

종단원	소단원	번호	문제	키워드	답
01. 논리 데이터 저장소 확인	1. 논리 데이터 모델 검증	1	논리 데이터 모델링의 특징	논리 데이터 모델링	1. 포용성 - 모든 엔티티 타입, 속성, 관계, 프로세스 등을 포함 2. 정규화 - 모든 데이터를 정규화하여 모델링 3. 완전성 - 모든 규칙과 관계를 완전하고 정확하게 표현 4. 독립성 - 성능, 제약사항에 독립적인 모델 - 특정 DBMS로부터 독립적인 성질
		2	논리 데이터 모델링의 속성	논리 데이터 모델링	1. 개체 2. 속성 3. 관계
		3	논리 데이터 모델링 개념 서술	논리 데이터 모델링	논리 데이터 모델링은 데이터베이스 설계 프로세스의 기초 설계 단계로, 비즈니스 정보의 구조와 규칙을 명확하게 표현할 수 있는 기법이다.
		4	논리 데이터 모델링의 속성	논리 데이터 모델링	1. 개체 (Entity) - 관리할 대상이 되는 실체 2. 속성 (Attribute) - 관리할 정보의 구체적 항목 3. 관계 - 개체 간의 대응 관계
		5	E-R모델 설명	E-R모델 (개체-관계 모델)	* 현실 세계에 존재하는 데이터와 그들 간의 관계를 사람이 이해할 수 있는 형태로 명확하게 표현하기 위해서 가장 널리 사용되고 있는 모델 * 요구사항으로부터 얻어낸 정보들을 개체(Entity), 속성(Attribute), 관계(Relation)로 기술한 모델
		6	속성(Attribute) 설명	E-R모델 (개체-관계 모델)	- 개체의 특성이나 상태를 기술한 것 - 관계의 열을 의미
		7	관계(Relation) 설명	E-R모델 (개체-관계 모델)	행과 열로 구성된 테이블
		8	정규화 설명	정규화 (Nomalization)	관계형 데이터베이스의 설계에서 중복을 최소화하게 데이터를 구조화하는 프로세스
		9	이상 현상 설명	이상 현상 (Anomaly)	데이터 중복성으로 인해 릴레이션(테이블)을 조작할 때 발생하는 비합리적 현상
		10	이상 현상의 종류	이상 현상 (Anomaly)	1. 삽입 이상 2. 삭제 이상 3. 갱신 이상
		11	삽입 이상 설명	이상 현상 (Anomaly)	정보 저장 시 해당 정보의 불필요한 세부정보를 입력해야 하는 경우
		12	삭제 이상 설명	이상 현상 (Anomaly)	정보 삭제 시 원치 않는 다른 정보가 같이 삭제되는 경우
		13	갱신 이상 설명	이상 현상 (Anomaly)	중복 데이터 중에서 특정 부분만 수정되어 중복된 값이 모순을 일으키는 경우
		14	정규화 개념	정규화 (Nomalization)	정규화는 관계형 데이터베이스의 설계에서 중복을 최소화하게 데이터를 구조화하는 프로세스이다

		15	정규화 종류	정규화 (Nomalization)	1. 1 정규형(1NF) - 원자값으로 구성 2. 2정규형(2NF) - 부분 함수 종속 제거 (완전 함수 종속 관계) 3. 3정규형(3NF) - 이행 함수 종속 제거 4. BC정규형(2NF) - 결정자 함수이면서 후보키가 아닌 것 제거 5. 4 정규형(4NF) - 다치(다중 값) 종속성 제거 6. 5정규형(5NF) - 조인 종속성 제거
		16	2정규형(2NF) 설명	정규화 (Nomalization)	1차 정규형을 만족하면서 부분 함수 종속된 것을 제거한 정규형이다
		17	보이스-코드 정규형(BCNF) 설명	정규화 (Nomalization)	3차 정규형을 만족하면서 모든 결정자가 후보키 집합에 속한 정규형이다
02. 물리 데이터 저장소 설계	1. 물리 데이터 모델 설계	1	모델링 기법 종류	모델링 기법	1. 논리 데이터 모델링 2. 물리 데이터 모델링
		2	물리 데이터 모델링 설명	물리 데이터 모델링	논리모델을 적용하고자 하는 기술에 맞도록 상세화해 가는 과정
		3	물리 데이터 모델링 절차	물리 데이터 모델링	1. 개체를 테이블로 변환 2. 속성을 컬럼으로 변환 3. UID를 기본키로 변환 4. 관계를 외래키로 변환 5. 컬럼 유형과 길이 정의 6. 반 정규화 수행
		4	시스템 성능 향상과 개발 및 운영의 단순화를 위해 데이터 모델을 통합하는 데이터베이스 성능향상 기법은 무엇인가	반 정규화	반 정규화 기법
		5	반 정규화 개념	반 정규화	시스템 성능 향상과 개발 및 운영의 단순화를 위해 데이터 모델을 통합하는 데이터베이스 성능향상 기법
	2. 물리 데이터 저장소 구성	1	인덱스 개념	인덱스 (Index)	* 검색 연산의 최적화를 위해 데이터베이스 내 열에 대한 정보를 구성한 데이터 구조 * 전체 데이터의 검색 없이 필요한 정보를 신속히 조회가 가능하게 하는 데이터 구조
		2	$(1 / \text{컬럼 값의 종류}) * 100$ $(\text{컬럼 값의 평균 Row 수} / \text{테이블의 총 Row 수}) * 100$ 위 수식은 인덱스 적용 기준과 관련된 수식이다. 어떤 개념에 대한 설명인가	인덱스 (Index)	분포도 - 특정 컬럼의 데이터가 테이블에 평균적으로 분포되어 있는 정도 - 인덱스로 적정한 분포도는 10%~15% 정도
		3	인덱스 컬럼 선정 기준	인덱스 (Index)	1. (분포도)가 좋은 컬럼은 단독적으로 생산한다 2. 자주 조합되어 사용되는 컬럼은 (결합 인덱스)로 생성한다
		4	인덱스 컬럼 선정 기준	인덱스 (Index)	1. 분포도가 좋은 컬럼은 단독적으로 생성한다 2. 자주 조합되어 사용되는 컬럼은 결합 인덱스로 생성한다 3. 결합 인덱스는 구성되는 컬럼 순서 선정(사용빈도, 유일성, 정렬 등)에 유의한다 4. 가능한 한 수정이 빈번하지 않은 컬럼을 선정한다
		5	인덱스 특징	인덱스 (Index)	* 지나치게 많은 인덱스는 오버헤드로 작용한다 * 인덱스는 추가적인 저장 공간이 필요함을 고려해야 한다 * 넓은 범위를 인덱스로 처리 시 오히려 전체 처리보다 많은 오버헤드를 발생시킬 수 있음에 유의해야 한다 * 인덱스와 테이블의 저장 공간을 적절히 분리될 수 있도록 설계해야 한다
		6	클러스터 설계 시 고려사항	클러스터 (Cluster)	* UNION, DISTINCT, ORDER BY, GROUP BY가 빈번한 컬럼이면 검토 대상 * 수정이 자주 발생하지 않는 컬럼은 검토 대상
		7	파티션 종류	파티션 (Partition)	1. 레인지 파티셔닝 (Range Partitioning) 2. 해시 파티셔닝 (Hash Partitioning) 3. 리스트 파티셔닝 (List Partitioning) 4. 컴포지트 파티셔닝 (Composite Partitioning)
		8	레인지 파티셔닝 (Range Partitioning) 개념	파티션 (Partition)	* 연속적인 숫자나 날짜를 기준으로 하는 파티셔닝 기법 * 손쉬운 관리 기법을 제공하여 관리 시간의 단축이 가능
		9	해시 파티셔닝 (Hash Partitioning) 개념	파티션 (Partition)	파티션 키의 해시값에 의한 균등한 데이터 분할 가능한 파티셔닝 기법

		10	파티셔닝 수행 시 장점	파티션 (Partition)	1. 성능 향상 2. 가용성 향상 3. 백업 가능 4. 경합 감소
		11	파티셔닝 수행 시 장점	파티션 (Partition)	1. 성능 향상 - 데이터 액세스 또는 접근 범위를 줄여 성능 향상 2. 가용성 향상 - 전체 데이터의 훼손 가능성이 감소 및 데이터 가용성 향상 3. 백업 가능 - 분할 영역을 독립적으로 백업하고 복구 가능 4. 경합 감소 - 디스크 스트라이핑으로 입출력 성능을 향상 - 디스크 컨트롤러에 대한 경합의 감소
03. 데이터 조작 프로시저 작성	1. 데이터 조작 프로시저 개발	1	프로시저 개념	프로시저 (Procedure)	* SQL 이용해 생성된 데이터를 조작하는 프로그램 * 데이터베이스 내부에 저장되고 일정한 조건이 되면 자동으로 수행됨
		2	* 표준 SQL을 기본으로 Oracle에서 개발한 데이터 조작 언어 * Oracle 기반의 모든 프로시저 작성에 사용되며 표준 SQL의 확장 기능이 우수 위에서 설명하는 데이터 조작 언어는?	프로시저 (Procedure)	PL/SQL - SQL과 함께 효과적으로 DB에 접근할 수 있는 방법을 제공하는 프로그래밍 언어
		3	데이터 저장소 연결 순서	프로시저 (Procedure)	1. 드라이버 로딩 2. 데이터베이스 연결 3. 쿼리 전달 4. 결과 수신
		4	생성에 해당하는 명령어	데이터 정의어 (DDL)	CREATE
		5	일정 기준을 두는 규칙, 제약사항 만들 때 사용하는 명령어	데이터 정의어 (DDL)	CONSTRAINT
		6	테이블 수정 쿼리	데이터 정의어 (DDL)	ALTER 테이블명 MODIFY (속성명 데이터형식);
		7	프로시저 문법 기초	데이터 조작 프로시저	CREATE OR REPLACE PROCEDURE 프로시저명 (파라미터 데이터 타입 [IN OUT INOUT]) IS[AS] 선언부 BEGIN 실행부 EXEPTION 예외 처리부 END;
		8	JDBC 개념	프로시저 (Procedure)	데이터베이스에 연결 및 작업을 하기 위한 자바 표준 인터페이스
		9	JDBC 개념	프로시저 (Procedure)	자바에서 데이터베이스를 사용할 수 있도록 연결해주는 응용 프로그램 인터페이스
		10	데이터베이스 연결을 위한 용어 채우기	프로시저 (Procedure)	Java 환경의 경우 구축된 데이터베이스로의 연결은 (JDBC)를 통해 연결한다
	2. 데이터 조작 프로시저 테스트	1	DBMS_OUTPUT 패키지 개념 서술	PL/SQL 테스트	메시지를 버퍼에 저장하고 버퍼로부터 메시지를 읽어오기 위한 인터페이스를 제공하는 패키지

04. 데이터 조작 프로시저 최적화	2. 데이터 조작 프로시저 성능개선	2	PL/SQL 처리 결과를 화면에 출력하기 위한 SQL 명령어	PL/SQL 테스트	SET SERVEROUTPUT ON
		3	PL/SQL의 명령어인 SHOW ERRORS의 역할 서술	PL/SQL 테스트	PL/SQL 처리 결과에 대한 오류내용을 출력
		1	옵티마이저 개념	옵티마이저	* SQL을 가장 빠르고 효율적으로 수행할 최적(최저비용)의 처리경로를 생성해 주는 DBMS 내부의 핵심엔진
		2	옵티마이저 유형	옵티마이저	* 규칙 기반 옵티마이저 <ul style="list-style-type: none"> - 통계 정보가 없는 상태에서 사전 등록된 규칙에 따라 질의 실행 계호기를 선택하는 옵티마이저 - 우선 순위 기반 - 평가기준 : 인덱스 구조, 연산자, 조건절 형태 등 * 비용 기반 옵티마이저 <ul style="list-style-type: none"> - 통계 정보로부터 모든 접근 경로를 고려한 질의실행 계획을 선택하는 옵티마이저 - 수행 시간 기반 - 평가기준 : 레코드 개수, 블록 개수, 평균 행 길이, 컬럼 값의 수, 컬럼 값 분포, 인덱스 높이, 클러스터링 팩터 등
		3	실행 계획 개념	옵티마이저	* 옵티마이저가 생성한 SQL 처리 경로 * 사용자가 작성한 SQL이 요구한 데이터를 추출하기 위해 옵티마이저가 작업의 방법과 순서를 결정하는 것 * EXPLAIN PLAN, SET AUTOTRACE, tkprof 등으로 확인 가능한 것
		4	실행 계획 개념	옵티마이저	* 실행계획은 옵티마이저가 생성한 SQL 처리 경로이다 * 사용자가 작성한 SQL이 요구한 데이터를 추출하기 위해 옵티마이저가 작업의 방법과 순서를 결정하는 것을 실행계획이라고 한다
		5	힌트 개념	옵티마이저 힌트	* SQL 튜닝의 핵심부분 * 일종의 지시구문 * SQL에 포함되어 쓰이고 옵티마이저의 실행 계획을 원하는 대로 바꿀 수 있게 함
		6	옵티마이저가 비정상적인 실행계획을 수립 시 액세스 경로 및 조인 순서를 제어할 수 있도록 하기 위한 기법	옵티마이저 힌트	힌트 또는 힌트 사용
		7	주요 옵티마이저 힌트	옵티마이저 힌트	/**RULE*/ : 규칙 기반 접근 방식을 사용하도록 지정 /**CHOOSE*/ : 오라클 옵티마이저 디폴트 값에 따름 /**INDEX(테이블명 인덱스명)*/ : 지정된 인덱스를 강제적으로 쓰게끔 지정 /**USE_HASH(테이블명)*/ : 지정된 테이블의 조인이 해시 조인(Hash-Join)형식으로 일어나도록 유도 /**USE_MERGE(테이블명)*/ : 지정된 테이블의 조인이 정렬 합병 조인(Sort-Merge)형식으로 일어나도록 유도 /**USE_NL(테이블명)*/ : 지정된 테이블의 조인이 중첩 반복 조인(Nested-Loop)형식으로 일어나도록 유도
		8	옵티마이저 힌트 사용 이유	옵티마이저 힌트	옵티마이저라고 해서 항상 최선의 실행 계획을 수립할 수는 없으므로 테이블이나 인덱스의 잘못된 실행 계획을 직접 바꿀 수 있도록 도와주는 것이 힌트의 역할
		9	지정된 테이블들의 조인이 Nested-Loop 형식으로 일어나도록 유도하기 위한 힌트를 작성하시오	옵티마이저 힌트	/**USE_NL(table_name)*/

