**문제1**

**문제**: 어떤 시스템의 설계 단계에서 초기 사용자 요구를 시각화하고 빠른 피드백을 얻기 위해 화면 정의 작업을 수행하고자 한다. 이 때 가장 적절한 도구 또는 산출물을 고르시오.  
**보기**:  
① UI 프로토타입  
② 와이어프레임  
③ 화면 흐름도  
④ 반응형 디자인  
⑤ 표준 컴포넌트  
**정답**: ①  
**해설**: 초기 사용자 피드백을 얻기 위한 시각화 도구로는 UI 프로토타입이 가장 적절하다.

**문제2**

**문제**: 스크럼 개발 방법론에서 매일 팀이 모여 작업 진행 상황을 공유하고, 문제를 논의하는 회의는 무엇인가?  
**보기**:  
① 스프린트 회고  
② 제품 백로그 회의  
③ 데일리 스크럼  
④ 스프린트 계획  
⑤ 리파인먼트 미팅  
**정답**: ③  
**해설**: 데일리 스크럼은 스크럼 팀이 매일 모여 진행 상황을 공유하고 문제를 논의하는 짧은 회의이다.

**문제3**

**문제**: 시스템 간 연계 인터페이스를 설계할 때, 비동기 방식으로 메시지를 전달하고 수신을 보장하기 위해 사용하는 시스템은?  
**보기**:  
① API Gateway  
② 메시지 브로커  
③ Web Server  
④ File Server  
⑤ Load Balancer  
**정답**: ②  
**해설**: 메시지 브로커는 비동기 통신을 위한 시스템으로, 송신자와 수신자가 동시에 연결되어 있지 않아도 메시지 전달을 보장한다.

**문제4**

**문제**: 소프트웨어 형상관리에서 변경 요청에 대해 승인하거나 거절하는 기능은 무엇인가?  
**보기**:  
① 형상 감사  
② 형상 식별  
③ 형상 통제  
④ 형상 보고  
⑤ 형상 계획  
**정답**: ③  
**해설**: 형상 통제는 형상 항목에 대한 변경 요청을 검토하고 승인 또는 거절하는 활동이다.

**문제5**

**문제**: 사용자가 메뉴를 클릭했을 때 시스템이 응답 중임을 보여주는 UI 원칙은?  
**보기**:  
① 일관성  
② 직관성  
③ 가시성  
④ 피드백  
⑤ 유연성  
**정답**: ④  
**해설**: 피드백은 사용자의 행위에 대해 시스템이 상태나 결과를 즉각적으로 알려주는 것을 의미한다.

**문제6**

**문제**: 소프트웨어 생명주기(SDLC)의 단계 중 구현 단계 직전에 수행되어야 하는 활동은?  
**보기**:  
① 요구사항 정의  
② 테스트 계획 수립  
③ 설계  
④ 유지보수  
⑤ 배포  
**정답**: ③  
**해설**: 소프트웨어 구현은 설계 후에 수행되며, 설계는 요구사항을 기반으로 시스템 구조를 정의하는 단계이다.

**문제7**

**문제**: REST API를 설계할 때 가장 먼저 고려해야 하는 요소는 무엇인가?  
**보기**:  
① 요청/응답 포맷  
② API 인증 방식  
③ 리소스 URI 설계  
④ HTTP 상태코드  
⑤ 응답 시간  
**정답**: ③  
**해설**: RESTful 설계에서 가장 먼저 고려해야 하는 것은 URI를 리소스 중심으로 설계하는 것이다.

**문제8**

**문제**: API 인증에서 OAuth2 방식은 어떤 방식으로 접근 토큰을 부여하는가?  
**보기**:  
① 사용자 직접 인증  
② API Key 발급  
③ 권한 위임을 통한 인증 서버 발급  
④ 세션 쿠키 부여  
⑤ 보안코드 직접 삽입  
**정답**: ③  
**해설**: OAuth2는 권한 위임 방식으로 인증 서버가 접근 토큰을 발급해주는 구조이다.

**문제9**

**문제**: 업무 프로세스를 분석할 때 현재 수행 중인 프로세스를 도식화하는 기법은?  
**보기**:  
① TO-BE 모델링  
② 시나리오 기법  
③ 유스케이스 다이어그램  
④ AS-IS 모델링  
⑤ 상태 다이어그램  
**정답**: ④  
**해설**: AS-IS 모델링은 현재 상태의 업무 프로세스를 분석하고 시각화하는 데 사용된다.

**문제10**

**문제**: 로그인 기능 테스트에서 사용자 입력 없이 로그인 버튼을 클릭했을 때 오류 메시지가 출력되는지 확인하는 테스트 유형은?  
**보기**:  
① 경계값 분석  
② 정상 시나리오  
③ 예외 시나리오  
④ 통합 테스트  
⑤ 회귀 테스트  
**정답**: ③  
**해설**: 예외 시나리오는 사용자 오류 또는 예외 조건에 대한 시스템의 처리 여부를 검증하는 테스트 방식이다.

**문제11**

**문제**: 사용자의 접근 빈도가 높은 데이터를 빠르게 제공하기 위해 캐싱을 도입하고자 한다. 이 때 가장 적절한 캐시 전략은?  
**보기**:  
① LRU 캐시  
② FIFO 캐시  
③ LFU 캐시  
④ Random 캐시  
⑤ ARC 캐시  
**정답**: ①  
**해설**: LRU(Least Recently Used)는 가장 오랫동안 사용되지 않은 데이터를 제거하는 전략으로 일반적인 캐시에서 효과적이다.

**문제12**

**문제**: 마이크로서비스 아키텍처(MSA)에서 서비스 간 독립성을 유지하기 위해 데이터베이스를 어떻게 구성하는 것이 가장 적절한가?  
**보기**:  
① 단일 통합 DB 사용  
② 모놀리식 DB 분할  
③ 서비스별 독립 DB 구성  
④ 클라우드 공유 DB 사용  
⑤ 읽기 전용 DB 사용  
**정답**: ③  
**해설**: MSA에서는 각 서비스가 독립적으로 배포 및 운영될 수 있도록 자체 DB를 갖도록 설계한다.

**문제13**

**문제**: 요구사항 명세 시 고객의 니즈를 정확히 반영하기 위한 활동으로 가장 적절한 것은?  
**보기**:  
① 기능 설계  
② 요구사항 인터뷰  
③ 테스트 시나리오 작성  
④ 모듈 분해  
⑤ 성능 측정  
**정답**: ②  
**해설**: 요구사항 인터뷰는 고객과 직접 커뮤니케이션을 통해 명확한 요구사항을 도출하는 기법이다.

**문제14**

**문제**: 소프트웨어 테스트 단계에서 결함 수정 후 기존 기능이 정상적으로 동작하는지 확인하는 테스트는?  
**보기**:  
① 회귀 테스트  
② 단위 테스트  
③ 통합 테스트  
④ 부하 테스트  
⑤ 탐색 테스트  
**정답**: ①  
**해설**: 회귀 테스트는 버그 수정 등 변경 이후 기존 기능의 이상 유무를 확인하기 위한 테스트이다.

**문제15**

**문제**: 시스템의 무결성과 일관성을 유지하기 위해 변경 이력과 접근을 통제하는 활동은?  
**보기**:  
① 보안 정책 설정  
② 형상관리  
③ 변경 요청  
④ 로깅 시스템 도입  
⑤ 감사 로그 검토  
**정답**: ②  
**해설**: 형상관리는 소프트웨어 구성요소의 변경 이력을 추적하고 통제하는 프로세스를 포함한다.

**문제16**

**문제**: 테스트 케이스 설계 시 정상적인 입력만을 고려하는 방식은?  
**보기**:  
① 경계값 분석  
② 예외 시나리오 설계  
③ 기능 기반 설계  
④ 정상 시나리오 설계  
⑤ 오류 추정  
**정답**: ④  
**해설**: 정상 시나리오는 일반적인 상황에서 예상되는 입력을 기준으로 테스트를 설계한다.

**문제17**

**문제**: 웹 시스템 설계 시, 다양한 해상도와 디바이스에서도 일관된 화면을 제공하기 위한 접근 방식은?  
**보기**:  
① 서버 사이드 렌더링  
② 정적 페이지  
③ 반응형 웹 디자인  
④ 하이브리드 앱  
⑤ 데스크톱 전용 설계  
**정답**: ③  
**해설**: 반응형 웹 디자인은 다양한 화면 크기에 맞춰 자동으로 레이아웃이 조정되도록 설계된다.

**문제18**

**문제**: 시스템 간 실시간 데이터 연계를 위한 가장 적절한 통신 방식은?  
**보기**:  
① 일괄 처리  
② FTP  
③ 파일 공유  
④ REST API  
⑤ 테이블 복제  
**정답**: ④  
**해설**: REST API는 경량의 HTTP 기반 통신으로 실시간 연계에 적합하다.

**문제19**

**문제**: 애플리케이션 배포 자동화를 위해 Jenkins와 같은 도구를 사용하는 이유는?  
**보기**:  
① 데이터 분석  
② 코드 컴파일  
③ 빌드 자동화 및 배포  
④ 모니터링  
⑤ 트래픽 분석  
**정답**: ③  
**해설**: CI/CD 도구는 코드 빌드부터 테스트, 배포까지 자동화된 파이프라인을 제공한다.

**문제20**

**문제**: 다음 중 UI 설계 시 직관성을 높이기 위한 가장 적절한 방법은?  
**보기**:  
① 색상 최소화  
② 사용자 흐름 단순화  
③ 기능 숨기기  
④ 복잡한 버튼 배치  
⑤ 상호작용 숨김  
**정답**: ②  
**해설**: 사용자 흐름이 단순할수록 사용자는 빠르게 UI를 이해하고 이용할 수 있다.

**문제21**

**문제**: 대규모 분산 시스템에서 장애 복구 시, 서비스 중단을 최소화하기 위해 설계에 포함해야 할 전략은?  
**보기**:  
① 단일 서버 아키텍처  
② 수동 백업 시스템  
③ 이중화 및 페일오버 구성  
④ 로컬 로그 저장  
⑤ 암호화 적용  
**정답**: ③  
**해설**: 이중화 및 페일오버는 장애 발생 시 다른 노드로 자동 전환되어 서비스 연속성을 보장한다.

**문제22**

**문제**: UI 요소가 서로 다른 기능을 수행하지만 동일한 색상이나 아이콘을 사용하는 경우 발생할 수 있는 문제는?  
**보기**:  
① 접근성 향상  
② 피드백 강화  
③ 혼란 유발  
④ 보안성 저하  
⑤ 일관성 향상  
**정답**: ③  
**해설**: 시각적으로 동일한 요소가 다른 기능을 수행하면 사용자의 혼란을 초래한다.

**문제23**

**문제**: 비기능 요구사항 중 시스템이 동시에 처리할 수 있는 사용자 수와 관련된 항목은?  
**보기**:  
① 보안성  
② 신뢰성  
③ 확장성  
④ 응답 시간  
⑤ 동시 처리성  
**정답**: ⑤  
**해설**: 동시 처리성은 시스템이 동시 사용자 요청을 얼마나 감당할 수 있는지를 의미한다.

**문제24**

**문제**: 업무 프로세스 개선을 위해 현재 프로세스를 분석한 후 미래 프로세스를 설계하는 접근 방법은?  
**보기**:  
① BPMN 작성  
② AS-IS 모델링  
③ TO-BE 모델링  
④ 워크플로우 분석  
⑤ 요구사항 수집  
**정답**: ③  
**해설**: TO-BE 모델링은 향후 도입될 개선된 프로세스를 정의하는 기법이다.

**문제25**

**문제**: MSA 환경에서 서비스 간 통신 시, 메시지 구조를 정의하고 공유하는 데 사용하는 형식은?  
**보기**:  
① YAML  
② Swagger  
③ JSON Schema  
④ Markdown  
⑤ PDF 문서  
**정답**: ③  
**해설**: JSON Schema는 API 요청 및 응답 메시지 구조를 명확히 정의하는 데 사용된다.

**문제26**

**문제**: UI 테스트에서 실제 사용자 흐름을 자동화하여 검증하는 테스트 기법은?  
**보기**:  
① 단위 테스트  
② 통합 테스트  
③ 회귀 테스트  
④ 시나리오 테스트  
⑤ 코드 리뷰  
**정답**: ④  
**해설**: 시나리오 테스트는 실제 사용자 행동을 따라가며 UI 동작을 검증한다.

**문제27**

**문제**: 운영 환경에 배포된 시스템의 이상 동작을 실시간으로 감지하고자 한다. 이 때 가장 적절한 방법은?  
**보기**:  
① 수동 점검  
② 주기적 백업  
③ 로그 분석 도구 활용  
④ 테스트 코드 작성  
⑤ 형상관리  
**정답**: ③  
**해설**: 로그 분석 도구는 실시간 모니터링과 경고 시스템을 통해 이상 동작을 감지한다.

**문제28**

**문제**: REST API에서 리소스를 식별하는 가장 기본적인 방식은?  
**보기**:  
① 쿼리 파라미터  
② 세션 토큰  
③ 쿠키 값  
④ URI 경로  
⑤ 응답 헤더  
**정답**: ④  
**해설**: REST 설계의 핵심은 리소스를 URI로 식별하는 것이다.

**문제29**

**문제**: 스프린트 종료 후 팀이 개선점을 도출하고 다음 작업에 반영하기 위해 수행하는 회의는?  
**보기**:  
① 스프린트 계획 회의  
② 데일리 스크럼  
③ 스프린트 리뷰  
④ 스프린트 회고  
⑤ 리파인먼트  
**정답**: ④  
**해설**: 스프린트 회고는 프로세스 개선을 위한 내부 피드백 회의이다.

**문제30**

**문제**: 백엔드 서버가 동시에 처리할 수 있는 요청 수를 증가시키기 위해 가장 먼저 고려해야 할 요소는?  
**보기**:  
① UI 색상 조정  
② 데이터베이스 인덱스  
③ 로깅 정책  
④ 스레드/이벤트 루프 모델  
⑤ REST 응답 포맷  
**정답**: ④  
**해설**: 서버는 스레드 기반 또는 이벤트 루프 방식으로 병렬 처리를 하며 성능에 직접적인 영향을 준다.

**문제31**

**문제**: 대용량 트래픽을 효율적으로 처리하기 위해 서버 앞단에 위치하여 요청을 분산 처리하는 구성 요소는?  
**보기**:  
① 메시지 큐  
② 데이터 웨어하우스  
③ API 게이트웨이  
④ 로드 밸런서  
⑤ 프록시 서버  
**정답**: ④  
**해설**: 로드 밸런서는 요청을 여러 서버에 분산시켜 부하를 분산하고 가용성을 높이는 역할을 한다.

**문제32**

**문제**: UI 프로토타입을 설계할 때 가장 중요한 고려 요소는?  
**보기**:  
① 정확한 코드 구현  
② 미적인 요소  
③ 사용자의 니즈 반영  
④ 데이터베이스 구조  
⑤ 브라우저 호환성  
**정답**: ③  
**해설**: UI 프로토타입은 사용자의 니즈와 기대를 빠르게 반영하고 피드백을 받기 위한 도구이다.

**문제33**

**문제**: 다음 중 인터페이스 정의서에 포함되어야 할 핵심 항목이 아닌 것은?  
**보기**:  
① 인터페이스 식별자  
② 전송 방식  
③ 오류 처리 방법  
④ 데이터베이스 인덱스  
⑤ 데이터 포맷  
**정답**: ④  
**해설**: 인터페이스 정의서는 시스템 간 통신 사양을 정의하며, DB 인덱스는 포함 대상이 아니다.

**문제34**

**문제**: 단위 테스트에서 주로 사용되는 테스트 대상은 무엇인가?  
**보기**:  
① 전체 시스템  
② 사용자 시나리오  
③ 개별 모듈/함수  
④ 통신 네트워크  
⑤ UI 화면 구성  
**정답**: ③  
**해설**: 단위 테스트는 함수나 메서드와 같은 최소 코드 단위를 대상으로 수행한다.

**문제35**

**문제**: 다음 중 마이크로서비스 아키텍처에서 발생할 수 있는 단점은?  
**보기**:  
① 빠른 배포 가능  
② 모듈 간 낮은 결합도  
③ 서비스 간 통신 복잡성 증가  
④ 유연한 기술 스택  
⑤ 독립적 확장 가능  
**정답**: ③  
**해설**: MSA는 독립성이 장점이지만 서비스 간 통신이 복잡해지는 단점이 있다.

**문제36**

**문제**: REST API 응답에서 리소스를 성공적으로 생성했을 때 주로 사용하는 HTTP 상태 코드는?  
**보기**:  
① 200 OK  
② 204 No Content  
③ 201 Created  
④ 400 Bad Request  
⑤ 500 Internal Server Error  
**정답**: ③  
**해설**: 201 Created는 POST 요청으로 리소스가 성공적으로 생성되었음을 나타낸다.

**문제37**

**문제**: 개발자가 Git을 사용할 때 브랜치를 활용하는 주된 이유는?  
**보기**:  
① 로그 저장  
② 코드 컴파일  
③ 독립적 기능 개발  
④ 리소스 감소  
⑤ 트래픽 분산  
**정답**: ③  
**해설**: 브랜치는 독립적인 기능 개발이나 버그 수정을 위해 사용되며, 병합을 통해 통합된다.

**문제38**

**문제**: 테스트 자동화의 주요 장점 중 하나는 무엇인가?  
**보기**:  
① 개발 시간 증가  
② 수동 오류 유발  
③ 반복 작업 자동화  
④ 테스트 생략  
⑤ 복잡도 증가  
**정답**: ③  
**해설**: 자동화는 반복적인 테스트를 빠르고 정확하게 수행할 수 있게 해준다.

**문제39**

**문제**: UI/UX 설계에서 피드백이 필요한 가장 대표적인 예는?  
**보기**:  
① 정적 화면  
② 동영상 배경  
③ 버튼 클릭 후 상태 변화  
④ 동일한 색상 테마  
⑤ 페이지 로딩 시간  
**정답**: ③  
**해설**: 사용자 행동에 대한 시스템의 응답을 시각적으로 제공하는 것이 피드백이다.

**문제40**

**문제**: 유지보수 단계에서 가장 중요한 활동 중 하나는?  
**보기**:  
① 코드 주석 제거  
② 기능 삭제  
③ 버그 수정 및 개선  
④ 데이터베이스 재설계  
⑤ 요구사항 축소  
**정답**: ③  
**해설**: 유지보수는 기존 시스템의 오류를 수정하고 성능을 향상시키는 것이 핵심이다.

**문제41**

**문제**: 인터페이스 설계에서 시스템 간 데이터 연계 시 필수적으로 고려해야 할 항목은?  
**보기**:  
① UI 디자인  
② 응답 시간  
③ 데이터 포맷  
④ 폰트 스타일  
⑤ 소스코드 구조  
**정답**: ③  
**해설**: 데이터 포맷은 서로 다른 시스템이 데이터를 정확히 주고받기 위해 반드시 정의되어야 한다.

**문제42**

**문제**: 다음 중 UI 설계 원칙에 해당하지 않는 것은?  
**보기**:  
① 일관성  
② 단순성  
③ 가시성  
④ 보안성  
⑤ 직관성  
**정답**: ④  
**해설**: 보안성은 시스템 설계 관점이고, UI 설계 원칙에는 포함되지 않는다.

**문제43**

**문제**: 마이크로서비스 간 통신 방식으로 REST 대신 비동기 메시징을 사용하는 주요 이유는?  
**보기**:  
① 사용자 인터페이스 단순화  
② 서버 부하 증가  
③ 실시간 응답 제공  
④ 느린 응답 처리  
⑤ 느슨한 결합 유지  
**정답**: ⑤  
**해설**: 비동기 메시징은 서비스 간 강한 연결 없이 메시지만 주고받아 느슨한 결합을 가능하게 한다.

**문제44**

**문제**: 요구사항 분석 단계에서 작성된 유스케이스 다이어그램은 주로 어떤 내용을 포함하는가?  
**보기**:  
① UI 컴포넌트 목록  
② 시스템 내부 클래스 구조  
③ 사용자와 시스템 간 상호작용  
④ 테이블 명세서  
⑤ 네트워크 구성  
**정답**: ③  
**해설**: 유스케이스 다이어그램은 사용자의 행위와 시스템의 응답 간 상호작용을 표현한다.

**문제45**

**문제**: API 명세서에 가장 일반적으로 포함되지 않는 항목은?  
**보기**:  
① 요청 방식 (GET/POST 등)  
② 파라미터 설명  
③ 응답 코드 설명  
④ 보안 로그 기록  
⑤ 예제 요청/응답  
**정답**: ④  
**해설**: 보안 로그는 시스템 운영과 관련된 항목으로, API 명세서에는 포함되지 않는다.

**문제46**

**문제**: 단위 테스트 수행 시 테스트 대상이 외부 의존성을 가지는 경우 가장 효과적인 해결 방법은?  
**보기**:  
① 실제 DB 연결  
② API 호출 생략  
③ 목(Mock) 객체 사용  
④ 테스트 건너뛰기  
⑤ 무작위 값 삽입  
**정답**: ③  
**해설**: 목(Mock)은 외부 의존성을 대체하여 단위 테스트가 독립적으로 수행될 수 있게 한다.

**문제47**

**문제**: 백엔드 아키텍처에서 트랜잭션의 원자성을 보장하기 위한 기법은?  
**보기**:  
① SQL Injection  
② 캐시 적중  
③ ACID 원칙  
④ 데이터 파이프라인  
⑤ 로깅 미들웨어  
**정답**: ③  
**해설**: 트랜잭션의 원자성, 일관성, 고립성, 지속성을 보장하는 원칙이 ACID이다.

**문제48**

**문제**: 운영 환경에서 배포 전 기능 이상 유무를 확인하기 위해 실제와 유사한 환경에서 수행하는 테스트는?  
**보기**:  
① 단위 테스트  
② 회귀 테스트  
③ 시스템 테스트  
④ 통합 테스트  
⑤ 스테이징 테스트  
**정답**: ⑤  
**해설**: 스테이징 테스트는 실제 운영 환경과 유사한 조건에서 수행하여 배포 전 이상 여부를 점검한다.

**문제49**

**문제**: REST API 응답에서 요청이 잘못되었음을 나타내는 HTTP 상태 코드는?  
**보기**:  
① 200 OK  
② 201 Created  
③ 204 No Content  
④ 400 Bad Request  
⑤ 500 Internal Server Error  
**정답**: ④  
**해설**: 400 Bad Request는 클라이언트의 요청이 잘못되었을 때 반환된다.

**문제50**

**문제**: 인터페이스 요구사항 중 데이터 전송에 있어 ‘속도’와 가장 밀접하게 관련된 비기능 요구사항은?  
**보기**:  
① 보안성  
② 신뢰성  
③ 응답성  
④ 확장성  
⑤ 가용성  
**정답**: ③  
**해설**: 응답성은 시스템이 요청에 얼마나 빠르게 응답하는지를 의미한다.

**문제51**

**문제**: 다음 중 요구사항 정의 단계에서 주로 작성하는 산출물은 무엇인가?  
**보기**:  
① 테스트 케이스  
② 클래스 다이어그램  
③ UI 디자인 시안  
④ 요구사항 명세서  
⑤ 운영 매뉴얼  
**정답**: ④  
**해설**: 요구사항 명세서는 사용자 요구를 정리하고 시스템이 제공해야 할 기능을 명확히 문서화한 결과물이다.

**문제52**

**문제**: 화면 설계 시 '일관성' 원칙을 가장 잘 설명한 것은?  
**보기**:  
① 모든 기능은 하나의 색상으로 표시한다.  
② 동일한 기능은 동일한 위치와 표현으로 배치한다.  
③ 최대한 많은 정보를 제공한다.  
④ 기능별로 매번 다른 디자인을 적용한다.  
⑤ 애니메이션 효과를 최소화한다.  
**정답**: ②  
**해설**: 일관성은 사용자가 시스템을 예측하고 쉽게 사용할 수 있도록 동일한 규칙을 적용하는 것이다.

**문제53**

**문제**: MSA 환경에서 공통 기능(인증, 로깅 등)을 처리하는 API Gateway의 역할 중 가장 적절한 것은?  
**보기**:  
① 데이터 분석  
② 사용자 인터페이스 처리  
③ 서비스별 로직 분기  
④ 공통 기능 집중 처리  
⑤ 데이터 마이그레이션  
**정답**: ④  
**해설**: API Gateway는 인증, 로깅, 라우팅 등 공통 기능을 처리하여 마이크로서비스 간 복잡성을 낮춰준다.

**문제54**

**문제**: UI 요소 간의 상호작용 흐름을 시각적으로 표현한 다이어그램은?  
**보기**:  
① 클래스 다이어그램  
② 상태 다이어그램  
③ 시퀀스 다이어그램  
④ 화면 흐름도  
⑤ 유스케이스 다이어그램  
**정답**: ④  
**해설**: 화면 흐름도는 UI 간 전환 및 사용자의 동선을 시각적으로 표현하는 도구이다.

**문제55**

**문제**: 다음 중 테스트 시 ‘정상 입력값 경계’를 검증하는 데 적절한 테스트 기법은?  
**보기**:  
① 동등 분할  
② 경계값 분석  
③ 상태 전이  
④ 탐색 테스트  
⑤ 회귀 테스트  
**정답**: ②  
**해설**: 경계값 분석은 입력값의 상한, 하한 경계에서 시스템이 올바르게 작동하는지를 확인하는 기법이다.

**문제56**

**문제**: 서비스 장애 발생 시 자동으로 복구되도록 설정하는 기능은?  
**보기**:  
① 리포트 기능  
② Failover  
③ 리팩토링  
④ 세션 클러스터링  
⑤ 데이터 암호화  
**정답**: ②  
**해설**: Failover는 장애 발생 시 자동으로 대체 시스템으로 전환하여 서비스를 지속할 수 있게 하는 기능이다.

**문제57**

**문제**: 테스트 계획 수립 시 가장 먼저 정의해야 할 항목은?  
**보기**:  
① 테스트 데이터  
② 테스트 도구  
③ 테스트 범위 및 목표  
④ 버그 추적 방식  
⑤ 리포트 양식  
**정답**: ③  
**해설**: 테스트의 전체 방향성과 목적을 결정하기 위해 가장 먼저 범위 및 목표를 설정한다.

**문제58**

**문제**: 비즈니스 로직 설계 시 가장 우선적으로 고려해야 할 것은?  
**보기**:  
① 데이터 암호화  
② 사용자 인터페이스  
③ 업무 흐름 및 규칙  
④ 보고서 양식  
⑤ 로깅 방식  
**정답**: ③  
**해설**: 비즈니스 로직은 업무 규칙을 반영해야 하므로, 가장 먼저 해당 흐름과 규칙을 고려해야 한다.

**문제59**

**문제**: 요구사항이 불완전하거나 모호할 경우 가장 바람직한 대응은?  
**보기**:  
① 그대로 개발 진행  
② 유사 프로젝트 참고  
③ 사용자와 추가 인터뷰 진행  
④ 개발자 추측으로 판단  
⑤ 문서 생략  
**정답**: ③  
**해설**: 불완전한 요구사항은 반드시 사용자와의 추가 커뮤니케이션을 통해 명확히 해야 한다.

**문제60**

**문제**: 단위 테스트를 자동화하기 위한 가장 일반적인 도구는?  
**보기**:  
① Jenkins  
② Selenium  
③ JUnit  
④ Grafana  
⑤ Git  
**정답**: ③  
**해설**: JUnit은 Java 기반 단위 테스트 자동화를 위한 대표적인 프레임워크이다.

**문제61**

**문제**: 사용자의 요구사항을 기능별로 정리하여 기능 구조도를 작성하는 과정은 어떤 분석 단계에 해당하는가?  
**보기**:  
① 요구사항 정의  
② 현행 시스템 분석  
③ 기능 정형화  
④ 인터페이스 정의  
⑤ 개발 방법론 적용  
**정답**: ①  
**해설**: 사용자의 요구를 기능적으로 정리하고 구조화하는 과정은 요구사항 정의 단계에서 수행된다.

**문제62**

**문제**: 다음 중 API 응답에서 '리소스를 찾을 수 없음'을 의미하는 상태 코드는?  
**보기**:  
① 200 OK  
② 204 No Content  
③ 301 Moved Permanently  
④ 404 Not Found  
⑤ 500 Internal Server Error  
**정답**: ④  
**해설**: 404는 요청한 리소스가 존재하지 않을 때 반환되는 HTTP 상태 코드이다.

**문제63**

**문제**: 시스템의 성능 요구사항에 해당하는 항목으로 가장 적절한 것은?  
**보기**:  
① API 인증 방식  
② 응답 시간  
③ 데이터 정합성  
④ UI 구성  
⑤ 접근 권한  
**정답**: ②  
**해설**: 응답 시간은 사용자의 요청에 대해 시스템이 얼마나 빠르게 응답하는지를 의미하는 성능 지표다.

**문제64**

**문제**: 단위 테스트와 통합 테스트의 가장 큰 차이점은?  
**보기**:  
① 테스트 도구 사용  
② 테스트 환경  
③ 테스트 대상 범위  
④ 테스트 데이터 보안  
⑤ 테스트 보고서 형식  
**정답**: ③  
**해설**: 단위 테스트는 모듈 단위, 통합 테스트는 모듈 간 연계 동작을 검증하는 테스트로 범위가 다르다.

**문제65**

**문제**: 마이크로서비스 설계 시 공통 기능을 각 서비스에 중복 구현하는 것이 바람직하지 않은 이유는?  
**보기**:  
① 유지보수 어려움  
② 개발 속도 증가  
③ 서비스 간 결합도 증가  
④ UI 설계 복잡도 증가  
⑤ 사용자 만족도 감소  
**정답**: ①  
**해설**: 공통 기능 중복 구현은 유지보수 시 여러 서비스에 영향을 주어 관리가 어렵다.

**문제66**

**문제**: 서비스 장애 발생 시, 자동 복구 또는 대체 서비스로 전환되는 기능을 무엇이라 하는가?  
**보기**:  
① 복제  
② Failover  
③ Load Balancing  
④ API Gateway  
⑤ 암호화  
**정답**: ②  
**해설**: Failover는 시스템 오류 발생 시 자동으로 다른 노드로 전환해 장애를 회피하는 기능이다.

**문제67**

**문제**: 다음 중 UI 설계 시 '가시성' 원칙에 대한 설명으로 가장 적절한 것은?  
**보기**:  
① 잘 보이지 않도록 처리  
② 사용자 중심 기능 숨김  
③ 필요한 정보는 눈에 잘 띄게 배치  
④ 모든 기능을 동일한 색상 사용  
⑤ 상태 변화 없이 화면 유지  
**정답**: ③  
**해설**: 가시성은 사용자에게 필요한 정보를 명확히 보여주도록 설계하는 원칙이다.

**문제68**

**문제**: 프로세스 모델링 시 업무 절차의 순서와 조건을 명확히 표현하기 위해 사용하는 기법은?  
**보기**:  
① 데이터 흐름도  
② BPMN  
③ 유스케이스 다이어그램  
④ 클래스 다이어그램  
⑤ 화면 흐름도  
**정답**: ②  
**해설**: BPMN은 업무 절차와 흐름, 조건 등을 시각적으로 명확히 표현하는 표준 모델링 언어이다.

**문제69**

**문제**: RESTful API 설계 원칙 중 '자원 중심' 설계란 무엇을 의미하는가?  
**보기**:  
① 사용자 인증 중심  
② 함수 기반 설계  
③ URL을 통해 자원을 명확히 식별  
④ 브라우저 호환성 고려  
⑤ 세션 기반 처리  
**정답**: ③  
**해설**: 자원 중심 설계는 URL이 자원을 명확히 표현하고, HTTP 메서드로 동작을 구분하는 REST 설계의 핵심이다.

**문제70**

**문제**: 다음 중 테스트 자동화의 주요 이점으로 보기 어려운 것은?  
**보기**:  
① 반복 작업의 효율성 증가  
② 일관된 테스트 수행  
③ 테스트 생략 가능  
④ 개발 속도 향상  
⑤ 오류 조기 발견  
**정답**: ③  
**해설**: 테스트 자동화는 테스트 생략이 아닌, 반복 수행을 자동화하여 신뢰성과 효율성을 높이는 목적이다.

**문제71**

**문제**: 시스템 간 인터페이스 설계에서 데이터 포맷, 호출 주기, 프로토콜 등을 정리한 문서는?  
**보기**:  
① 운영 매뉴얼  
② 서비스 명세서  
③ 인터페이스 명세서  
④ 기능 정의서  
⑤ 성능 보고서  
**정답**: ③  
**해설**: 인터페이스 명세서는 시스템 간 연계를 위한 상세 사양을 정의한 문서로, 포맷과 프로토콜 등이 포함된다.

**문제72**

**문제**: 웹 시스템에서 입력값 검증을 누락했을 때 발생할 수 있는 대표적인 보안 취약점은?  
**보기**:  
① 데이터 중복  
② 화면 깜빡임  
③ SQL Injection  
④ 로딩 지연  
⑤ 사용자 혼란  
**정답**: ③  
**해설**: 입력값 검증이 없으면 악의적 쿼리를 통한 SQL Injection 공격에 노출될 수 있다.

**문제73**

**문제**: 애플리케이션에서 비즈니스 규칙에 따라 데이터를 처리하는 핵심 부분은 어디인가?  
**보기**:  
① 인터페이스 계층  
② 프레젠테이션 계층  
③ 비즈니스 로직 계층  
④ 데이터베이스  
⑤ 캐시 계층  
**정답**: ③  
**해설**: 비즈니스 로직 계층은 도메인 규칙에 따라 데이터를 가공하고 처리하는 핵심 부분이다.

**문제74**

**문제**: 단위 테스트를 수행할 때 테스트 결과를 정리하여 공유하는 문서는?  
**보기**:  
① 테스트 케이스 명세서  
② 테스트 결과 보고서  
③ 결함 추적표  
④ 요구사항 명세서  
⑤ 회고 보고서  
**정답**: ②  
**해설**: 테스트 결과 보고서는 수행한 테스트의 결과와 통계, 결함 등을 요약하여 정리한 문서이다.

**문제75**

**문제**: UI 설계 시 ‘직관성’을 가장 잘 설명한 예는?  
**보기**:  
① 같은 기능이 여러 버튼에 중복  
② 버튼 클릭 시 아무 반응 없음  
③ 자주 쓰는 기능이 눈에 띄게 배치됨  
④ 에러 메시지가 영어로 출력됨  
⑤ 복잡한 메뉴 구조  
**정답**: ③  
**해설**: 직관성은 사용자가 별도 학습 없이도 바로 이해하고 사용할 수 있도록 설계하는 것을 의미한다.

**문제76**

**문제**: MSA에서 서비스 간 데이터 일관성 보장을 위해 사용하는 패턴은?  
**보기**:  
① API Gateway  
② Saga 패턴  
③ Load Balancer  
④ Proxy 서버  
⑤ Singleton 패턴  
**정답**: ②  
**해설**: Saga 패턴은 분산된 트랜잭션을 순차적으로 처리하면서 서비스 간 데이터 일관성을 유지하는 방법이다.

**문제77**

**문제**: 다음 중 시스템 기능 테스트에 해당하지 않는 항목은?  
**보기**:  
① 로그인 기능 검증  
② 메뉴 이동 검증  
③ UI 배치 정렬  
④ 파일 업로드 기능 확인  
⑤ 데이터 저장 확인  
**정답**: ③  
**해설**: UI 배치 정렬은 기능보다는 사용자 인터페이스의 미관 및 편의성 테스트 항목이다.

**문제78**

**문제**: 시스템의 현재 상태를 유지하며 요청을 처리하기 위해 세션을 유지하는 대표적 기술은?  
**보기**:  
① HTTP  
② Stateless API  
③ JWT  
④ 쿠키/세션  
⑤ GET 방식  
**정답**: ④  
**해설**: 쿠키와 세션은 클라이언트 상태를 서버에서 유지하며 인증 및 요청 정보를 보존하는 데 사용된다.

**문제79**

**문제**: MSA 환경에서 각각의 마이크로서비스가 자신의 데이터베이스를 갖도록 하는 설계 원칙은?  
**보기**:  
① 공통 데이터베이스  
② 공유 인프라 구조  
③ DB 분리 원칙  
④ 데이터 마이그레이션  
⑤ 샤딩 기술  
**정답**: ③  
**해설**: 마이크로서비스는 서비스의 독립성과 결합도 감소를 위해 DB를 분리하여 독립적으로 운용한다.

**문제80**

**문제**: 통합 테스트에서 검증해야 할 주요 내용은 무엇인가?  
**보기**:  
① 개별 함수의 정확성  
② 사용자 인터페이스 색상  
③ 모듈 간 연계 동작  
④ 단일 서비스의 성능  
⑤ 로깅 경로  
**정답**: ③  
**해설**: 통합 테스트는 여러 모듈이 함께 동작할 때 인터페이스와 데이터 흐름이 정확한지를 검증한다.

**문제81**

**문제**: 업무 시스템을 설계할 때, 사용자에게 표시되는 정보를 명확하게 정의하는 작업은?  
**보기**:  
① 데이터 모델링  
② 화면 정의  
③ 인터페이스 설계  
④ 비즈니스 로직 정의  
⑤ 프로세스 분석  
**정답**: ②  
**해설**: 화면 정의는 각 화면에 어떤 정보가 표시되는지, 사용자 입력 및 출력 요소를 정의하는 작업이다.

**문제82**

**문제**: API 호출 중 트래픽이 몰리는 특정 시간대에 대응하기 위해 도입할 수 있는 전략은?  
**보기**:  
① 페이징 처리  
② 응답 데이터 축소  
③ 캐싱 적용  
④ UI 개선  
⑤ DB 정규화  
**정답**: ③  
**해설**: 캐싱은 동일한 요청에 대한 응답을 미리 저장해두어 서버 부하를 줄이는 데 효과적이다.

**문제83**

**문제**: 비즈니스 로직이 복잡하고 기능이 많은 시스템에서 기능별로 코드를 구분하기 위한 대표적 설계 패턴은?  
**보기**:  
① MVC 패턴  
② Singleton 패턴  
③ RESTful 설계  
④ 파일 시스템 분리  
⑤ 동기화 설계  
**정답**: ①  
**해설**: MVC 패턴은 Model, View, Controller로 기능을 분리해 코드의 유지보수성과 확장성을 높인다.

**문제84**

**문제**: 시스템 내 모든 모듈 간의 흐름과 연결을 도식화하여 전체 구조를 파악하는 데 가장 적절한 다이어그램은?  
**보기**:  
① 클래스 다이어그램  
② ERD  
③ 시퀀스 다이어그램  
④ 컴포넌트 다이어그램  
⑤ 유스케이스 다이어그램  
**정답**: ④  
**해설**: 컴포넌트 다이어그램은 시스템을 구성하는 모듈(컴포넌트) 간의 의존 관계와 구조를 표현한다.

**문제85**

**문제**: UI/UX 설계 시 사용자의 반복적인 입력을 줄이기 위한 대표적인 기법은?  
**보기**:  
① 다크모드 적용  
② 캐시 무효화  
③ 자동완성 기능  
④ 서버 간 로드 밸런싱  
⑤ 에러 페이지 설정  
**정답**: ③  
**해설**: 자동완성 기능은 이전 입력을 기억하여 사용자의 반복 작업을 줄이고 편의성을 높인다.

**문제86**

**문제**: 다음 중 통합 테스트 시 주로 검증하는 항목은?  
**보기**:  
① DB 정규화 수준  
② API 응답 포맷  
③ UI 색상 배치  
④ 기능 간 상호 연동  
⑤ 로그 보존 정책  
**정답**: ④  
**해설**: 통합 테스트는 여러 기능이 함께 동작할 때 정확하게 연동되는지를 검증하는 단계이다.

**문제87**

**문제**: 소프트웨어 개발에서 '요구사항 추적성'이 필요한 주요 이유는?  
**보기**:  
① 개발 속도 향상  
② 테스트 자동화  
③ 변경 영향 분석  
④ 디자인 다양성 확보  
⑤ 유지보수 편의성 감소  
**정답**: ③  
**해설**: 요구사항 추적성은 변경 시 어떤 기능과 테스트에 영향을 주는지를 파악하기 위해 필요하다.

**문제88**

**문제**: 데이터 품질 관리에서 정확성 확보를 위해 필요한 조치는?  
**보기**:  
① 데이터 압축  
② 이중화 구성  
③ 포맷 검증 및 유효성 검사  
④ 암호화  
⑤ UI 설계 일관성  
**정답**: ③  
**해설**: 정확성은 올바른 데이터 값이 입력되었는지를 확인하기 위한 포맷 및 유효성 검사가 핵심이다.

**문제89**

**문제**: 다음 중 단위 테스트 도구가 아닌 것은?  
**보기**:  
① JUnit  
② Mockito  
③ TestNG  
④ Postman  
⑤ NUnit  
**정답**: ④  
**해설**: Postman은 API 테스트 도구이며, 단위 테스트 프레임워크는 아니다.

**문제90**

**문제**: 사용자의 접근 권한을 기준으로 화면에 노출되는 정보가 달라지도록 설계하는 것은 어떤 원칙에 해당하는가?  
**보기**:  
① 최소 권한 원칙  
② UI 일관성  
③ 사용자 중심 설계  
④ API 라우팅  
⑤ 인터페이스 통합  
**정답**: ①  
**해설**: 최소 권한 원칙은 사용자가 필요한 범위 내에서만 정보를 접근할 수 있도록 제한하는 보안 설계 기준이다.

**문제91**

**문제**: API 요청 시 인증되지 않은 사용자의 접근을 차단하기 위한 가장 일반적인 방식은?  
**보기**:  
① JSON 포맷 사용  
② 응답 시간 단축  
③ 인증 토큰 활용  
④ 로그 압축  
⑤ 페이징 처리  
**정답**: ③  
**해설**: 인증 토큰(JWT 등)은 API 요청 시 사용자 인증을 확인하는 데 가장 널리 사용되는 방식이다.

**문제92**

**문제**: 사용자가 입력한 데이터의 형식이나 유효성을 사전에 검사하는 프로세스는?  
**보기**:  
① 캐시 처리  
② 입력값 유효성 검증  
③ 데이터 백업  
④ 세션 클러스터링  
⑤ 로드 밸런싱  
**정답**: ②  
**해설**: 입력값 유효성 검증은 잘못된 데이터가 시스템에 들어가지 않도록 사전에 필터링하는 절차이다.

**문제93**

**문제**: 시스템의 이상 여부를 판단하고 장애를 조기 탐지하기 위해 필요한 구성요소는?  
**보기**:  
① 소스코드 주석  
② 사용자 교육  
③ 모니터링 도구  
④ UI 가이드  
⑤ 화면 흐름도  
**정답**: ③  
**해설**: 모니터링 도구는 시스템 상태를 실시간으로 점검하고 이상을 탐지하는 데 사용된다.

**문제94**

**문제**: 다음 중 데이터 모델링 과정에서 '엔터티'를 정의하는 가장 적절한 설명은?  
**보기**:  
① 사용자의 이름  
② 테이블 간 조인 관계  
③ 독립적으로 존재하는 정보 단위  
④ 테이블의 속성 값  
⑤ 데이터 중복 방지 기술  
**정답**: ③  
**해설**: 엔터티는 고유하게 정의되고 독립적으로 존재할 수 있는 객체나 정보 단위를 의미한다.

**문제95**

**문제**: 시스템 설계 시 여러 사용자 요구를 수용하다 보면 복잡해지는 문제를 방지하기 위해 적용할 수 있는 개념은?  
**보기**:  
① 기능 누락  
② 요구사항 우선순위 지정  
③ 코드 중복  
④ 무분별한 기능 통합  
⑤ API 외부 공개  
**정답**: ②  
**해설**: 요구사항 우선순위 지정은 제한된 자원 내에서 핵심 기능부터 개발하기 위한 합리적 접근이다.

**문제96**

**문제**: MSA 환경에서 서비스 간 동기 호출로 인해 발생할 수 있는 문제는?  
**보기**:  
① 빠른 응답  
② 서비스 확장성 증가  
③ 높은 결합도  
④ 낮은 지연 시간  
⑤ 통신 단절 방지  
**정답**: ③  
**해설**: 동기 호출은 호출 간 종속성이 생겨 결합도가 높아지고, 장애 전파 가능성이 커진다.

**문제97**

**문제**: 다음 중 데이터 품질의 주요 요소로 보기 어려운 것은?  
**보기**:  
① 정확성  
② 일관성  
③ 최신성  
④ 가용성  
⑤ 가시성  
**정답**: ⑤  
**해설**: 데이터 품질은 정확성, 일관성, 적시성(최신성) 등을 포함하지만 가시성은 일반적으로 포함되지 않는다.

**문제98**

**문제**: 데이터 전환 작업에서 가장 먼저 수행해야 할 단계는?  
**보기**:  
① 정제 대상 확인  
② 데이터 포맷 설정  
③ 전환 스크립트 실행  
④ 백업 수행  
⑤ 테스트 케이스 작성  
**정답**: ①  
**해설**: 데이터 전환은 정제 대상 및 범위를 명확히 정의하는 것으로 시작된다.

**문제99**

**문제**: 시스템 간 연계에서 실시간 데이터 송수신이 필요한 경우 주로 사용하는 방식은?  
**보기**:  
① 정기 배치 처리  
② CSV 파일 연계  
③ REST API  
④ 오프라인 전송  
⑤ 종이 문서  
**정답**: ③  
**해설**: REST API는 네트워크를 통해 실시간으로 데이터를 송수신할 수 있는 대표적인 연계 방식이다.

**문제100**

**문제**: 다음 중 화면 정의서에 포함되어야 할 필수 항목은?  
**보기**:  
① 네트워크 대역폭  
② 사용자 반응시간  
③ UI 구성 요소 설명  
④ 개발자 역할  
⑤ API 성능 수치  
**정답**: ③  
**해설**: 화면 정의서는 각 화면의 구성 요소, 입력 필드, 버튼 등을 명확히 기술한 문서이다.

[ 데이터 모델링 ]

**문제1**  
문제: 다음 중 엔터티의 특징으로 올바르지 않은 것은?  
보기:  
① 업무에서 독립적  
② 고유 식별자 존재  
③ 속성을 가짐  
④ 관계 없이 존재 가능  
⑤ 논리 모델에서 사용  
정답: ④  
해설: 엔터티는 다른 엔터티와 관계를 통해 의미가 완성되는 경우가 많다.

**문제2**  
문제: 다음 중 관계형 데이터 모델의 기본 요소가 아닌 것은?  
보기:  
① 엔터티  
② 속성  
③ 관계  
④ 객체  
⑤ 기본키  
정답: ④  
해설: 객체는 객체지향 개념으로, 관계형 모델에서는 사용되지 않는다.

**문제3**  
문제: 데이터 전환 설계 시 가장 먼저 고려할 항목은?  
보기:  
① 전환 대상 식별  
② 스크립트 자동화  
③ 로그 정리  
④ 테스트 시나리오 작성  
⑤ UI 변경  
정답: ①  
해설: 전환 설계는 어떤 데이터를 옮길 것인지를 먼저 명확히 해야 한다.

**문제4**  
문제: 데이터 모델링의 3단계 중 논리 모델링의 주요 목적은 무엇인가?  
보기:  
① 사용자 요구사항 도출  
② 현행 시스템 분석  
③ 업무 중심의 데이터 구조 설계  
④ 물리적 저장 장치 설계  
⑤ SQL 성능 최적화  
정답: ③  
해설: 논리 모델링은 업무 중심으로 데이터 구조를 정제하여 설계하는 단계이다.

**문제5**  
문제: 데이터 품질 관리에서 정확성 확보를 위한 활동은?  
보기:  
① 포맷 검사 및 유효성 체크  
② 데이터 압축  
③ 백업 주기 단축  
④ 로그 분석  
⑤ 모델링 자동화  
정답: ①  
해설: 정확성은 입력값의 형식과 유효 여부를 검사하여 확보할 수 있다.

**문제6**  
문제: 정규화의 목적 중 가장 적절한 것은?  
보기:  
① 데이터 무결성 확보  
② 데이터 중복 증가  
③ 관계형 모델 제거  
④ 비정규형 유지  
⑤ 트랜잭션 분리  
정답: ①  
해설: 정규화는 중복을 제거하고 무결성을 유지하기 위한 데이터 구조 정제 과정이다.

**문제7**  
문제: ERD에서 다대다 관계를 표현하기 위해 추가로 필요한 구성 요소는?  
보기:  
① 뷰 테이블  
② 조인 테이블  
③ 정규 테이블  
④ 트랜잭션 로그  
⑤ 시퀀스  
정답: ②  
해설: 다대다 관계는 관계 테이블(조인 테이블)을 통해 1:N, N:1 관계로 분해한다.

**문제8**  
문제: 제3정규형(3NF)에 해당하는 조건은?  
보기:  
① 모든 속성이 후보키에 이행적 함수 종속  
② 모든 속성이 기본키에 완전 함수 종속  
③ 모든 속성이 기본키에 부분 종속  
④ 중복 속성 허용  
⑤ 기본키 존재하지 않음  
정답: ②  
해설: 제3정규형은 이행적 함수 종속 제거 및 기본키에 완전 종속된 구조를 갖는다.

**문제9**  
문제: 물리 모델링에서 인덱스를 사용하는 주요 목적은?  
보기:  
① 트랜잭션 처리량 감소  
② 데이터 입력 속도 향상  
③ 조회 성능 향상  
④ 정규화 수준 낮춤  
⑤ 데이터 보안 강화  
정답: ③  
해설: 인덱스는 조회 속도를 빠르게 하기 위해 사용하는 물리적 설계 요소이다.

**문제10**  
문제: 속성과 도메인 정의가 가장 핵심적인 모델링 활동은?  
보기:  
① 개념 모델링  
② 물리 모델링  
③ 데이터 표준화  
④ 인터페이스 정의  
⑤ 현행 분석  
정답: ③  
해설: 데이터 표준화는 속성명, 도메인, 정의 등을 통일하여 데이터 품질을 확보한다.

**문제11**  
문제: 물리 데이터 모델링에서 테이블 분할을 고려하는 주요 이유는?  
보기:  
① 보안 정책 강화  
② 정규화 해제  
③ 조회 성능 향상  
④ 트랜잭션 관리  
⑤ 데이터 마스킹  
정답: ③  
해설: 데이터가 많을 경우 테이블을 수평 또는 수직 분할하여 조회 성능을 개선할 수 있다.

**문제12**  
문제: 속성의 도메인을 정의하는 가장 주요한 목적은?  
보기:  
① 외래키 식별  
② 데이터 무결성 유지  
③ 테이블 이름 정리  
④ 관계 정의  
⑤ 논리 구조 삭제  
정답: ②  
해설: 도메인은 속성의 허용 값 범위를 지정하여 무결성을 유지하는 데 사용된다.

**문제13**  
문제: 데이터 모델링에서 기본키(Primary Key)의 가장 중요한 역할은?  
보기:  
① 속성 타입 결정  
② 데이터 암호화  
③ 엔터티 식별  
④ 뷰 생성  
⑤ 테이블 정렬  
정답: ③  
해설: 기본키는 엔터티 내에서 레코드를 유일하게 식별하는 데 사용된다.

**문제14**  
문제: 개념 모델에서 가장 중심이 되는 요소는?  
보기:  
① 속성  
② 인덱스  
③ 엔터티  
④ 파티션  
⑤ 뷰  
정답: ③  
해설: 개념 모델은 업무 중심 엔터티와 그들 간의 관계를 정의하는 데 초점을 맞춘다.

**문제15**  
문제: 다음 중 관계형 데이터베이스 설계 시 주의해야 할 사항은?  
보기:  
① 동일한 속성명 반복  
② 테이블 간 중복 제거  
③ 데이터 중복 허용  
④ 기본키 없이 설계  
⑤ 테이블 수 최대화  
정답: ②  
해설: 테이블 간 중복을 최소화하여 데이터의 일관성과 무결성을 유지해야 한다.

**문제16**  
문제: 엔터티 간 1:N 관계를 물리적으로 표현할 때 일반적으로 사용하는 방법은?  
보기:  
① 중복 테이블 생성  
② 다중 기본키 설정  
③ 외래키 사용  
④ 뷰 정의  
⑤ 로그 테이블 생성  
정답: ③  
해설: 1:N 관계는 자식 테이블에 외래키를 추가하여 표현한다.

**문제17**  
문제: 정규화 과정에서 부분 함수 종속을 제거하는 단계는?  
보기:  
① 제1정규형  
② 제2정규형  
③ 제3정규형  
④ BCNF  
⑤ 비정규형  
정답: ②  
해설: 제2정규형은 부분 함수 종속을 제거하여 기본키 전체에 완전 종속되도록 한다.

**문제18**  
문제: 물리 모델링 시 데이터 접근 속도 개선을 위한 가장 일반적인 기법은?  
보기:  
① 테이블명 축약  
② 인덱스 설계  
③ ERD 단순화  
④ 속성명 정렬  
⑤ 트리거 제거  
정답: ②  
해설: 인덱스는 검색 속도를 높이기 위한 대표적인 물리 모델링 기법이다.

**문제19**  
문제: 다음 중 엔터티의 분류에 해당하지 않는 것은?  
보기:  
① 주요 엔터티  
② 행위 엔터티  
③ 중심 엔터티  
④ 행위 중심 엔터티  
⑤ 참조 엔터티  
정답: ②  
해설: 엔터티는 중심, 행위 중심, 참조 엔터티로 나뉘며 ‘행위 엔터티’는 올바른 분류가 아니다.

**문제20**  
문제: ERD 상의 관계 설정 시 유의해야 할 사항으로 적절하지 않은 것은?  
보기:  
① 관계의 참여도 명시  
② 선택성과 필수성 구분  
③ 다대다 관계 허용  
④ 관계 이름 명확화  
⑤ 관계 유형 구분  
정답: ③  
해설: 다대다 관계는 물리 모델링 시 직접 사용하지 않고 조인 테이블로 분해하여 표현한다.

**문제21**  
문제: 데이터 정규화 이전에 비정규형(UNF) 테이블의 가장 큰 문제점은?  
보기:  
① 데이터 중복 최소화  
② 속성 수 제한  
③ 데이터 중복 및 반복 그룹 존재  
④ 기본키 과다 사용  
⑤ 트랜잭션 처리 속도 저하  
정답: ③  
해설: 비정규형 테이블은 반복 그룹 등으로 인해 중복이 발생하고 구조가 불안정하다.

**문제22**  
문제: 제1정규형(1NF)의 주요 요건은 무엇인가?  
보기:  
① 반복 속성 제거  
② 기본키 제거  
③ 속성 그룹화  
④ 비관계형 테이블 생성  
⑤ 속성별 보안 설정  
정답: ①  
해설: 제1정규형은 반복 속성을 제거하고 원자값(단일 값)을 유지하는 것이 핵심이다.

**문제23**  
문제: 데이터 모델에서 속성(Attribute)에 해당하는 예시는?  
보기:  
① 주문 테이블  
② 고객 이름  
③ 판매 관계  
④ 외래키  
⑤ ERD  
정답: ②  
해설: 속성은 엔터티의 세부정보를 설명하는 항목으로 '고객 이름'과 같은 정보이다.

**문제24**  
문제: 데이터 표준화의 주요 목적은?  
보기:  
① 사용자 인터페이스 개선  
② 시스템 부하 분산  
③ 속성명, 정의, 도메인 등을 일관되게 정의  
④ 보고서 자동화  
⑤ 트랜잭션 관리  
정답: ③  
해설: 데이터 표준화는 데이터 항목이 일관되게 사용되도록 명명규칙과 도메인 등을 정의한다.

**문제25**  
문제: 다음 중 참조 무결성 제약조건의 설명으로 가장 적절한 것은?  
보기:  
① 기본키 중복을 허용한다  
② 외래키가 참조하는 기본키가 존재해야 한다  
③ 속성 간 계산 가능성을 제시  
④ 속성 정의에 따라 도메인을 결정  
⑤ 트리거 기반 연산을 수행  
정답: ②  
해설: 참조 무결성은 외래키가 참조하는 기본키가 테이블에 존재해야 함을 보장하는 제약이다.

**문제26**  
문제: 모델링 과정에서 속성 정의 시 고려해야 할 요소가 아닌 것은?  
보기:  
① 데이터 길이  
② 속성 정의  
③ 기본값  
④ 속성 배치 순서  
⑤ 데이터 타입  
정답: ④  
해설: 속성 배치 순서는 의미상 큰 영향이 없으며, 명확한 정의, 타입, 길이 등이 중요하다.

**문제27**  
문제: 정규화된 데이터 모델이 가진 장점은?  
보기:  
① 데이터 중복 증가  
② 속성 수 감소  
③ 무결성 유지 용이  
④ 비정규형 유지  
⑤ 물리 모델 복잡성 증가  
정답: ③  
해설: 정규화를 통해 데이터 구조를 체계화함으로써 무결성을 유지하기 쉽다.

**문제28**  
문제: 속성의 유일성과 존재 여부를 보장하는 제약조건은?  
보기:  
① DEFAULT  
② NOT NULL  
③ CHECK  
④ PRIMARY KEY  
⑤ INDEX  
정답: ④  
해설: PRIMARY KEY는 유일성 및 NOT NULL 조건을 모두 만족시키는 제약조건이다.

**문제29**  
문제: 관계형 모델에서 한 테이블의 외래키가 다른 테이블의 기본키를 참조하는 이유는?  
보기:  
① 데이터 중복 허용  
② 관계 설정 및 무결성 유지  
③ 속성명 단순화  
④ 인덱스 생성 용도  
⑤ 속성 암호화  
정답: ②  
해설: 외래키는 관계 설정과 참조 무결성을 유지하기 위한 필수적인 도구이다.

**문제30**  
문제: 물리 데이터베이스 설계 시 성능을 고려하여 속성에 추가할 수 있는 요소는?  
보기:  
① 트랜잭션  
② 로그 파일  
③ 인덱스  
④ 정규화  
⑤ 외부키  
정답: ③  
해설: 인덱스는 조회 성능 향상을 위한 물리적 설계 요소로 활용된다.

**문제31**  
문제: 데이터 모델링에서 개체 간 관계의 참여도를 정의할 때 사용하는 개념은?  
보기:  
① 도메인  
② 기본키  
③ 선택성  
④ 배타성  
⑤ 카디널리티  
정답: ⑤  
해설: 카디널리티(Cardinality)는 관계에 참여하는 개체 수를 나타내며 1:1, 1:N, N:M 형태로 정의된다.

**문제32**  
문제: 다음 중 데이터베이스에서 무결성을 확보하기 위한 제약조건에 해당하지 않는 것은?  
보기:  
① PRIMARY KEY  
② FOREIGN KEY  
③ UNIQUE  
④ DEFAULT  
⑤ ENCRYPT  
정답: ⑤  
해설: ENCRYPT는 보안 처리이며, 무결성 제약조건에는 포함되지 않는다.

**문제33**  
문제: 속성 값의 유효 범위를 사전에 정해두고 이를 벗어나는 값을 입력하지 못하게 하는 제약조건은?  
보기:  
① UNIQUE  
② NOT NULL  
③ DEFAULT  
④ CHECK  
⑤ FOREIGN KEY  
정답: ④  
해설: CHECK 제약조건은 속성 값의 허용 범위를 정의하여 데이터 정확성을 확보한다.

**문제34**  
문제: 모델링에서 개념 모델의 주요 특징으로 적절한 것은?  
보기:  
① 물리적 저장 구조를 반영  
② 인덱스 설계 포함  
③ 비즈니스 중심 구조 표현  
④ SQL 코드 포함  
⑤ 디스크 용량 고려  
정답: ③  
해설: 개념 모델은 비즈니스 개념을 중심으로 엔터티와 관계를 추상화하여 표현하는 것이 특징이다.

**문제35**  
문제: 제2정규형의 조건으로 올바른 것은?  
보기:  
① 반복 속성 제거  
② 기본키에 이행적 종속 제거  
③ 후보키에 완전 함수 종속  
④ 속성 간 계산 가능성 정의  
⑤ 기본키 없이 관계 구성  
정답: ③  
해설: 제2정규형은 부분 함수 종속을 제거하고 기본키 전체에 완전 종속되도록 구성하는 것이다.

**문제36**  
문제: 데이터 전환 계획 수립 시 가장 먼저 수행해야 할 작업은?  
보기:  
① 전환 툴 개발  
② 전환 일정 작성  
③ 전환 대상 목록 정리  
④ 백업 정책 수립  
⑤ 테스트 스크립트 생성  
정답: ③  
해설: 어떤 데이터를 전환할지 목록을 정리해야 전체 범위와 방법을 구체화할 수 있다.

**문제37**  
문제: 속성 간 함수 종속이란 어떤 개념을 의미하는가?  
보기:  
① 속성명 중복  
② 속성 값에 대한 보안  
③ 하나의 속성이 다른 속성 값을 결정  
④ 속성 길이 제한  
⑤ 속성명 영문화  
정답: ③  
해설: 함수 종속은 어떤 속성의 값이 다른 속성 값을 유일하게 결정하는 관계를 의미한다.

**문제38**  
문제: 정규화된 테이블을 의도적으로 비정규화하는 가장 일반적인 목적은?  
보기:  
① 보안 강화  
② 데이터 압축  
③ 성능 향상  
④ 트랜잭션 분리  
⑤ 스키마 단순화  
정답: ③  
해설: 성능 향상을 위해 조인 비용이 높은 테이블을 비정규화하여 속도를 개선하는 경우가 있다.

**문제39**  
문제: 하나의 테이블 내에 정의된 속성 중 유일성을 보장하지만 기본키는 아닌 속성을 정의할 수 있는 제약조건은?  
보기:  
① CHECK  
② NOT NULL  
③ UNIQUE  
④ FOREIGN KEY  
⑤ DEFAULT  
정답: ③  
해설: UNIQUE 제약조건은 중복되지 않도록 하지만 기본키는 아닐 수 있다.

**문제40**  
문제: 데이터 모델 품질을 평가하는 항목으로 가장 적절하지 않은 것은?  
보기:  
① 완전성  
② 정합성  
③ 중복성  
④ 유연성  
⑤ 추적성  
정답: ③  
해설: 중복성은 낮을수록 좋은 품질을 의미하며, 중복성 자체는 품질 요소로 간주되지 않는다.

**문제41**  
문제: 다음 중 정규화의 목적과 가장 거리가 먼 것은?  
보기:  
① 데이터 중복 제거  
② 무결성 확보  
③ 저장 용량 감소  
④ 데이터 구조의 명확화  
⑤ 관계 구조 정리  
정답: ③  
해설: 정규화는 저장 용량 감소보다는 데이터 중복 제거와 구조 정리를 목적으로 한다.

**문제42**  
문제: 데이터 모델링에서 '관계(Relationship)'의 주요 목적은?  
보기:  
① 속성 정의  
② 개체 간 연결  
③ 기본키 생략  
④ 제약조건 해제  
⑤ 도메인 정의  
정답: ②  
해설: 관계는 서로 다른 개체 간의 연관성을 나타내기 위한 모델링 구성 요소이다.

**문제43**  
문제: 다음 중 ERD 작성 시 포함되지 않는 요소는?  
보기:  
① 엔터티  
② 속성  
③ 관계  
④ 뷰  
⑤ 카디널리티  
정답: ④  
해설: 뷰는 논리적 가상 테이블로 ERD 작성 시 직접 포함되지는 않는다.

**문제44**  
문제: 논리 모델링 단계에서 결정해야 하는 사항이 아닌 것은?  
보기:  
① 테이블 간 관계  
② 속성 도메인  
③ 물리 저장 위치  
④ 기본키 설정  
⑤ 속성 정의  
정답: ③  
해설: 물리 저장 위치는 물리 모델링 단계에서 고려하는 요소이다.

**문제45**  
문제: 다음 중 물리 모델링에서 고려하는 요소가 아닌 것은?  
보기:  
① 인덱스 설계  
② 파티셔닝  
③ 데이터 타입  
④ 관계 설정  
⑤ 테이블 스페이스  
정답: ④  
해설: 관계 설정은 논리 모델링 단계에서 수행되며, 물리 모델링에서는 성능과 저장 구조가 중심이다.

**문제46**  
문제: 데이터 전환 수행 단계에서 가장 중요한 작업은?  
보기:  
① 인터뷰 진행  
② 테스트 자동화  
③ 전환 스크립트 실행 및 검증  
④ 데이터 백업  
⑤ 문서 버전 관리  
정답: ③  
해설: 전환 스크립트를 실행하고 그 결과를 검증하는 것이 전환 작업의 핵심이다.

**문제47**  
문제: 속성 A가 속성 B를 결정하고, B가 C를 결정할 때, A가 C를 결정하는 종속 관계는?  
보기:  
① 직접 종속  
② 전체 종속  
③ 이행적 종속  
④ 역방향 종속  
⑤ 함수적 비종속  
정답: ③  
해설: A → B, B → C이면 A → C는 이행적 종속 관계를 나타낸다.

**문제48**  
문제: 정규화된 모델을 비정규화하는 경우, 고려해야 할 주요 요소는?  
보기:  
① 테이블 개수  
② UI 구성  
③ 데이터 중복 허용성  
④ 정합성 제거  
⑤ 기본키 제거  
정답: ③  
해설: 비정규화 시 데이터 중복이 허용되므로 중복에 따른 영향과 처리를 고려해야 한다.

**문제49**  
문제: 물리 모델링 시 속성의 데이터 타입을 결정할 때 가장 우선 고려해야 할 요소는?  
보기:  
① DB 용량  
② 속성 의미  
③ UI 디자인  
④ 정규화 정도  
⑤ 관계 개수  
정답: ②  
해설: 속성의 의미에 따라 그에 적절한 데이터 타입을 지정해야 데이터 무결성과 효율성이 확보된다.

**문제50**  
문제: 하나의 엔터티가 여러 다른 엔터티와 관계를 맺을 때 사용하는 설계 방식은?  
보기:  
① 슈퍼 엔터티  
② 서브타입 관계  
③ 중복 테이블 분할  
④ 다대다 연계  
⑤ 공통 부모 엔터티 도출  
정답: ⑤  
해설: 여러 엔터티와의 공통 속성을 가진 부모 엔터티를 도출하여 관계를 효율적으로 관리할 수 있다.

**문제51**  
문제: 데이터 모델링 시 엔터티 간 관계를 표현하는 주요 기호 중 ‘|’는 어떤 의미인가?  
보기:  
① 필수 참여  
② 선택 참여  
③ 다대다 관계  
④ 반복 속성  
⑤ 엔터티 병합  
정답: ①  
해설: '|' 기호는 해당 엔터티가 관계에 반드시 참여해야 함을 의미한다(필수 참여).

**문제52**  
문제: 논리 모델에서 속성 정의 시 일반적으로 포함되지 않는 항목은?  
보기:  
① 데이터 타입  
② 속성 정의  
③ 도메인  
④ UI 배치 정보  
⑤ NULL 여부  
정답: ④  
해설: UI 배치 정보는 화면 설계 단계에서 다루며, 논리 모델에서는 포함되지 않는다.

**문제53**  
문제: 정규화가 지나치게 되었을 때 발생할 수 있는 문제는?  
보기:  
① 데이터 중복  
② 성능 저하  
③ 무결성 감소  
④ 테이블 병합  
⑤ 속성 누락  
정답: ②  
해설: 지나친 정규화는 많은 조인으로 인해 오히려 성능 저하를 초래할 수 있다.

**문제54**  
문제: 관계형 데이터 모델에서 모든 속성이 원자값을 가지도록 하는 정규형은?  
보기:  
① 제1정규형  
② 제2정규형  
③ 제3정규형  
④ BCNF  
⑤ 제5정규형  
정답: ①  
해설: 제1정규형(1NF)은 모든 속성이 원자값(Atomic Value)을 가져야 한다는 조건을 가진다.

**문제55**  
문제: 하나의 기본키를 기준으로 여러 속성이 종속될 때, 이 구조를 무엇이라 하는가?  
보기:  
① 전체 함수 종속  
② 부분 함수 종속  
③ 이행 함수 종속  
④ 상호 종속  
⑤ 반복 종속  
정답: ①  
해설: 전체 함수 종속은 기본키 전체에 의존하는 관계로, 제2정규형에서 요구된다.

**문제56**  
문제: 비즈니스 규칙을 기반으로 도출된 데이터 요소를 논리적으로 그룹화한 모델은?  
보기:  
① 화면 모델  
② 물리 모델  
③ 개념 모델  
④ UI 모델  
⑤ 사용자 모델  
정답: ③  
해설: 개념 모델은 비즈니스 중심으로 데이터 요소를 그룹화하고 관계를 표현한 모델이다.

**문제57**  
문제: 다음 중 데이터 모델 품질을 높이기 위한 활동으로 가장 적절한 것은?  
보기:  
① 데이터 추출 생략  
② 도메인 정의 생략  
③ 속성명 중복 허용  
④ 표준화된 명명 규칙 적용  
⑤ 무결성 제약 생략  
정답: ④  
해설: 표준화된 명명 규칙은 일관성 있고 유지보수가 용이한 모델 설계를 가능하게 한다.

**문제58**  
문제: 모델링 시 도출된 엔터티 간 관계가 순환 구조를 가지는 경우 발생할 수 있는 주요 문제는?  
보기:  
① 속성 누락  
② 정규화 실패  
③ 종속성 모호성  
④ 데이터 마스킹  
⑤ 기본키 제거  
정답: ③  
해설: 순환 구조에서는 어느 엔터티가 부모/자식인지 모호해져 종속성 정의가 어려워진다.

**문제59**  
문제: 다음 중 속성 도메인을 정의할 때 고려하지 않아도 되는 항목은?  
보기:  
① 값의 길이  
② 허용 값 범위  
③ 값의 의미  
④ 사용자 권한  
⑤ 데이터 타입  
정답: ④  
해설: 도메인은 속성의 형식, 범위, 타입 등을 정의하는 것이며 권한과는 무관하다.

**문제60**  
문제: 논리 모델링 단계에서 수행되는 작업으로 가장 적절한 것은?  
보기:  
① 테이블 파티셔닝  
② SQL 튜닝  
③ 데이터 타입 확정  
④ 속성 정제 및 관계 설정  
⑤ 로그 백업 정책 수립  
정답: ④  
해설: 논리 모델링 단계에서는 엔터티와 속성을 정의하고 관계 및 도메인을 설정하는 작업이 수행된다.

계속해서 **61번부터 70번까지** 보시려면 “다음 문제 보여줘”라고 말씀해주세요.

**나의 말:**

**문제61**  
문제: 데이터 모델에서 하나의 엔터티가 다른 엔터티의 기본키를 포함할 때의 관계는?  
보기:  
① 동등 관계  
② 강한 관계  
③ 약한 관계  
④ 역관계  
⑤ 중첩 관계  
정답: ③  
해설: 약한 관계는 자식 엔터티가 부모의 기본키를 외래키로 포함하고 있으며 독립적으로 존재하지 못한다.

**문제62**  
문제: 다음 중 외래키 제약조건에 해당하는 설명으로 가장 적절한 것은?  
보기:  
① 엔터티에 중복 허용  
② 기본키 자동 증가  
③ 참조 무결성 유지  
④ 속성 형식 무시  
⑤ 관계 제거  
정답: ③  
해설: 외래키는 다른 테이블의 기본키를 참조하여 참조 무결성을 유지하는 제약조건이다.

**문제63**  
문제: 물리 모델링에서 저장소 효율을 위해 자주 사용되는 기법은?  
보기:  
① 속성 정규화  
② 뷰 활용  
③ 파티셔닝  
④ 속성 병합  
⑤ 관계 회피  
정답: ③  
해설: 파티셔닝은 데이터를 물리적으로 분할해 저장소 효율성과 조회 성능을 높인다.

**문제64**  
문제: 데이터 정규화 후 테이블이 과도하게 분리되었을 때 이를 해결하기 위해 사용하는 설계 기법은?  
보기:  
① 도메인 분석  
② 정규화  
③ 뷰 정의  
④ 비정규화  
⑤ 속성 제거  
정답: ④  
해설: 비정규화는 성능 개선이나 설계 단순화를 위해 테이블을 합치는 과정이다.

**문제65**  
문제: 데이터 품질 기준 중 데이터가 일관성 있게 유지되어야 한다는 것을 의미하는 것은?  
보기:  
① 정확성  
② 완전성  
③ 정합성  
④ 적시성  
⑤ 추적성  
정답: ③  
해설: 정합성은 데이터가 논리적으로 일관된 상태를 유지함을 의미한다.

**문제66**  
문제: 정규화 단계 중 제1정규형에서 제거되는 데이터 구조는?  
보기:  
① 이행 종속  
② 반복 속성  
③ 기본키 중복  
④ 속성 분리  
⑤ 외래키 생략  
정답: ②  
해설: 제1정규형에서는 반복 속성을 제거하여 속성이 원자값을 갖도록 한다.

**문제67**  
문제: 다음 중 엔터티 정의 시 일반적으로 고려하지 않는 항목은?  
보기:  
① 엔터티 명  
② 엔터티 설명  
③ 기본키  
④ 속성 목록  
⑤ 물리 저장소 경로  
정답: ⑤  
해설: 물리 저장소 경로는 물리 설계 단계에서 정의되며, 엔터티 정의에는 포함되지 않는다.

**문제68**  
문제: 데이터 모델의 ‘완전성’ 품질 기준이 의미하는 것은?  
보기:  
① 중복 허용  
② 필요한 데이터가 빠짐없이 존재  
③ 암호화 적용  
④ 백업 정책  
⑤ 트리거 설정  
정답: ②  
해설: 완전성은 데이터 모델이 표현해야 할 모든 데이터를 빠짐없이 포함하고 있음을 의미한다.

**문제69**  
문제: 다음 중 속성 간에 발생하는 이행적 함수 종속을 제거하는 정규화 단계는?  
보기:  
① 제1정규형  
② 제2정규형  
③ 제3정규형  
④ BCNF  
⑤ 제5정규형  
정답: ③  
해설: 제3정규형은 이행적 함수 종속을 제거하여 구조의 명확성과 무결성을 확보한다.

**문제70**  
문제: 다음 중 ERD(Entity Relationship Diagram)의 목적이 아닌 것은?  
보기:  
① 데이터베이스 구조 시각화  
② 개체 간 관계 정의  
③ 논리적 흐름 분석  
④ 속성 구조 설계  
⑤ UI 흐름 설계  
정답: ⑤  
해설: ERD는 데이터 모델을 시각적으로 표현하는 도구이며, UI 흐름은 설계 범위에 포함되지 않는다.

**문제81**  
문제: 데이터 모델링에서 속성명이 중복되었을 경우 발생할 수 있는 문제는?  
보기:  
① 속성 정렬 오류  
② 도메인 충돌  
③ 관계 정의 증가  
④ 무결성 향상  
⑤ 키 제약 해제  
정답: ②  
해설: 동일한 속성명이 다른 의미로 사용되면 도메인 충돌과 데이터 품질 문제가 발생할 수 있다.

**문제82**  
문제: 물리 모델링 단계에서 속성 정의 시 도메인을 설정하는 주요 목적은?  
보기:  
① 속성 분할  
② 관계 회피  
③ 형식 및 범위 일관성 유지  
④ 키 자동 생성  
⑤ 모델 정규화  
정답: ③  
해설: 도메인은 속성의 값 형식과 범위를 정의해 일관된 데이터를 유지하게 한다.

**문제83**  
문제: 속성 값이 특정 값들만 허용되도록 제약을 두는 것은 어떤 제약조건에 해당하는가?  
보기:  
① FOREIGN KEY  
② PRIMARY KEY  
③ DEFAULT  
④ CHECK  
⑤ UNIQUE  
정답: ④  
해설: CHECK 제약조건은 속성의 허용 가능한 값의 범위를 정의하여 제한한다.

**문제84**  
문제: 데이터 모델링에서 속성을 정의할 때 고려하지 않아도 되는 것은?  
보기:  
① 속성명  
② 데이터 타입  
③ 기본값  
④ 파티션 이름  
⑤ NULL 여부  
정답: ④  
해설: 파티션은 물리적 설계에서 고려되는 요소로 속성 정의에는 포함되지 않는다.

**문제85**  
문제: 정규화된 테이블에서 삽입 이상이 발생하지 않도록 하기 위한 방법은?  
보기:  
① 속성 최소화  
② 외래키 제거  
③ 이행 종속 제거  
④ 도메인 통일  
⑤ 기본키 축소  
정답: ③  
해설: 이행 종속을 제거하는 것은 제3정규형의 조건이며, 삽입 이상을 방지한다.

**문제86**  
문제: 정규화와 가장 직접적인 관련이 있는 개념은?  
보기:  
① 파티션  
② 트랜잭션  
③ 함수 종속성  
④ 암호화  
⑤ 트리거  
정답: ③  
해설: 정규화는 함수 종속성 기반으로 속성 간 관계를 분석해 테이블 구조를 최적화한다.

**문제87**  
문제: 다음 중 데이터 표준화 작업에 해당하지 않는 것은?  
보기:  
① 도메인 설정  
② 속성명 명명 규칙 수립  
③ 테이블 병합  
④ 데이터 정의서 작성  
⑤ 코드 값 일관화  
정답: ③  
해설: 테이블 병합은 비정규화 작업이며, 표준화에는 포함되지 않는다.

**문제88**  
문제: 논리 모델링과 물리 모델링의 가장 큰 차이점은?  
보기:  
① 데이터 타입 존재 여부  
② 기본키 사용 여부  
③ 속성 수  
④ 테이블 수  
⑤ 사용자 권한 설정  
정답: ①  
해설: 논리 모델링에서는 데이터 타입이 구체적으로 정의되지 않지만, 물리 모델링에서는 필수적으로 정의된다.

**문제89**  
문제: 정규화를 수행한 테이블에서 중복이 줄어들면 얻을 수 있는 직접적인 효과는?  
보기:  
① 성능 저하  
② 보안 강화  
③ 무결성 향상  
④ 저장 공간 증가  
⑤ 속성 수 증가  
정답: ③  
해설: 중복 제거로 인해 데이터 변경 시 일관성을 유지할 수 있어 무결성이 향상된다.

**문제90**  
문제: 정규화의 단점으로 적절한 것은?  
보기:  
① 무결성 저하  
② 데이터 중복 증가  
③ 복잡한 조인 발생  
④ 속성 일관성 감소  
⑤ 도메인 정의 불가  
정답: ③  
해설: 테이블이 분리되어 조인 연산이 많아지면 성능 저하로 이어질 수 있다.

**문제91**  
문제: 물리 모델링에서 테이블 간 조인 성능을 개선하기 위해 고려해야 할 요소는?  
보기:  
① 테이블 설명  
② 트랜잭션 분리  
③ 인덱스 설계  
④ 뷰 정의  
⑤ 도메인 분할  
정답: ③  
해설: 조인 성능 개선을 위해서는 인덱스를 적절하게 설계하는 것이 중요하다.

**문제92**  
문제: 개념 데이터 모델링의 주요 목적은?  
보기:  
① 테이블 간의 성능 최적화  
② 비즈니스 규칙 기반 데이터 구조 정의  
③ SQL 작성  
④ 관계형 DB 설치  
⑤ 암호화 정책 수립  
정답: ②  
해설: 개념 모델링은 비즈니스 규칙에 기반하여 데이터의 구조를 고수준으로 정의하는 데 목적이 있다.

**문제93**  
문제: 외래키(Foreign Key)가 참조하는 기본키가 삭제될 경우 오류를 방지하기 위한 옵션은?  
보기:  
① CASCADE  
② UNIQUE  
③ CHECK  
④ DEFAULT  
⑤ NOT NULL  
정답: ①  
해설: CASCADE 옵션은 참조된 행이 삭제될 경우 자동으로 외래키를 가진 행도 삭제되도록 설정한다.

**문제94**  
문제: 다음 중 데이터 품질 확보를 위한 요소로 가장 적절하지 않은 것은?  
보기:  
① 정확성  
② 일관성  
③ 중복성  
④ 적시성  
⑤ 완전성  
정답: ③  
해설: 중복성은 낮을수록 좋은 품질을 의미하며, 품질 확보 항목에는 포함되지 않는다.

**문제95**  
문제: 엔터티-관계 모델에서 두 엔터티가 서로 선택적으로 연결되는 경우의 관계 표현은?  
보기:  
① 필수 참여  
② 상호 배타  
③ 선택 참여  
④ 다대다  
⑤ 역관계  
정답: ③  
해설: 선택 참여는 엔터티가 관계에 반드시 참여하지 않아도 되는 상황을 표현한다.

**문제96**  
문제: 논리 모델링 산출물 중 속성 목록을 포함하고 있는 문서는?  
보기:  
① 관계 정의서  
② 속성 정의서  
③ 도메인 정의서  
④ 데이터 흐름도  
⑤ 시퀀스 다이어그램  
정답: ②  
해설: 속성 정의서는 각 속성의 이름, 타입, 길이, 정의 등을 포함한 논리 모델의 핵심 산출물이다.

**문제97**  
문제: 다음 중 데이터 모델링의 세부 단계 순서로 올바른 것은?  
보기:  
① 개념 → 논리 → 물리  
② 논리 → 개념 → 물리  
③ 물리 → 논리 → 개념  
④ 논리 → 물리 → 개념  
⑤ 개념 → 물리 → 논리  
정답: ①  
해설: 데이터 모델링은 개념 → 논리 → 물리 순으로 정제되어 가며 구체화된다.

**문제98**  
문제: 다음 중 속성에 대한 도메인 설정 시 가장 우선 고려해야 할 항목은?  
보기:  
① 데이터베이스 용량  
② 테이블 수  
③ 속성의 의미  
④ 속성명 길이  
⑤ 속성 정의 순서  
정답: ③  
해설: 도메인은 속성의 의미에 맞춰 설정되며, 값의 허용 범위와 형식을 결정한다.

**문제99**  
문제: 정규화된 모델에서 제3정규형(3NF)의 특징으로 가장 적절한 것은?  
보기:  
① 부분 함수 종속 제거  
② 이행 함수 종속 제거  
③ 속성 제거  
④ 중복 허용  
⑤ 정합성 해제  
정답: ②  
해설: 제3정규형은 이행 종속을 제거하여 테이블 구조를 안정화하고 무결성을 높인다.

**문제100**  
문제: 다음 중 ERD 작성 시 고려해야 할 항목으로 가장 적절한 것은?  
보기:  
① DB 트리거  
② 뷰 성능  
③ 관계 참여도  
④ 백업 정책  
⑤ SQL 문법  
정답: ③  
해설: 관계 참여도는 ERD에서 엔터티 간의 연결 여부와 필수성을 표현하는 핵심 요소이다.