자일 방법론에 해당하지 않는 것은**?**

1. 기능 중심 개발
2. 스크럼
3. 익스트림 프로그래밍
4. 모듈 중심 개발

정답 :

해설 :

애자일 모형

'민첩한', '기민한' 이라는 의미로 고객의 요구사항 변화에 유연하게 대응할 수 있도록 일정한 주기를 반복하면서 개발과정을 진행하는 개발 방법론

애자일 모형

스크럼(Scrum)
XP(eXtreme Programming)
칸반(Kanban)
Lean
크리스탈(CrystaL)
기능중심개발(FDD: Feature Driven Development)
DSDM(Dynamic System Development Method)
DAD(Disciplined Agile Delivery)

Section 2: 스크럼(Scrum) 기법

Q. 다음의 스크럼**(Scrum)** 개발 과정을 진행 순서에 맞게 올바르게 나열한 것은**?**

ㄱ. 스프린트(sprint)
ㄴ. 스프린트 회고(Spring Retrospective)
ㄷ. 일일 스크럼 회의(Daily Scrum Meet)
ㄹ. 스프린트 검토 회의
ㅁ. 스프린트 계획 회의

1. ㅁ - ㄷ - ㄱ - ㄴ - ㄹ
2. ㅁ - ㄱ - ㄷ - ㄹ - ㄴ
3. ㅁ - ㄷ - ㄱ - ㄹ - ㄴ
4. ㅁ - ㄹ - ㄱ - ㄴ - ㄷ

정답 :

해설 : 계획한 내용을 토대로 일정 기간 동안 스프린트를 수행하면서 진행 상황을
매일 점검하고 하나의 스프린트가 끝나면 검토한 후 진행을 되돌아본다.

Section 3: XP(eXtreme Programming) 기법

Q. XP(extreme Programming)의 기본 원리로 볼 수 없는 것은?

1. Linear Sequential Method
2. Pair Programming
3. Collective Ownership
4. Continuous Integration

정답 :

해설 : XP의 5가지 핵심가치(의사소통, 단순성, 용기, 존중, 피드백),
XP 의 주요실천방법 - 짝 프로그래밍, 공동 코드 소유, 테스트 주도 개발
전체 팀, 계속적인 통합, 리팩토링, 소규모 릴리즈 등
문제의 Linear Sequential Method: 순차적 방법으로 기본 원리가 아니다
Pair Programming: 개발 코드에 대한 권한과 책임을 공동으로 소유
Collective Ownership: 소스에 대한 팀의 공통책임, 누구든 수정 가능함
Continuous Integration: 컴포넌트 또는 모듈 단위로 지속적으로 통합,

Section 9: UML(Unified Modeling Language)

Q. 럼바우**(Rumbaugh)** 객체지향 분석 기법에서 동적 모델링에 활용되는 다이어그램은**?**

1. 객체 다이어그램(Object Diagram)
2. 패키지 다이어그램(Package Diagram)
3. 상태 다이어그램(State Diagram)
4. 자료 흐름도(Data Flow Diagram)

정답 :
해설 : 럼바우 객체지향 분석 기법과 관련된 다이어그램 두가지

- 정적모델링에 활용되는 구조적 다이어그램 : 객체 다이어그램(Object Diagram)
- 동적 모델링에 활용되는 행위 다이어그램 : 상태(State Diagram)

Q. UML(Unified Modeling Language)에 대한 설명 중 틀린 것은**?**

1. 기능적 모델은 사용자 측면에서 본 시스템 기능이며 UML에서는
Use case Diagram을 사용한다.

2. 정적 모델은 객체, 속성, 연관관계, 오퍼레이션의 시스템 구조를 나타내며, UML에서는 Class Diagram을 사용한다.
3. 동적 모델은 시스템의 내부 동작을 말하며, UML에서는 Sequence D다.
4. 정적 모델은 객체, 속성, 연관관계, 오퍼레이션의 시스템 구조를 나타내며, UML에서는 Class Diagram을 사용한다.

Section 10: 주요 UML 다이어그램

Q. UML에서 시퀀스 다이어그램의 구성 항목에 해당하지 않는 것은**?**

1. 생명선
2. 실행
3. 확장
4. 메시지

정답 :

해설 : 시퀀스다이어 다이어 그램(순차 다이어그램) - 시간의 흐름에 따른 상호작용 과정을 표현
구성요소 - 액터(Actor) : 서비스를 요청하는 외부요소, 사람이나 외부 시스템
객체(object) : 메시지를 주고받는 주체
생명선(Lifeline) : 객체가 메모리에 존재하는 기간, 객체 아래쪽에 점선을 그어 표현
실행 상자(Active Box): 객체가 메시지를 주고받으며 구동되고 있음을 표현함
메시지(Message): 객체가 상호 작용을 위해 주고받는 메시지

Chapter 2: 화면 설계

Section 13: UI 설계 도구

Q. 다음 내용이 설명하는 **UI** 설계 도구는**?**

- 디자인, 사용 방법 설명, 평가 등을 위해 실제 화면과 유사하게 만든 정적인 형태의 모형
- 시각적으로만 구성 요소를 배치하는 것으로 일반적으로 실재로 구현되지는 않음

1. 스토리보드(StoryBoard)
2. 목업(Mockup)
3. 프로토타입(Prototype)
4. 유스케이스(Usecase)

정답 :

해설 : 핵심은 “시각적으로만 배치하는 것으로 실제 기능은 구현되지 않는다”는 것

- 와이어프레임 : 기획 초기에 제작하는 것으로 페이지에 대한 개략적인 레이아웃, UI 요소 등에 대한 뼈대를 설계하는 단계
- 스토리보드 : 와이어 프레임에 콘텐츠에 대한 설명, 페이지 간 이동 흐름 등을 추가한 문서
- 목업: 디자인, 사용방법 설명, 평가 등을 위해 와이어프레임보다 좀 더 실제화면과 유사하게 만든 정적인 형태의 모형
- 프로토타입 : 와이어프레임이나 스토리보드 등에 “인터렉션”을 적용함으로써 실제 구현된 것처럼 테스트 가능한 동적인 형태의 모형이다.
- 유스케이스 : 사용자 측면에서의 요구사항으로, 사용자가 원하는 목표를 달성하기 위해 수행할 내용을 기술한다.

Section 15: 품질 요구 사항

Q. ISO/IEC 9126의 소프트웨어 품질 특성 중 기능성**(Functionality)**의 하위 특성으로 옳지 않은 것은**?**

1. 학습성
2. 적합성
3. 정확성
4. 보안성

정답 :

해설 : 기능성의 하위 특성에는 적합성, 정확성, 상호 운용성, 보안성, 준수성 등이 있다.

Q. 패키지 소프트웨어의 일반적인 제품 품질 요구사항 및 테스트를 위한 국제 표준은**?**

1. ISO/IEC 2192
2. IEEE 19554
3. ISO/IEC 12119
4. ISO/IEC 14959

정답 :

해설 : “테스트 절차”가 포함된 국제 표준은 ISO/IEC 12119이다.

Q. 소프트웨어 품질 목표중 하나 이상의 하드웨어 환경에서 운용되기 위해 쉽게 수정될 수 있는 시스템 능력을 의미하는 것은**?**

1. Portability
2. Efficiency
3. Usability
4. Correctness

정답 :

해설 : Portability - 이식성, Efficiency - 효율성, Usability - 유용성, Correctness - 정확성

Chapter 3: 애플리케이션 설계

Section 21: 소프트웨어 아키텍처

Q. 다음 **( )** 안에 들어갈 내용으로 옳은 것은**?**

컴포넌트 설계 시 “( )에 의한 설계” 를 따를 경우 해당 명세서에는

(1) 컴포넌트의 오퍼레이션 사용전에 참이 되어야할 선행조건
(2) 사용 후 만족 되어야 할 결과조건
(3) 오퍼레이션이 실행되는 동안 항상 만족되어야 할 불변조건 등이 포함되어야 한다.

1. 협약(Contract)
2. 프로토콜(Protocol)
3. 패턴(Pattern)
4. 관계(Relation)

정답 :

해설 : 협약(Contract)에 의한 설계 - 컴포넌트를 설계할 때 클래스에 대한 여러 가정을 공유할 수 있도록 명세한 것으로, 소프트웨어 컴포넌트에 대한 정확한 인터페이스를 명세한다.
협약에 의한 설계 시 명세에 포함될 조건에는 선행 조건, 결과 조건, 불변 조건이 있다.

Q. 소프트웨어 아키텍처 설계에서 시스템 품질 속성이 아닌 것은**?.**

1. 가용성(Availability)
2. 독립성(Isolation)

3. 변경 용이성(Modifiability)
4. 사용성(Usability)

정답 :

해설 : 시스템 품질 속성에는 성능, 변경 용이성, 사용성, 기능성, 가용성, 확장성, 보안성 등이 있다.

Q. 아키텍처 설계 과정이 올바른 순서로 나열된 것은**?**

(가) 설계 목표 설정
(나) 시스템 타입 결정
(다) 스타일 적용 및 커스터마이즈
(라) 서브시스템의 기능, 인터페이스 동작 작성
(마) 아키텍처 설계 검토

1. (가) - (나) - (다) - (라) - (마)
2. (마) - (가) - (나) - (라) - (다)
3. (가) - (마) - (나) - (라) - (다)
4. (가) - (나) - (다) - (마) - (라)

정답 :

해설 : 먼저 목표를 설정하고 타입을 결정한 후 스타일**(**패턴**)**을 적용한다. 이어서 서브시스템을 구체화하고 아키텍처를 최종 검토한다.

Section 22: 아키텍처 패턴

Q. 서브시스템이 입력 데이터를 받아 처리하고 결과를 다른 시스템에 보내는 작업이 반복되는 아키텍처 스타일은**?**

1. 클라이언트 서버 구조
2. 계층 구조
3. MVC 구조
4. 파이프 필터 구조

정답 :

해설 : 시스템이 파이프처럼 연결되어 있어서 앞 시스템의 처리 결과물을 파이프를 통해 전달받아 처리한 후 다음 시스템으로 넘겨주는 패턴을 반복하는 아키텍처 스타일을 파이프-필터 구조라고한다.

Q. 네트워크 프로토콜의 **OSI** 참조 모델과 가장 관련이 깊은 아키텍처 모델은**?**

1. Peer-To-Peer Model
2. Mvc Model
3. Layers Model
4. Client-Server Model

정답 :

해설 : OSI 참조 모델은 네트워크 프로토콜을 계층(Layer)별로 구분한 모델이다.

Section 24: 객체지향 분석 및 설계

Q. 객체지향 분석 방법론 중 **Coad-Yourdon** 방법에 해당하는 것은**?**

1. E-R 다이어그램을 사용하여 객체의 행위를 데이터 모델링하는데 초점을 둔 방법이다.
2. 객체, 동적, 기능 모델로 나누어 수행하는 방법이다.
3. 미시적 개발 프로세스와 거시적 개발 프로세스를 모두 사용하는 방법이다.
4. Use-Case를 강조하여 사용하는 방법이다.

정답 :

해설 : E-R 다이어그램은 Coad-Yourdon, 미시적과 거시적은 Booch,
Use-Case 는 Jacobson, 객체-동적-기능 모델은 Rumbaugh

Section 25: 모듈

Q. 다음 중 가장 강한 응집도**(Cohesion)**는**?**

1. Sequential Cohesion
2. Procedural Cohesion
3. Logical Cohesion
4. Coincidental Cohesion

정답 :

해설 : 응집도가 강한것부터(기능적 - 순차적 - 교환(통환)적 - 절차적 - 시간적 - 논리적 - 우연적)
Functional - Sequential - Communication - Procedural - Temporal - Logical

결합도의 경우 약한것부터(자료-스탬프(검인) - 제어 - 외부 - 공통(공유) - 내용)
Data - Stamp - Control - External - Common - Content

Section 28: 디자인 패턴

Q. GoF(Gang of Four) 디자인 패턴을 생성**,** 구조**,** 행동 패턴의 세그룹으로 분류할 때**,**
구조 패턴이 아닌 것은**?**

1. Adapter 패턴
2. Bridge 패턴
3. Builder 패턴
4. Proxy 패턴

정답 :

해설 : Builder는 '생성' 패턴이다.

Chapter 4: 인터페이스 설계

Section 30: 인터페이스 요구사항 검증

Q. 인터페이스 요구사항 검토 방법에 대한 설명이 옳은 것은**?**

1. 리팩토링 : 작성자 이외의 전문 검토 그룹이 요구사항 명세서를 상세히 조사하여 결함, 표준 위배, 문제점 등을 파악
2. 동료검토 : 요구사항 명세서 작성자가 요구사항 명세서를 설명하고 이해관계자들이 설명을 들으면서 결함을 발견
3. 인스펙션 : 자동화된 요구사항 관리 도구를 이용하여 요구사항 추적성과 일관성을 검토토
4. CASE 도구 : 검토 자료를 회의 전에 배포해서 사전 검토한 후 짧은 시간 동안 검토 회의를 진행하면서 결함을 발견견

정답 :

해설 : 요구사항 검토 방법의 핵심 키워드
동료검토 - '작성자가 명세서 내용을 직접 설명'
워크스루 - '명세서를 미리 배포'
인스펙션 - '검토 전문가들이 명세서 확인'
프로토타이핑 - '견본품(Prototype)을 통한 결과물 예측'
테스트 설계 - '테스트 케이스를 생성'

CASE 도구 - ‘일관성 분석, 추적 및 분석, 관리’

Section 35: 미들웨어 솔루션 명세

Q. 메시지 지향 미들웨어**(Message-Oriented Middleware, MOM)**에 대한 설명으로 틀린 것은**?**

1. 느리고 안정적인 응답보다는 즉각적인 응답이 필요한 온라인 업무에 적합한다.
2. 독립적인 애플리케이션을 하나의 통합된 시스템으로 묶기 위한 역할을 한다.
3. 송신측과 수신측의 연결 시 메시지 큐를 활용하는 방법이 있다.
4. 상이한 애플리케이션 간 통신을 비동기 방식으로 지원한다.

정답 :

해설 : MOM은 온라인 업무보다는 이기종 분산 데이터 시스템의 데이터 동기를 위해 많이 사용된다.

2과목 **:** 소프트웨어 개발

Chapter 1: 데이터 입출력 구현

Section 36: 자료 구조

Q. 효율적인 프로그램을 작성할 때 가장 우선적인 고려사항은 저장 공간의 효율성과 실행시간의 신속성이다**.** 자료 구조의 선택은 프로그램 실행시간에 직접적인 영향을 준다**.** 자료 구조에 관한 설명으로 거리가 먼 것은**?**

1. 자료 구조는 자료의 표현과 그것과 관련된 연산이다.
2. 자료 구조는 일련의 자료들을 조직하고 구조화하는 것이다.
3. 어떠한 자료 구조에서도 필요한 모든 연산들을 처리하는 것이 가능하다.
4. 처리할 문제가 주어지면 평소에 주로 사용하던 자료 구조를 적용하는 것이 좋다.

정답 :

해설 : 자료 구조에 따라 프로그램 실행시간이 달라지므로 처리할 문제에 어울리는 적절한 자료 구조를 선택하는것이 좋다.

Q. 다음 중 스택을 이용한 연산과 거리가 먼 것은**?**

1. 선택 정렬
2. 재귀 호출
3. 후위 표현(Post-Fix Expression)의 연산
4. 깊이 우선 탐색

정답 :

해설 : 스택(Stack)을 이용한 연산은 '재귀 호출, 후위(Postfix) 표기법, 깊이 우선 탐색'과 같이 왔던 길을 되돌아가는 경우에 사용합니다.

Q. 순서가 **A, B, C, D**로 정해진 입력 자료를 스택에 입력한 후 출력한 결과로 불가능한 것은**?**

1. D, C, B, A
2. B, C, D, A
3. C, B, A, D
4. D, B, C, A

정답 :

해설 : 직접 해볼것

Section 40: 데이터베이스 개요

Q. 데이터베이스 관리 시스템**(DBMS)**의 주요 필수 기능과 거리가 먼 것은**?**

1. 데이터베이스 구조를 정의할 수 있는 정의 기능
2. 데이터 사용자의 통제 및 보안 기능
3. 데이터베이스 내용의 정확성과 안정성을 유지할 수 있는 제어 기능
4. 데이터 조작어로 데이터베이스를 조작할 수 있는 조작 기능

정답 :

해설 : 주요 필수 기능 - 정의(DDL), 조작(DML), 제어(DCL)

Q. 데이터베이스 관리 시스템**(DBMS)**의 필수 기능 중 제어 기능에 대한 설명으로 거리가 먼 것은**?**

1. 데이터베이스를 접근하는 갱신, 삽입, 삭제 작업이 정확하게 수행되어 데이터의 무결성이 유지되도록 제어해야 한다.
2. 데이터의 논리적 구조와 물리적 구조 사이에 변환이 가능하도록, 두 구조 사이의 사상(Mapping)을 명시하여야 한다.
3. 정당한 사용자가 허가된 데이터만 접근할 수 있도록 보안(Security)을 유지하고 권한(Authority)을 검사할 수 있어야 한다.
4. 여러 사용자가 데이터베이스를 동시에 접근하여 데이터를 처리할 때 처리 결과가 항상 정확성을 유지하도록 병행 제어(Concurrency Control)를 할 수 있어야 한다.

정답 :

해설 : 정답은 데이터베이스를 생성하기 위한 '정의' 기능에 해당,
'제어' 기능의 핵심은 무결성, 보안, 권한, 병행 제어이다.

Chapter 2: 제품 소프트웨어 패키징

Section 46: 소프트웨어 패키징

Q. 소프트웨어 패키징에 대한 설명으로 틀린 것은**?**

1. 패키징은 개발자 중심으로 진행한다.
2. 신규 및 변경 개발소스를 식별하고, 이를 모듈화하여 상용제품으로 패킹지한다.
3. 고객의 편의성을 위해 메뉴얼 및 버전관리를 지속적으로 한다.
4. 범용 환경에서 사용이 가능하도록 일반적인 배포 형태로 패키징이 진행된다.

정답 :

해설 : 소프트웨어를 설계하거나 개발할 떄 그리고 개발된 소프트웨어를 패키징 할 때 까지도 모든 과정에서 가장 먼저 고려되어야할 대상은 소프트웨어를 사용할 사용자이다.

Section 48: 디지털 저작권 관리(DRM)

Q. 디지털 저작권 관리**(DRM)** 구성 요소가 아닌 것은**?**

1. Dataware House
2. DRM Controller
3. Packager
4. Contents Distributor

정답 :

해설 : 디지털 저작권 관리의 구성 요소는 클리어링 하우스(Clearing House)이다.

Section 51: 소프트웨어 버전 등록

Q. 소프트웨어 형상 관리에 대한 설명으로 거리가 먼 것은**?**

1. 소프트웨어에 가해지는 변경을 제어하고 관리한다.
2. 프로젝트 계획, 분석서, 설계서, 프로그램, 테스트 케이스 모두 관리 대상이다.
3. 대표적인 형상 관리 도구로 Ant, Maven, Gradle 등이 있다.
4. 유지 보수 단계뿐만 아니라 개발 단계에도 적용할 수 있다.

정답 :

해설 : Ant, Maven, Gradle은 빌드 자동화 도구이다.

Q. 소프트웨어 형상 관리**(Configuration Management)**에 관한 설명으로 틀린 것은**?**

1. 소프트웨어에서 일어나는 수정이나 변경을 알아내고 제어하는 것을 의미한다.
2. 소프트웨어 개발의 전체 비용을 줄이고, 개발 과정의 여러 방해 요인이 최소화되도록 보증하는 것을 목적으로 한다.
3. 형상 관리를 위하여 구성된 팀을 “Chief Programmer Team” 이라고 한다.
4. 형상 관리의 기능 중 하나는 버전 제어 기술이다.

정답 :

해설 : ‘Chief Programmer Team’은 효율성을 증대시키기 위해 경험과 능력이 풍부한 책임 프로그래머를 중심으로 구성한 개발 팀의 구성 방식 중 하나이다.

Section 52: 소프트웨어 버전 관리 도구

Q. 다음은 버전 관리 도구인 **Subversion**에 대한 설명이다**.** 잘못된 것은**?**

1. 클라이언트 / 서버 구조로, 서버에는 최신 버전과 버전의 변화를 저장한다.
2. 클라이언트에서는 서버의 자료를 복사해와 작업한 후 변경된 내용을 서버에 반영(Commit)한다.
3. 모든 개발작업은 trunk 디렉터리에서 수행되면, 부가적인 추가 작업은 branches 디렉터리 안에

별도의 디렉터리를 만들어 작업을 완료한 후 trunk 디렉터리의 작업과 병합한다.

4. 커밋(Commit)할 때마다 커밋의 버전이라고 할 수 있는 스냅샷(Snapshot)이 일정하게 증가한다.

정답 :

해설 : Subversion은 Revision으로 번호를 1씩 증가하면 버전을 관리하고, Git은 스냅샷 (영문자와 숫자가 혼합된 40자리 문자열)로 관리한다.

Section 60: 테스트 케이스 / 테스트 시나리오 / 테스트 오라클

Q. 다음 중 테스트 오라클에 대한 설명으로 옳지 않은 것은**?**

1. 샘플링 오라클 : 특정한 몇몇 테스트 케이스의 입력값들에 대해서만 기대하는 결과를 제공하는 오라클이다.
2. 토탈 오라클 : 모든 테스트 케이스의 입력 값에 대해 기대하는 결과를 제공하는 오라클이다.
3. 휴리스틱 오라클 : 특정 테스트 케이스의 입력 값에 대해 기대하는 결과를 제공하고, 나머지 입력 값들에 대해서는 추정으로 처리하는 오라클이다.
4. 일관성 검사 오라클 : 애플리케이션의 변경이 있을 경우 테스트 케이스의 수행 전과 후의 결과값이 동일한지를 확인하는 오라클이다.

정답 :

해설 : 오라클의 종류 - 참(True), 샘플링(Sampling), 추정(Heuristic), 일관성(Consistent) 가 있다. 2번은 참 오라클이다.

Section 61: 테스트 자동화 도구

Q. 테스트 케이스 자동 생성 도구를 이용하여 테스트 데이터를 찾아 내는 방법이 아닌 것은**?**

1. 스텁(Stub)과 드라이버(Driver)
2. 입력 도메인 분석
3. 랜덤(Random) 테스트
4. 자료 흐름도

정답 :

해설 : 테스트 케이스 생성도구 - 자료 흐름도, 기능 테스트, 랜덤 테스트, 입력 도메인 분석

Section 65: 애플리케이션 성능 개선

Q. 소스 코드 품질 분석 도구 중 정적 분석 도구가 아닌 것은**?**

1. pmd
2. checkstyle
3. valMeter
4. cppcheck

정답 :

해설 : 정적 분석 도구 - pmd, cppcheck, SonarQube, checkstyle, ccm, cobertuna 등

Section 67: 모듈 연계를 위한 인터페이스 기능 식별

Q. 내**,** 외부 모듈 연계 방법 중 **ESB(Enterprise Service Bus)**에 대한 설명으로 가장 옳지 않은 것은**?**

1. ESB는 애플리케이션과의 결합도(Coupling)을 약하게(Loosely) 유지한다.
2. ESB는 크게 Point-to-Point 형, Hub & Spoke 방식, Hybrid 형태의 구성으로 분류될 수 있다.
3. 높은 수준의 품질 지원이 가능하다.
4. 관리 및 보안이 쉽다.

정답 :

해설 : EAI의 구축 유형 4가지는 Point-to-Point, Hub & Spoke, Message Bus, Hybrid 이다.

Section 72: 인터페이스 보안

Q. 크래커가 침입하여 백도어를 만들어 놓거나**,** 설정 파일을 변경했을 때 분석하는 도구는**?**

1. trace
2. tripwire
3. udpdump
4. cron

정답 :

해설 : 크래커가 침입하여 백도어를 만들거나, 설정 파일을 변경했을 때 분석하는 도구를 데이터 무결성 도구라고 한다. 종류 - Tripwire, AIDE, Samhain, Claymore, Slipwire, Fcheck 등

Section 76: 데이터베이스 설계

Q. 데이터베이스에서 개념적 설계 단계에 대한 설명으로 틀린 것은**?**

1. 산출물로 E-R Diagram을 만들 수 있다.
2. DBMS에 독립적인 개념 스키마를 설계한다.
3. 트랜잭션 인터페이스를 설계 및 작성한다.
4. 논리적 설계 단계의 앞 단계에서 수행된다.

정답 :

해설 : 트랜잭션 인터페이스를 설계 및 작성하는 단계는 논리적 설계 단계 이다..

Q. 데이터베이스 설계 단계와 그 단계에서 수행되는 결과의 연결이 잘못된 것은**?**

1. 개념적 설계 단계 - 트랜잭션 모델링
2. 물리적 설계 단계 - 목표 DBMS에 맞는 물리적 구조 설계
3. 논리적 설계 단계 - 목표 DBMS에 독립적인 논리 스키마 설계
4. 구현 단계 - 목표 DBMS DDL로 스키마 작성

정답 :

해설 : 논리적 설계 단계는 목표 DBMS에 맞는, 즉 독립적인이 아니라 종속적인 스키마를 설계하는 단계이다.

Q. 데이터베이스의 논리적 설계**(Logical Design)** 단계에서 수행하는 작업이 아닌 것은**?**

1. 레코드 집중의 분석 및 설계
2. 논리적 데이터베이스 구조로 매핑(mapping)
3. 트랜잭션 인터페이스 설계
4. 스키마의 평가 및 정제

정답 :

해설 : 레코드 집중의 분석 및 설계는 물리적 설계 단계에서 수행하는 작업이다.

Q. 데이터베이스 설계 단계 중 저장 레코드 양식 설계**,** 레코드 집중의 분석 및 설계**,** 접근 경로 설계와

관계되는 것은**?**

1. 논리적 설계
2. 요구 조건 분석
3. 개념적 설계
4. 물리적 설계

정답 :

해설 : 레코드 양식, 집중 분석, 접근 경로 설계 - 물리적 설계

Q. 물리적 데이터베이스 구조의 기본 데이터 단위의 저장 레코드의 양식을 설계할 때

고려 사항이 아닌 것은**?**

1. 데이터 타입
2. 데이터 값의 분포
3. 트랜잭션 모델링
4. 접근 빈도

정답 :

해설 : 트랜잭션 모델링은 개념적 설계 단계에서 수행한다.

Section 77: 데이터 모델의 개념

구글 문서: **https://docs.google.com/document/d/1s4K8SeievmV3C3Z4ukC25vVxjyMo6fsPg-7_V4hpyac/edit**