**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.