

3과목-데이터베이스 구축

(Part 2. 물리 데이터베이스 설계 - I)

데이터베이스 구축 총 파트

데이터베이스 구축 3과목은 총 5Part로 이루어져 있다.

1장 논리 데이터베이스 설계(49.72%)

2장 물리 데이터베이스 설계(22.91%)

3장 SQL 응용(26.26%)

4장 SQL 활용(1.12%)

5장 데이터 전환(0.00%)

물리 데이터베이스 설계

물리 데이터베이스 설계 Part는 16개의 섹션으로 구성되어 있다.

- 01 사전 조사분석
- 02 데이터베이스 저장공간설계
- 03 트랜잭션 분석 / CRUD 분석
- 04 인덱스 설계
- 05 뷰(View) 설계
- 06 클러스터 설계
- 07 파티션 설계
- 08 데이터베이스 용량 설계
- 09 분산 데이터베이스 설계
- 10 데이터베이스 이중화 / 서버 클러스터링
- 11 데이터베이스 보안 / 암호화
- 12 데이터베이스 보안 - 접근통제
- 13 데이터베이스 백업
- 14 스토리지
- 15 논리 데이터 모델의 물리 데이터 모델 변환
- 16 물리 데이터 모델 품질 검토

2. 물리 데이터베이스 설계-SEC_01(사전 조사 분석)

이 장에서 꼭 알아야 할 키워드

물리 데이터베이스 설계, 트랜잭션, 인덱스, 뷰, 클러스터, 파티션, 분산 데이터베이스, 접근통제, 스토리지, 이중화

1) 물리 데이터베이스 설계

; 물리 데이터베이스 설계는 논리적 구조로 표현된 논리적 데이터베이스를 디스크 등의 물리적 저장장치에 저장할 수 있는 물리적 구조의 데이터로 변환하는 과정이다.

- 물리적 데이터베이스 구조의 기본적인 데이터 단위는 저장 레코드(Stored Record)이다.
- 물리적 설계 단계에 꼭 포함되어야 할 것은 저장 레코드의 양식 설계, 레코드 집중(Record Clustering)의 분석 및 설계, 접근 경로 설계 등이다.
- 물리적 데이터베이스 구조는 여러 가지 타입의 저장 레코드 집합이라는 면에서 단순한 파일과 다르다.
- 물리적 데이터베이스 구조는 데이터베이스 시스템의 성능에 중대한 영향을 미친다.

Clustering : 사전적 의미로는 '송이를 이루다', '떼를 짓게 하다'라는 뜻으로 여기서는 '군집'이란 의미를 가지고 있다.

2. 물리 데이터베이스 설계-SEC_01(사전 조사 분석)

1) 물리 데이터베이스 설계

- 물리적 설계 시 고려 사항

- 인덱스 구조
- 레코드 크기
- 파일에 존재하는 레코드 개수
- 파일에 대한 트랜잭션의 갱신과 참조 성향
- 성능 향상을 위한 개념 스키마의 변경 여부 검토
- 빈번한 질의와 트랜잭션들의 수행속도를 높이기 위한 고려
- 시스템 운용 시 파일 크기의 변화 가능성

- 물리적 설계 전에 기존 시스템을 분석하여 데이터 명명 규칙, 시스템 자원, 데이터베이스 관리 요소 등을 파악해야 한다.

트랜잭션(Transaction) : 데이터베이스의 상태를 변환시키는 하나의 논리적 기능을 수행하기 위한 작업의 단위 또는 한꺼번에 모두 수행되어야 할 일련의 연산들을 의미한다.

개념 스키마(Conceptual Schema) : 데이터베이스의 전체적인 논리적 구조로서, 모든 응용 프로그램이나 사용자들이 필요로 하는 데이터를 종합한 조직 전체의 데이터베이스로 하나만 존재한다.

2. 물리 데이터베이스 설계-SEC_01(사전 조사 분석)

2) 물리적 설계 옵션

; 물리적 설계 옵션이란 특정 DBMS에서 제공되는 것으로, 데이터베이스 파일에 대한 저장 구조와 접근 경로에 대한 다양한 옵션을 말한다.

- 반응시간(Response Time) : 트랜잭션 수행을 요구한 시점부터 처리 결과를 얻을 때까지의 경과 시간
- 공간 활용도(Space Utilization) : 데이터베이스 파일과 액세스 경로 구조에 의해 사용되는 저장 공간의 양
- 트랜잭션 처리량(Transaction Throughput) : 단위시간 동안 데이터베이스 시스템에 의해 처리될 수 있는 트랜잭션의 평균 개수

2. 물리 데이터베이스 설계-SEC_01(사전 조사 분석)

3) 데이터 명명 규칙 파악

; 데이터 명명 규칙은 물리 데이터 모델에 적용해야 하는 규칙으로, 조직마다 다를 수 있으므로 물리 데이터 모델의 설계 전에 파악해야 한다.

예) 데이터 명명 규칙 예시

업무코드	•업무코드 3자리, 세부업무코드 3자리로 구성된다. •업무코드와 세부업무코드를 기준으로 거래코드, 프로그램 등이 명명되므로 표준을 준수해야 한다.
리소스 그룹	업무구분 3자리 + 세부업무구분 2자리로 명명한다.
거래코드	•거래코드는 입력되는 데이터의 유형에 따라 부여된다. •거래코드 명명규칙 : 서비스명과 동일 •거래처 코드와 서비스 명은 1:1 관계로 명명한다.

- 데이터 명명 규칙은 데이터 표준화 및 논리 데이터베이스 설계의 결과물 등을 통해 파악한다.
- 물리 데이터베이스 설계와 논리 데이터베이스 설계에 적용되는 명명 규칙은 서로 일관성을 유지해야 한다.
- 데이터 명명 규칙은 논리적 데이터 요소를 물리적 데이터 요소로 전환할 때 동일 명칭 부여의 근거로 사용된다.

2. 물리 데이터베이스 설계-SEC_01(사전 조사 분석)

3) 데이터 명명 규칙 파악

- 데이터 명명 규칙을 통해 중복 구축 등을 방지할 수 있다.
- 명명 규칙을 파악하려면 도메인과 데이터 사전에 대한 지식이 필요하다.
- 도메인 : 객체(Entity)에 포함된 속성들의 데이터 타입, 크기 등을 표준화 규칙에 따라 일관성 있게 정의한 것을 의미한다.

예) 도메인 예시

도메인 분류	도메인명	데이터 타입
코드	사원코드	VARCHAR2(10)
	매장코드	VARCHAR2(10)
	상품코드	VARCHAR2(10)
이름	사원이름	VARCHAR2(20)
	매장이름	VARCHAR2(20)
	상품이름	VARCHAR2(20)
수	사원수	NUMBER(10)
	상품수	NUMBER(10)

CHAR : 데이터 타입은 고정된 문자열을 저장하는 데 사용. 테이블을 만들 때, 무조건 고정된 길이를 지정해줘야 하고 범위는 1~2000 bytes이며 한글의 경우 글자당 2byte이므로 1000글자가 저장 가능하다. 영문, 숫자의 경우 1byte.

VARCHAR, VARCHAR2 : 가변 길이로 문자열을 저장할 수 있는 데이터 타입. 만약 우리가 VARCHAR2로 변수를 만들면 1~4000byte까지의 길이를 지정해줄 수 있다.

VARCHAR와 VARCHAR2 차이점 : 실제로 오라클 공식 문서에는 두 데이터 타입이 동의어라고 나와 있다.

하지만 VARCHAR2를 사용하는 것을 추천. 이유는 VARCHAR는 나중에 다른 방식의 데이터 타입으로 변경될 수 있기 때문

2. 물리 데이터베이스 설계-SEC_01(사전 조사 분석)

3) 데이터 명명 규칙 파악

- 데이터 사전(데이터 용어 사전)

- 전체 프로젝트 과정에서 일관성 있는 데이터 이름과 인터페이스를 제공하기 위해 데이터 속성의 논리명(Logical Name), 물리명(Physical Name), 용어 정의(Definition)를 기술해 놓은 것이다.
- 데이터 사전은 프로젝트에서 사용하는 명칭 부여의 근거로 사용된다.

2. 물리 데이터베이스 설계-SEC_01(사전 조사 분석)

4) 시스템 자원 파악

; 시스템 자원은 데이터베이스 설치에 영향을 미칠 수 있는 물리적인 요소들로, 사전에 미리 파악해야 한다.

- 시스템 자원은 하드웨어 자원, 운영체제 및 DBMS 버전, DBMS 파라미터 정보 등으로 구분한다.
- 하드웨어 자원

중앙처리장치	중앙처리장치(CPU)의 성능과 집중적인 부하 발생시간 등을 파악한다.
메모리	시스템의 전체 메모리 규모, 사용 중인 메모리 영역, 사용 가능한 메모리 영역 등 확보된 자원이나 실질적인 시스템 활용 정도 등을 파악한다.
디스크	전체 디스크의 크기, 확보된 디스크 지원, 디스크 분할 형태, 현재 디스크 활용률, 사용 가능한 디스크 공간 등을 파악한다.
I/O Controller	현행 시스템의 입·출력 컨트롤러의 성능, 운용의 적절성 등을 파악한다.
네트워크	네트워크의 처리량 처리 속도, 집중적인 부하 발생 시간, 동시 접속 가능 정도 등을 파악한다.

- 운영체제 및 DBMS 버전 : 운영체제와 DBMS 버전은 데이터베이스 운영에 영향을 미칠 수 있으므로 관련 요소 등을 파악하고 적절하게 관리해야 한다.
- DBMS 파라미터(Parameter) 정보
 - DBMS 파라미터는 데이터베이스 관리 시스템 별로 차이가 많고 관리 방법도 제 각각이므로 시스템 별 DBMS 파라미터의 종류 및 관리 대상 등을 파악한다.
 - DBMS의 저장 공간, 메모리 등에 대한 파라미터, 쿼리에서 활용하는 **옵티마이저(Optimizer)**의 사용 방법 등을 파악한다.

2. 물리 데이터베이스 설계-SEC_01(사전 조사 분석)

5) 데이터베이스 관리 요소 파악

; 데이터베이스 관리 요소는 데이터베이스 운영과 관련된 관리 요소로, 데이터베이스 시스템의 환경에 따라 달라질 수 있으므로 미리 파악해야 한다.

- 데이터베이스 관리 요소를 파악한 후 이를 기반으로 데이터베이스 시스템 조사 분석서를 작성한다.

예) 데이터베이스 시스템 조사 분석서 예시

구분	환경 변수	중앙(Center)	지역(Local)
운영체제 환경	IP Address	192,254,250,20	192,254,250,20
	CPU	4G	4G
	DISK SPACE	100G	100G
데이터베이스	SGA	40M	40M
	MAX DATA FILES	50	50
	DB BLOCK SIZE	16K	16K
	DB FILES	40	40
	LOG BUFFER	45215	45215
	PROCESS	40	40
	.	.	.
	.	.	.
	.	.	.

2. 물리 데이터베이스 설계-SEC_01(사전 조사 분석)

5) 데이터베이스 관리 요소 파악

- 시스템 조사 분석서를 기반으로 데이터베이스 구조, 이중화 구성, 분산 데이터베이스, 접근제어/접근통제, DB암호화 등의 범위와 특성을 파악한다.

데이터베이스구조	데이터베이스의 구조에 따라 문제 발생 시 대응 방법이 다르므로 서버와 데이터베이스의 구조 등을 파악한다.
이중화 구성	문제 발생에 대비하여 동일한 데이터베이스를 복제하여 관리하는 이중화 구성을 파악한다.
분산 데이터베이스	분산 데이터베이스는 물리적 재해 및 파손 등 데이터 유실을 최소화할 수 있고, 장애로 인한 데이터 유실 복구에 효과적이므로 데이터베이스의 분산 구조를 파악한다.
접근 제어/접근 통제	데이터베이스는 접근 가능한 사용자의 권한 남용으로 인한 정보 유출 및 변조가 가장 빈번하게 발생하므로 데이터베이스의 접근 제어 방법 등을 파악한다.
DB암호화	DB암호화는 데이터 암호화, 암호 키에 대한 인증 등을 통해 데이터 유출 시 데이터의 복호화를 어렵게 하므로 DB암호화의 특성을 파악한다.

물리 데이터베이스 설계-SEC_01(사전 조사 분석) 출제 예상 문제

출제 예상 문제(사전 조사 분석)

1. 데이터베이스 설계 시 물리적 설계 단계에서 수행하는 사항이 아닌 것은?

- ① 저장 레코드 양식 설계
- ② 레코드 집종의 분석 및 설계
- ③ 접근 경로 설계
- ④ 목표 DBMS에 맞는 스키마 설계

목표 DBMS에 맞는 스키마 설계는 논리적 설계 단계에서 수행할 사항이다.

물리 데이터베이스 설계

물리 데이터베이스 설계는 논리적 구조로 표현되어진 논리적 데이터베이스를 디스크 등의 물리적 저장장치에 저장할 수 있는 물리적 구조의 데이터로 변환하는 과정이다.

- 물리적 데이터베이스 구조의 기본적인 데이터 단위는 저장 레코드이다.
- 물리적 설계 단계에 꼭 포함되어야 할 것은 저장 레코드의 양식 설계, 레코드 집중(Record Clustering)의 분석 및 설계, 접근 경로 설계 등이다.

3. 데이터 명명 규칙에 대한 설명으로 틀린 것은?

- ① 데이터 명명 규칙은 조직마다 다를 수 있으므로 설계 전에 미리 파악해야 한다.
- ② 물리 데이터베이스 설계와 논리 데이터베이스 설계의 명명 규칙은 각 단계의 특성에 따라 다양하게 지정해야 한다.
- ③ 데이터 명명 규칙은 물리 데이터베이스 설계 시 중복 구축을 방지하는데 도움을 준다.
- ④ 데이터 명명 규칙은 논리적 데이터 요소를 물리적 요소로 전환할 때 동일 명칭 부여의 근거로 사용된다.

물리 데이터베이스는 논리 데이터베이스를 변환하여 구성하는 것으로 두 단계에 적용되는 명명 규칙은 일관성을 유지해야 한다.

데이터 명명 규칙

데이터 명명 규칙은 물리 데이터 모델에 적용해야 하는 규칙으로, 조직마다 다를 수 있기 때문에 물리 데이터 모델의 설계 전에 미리 파악을 해야 한다.

- 데이터 명명 규칙은 데이터 표준화 및 논리 데이터베이스 설계의 결과물 등을 통해 파악한다.
- 물리 데이터베이스 설계와 논리 데이터베이스 설계에 적용되는 명명

물리 데이터베이스 설계-SEC_01(사전 조사 분석) 출제 예상 문제

출제 예상 문제(사전 조사 분석)

5. 데이터베이스 설계 단계 중 물리적 설계에 대한 설명으로 옳지 않은 것은?

① 개념적 설계 단계에서 만들어진 정보 구조로부터 특정 목표 DBMS가 처리할 수 있는 스키마를 생성한다.

② 다양한 데이터베이스 응용에 대해서 처리 성능을 얻기 위해 데이터베이스 파일의 저장 구조 및 액세스 경로를 결정한다.

③ 물리적 저장장치에 저장할 수 있는 물리적 구조의 데이터로 변환하는 과정이다.

④ 물리적 설계에서 옵션 선택 시 응답 시간, 저장 공간의 효율화, 트랜잭션 처리율 등을 고려하여야 한다.

물리 데이터베이스 설계는 논리적 구조로 표현된 논리적 데이터베이스를 디스크 등의 물리적 저장장치에 저장할 수 있는 물리적 구조의 데이터로 변환하는 과정이다.

6. 물리적 데이터베이스 설계 시 고려 사항으로 가장 거리가 먼 것은?

① 레코드의 크기

② 파일에 대한 트랜잭션의 갱신과 참조 성향

7. 다음 중, 물리적 설계 단계 중 시스템 자원 파악에 해당하지 않는 것은?

① 하드웨어 자원

② 트랜잭션

③ 운영체제 및 DBMS 버전

④ DBMS파라미터

시스템 자원은 하드웨어 자원, 운영체제 및 DBMS버전, DBMS 파라미터 정보 등으로 구분한다.

하드웨어 자원

1. 중앙처리장치(CPU) : CPU의 성능과 집중적인 부하 발생시간 파악

2. 메모리 : 전체 메모리 규모, 사용 중인 메모리 영역, 사용 가능한 메모리 영역 등 확보된 자원이나 실질적인 시스템 활용 정도 등을 파악

3. 디스크 : 전체 디스크 크기, 확보된 디스크 자원, 디스크 분할 형태, 현재 디스크 활용률, 사용 가능한 디스크 공간 등을 파악

4. I/O Controller : 현행 시스템의 입, 출력 컨트롤러의 성능, 운용의 적절성 등을 파악

5. 네트워크 : 네트워크 처리량, 처리 속도, 집중적인 부하 발생 시간, 동시 접속 가능 정도 등을 파악

2. 물리 데이터베이스 설계-SEC_02(데이터베이스 저장 공간 설계)

1) 테이블(Table)

; 테이블은 데이터베이스의 가장 기본적인 객체로 로우(Row, 행)와 컬럼(Column, 열)으로 구성되어 있다.

- 데이터베이스의 모든 데이터는 테이블에 저장된다.
- 테이블은 논리 설계 단계의 개체(Entity)에 대응하는 객체이다.
- DBMS 종류에 따라 테이블의 명칭과 기능 등은 약간씩 차이가 있다.
- 테이블의 종류에는 일반 테이블, 클러스터 인덱스 테이블, 파티셔닝 테이블, 외부 테이블, 임시 테이블 등이 있다.

2. 물리 데이터베이스 설계-SEC_02(데이터베이스 저장 공간 설계)

2) 일반 테이블

; 일반 테이블은 현재 사용되는 대부분의 DBMS에서 표준 테이블로 사용되는 테이블 형태이다.

- 테이블에 저장되는 데이터의 로우(Row) 위치는 속성 값에 상관없이 데이터가 저장되는 순서에 따라 결정된다.

예) <사원> 테이블

사원번호	이름	입사일자	부서	직급	연봉
45S051	유순자	2018-05-09	영업부	사원	25,000,000
37S042	남민호	2019-01-30	기획부	과장	37,000,000
43S039	채진욱	2018-09-25	기획부	사원	28,000,000
36S201	박철수	2019-06-23	총무부	대리	35,000,000

데이터가 일정한 기준 없이 입력되는 순서에 따라 테이블에 저장된다.

2. 물리 데이터베이스 설계-SEC_02(데이터베이스 저장 공간 설계)

3) 클러스터드 인덱스 테이블(Clustered Index Table)

; 클러스터드 인덱스 테이블은 기본키(Primary Key)나 인덱스 키의 순서에 따라 데이터가 저장되는 테이블이다.

- 클러스터 인덱스 테이블은 일반적인 인덱스를 사용하는 테이블에 비해 접근 경로가 단축된다.

예) <사원> 테이블

사원번호	이름	입사일자	부서	직급	연봉
32S122	안병수	2018-05-05	영업부	과장	41,000,000
32S219	임현희	2018-12-11	총무부	부장	55,000,000
36S201	박철수	2019-06-23	총무부	대리	35,000,000
37S042	남민호	2019-01-30	기획부	과장	37,000,000
42S235	오동근	2019-04-19	기획부	사원	29,000,000
43S039	채진욱	2018-09-25	기획부	사원	28,000,000
45S051	유순자	2018-05-09	영업부	사원	25,000,000

데이터가 기본키인 '사원번호' 필드를 기준으로 오름차순 정렬되어 테이블에 저장된다.

2. 물리 데이터베이스 설계-SEC_02(데이터베이스 저장 공간 설계)

4) 파티셔닝 테이블(Partitioning Table)

; 파티셔닝 테이블은 대용량의 테이블을 작은 논리적 단위인 파티션(Partition)으로 나눈 테이블이다.

- 파티셔닝 테이블은 대용량의 데이터를 효과적으로 관리할 수 있지만 파티션 키를 잘못 구성하면 성능 저하 등의 역효과를 초래할 수 있다.
- 파티셔닝 방식에 따라 범위 분할(Range Partitioning), 해시 분할(Hash Partitioning), 조합 분할(Composite Partitioning) 등으로 나뉜다. 아래는 '입사일자' 필드를 기준으로 연도별 파티셔닝 되었다.

예) <파트1> 테이블

사원번호	이름	입사일자	부서	직급	연봉
32S122	안병수	2018-05-05	영업부	과장	41,000,000
45S051	유순자	2018-05-09	영업부	사원	25,000,000
43S039	채진욱	2018-09-25	기획부	사원	28,000,000

<파트2> 테이블

사원번호	이름	입사일자	부서	직급	연봉
37S042	남민호	2019-01-30	기획부	과장	37,000,000
42S235	오동근	2019-04-19	기획부	사원	29,000,000

범위 분할(Range Partitioning) : 지정한 열의 값을 기준으로 분할함.

해시 분할(Hash Partitioning) : 해시 함수를 적용한 결과 값에 따라 데이터를 분할함.

조합 분할(Composite Partitioning) : 범위 분할로 분할한 다음 해시 함수를 적용하여 다시 분할하는 방식

2. 물리 데이터베이스 설계-SEC_02(데이터베이스 저장 공간 설계)

5) 외부 테이블(External Table)

; 외부 테이블은 데이터베이스에서 일반 테이블처럼 이용할 수 있는 외부 파일로, 데이터베이스 내에 객체로 존재한다.

- 외부 테이블은 데이터웨어하우스(Data Warehouse)에서 ETL(Extraction Transformation Loading) 등의 작업에 유용하게 사용된다.

6) 임시 테이블(Temporary Table)

; 임시 테이블은 트랜잭션이나 세션 별로 데이터를 저장하고 처리할 수 있는 테이블이다.

- 임시 테이블에 저장된 데이터는 트랜잭션이 종료되면 삭제된다.
- 임시 테이블은 절차적인 처리를 위해 임시로 사용하는 테이블이다.

데이터웨어하우스(Data Ware-house) : 데이터웨어하우스란 조직이나 기업체의 중심이 되는 주요 업무 시스템에서 추출되어 새로이 생성된 데이터베이스로서, 의사결정지원 시스템을 지원하는 주체적, 통합적, 시간적 데이터의 집합체를 말한다.

ETL은 데이터웨어하우스를 사용하여 추출, 변환, 적재하는 일련의 모든 과정을 의미한다.

세션(session)이란 웹 사이트의 여러 페이지에 걸쳐 사용되는 사용자 정보를 저장하는 방법을 의미한다. 사용자가 브라우저를 닫아 서버와의 연결을 끝내는 시점까지를 세션이라고 한다.

2. 물리 데이터베이스 설계-SEC_02(데이터베이스 저장 공간 설계)

7) 컬럼(Column)

; 컬럼은 테이블의 열을 구성하는 요소로 데이터 타입(Data Type), 길이(Length) 등 으로 정의된다.

- 데이터 타입은 데이터의 일관성 유지를 위해 사용되는 가장 기본적인 것으로, 도메인을 정의한 경우 도메인에 따라 데이터의 타입과 길이가 정의된다.
- 두 컬럼을 비교하는 연산에서 두 컬럼의 데이터 타입이나 길이가 다르면 DBMS 내부적으로 데이터 타입을 변환한 후 비교 연산을 수행한다.
- 참조 관계인 컬럼들은 데이터 타입과 길이가 일치해야 한다.
- 데이터 타입과 길이 지정 시 고려 사항
 - 가변 길이 데이터 타입 : 예상되는 최대 길이로 정의
 - 고정 길이 데이터 타입 : 최소 길이로 지정
 - 소수점 이하 자릿수 : 소수점 이하 자릿수는 반올림되어 저장
- 데이터 타입에 따른 컬럼의 물리적인 순서
 - 고정 길이 컬럼이고 NOT Null인 컬럼 : 앞쪽
 - 가변 길이 컬럼 : 뒤쪽
 - Null 값이 많을 것으로 예상되는 컬럼 : 뒤쪽

2. 물리 데이터베이스 설계-SEC_02(데이터베이스 저장 공간 설계)

8) 테이블스페이스(Tablespace)

; 테이블스페이스는 테이블이 저장되는 논리적인 영역으로, 하나의 테이블스페이스에 하나 또는 그 이상의 테이블을 저장할 수 있다.

- 테이블을 저장하면 논리적으로는 테이블스페이스에 저장되고, 물리적으로는 해당 테이블스페이스와 연관된 데이터 파일(Data File)에 저장된다.
- 데이터베이스를 테이블, 테이블스페이스, 데이터 파일로 나눠 관리하면 논리적 구성이 물리적 구성에 종속되지 않아 투명성이 보장된다.
- 테이블스페이스는 데이터베이스에 저장되는 내용에 따라 테이블, 인덱스, 임시(Temporary) 등의 용도로 구분하여 설계한다.
- 테이블스페이스 설계 시 고려사항
 - 테이블스페이스는 업무별로 구분하여 지정한다.
 - 대용량 테이블은 하나의 테이블스페이스에 독립적으로 저장한다.
 - 테이블과 인덱스는 분리하여 저장한다.
 - LOB(Large Object)타입의 데이터는 독립적인 공간으로 지정한다.

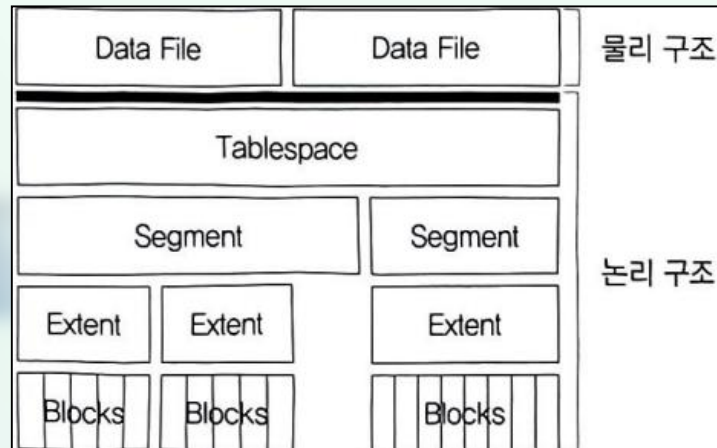
투명성 : 투명성이란 어떠한 사실이 존재함에도 마치 투명하여 보이지 않는 것처럼, 사실의 존재 여부를 염두에 두지 않아도 되는 성질을 말한다.

LOB(Large Object)는 대용량의 데이터를 저장할 수 있는 데이터 타입이다.

2. 물리 데이터베이스 설계-SEC_02(데이터베이스 저장 공간 설계)

8) 테이블스페이스(Tablespace)

예) Oracle의 물리/논리 저장 구조 계층



Oracle 프로그램은 데이터베이스를 Block, Extent, Segment, Tablespace, Data File로 나눠 관리한다.

Block, Extent, Segment, Tablespace는 논리적인 영역이고, Data File은 물리적인 영역이다.

Segment는 테이블, Extent는 데이터 블록의 모임, Block은 입, 출력 단위에 해당한다.

Data Block : Logical Block, ORACLE Block 또는 page라는 용어로 불리워지는데, OS Disk상의 일정한 Byte단위의 Space를 말한다.

Extent : 일정한 수의 연속된 ORACLE block 들을 말한다. 일정한 수라는 의미는 사용자가 지정한 값을 말한다.

Segment : 데이터베이스 내에 생성되는 모든 객체(테이블, 인덱스, 뷰, 시퀀스 등)들을 세그먼트(Segment)라고 한다.

물리 데이터베이스 설계-SEC_02(데이터베이스 저장 공간 설계) 출제 예상 문제

출제 예상 문제(데이터베이스 저장 공간 설계)

1. 데이터베이스의 모든 데이터가 저장되는 테이블에 대한 설명으로 틀린 것은?

- ① 테이블은 로우(Row)와 컬럼(Column)으로 구성되어 있다.
- ② 테이블은 논리 데이터 모델의 속성에 대응하는 객체이다.
- ③ 테이블의 종류에는 일반 유형 테이블, 클러스터 인덱스 테이블, 파티셔닝 테이블 등이 있다.
- ④ 테이블의 구성 요소인 컬럼은 데이터 타입과 길이로 정의된다.

테이블(Table)

테이블은 데이터베이스의 가장 기본적인 객체로 로우(Row, 행)와 컬럼(Column, 열, 속성)으로 구성되어 있다.

- 데이터베이스의 모든 데이터는 테이블에 저장된다.
- 테이블은 논리 설계 단계의 개체(Entity)에 대응하는 객체이다.
- DBMS 종류에 따라 테이블의 명칭과 기능 등은 약간씩 차이가 있다.
- 테이블의 종류에는 일반 테이블, 클러스터 인덱스 테이블, 파티셔닝 테이블, 외부 테이블, 임시 테이블이 있다.

2. 다음 중 테이블에 대한 설명으로 가장 옳지 않은 것은?

3. 다음 중 컬럼에 대한 설명으로 잘못된 것은?

- ① 컬럼은 테이블을 구성하는 요소이다.
- ② 컬럼은 데이터 타입과 길이로 정의된다.
- ③ 데이터 타입과 길이가 다른 컬럼의 값은 비교할 수 없다.
- ④ 참조 관계인 컬럼들은 서로 데이터 타입과 길이가 동일해야 한다.

컬럼(Column)

컬럼은 테이블의 열을 구성하는 요소로 데이터 타입, 길이 등으로 정의된다.

- 데이터 타입은 데이터의 일관성 유지를 위해 사용되는 가장 기본적인 것으로, 도메인을 정의한 경우 도메인에 따라 데이터의 타입과 길이가 정의된다.
- 두 컬럼을 비교하는 연산에서 두 컬럼의 데이터 타입이나 길이가 다르면 DBMS 내부적으로 데이터 타입을 변환한 후 비교 연산을 수행한다.
- 참조 관계인 컬럼들은 데이터 타입과 길이가 반드시 일치해야 한다.

4. 다음 중 테이블스페이스에 대한 설명으로 틀린 것은?

- ① 테이블스페이스는 테이블이 저장되는 물리적인 영역이다.
- ② 하나의 테이블스페이스에 하나 또는 그 이상의 테이블을 저장할 수 있다.

물리 데이터베이스 설계-SEC_02(데이터베이스 저장 공간 설계) 출제 예상 문제

출제 예상 문제(데이터베이스 저장 공간 설계)

5. 데이터베이스를 구성하는 테이블의 종류가 아닌 것은?

- ① 일반 유형 테이블
- ② 클러스터 인덱스 테이블
- ③ 인터럽트 테이블
- ④ 파티셔닝 테이블

테이블의 종류에는 일반 테이블, 클러스터 인덱스 테이블, 파티셔닝 테이블, 외부 테이블, 임시 테이블 등이 있다.

파티셔닝 테이블(Partition Table)

파티셔닝 테이블은 대용량의 테이블을 작은 논리적 단위인 파티션으로 나눈 테이블이다.

- 파티셔닝 테이블은 대용량의 데이터를 효과적으로 관리할 수 있지만 파티션 키를 잘못 구성하면 성능 저하 등의 역효과를 초래할 수 있다.

- 파티셔닝 방식에 따라 범위 분할(Range Partitioning), 해시 분할(Hash Partitioning), 조합 분할(Composite Partitioning) 등으로 나뉜다.

범위 분할 : 지정한 열의 값을 기준으로 분할함.

7. 다음 중, 테이블스페이스 설계 시 고려사항으로 틀린 것은?

- ① 테이블스페이스는 업무별로 구분하여 지정한다.
- ② 대용량 테이블은 하나의 테이블스페이스에 종속적으로 저장한다.
- ③ 테이블과 인덱스는 분리하여 저장한다.
- ④ LOB(Large Object)타입의 데이터는 독립적인 공간으로 지정한다.

테이블스페이스 설계 시 고려사항

- 테이블스페이스는 업무별로 구분하여 지정한다.
- 대용량 테이블은 하나의 테이블스페이스에 독립적으로 저장한다.
- 테이블과 인덱스는 분리하여 저장한다.
- LOB(Large Object)타입의 데이터는 독립적인 공간으로 지정한다.

2. 물리 데이터베이스 설계-SEC_03(트랜잭션 분석 / CRUD 분석)

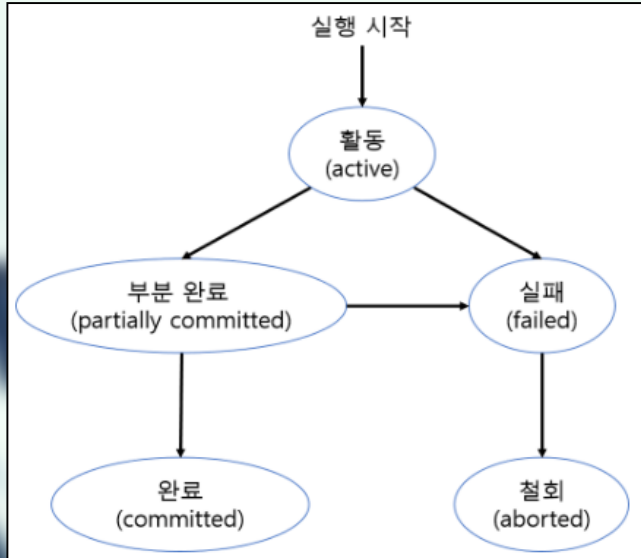
1) 트랜잭션(Transaction) 정의

; 트랜잭션은 데이터베이스의 상태를 변환시키는 하나의 논리적 기능을 수행하기 위한 작업의 단위 또는 한꺼번에 모두 수행되어야 할 일련의 연산들을 의미한다.

- 트랜잭션은 데이터베이스 시스템에서 병행 제어 및 회복 작업 시 처리되는 작업의 논리적 단위로 사용된다.
- 트랜잭션은 사용자가 시스템에 대한 서비스 요구 시 시스템이 응답하기 위한 상태 변환 과정의 작업 단위로 사용된다.

2. 물리 데이터베이스 설계-SEC_03(트랜잭션 분석 / CRUD 분석)

2) 트랜잭션의 상태



활동(Active)	트랜잭션이 실행 중인 상태이다.
실패(Failed)	트랜잭션 실행에 오류가 발생하여 중단된 상태이다.
철회(Aborted)	트랜잭션이 비정상적으로 종료되어 Rollback 연산을 수행한 상태이다.
부분 완료 (Partially Committed)	트랜잭션을 모두 성공적으로 실행한 후 Commit 연산이 실행되기 직전인 상태이다.
완료(Committed))	트랜잭션을 모두 성공적으로 실행한 후 Commit 연산을 실행한 후의 상태이다.

2. 물리 데이터베이스 설계-SEC_03(트랜잭션 분석 / CRUD 분석)

3) 트랜잭션의 특성

; 다음은 데이터의 무결성(Integrity)을 보장하기 위하여 DBMS의 트랜잭션이 가져야 할 특성이다.

Atomicity(원자성)	<ul style="list-style-type: none">•트랜잭션의 연산은 데이터베이스에 모두 반영되도록 완료(Commit)되든지 아니면 전혀 반영되지 않도록 복구(Rollback)되어야 한다.•트랜잭션 내의 모든 명령은 반드시 완벽히 수행되어야 하며, 모두가 완벽히 수행되지 않고 어느 하나라도 오류가 발생하면 트랜잭션 전부가 취소되어야 한다.
Consistency(일관성)	<ul style="list-style-type: none">•트랜잭션이 그 실행을 성공적으로 완료하면 언제나 일관성 있는 데이터베이스 상태로 변환한다.•시스템이 가지고 있는 고정 요소는 트랜잭션 수행 전과 트랜잭션 수행 완료 후의 상태가 같아야 한다.
Isolation (독립성,격리성,순차성)	<ul style="list-style-type: none">•둘 이상의 트랜잭션이 동시에 병행 실행되는 경우 어느 하나의 트랜잭션 실행 중에 다른 트랜잭션의 연산이 끼어들 수 없다.•수행중인 트랜잭션은 완전히 완료될 때까지 다른 트랜잭션에서 수행 결과를 참조할 수 없다.
Durability (영속성, 지속성)	성공적으로 완료된 트랜잭션의 결과는 시스템이 고장나더라도 영구적으로 반영되어야 한다.

2. 물리 데이터베이스 설계-SEC_03(트랜잭션 분석 / CRUD 분석)

4) CRUD 분석

; CRUD는 '생성(Create), 읽기(Read), 갱신(Update), 삭제>Delete)'의 앞 글자만 모아서 만든 용어이며, CRUD 분석은 데이터베이스 테이블에 변화를 주는 트랜잭션의 CRUD 연산에 대해 CRUD 매트릭스를 작성하여 분석하는 것이다.

- CRUD 분석으로 테이블에 발생하는 트랜잭션의 주기별 발생 횟수를 파악하고 연관된 테이블들을 분석하면 테이블에 저장되는 데이터의 양을 유추할 수 있다.
- CRUD 분석을 통해 많은 트랜잭션이 물리는 테이블을 파악할 수 있으므로 디스크 구성 시 유용한 자료로 활용할 수 있다.
- CRUD 분석을 통해 외부 프로세스 트랜잭션의 부하가 집중되는 데이터베이스 채널을 파악하고 분산 시킴으로써 연결 지연이나 타임아웃 오류를 방지할 수 있다.

CRUD 매트릭스는 2차원 형태의 표로서, 행에는 프로세스를, 열에는 테이블을, 행과 열이 만나는 위치에는 프로세스가 테이블에 발생시키는 변화를 표시하는 업무 프로세스와 데이터 간 상관 분석표이다.

2. 물리 데이터베이스 설계-SEC_03(트랜잭션 분석 / CRUD 분석)

5) CRUD 매트릭스

; CRUD 매트릭스는 2차원 형태의 표로서, 행(Row)에는 프로세스를, 열(Column)에는 테이블을, 행과 열이 만나는 위치에는 프로세스가 테이블에 발생시키는 변화를 표시하는 업무 프로세스와 데이터 간 상관 분석표이다.

- CRUD 매트릭스를 통해 프로세스의 트랜잭션이 테이블에 수행하는 작업을 검증한다.
- CRUD 매트릭스의 각 셀에는 Create, Read, Update, Delete의 앞 글자가 들어가며, 복수의 변화를 줄 때는 기본적으로 'C>D>U>R'의 우선순위를 적용하여 한 가지만 적지만, 활용 목적에 따라 모두 기록할 수 있다.

예) '주문 변경' 프로세스를 실행하려면 테이블의 데이터를 읽은(Read) 다음 수정(Update) 해야 하므로 R(Read)과 U(Update)가 필요하지만 CRUD 매트릭스에는 우선순위가 높은 'U'만 표시한다.

- CRUD 매트릭스가 완성되었다면 C, R, U, D 중 어느 것도 적히지 않은 행이나 열, C나 R이 없는 열을 확인하여 불필요하거나 누락된 테이블 또는 프로세스를 찾는다.

2. 물리 데이터베이스 설계-SEC_03(트랜잭션 분석 / CRUD 분석)

5) CRUD 매트릭스

예) 온라인 쇼핑몰의 CRUD 매트릭스 예시

테이블 프로세스	회원	상품	주문	주문목록	제조사
신규 회원 등록	C				
회원정보 변경	R, U				
주문 요청	R	R	C	C	
주문 변경			R	R, U	
주문 취소			R, D	R, D	
상품 등록		C			C, R
상품정보 변경		R, U			R, U

2. 물리 데이터베이스 설계-SEC_03(트랜잭션 분석 / CRUD 분석)

6) 트랜잭션 분석

; 트랜잭션 분석의 목적은 CRUD 매트릭스를 기반으로 테이블에 발생하는 트랜잭션 양을 분석하여 테이블에 저장되는 데이터의 양을 유추하고 이를 근거로 DB 용량을 산정하고 DB 구조를 최적화하는 것이다.

- 트랜잭션 분석은 업무 개발 담당자가 수행한다.
- 트랜잭션 분석을 통해 프로세스가 과도하게 접근하는 테이블을 확인하여 여러 디스크에 배치함으로써 디스크 입·출력 분산을 통한 성능 향상을 가져올 수 있다.

데이터 양 유추 : 직원이 10,000명이고, 직원 한 명당 발령은 2번, 상여금 변경은 20번 발생한다면 발령 테이블에는 $10,000 \times 2 = 20,000$ 건, 상여금 테이블에는 $10,000 \times 20 = 200,000$ 건의 데이터가 발생한다는 것을 유추할 수 있다.
※ 이 계산은 정확한 데이터 양을 계산한 것이 아니라 저장장치나 파일 등의 크기를 지정하기 위해 대략적으로 유추 한 것이다.
업무 개발 담당자는 애플리케이션을 분석 및 설계하는 일을 담당한다.

2. 물리 데이터베이스 설계-SEC_03(트랜잭션 분석 / CRUD 분석)

7) 트랜잭션 분석서

; 트랜잭션 분석서는 단위 프로세스와 CRUD 매트릭스를 이용하여 작성하며, 구성 요소에는 단위 프로세스, CRUD 연산, 테이블명, 컬럼명, 테이블 참조 횟수, 트랜잭션 수, 발생 주기 등이 있다.

- 단위 프로세스 : 업무를 발생시키는 가장 작은 단위의 프로세스
- CRUD 연산 : 프로세스의 트랜잭션이 데이터베이스 테이블에 영향을 주는 C, R, U, D의 4가지 연산
- 테이블명, 컬럼명 : 프로세스가 접근하는 데이터베이스의 테이블명을 기록한다. 필요한 경우 테이블의 컬럼명을 적는다. 컬럼명을 적을 때는 마침표로 연결하여 '테이블.컬럼명'과 같이 적는다.
- 테이블 참조 횟수 : 프로세스가 테이블을 참조하는 횟수
- 트랜잭션 수 : 주기별로 수행되는 트랜잭션 횟수
- 발생 주기 : 연, 분기, 월, 일, 시간 등 트랜잭션 횟수를 측정하기 위한 발생 주기

예) '주문요청' 프로세스에 대한 트랜잭션 분석서 예시

프로세스	CRUD	테이블명	컬럼명	참조횟수	트랜잭션 수	주기
주문요청	R	회원	회원번호, 회원명, 주소	1	150	일
	R	상품	상품번호, 상품명, 재고량	1	150	
	C	주문	주문번호, 일자, 회원번호	3	450	
	C	주문목록	주문번호, 상품번호, 수량, 가격	5	750	

물리 데이터베이스 설계- SEC_03(트랜잭션 분석/CRUD 분석) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(트랜잭션 분석/CRUD 분석)

1. 다음 설명과 관련 있는 트랜잭션의 특징은?

트랜잭션의 연산은 모두 실행되거나, 모두 실행되지 않아야 한다.

- ① Durability ② Isolation
- ③ Consistency ④ Atomicity

원자성(Atomicity)은 All Or Nothing, 즉 전부가 아니면 아무것도 아니라는 뜻이다. 완전히 수행하든지 하나도 수행하지 말든지 일부는 인정하지 않겠다는 뜻이다.

Atomicity(원자성)

- 트랜잭션의 연산은 데이터베이스에 모두 반영되도록 완료(Commit)되든지 아니면 전혀 반영되지 않도록 복구(Rollback)되어야 한다.
- 트랜잭션 내의 모든 명령은 반드시 완벽히 수행되어야 하며, 모두가 완벽히 수행되지 않고 어느 하나라도 오류가 발생하면 트랜잭션 전부가 취소되어야 한다.

Consistency(일관성)

- 트랜잭션이 그 실행을 성공적으로 완료하면 언제나 일관성 있는

3. 데이터베이스에 영향을 주는 생성, 읽기, 갱신, 삭제 연산으로 프로세스와 테이블 간에 매트릭스를 만들어서 트랜잭션을 분석하는 것은?

- ① CASE 분석 ② 일치 분석
- ③ **CRUD 분석** ④ 연관성 분석

CRUD 분석

CRUD는 생성(Create), 읽기(Read), 갱신(Update), 삭제>Delete)의 앞 글자만 모아서 만든 용어이며, CRUD 분석은 데이터베이스 테이블에 변화를 주는 트랜잭션의 CRUD 연산에 대해 CRUD 매트릭스를 작성하여 분석하는 것이다.

- CRUD 분석으로 테이블에 발생하는 트랜잭션의 주기별 발생 횟수를 파악하고 연관된 테이블들을 분석하면 테이블에 저장되는 데이터의 양을 유추할 수 있다.
- CRUD 분석을 통해 많은 트랜잭션이 물리는 테이블을 파악할 수 있으므로 디스크 구성 시 유용한 자료로 활용할 수 있다.
- CRUD 분석을 통해 외부 프로세스 트랜잭션의 부하가 집중되는 데이터베이스 채널을 파악하고 분산 시킴으로써 연결 지연이나 타임아웃 오류를 방지할 수 있다.

물리 데이터베이스 설계- SEC_03(트랜잭션 분석/CRUD 분석) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(트랜잭션 분석/CRUD 분석)

5. 무결성을 보장하기 위해 트랜잭션이 가져야 할 특성에 대한 설명으로 옳지 않은 것은?

- ① 트랜잭션 내의 모든 명령은 반드시 완벽히 수행되어야 하며, 모두가 완벽히 수행되지 않고 어느 하나라도 오류가 발생하면 트랜잭션 전부가 취소되어야 한다.
- ② 트랜잭션의 수행과 관계 없이 데이터베이스가 가지고 있는 고정 요소는 일관되어야 한다.
- ③ 둘 이상의 트랜잭션이 동시에 병행 실행되는 경우 어느 하나의 트랜잭션 실행 중에 다른 트랜잭션의 연산이 끼어들 수 없다.
- ④ Commit과 Rollback 명령어에 의해 보장 받는 트랜잭션의 특성은 일관성이다.

트랜잭션 특성에 대한 설명에서 Commit, Rollback이 있다면 바로 원자성을 떠올려야 한다.

6. 데이터베이스에서 하나의 논리적 기능을 수행하기 위한 작업의 단위 또는 한꺼번에 모두 수행되어야 할 일련의 연산들을 의미하는 것은?

- ① 트랜잭션
- ② 뷰

7. 트랜잭션의 주요 특성 중 하나로, 둘 이상의 트랜잭션이 동시에 병행 실행되는 경우 어느 하나의 트랜잭션 실행 중에 다른 트랜잭션의 연산이 끼어들 수 없음을 의미하는 것은?

- ① Log
- ② Consistency
- ③ Isolation
- ④ Durability

다른 트랜잭션의 개입 없이 독립적으로 실행한다는 특성은 독립성(Isolation)이다.

8. 다음과 같은 트랜잭션의 특성은?

시스템이 가지고 있는 고정 요소는 트랜잭션 수행 전과 트랜잭션 수행 완료 후의 상태가 같아야 한다.

- ① 원자성(atomicity)
- ② 일관성(consistency)
- ③ 격리성(isolation)
- ④ 영속성(durability)

트랜잭션 수행 전과 후가 일관성 있게 동일해야 한다는 특성은 일관성이다.

물리 데이터베이스 설계- SEC_03(트랜잭션 분석/CRUD 분석) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(트랜잭션 분석/CRUD 분석)

9. 무결성을 보장하기 위해 트랜잭션이 가져야 할 특성에 대한 트랜잭션의 정의 및 특징이 아닌 것은?

- ① 한꺼번에 수행되어야 할 일련의 데이터베이스 연산집합
- ② 사용자의 시스템에 서비스 요구 시스템의 상태 변환 과정의 작업 단위
- ③ 병행제어 및 회복 작업의 논리적 작업 단위
- ④ 트랜잭션의 연산이 데이터베이스에 모두 반영되지 않고 일부만 반영시키는 원자성의 성질

10. 트랜잭션(Transaction)은 보통 일련의 연산 작업이란 의미로 사용되며, 하나의 논리적 기능을 수행하는 작업의 단위이다. 트랜잭션(Transaction)이 가져야 할 특성에 해당하지 않는 것은?

- ① 원자성(Atomicity) ② 투명성(Transparency)
- ③ 일관성(Consistency) ④ 격리성(Isolation)

투명성이란 어떠한 사실이 존재함에도 마치 투명하여 보이지 않는 것처럼, 사실의 존재 여부를 염두에 두지 않아도 되는 성질을 의미한다.

11. A 은행에서 B라는 사람이 고객 인증 절차를 거쳐 잔액을 조회한 후, 타인에게 송금하는 도중에 장애가 발생하였을 경우 문제가 발생한다. 이러한 경우의 부작용을 방지할 수 있는 트랜잭션의 특성은?

- ① 일관성(Consistency) ② 고립성(Isolation)
- ③ 지속성(Duration) ④ 원자성(Atomicity)

12. 트랜잭션 분석서를 작성하는 과정에서 필요 없는 항목에 해당하는 것은?

- ① 단위 프로세스 ② 연관 모듈
- ③ CRUD 연산 ④ 테이블 참조 횟수

트랜잭션 분석서

트랜잭션 분석서는 단위 프로세스와 CRUD 매트릭스를 이용하여 작성하며, 구성 요소에는 단위 프로세스, CRUD 연산, 테이블명, 컬럼명, 테이블 참조 횟수, 트랜잭션 수, 발생 주기 등이 있다.

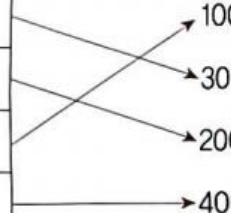
- 단위 프로세스 : 업무를 발생시키는 가장 작은 단위의 프로세스
- CRUD 연산 : 프로세스의 트랜잭션이 데이터베이스 테이블에 영향을 주는 C, R, U, D의 4가지 연산
- 테이블명, 컬럼명 : 프로세스가 접근하는 데이터베이스의 테이블명을

2. 물리 데이터베이스 설계-SEC_04(인덱스 설계)

1) 인덱스(Index)의 개념

; 인덱스는 데이터 레코드를 빠르게 접근하기 위해 <키 값, 포인터> 쌍으로 구성되는 데이터 구조이다.

예) 학생 릴레이션의 학번 속성에 대한 인덱스

〈인덱스〉		〈학생〉				
학번(키 값)	주소(포인터)	주소	학번	이름	학년	학과
17083	300		18065	강홍구	3	컴퓨터공학과
18032	200		17083	김동오	4	경영학과
18065	100		18032	정연중	3	영문과
19054	400	→ 400	19054	심희정	2	방송학과

위의 인덱스에서 키 값은 '학번'이고 포인터는 해당 '학번'이 저장된 레코드의 물리적인 주소이다.

키 값인 '학번'이 정렬되어 있기 때문에 인덱스를 통해 레코드를 빠르게 접근할 수 있다.

2. 물리 데이터베이스 설계-SEC_04(인덱스 설계)

1) 인덱스(Index)의 개념

- 인덱스는 데이터가 저장된 물리적 구조와 밀접한 관계가 있다.
- 인덱스는 레코드가 저장된 물리적 구조에 접근하는 방법을 제공한다.
- 인덱스를 통해서 파일의 레코드에 대한 액세스를 빠르게 수행할 수 있다.
- 레코드의 삽입과 삭제가 수시로 일어나는 경우에는 인덱스의 개수를 최소로 하는 것이 효율적이다.
- 데이터 정의어(DDL)를 이용하여 사용자가 생성, 변경, 제거할 수 있다.
- 인덱스가 없으면 특정한 값을 찾기 위해 모든 데이터 페이지를 확인하는 TABLE SCAN이 발생한다.
- 기본키를 위한 인덱스를 기본 인덱스라 하고, 기본 인덱스가 아닌 인덱스들을 보조 인덱스라고 한다. ●

대부분의 관계형 데이터베이스 관리 시스템에서는 모든 기본키에 대해서 자동적으로 기본 인덱스를 생성한다.

- 레코드의 물리적 순서가 인덱스의 엔트리 순서와 일치하게 유지되도록 구성되는 인덱스를 클러스터드 (Clustered) 인덱스라고 한다.
- 인덱스는 인덱스를 구성하는 구조나 특징에 따라 트리 기반 인덱스, 비트맵 인덱스, 함수 기반 인덱스, 비트맵 조인 인덱스, 도메인 인덱스 등으로 분류된다.

TABLE SCAN은 테이블에 있는 모든 레코드를 순차적으로 읽는 것으로 FULL TABLE SCAN이라고도 한다. 일반적으로 적용 가능한 인덱스가 없거나 분포도가 넓은 데이터를 검색할 때는 FULL TABLE SCAN을 사용한다.

2. 물리 데이터베이스 설계-SEC_04(인덱스 설계)

2) 클러스터드 인덱스/ 논 클러스터드 인덱스

- 클러스터드 인덱스(Clustered Index)

- > 인덱스 키의 순서에 따라 데이터가 정렬되어 저장되는 방식이다.
- > 실제 데이터가 순서대로 저장되어 있어 인덱스를 검색하지 않아도 원하는 데이터를 빠르게 찾을 수 있다.
- > 데이터 삽입, 삭제 발생 시 순서를 유지하기 위해 데이터를 재정렬해야 한다.
- > 한 개의 릴레이션에 하나의 인덱스만 생성할 수 있다.

- 논 클러스터드 인덱스(Non-Clustered Index)

- > 인덱스의 키 값만 정렬되어 있을 뿐 실제 데이터는 정렬되지 않는 방식이다.
- > 데이터를 검색하기 위해서는 먼저 인덱스를 검색하여 실제 데이터의 위치를 확인해야 하므로 클러스터드 인덱스에 비해 검색 속도가 떨어진다.
- > 한 개의 릴레이션에 여러 개의 인덱스를 만들 수 있다.

2. 물리 데이터베이스 설계-SEC_04(인덱스 설계)

3) 트리 기반 인덱스

; 트리 기반 인덱스는 인덱스를 저장하는 블록들이 트리 구조를 이루고 있는 것으로, 상용 DBMS에서는 트리 구조 기반의 B+ 트리 인덱스를 주로 활용한다.

● B트리 인덱스

- 일반적으로 사용되는 인덱스 방식으로, 루트 노드에서 하위 노드로 키 값의 크기를 비교해 나가면서 단말 노드에서 찾고자 하는 데이터를 검색한다.
- 키 값과 레코드를 가리키는 포인터들이 트리 노드에 **오름차순으로 저장**된다.
- 모든 리프 노드는 같은 레벨에 있다.
- **브랜치 블록(Branch Block)**과 **리프 블록(Leaf Block)**으로 구성된다.
 - ▶ 브랜치 블록 : 분기를 위한 목적으로 사용되고, 다음 단계를 가리키는 포인터를 가지고 있음
 - ▶ 리프 블록: 인덱스를 구성하는 컬럼 데이터와 해당 데이터의 행 위치를 가리키는 레코드 식별자로 구성됨

2. 물리 데이터베이스 설계-SEC_04(인덱스 설계)

3) 트리 기반 인덱스

● B+ 트리 인덱스

- B+ 트리는 B 트리의 변형으로 단말 노드가 아닌 노드로 구성된 인덱스 세트(Index Set)와 단말 노드로만 구성된 순차세트(Sequence Set)로 구분된다.
- 인덱스 세트에 있는 노드들은 단말 노드에 있는 키 값을 찾아갈 수 있는 경로로만 제공되며, 순차 세트에 있는 단말 노드가 해당 데이터 레코드의 주소를 가리킨다.
- 인덱스 세트에 있는 모든 키 값이 단말 노드에 다시 나타나므로 단말 노드만을 이용한 순차 처리가 가능하다.

2. 물리 데이터베이스 설계-SEC_04(인덱스 설계)

3) 트리 기반 인덱스

● B-tree VS B+ tree

- 아래서 말하는 데이터는 자료구조 상 value를 가리킴.

구분	B-tree	B+tree
데이터 저장	리프 노드, 브랜치 노드 모두 데이터 저장 가능	오직 리프 노드에만 데이터 저장 가능
트리의 높이	높음	낮음(한 노드 당 key를 많이 담을 수 있음)
풀 스캔 시, 검색 속도	모든 노드 탐색	리프 노드에서 선형 탐색
키 중복	없음	있음(리프 노드에 모든 데이터가 있기 때문)
검색	자주 access 되는 노드를 루트 노드 가까이 배치할 수 있고, 루트 노드에서 가까운 경우, 브랜치 노드에도 데이터가 존재하기 때문에 빠름	리프 노드까지 가야 데이터 존재
링크드 리스트	없음	리프 노드끼리 링크드 리스트로 연결

2. 물리 데이터베이스 설계-SEC_04(인덱스 설계)

4) 비트맵 인덱스

; 비트맵 인덱스는 인덱스 컬럼의 데이터를 Bit 값인 0 또는 1로 변환하여 인덱스 키로 사용하는 방법이다.

- 비트맵 인덱스의 목적은 키 값을 포함하는 로우(Row)의 주소를 제공하는 것이다.
- 비트맵 인덱스는 분포도가 좋은 컬럼에 적합하며, 성능 향상 효과를 얻을 수 있다.
- 데이터가 Bit로 구성되어 있기 때문에 효율적인 논리 연산이 가능하고 저장 공간이 작다.
- 비트맵 인덱스는 다중 조건을 만족하는 튜플의 개수 계산에 적합하다.
- 비트맵 인덱스는 동일한 값이 반복되는 경우가 많아 압축 효율이 좋다.

로우(Row)의 주소 : 비트맵에서 비트의 위치는 테이블에서 로우(Row)의 상대적인 위치를 의미한다. 해당 테이블이 시작되는 물리적인 주소를 기반으로 실제 로우의 물리적 위치를 계산할 수 있다.

분포도, 선택성(Selectivity) : $(\text{조건에 맞는 레코드 수} / \text{전체 릴레이션 레코드 수}) \times 100$,
전체 레코드 중 조건에 맞는 레코드의 숫자가 적은 경우 분포도가 좋다고 한다.

분포도가 10~15%인 경우 효율적인 인덱스 검색을 할 수 있다.

분포도를 선택성(Selectivity)이란 용어로 사용하기도 한다.

2. 물리 데이터베이스 설계-SEC_04(인덱스 설계)

5) 함수 기반 인덱스

; 함수 기반 인덱스는 컬럼의 값 대신 컬럼에 특정 함수(Function)나 수식(Expression)을 적용하여 산출된 값을 사용하는 것으로, B+ 트리 인덱스 또는 비트맵 인덱스를 생성하여 사용한다.

- 함수 기반 인덱스는 데이터를 입력하거나 수정할 때 함수를 적용해야 하므로 부하가 발생할 수 있다.
- 사용된 함수가 사용자 정의 함수일 경우 시스템 함수보다 부하가 더 크다.
- 함수 기반 인덱스는 대소문자, 띄어쓰기 등에 상관없이 조회할 때 유용하게 사용된다.
- 적용 가능한 함수의 종류 : 산술식(Arithmetic Expression), 사용자 정의 함수, PL/SQL Function, SQL Function, Package, C callout 등

C callout은 어떤 함수를 호출한다라는 의미이다.

2. 물리 데이터베이스 설계-SEC_04(인덱스 설계)

6) 비트맵 조인 인덱스

; 비트맵 조인 인덱스는 다수의 조인된 객체로 구성된 인덱스로, 단일 객체로 구성된 일반적인 인덱스와 액세스 방법이 다르다.

- 비트맵 조인 인덱스는 비트맵 인덱스와 물리적 구조가 동일하다.
- 오라클 9i에서는 이 비트맵 인덱스를 한 단계 더 발전시켜 비트맵 조인 인덱스(Bitmap Join Index) 기능을 제공하고 있다.
- 테이블들 사이의 조인 결과에 비트맵 인덱스를 생성하는 것으로 질의 처리를 위한 조인을 피함으로써 성능 향상을 제공한다.
- Bitmap Join Index를 사용하면 실제 join 작업이 발생하지 않도록 할 수 있다.

7) 도메인 인덱스

; 도메인 인덱스는 개발자가 필요한 인덱스를 직접 만들어 사용하는 것으로, 확장형 인덱스(Extensible Index)라고도 한다.

- 개발자가 필요에 의해 만들었지만 프로그램에서 제공하는 인덱스처럼 사용할 수도 있다.

도메인 인덱스 : 오라클8i에서부터 새롭게 도입된 개념으로 개발자가 자신이 원하는 인덱스 타입을 생성할 수 있게 함으로써 오라클의 인덱스 시스템에 많은 확장을 가져다 주었다.

즉, 데이터베이스에는 아직 존재하지도 않는 새로운 인덱스 타입을 자신이 스스로 정의하여 오라클에서 지원하는 인덱스처럼 사용할 수가 있다는 것이다.

2. 물리 데이터베이스 설계-SEC_04(인덱스 설계)

8) 인덱스 설계

; 인덱스를 설계할 때는 분명하게 드러난 컬럼에 대해 기본적인 인덱스를 먼저 지정한 후 개발 단계에서 필요한 인덱스의 설계를 반복적으로 진행한다.

● 인덱스 설계 순서

- ① 인덱스의 대상 테이블이나 컬럼 등을 선정한다.
- ② 인덱스의 효율성을 검토하여 인덱스 최적화를 수행한다.
- ③ 인덱스 정의서를 작성한다.

예) 인덱스 정의서 예시

엔티 티명	테이블명	인덱스명	컬럼명	타입	테이블 스페이스	인덱스 유형	정렬
사원	EMP	PK_EMP	EMPNO	NUMBER(5)	ISTEXT01	UNIQUE	ASC
		1_EMP2	DEPTNO	NUMBER(5)	ISTEXT01	NOT UNIQUE	ASC
고객	CUST	PK_CUST	CUSTNO	NUMBER(5)	ISTEXT01	UNIQUE	ASC
부서	DEPT	PK_DEPT	DEPTNO	NUMBER(3)	ISTEXT01	UNIQUE	ASC
주문	ORDER	PK_ORDER	ORDERNO	NUMBER(6)	ISTEXT01	UNIQUE	ASC

인덱스 설계 : 프로젝트 진행 중에는 데이터나 프로세스 내용 등에 변경이 많기 때문에 처음부터 모든 인덱스를 한꺼번에 정의하는 것은 불가능하다. 인덱스 설계 단계에서는 기본적인 인덱스를 지정하고 프로젝트를 진행하면서 SQL 문장 구조 등을 검토하여 반복적으로 인덱스 설계를 진행해야 한다.

2. 물리 데이터베이스 설계-SEC_04(인덱스 설계)

9) 인덱스 대상 테이블 선정 기준

- MULTI BLOCK READ 수에 따라 판단

예) MULTI BLOCK READ가 16이면, 테이블의 크기가 16블록 이상일 경우 인덱스 필요

- 랜덤 액세스가 빈번한 테이블
- 특정 범위나 특정 순서로 데이터 조회가 필요한 테이블
- 다른 테이블과 순차적 조인이 발생하는 테이블

MULTI BLOCK READ란 테이블 액세스 시 메모리에 한 번에 읽어 들일 수 있는 블록의 수를 말한다.

2. 물리 데이터베이스 설계-SEC_04(인덱스 설계)

10) 인덱스 대상 컬럼 선정 기준

- 인덱스 컬럼의 분포도가 10~15% 이내인 컬럼
 - 분포도 = (컬럼 값의 평균 Row 수 / 테이블의 총 Row 수) × 100
- 분포도가 10~15% 이상이어도 부분 처리를 목적으로 하는 컬럼
- 입·출력 장표 등에서 조회 및 출력 조건으로 사용되는 컬럼
- 인덱스가 자동 생성되는 기본키와 Unique키 제약 조건을 사용한 컬럼
- 가능한 한 수정이 빈번하지 않은 컬럼
- ORDER BY, GROUP BY, UNION이 빈번한 컬럼
- 분포도가 좁은 컬럼은 단독 인덱스로 생성
- 인덱스들이 자주 조합되어 사용되는 경우 하나의 결합 인덱스(Concatenate Index)로 생성

ORDER BY, GROUP BY, UNION : ORDER BY는 정렬, GROUP BY는 그룹 지정, UNION은 통합 지정 시 사용하는 SQL명령어이다.

결합 인덱스 : 결합 인덱스란 한 릴레이션 내에 존재하는 여러 컬럼들을 묶어 하나의 인덱스로 만드는 것을 말한다. 결합 인덱스는 컬럼 순서에 따라 액세스하는 범위가 달라질 수 있으므로 유의해야 한다.

- 컬럼 순서 우선순위
 - ① 항상 사용되는 컬럼
 - ② '=' 연산이 되는 컬럼
 - ③ 분포도가 좋은 컬럼
 - ④ 정렬이 자주 발생하는 컬럼

2. 물리 데이터베이스 설계-SEC_04(인덱스 설계)

11) 인덱스 설계 시 고려사항

- 새로 추가되는 인덱스는 기존 액세스 경로에 영향을 미칠 수 있다.
- 인덱스를 지나치게 많이 만들면 오버헤드가 발생한다.
- 넓은 범위를 인덱스로 처리하면 많은 오버헤드가 발생한다.
- 인덱스를 만들면 추가적인 저장 공간이 필요하다.
- 인덱스와 테이블 데이터의 저장 공간이 분리되도록 설계한다.

인덱스와 테이블 분리
인덱스와 테이블을 분리하는 형태는 데이터베이스의 가장 일반적인 형태로, 데이터 저장 시 인덱스의 영향을 받지 않아 저장이 빠르다.

물리 데이터베이스 설계-SEC_04(인덱스 설계) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(인덱스 설계)

1. 인덱스(Index)에 대한 설명으로 부적절한 것은?

- ① 인덱스는 데이터베이스의 물리적 구조와 밀접한 관계가 있다.
- ② 인덱스는 하나 이상의 필드로 만들어도 된다.
- ③ 레코드의 삽입과 삭제가 수시로 일어나는 경우는 인덱스의 개수를 최대한 많이 한다.
- ④ 인덱스를 통해서 테이블의 레코드에 대한 액세스를 빠르게 수행할 수 있다.

인덱스(Index)의 개념

인덱스는 데이터 레코드를 빠르게 접근하기 위해 <키 값, 포인터> 쌍으로 구성되는 데이터 구조이다.

- 인덱스는 데이터가 저장된 물리적 구조와 밀접한 관계가 있다.
- 인덱스는 레코드가 저장된 물리적 구조에 접근하는 방법을 제공한다.
- 인덱스를 통해서 파일의 레코드에 대한 액세스를 빠르게 수행할 수 있다.
- 레코드의 삽입과 삭제가 수시로 일어나는 경우에는 인덱스의 개수를 최소로 하는 것이 효율적이다.

3. 데이터베이스에서 인덱스(Index)와 관련한 설명으로 틀린 것은?

- ① 인덱스의 기본 목적은 검색 성능을 최적화하는 것으로 볼 수 있다.
- ② B-트리 인덱스는 분기를 목적으로 하는 Branch Block을 가지고 있다.
- ③ BETWEEN 등 범위(Range) 검색에 활용될 수 있다.
- ④ 시스템이 자동으로 생성하여 사용자가 변경할 수 없다.

데이터 정의어(DDL)를 이용하여 사용자가 생성, 변경, 제거할 수 있다.

4. 다음 중 함수 기반 인덱스에 대한 설명으로 옳지 않은 것은?

- ① 함수 기반 인덱스는 컬럼 값 자체가 아니라 컬럼에 특정 함수를 적용한 값이다.
- ② 시스템 함수를 사용하면 사용자 정의 함수를 사용할 때 보다 더 많은 부하가 발생한다.
- ③ 함수 기반 인덱스는 데이터를 입력하거나 수정할 때 함수를 적용 해야 하므로 부하가 발생할 수 있다.
- ④ 함수 기반 인덱스로 B+ 트리 인덱스 또는 비트맵 인덱스를 생성하여 사용한다.

함수 기반 인덱스

함수 기반 인덱스는 컬럼의 값 대신 컬럼에 특정 함수나 수식을 적용하여 산출된 값을 사용하는 것으로 B+ 트리 인덱스 또는 비트맵

물리 데이터베이스 설계-SEC_04(인덱스 설계) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(인덱스 설계)

5. 다음 중 비트맵 인덱스에 대한 설명으로 옳지 않은 것은?

- ① 비트맵 인덱스란 인덱스 컬럼의 데이터를 Bit 값인 0 또는 1로 변환하여 인덱스 키로 사용하는 방법이다.
- ② 비트맵 인덱스는 분포도가 넓은 컬럼에 적합하다.
- ③ 비트맵 인덱스의 목적은 키 값을 포함하는 로우의 주소를 제공하는 것이다.
- ④ 비트맵 인덱스는 동일한 값이 반복될 확률이 높아 압축 효율이 매우 좋다.

비트맵 인덱스

비트맵 인덱스는 인덱스 컬럼의 데이터를 Bit 값인 0 또는 1로 변환하여 인덱스 키로 사용하는 방법이다.

- 비트맵 인덱스의 목적은 키 값을 포함하는 로우(Row)의 주소를 제공하는 것이다.
- 비트맵 인덱스는 분포도가 좋은 컬럼에 적합하며, 성능 향상 효과를 얻을 수 있다.
- 데이터가 bit로 구성되어 있기 때문에 효율적인 논리 연산이 가능하고 저장 공간이 작다.

7. 다음 중, 인덱스 대상 컬럼 선정 기준으로 틀린 것은?

- ① 인덱스 컬럼의 분포도가 10~15% 이내인 컬럼
- ② 인덱스가 자동 생성되는 기본키와 Unique키 제약 조건을 사용한 컬럼
- ③ 가능한 한 수정이 많은 컬럼
- ④ 분포도가 10~15% 이상이어도 부분 처리를 목적으로 하는 컬럼

인덱스 대상 컬럼 선정 기준

- 인덱스 컬럼의 분포도가 10~15% 이내인 컬럼

분포도 = (컬럼 값의 평균 Row 수 / 테이블의 총 Row 수) * 100

- 분포도가 10~15% 이상이어도 부분 처리를 목적으로 하는 컬럼
- 입,출력 장표 등에서 조회 및 출력 조건으로 사용되는 컬럼
- 가능한 한 수정이 빈번하지 않은 컬럼
- ORDER BY, GROUP BY, UNION이 빈번한 컬럼
- 분포도가 좁은 컬럼은 단독 인덱스로 생성
- 인덱스들이 자주 조합되어 사용하는 경우 하나의 결합 인덱스로 생성

8. 다음 중 클러스터드 인덱스, 넌 클러스터드 인덱스에 대한 설명으로 틀린 것은?

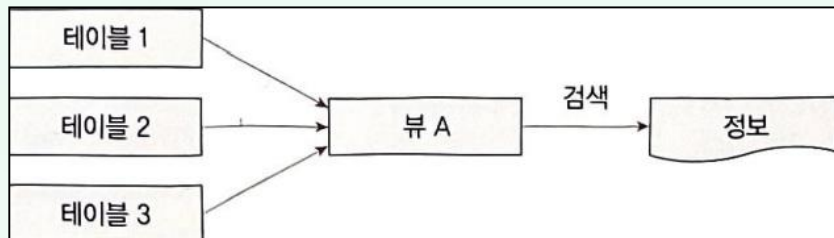
- ① 클러스터드 인덱스(Clustered Index)는 인덱스 키의 순서에 따라

2. 물리 데이터베이스 설계-SEC_05(뷰(View) 설계)

1) 뷰(View)의 개념

; 뷰는 사용자에게 접근이 허용된 자료만을 제한적으로 보여주기 위해 하나 이상의 기본 테이블로부터 유도된, 이름을 가지는 가상 테이블이다.

- 뷰는 저장장치 내에 물리적으로 존재하지 않지만, 사용자에게는 있는 것처럼 간주된다.
- 뷰는 데이터 보정 작업, 처리 과정 시험 등 임시적인 작업을 위한 용도로 활용된다.
- 뷰는 조인문의 사용 최소화로 사용상의 편의성을 최대화한다.
- 뷰를 생성하면 뷰 정의가 시스템 내에 저장되었다가 생성된 뷰 이름을 질의어(예를 들면 SQL)에서 사용할 경우 질의어가 실행될 때 뷰에 정의된 기본 테이블로 대체되어 기본 테이블에 대해 실행된다.
- 다음 그림은 뷰 A가 테이블1, 테이블2, 테이블3에서 유도되어 생성되며, 뷰 A를 통해 테이블1, 테이블2, 테이블3에 대한 데이터에 접근할 수 있음을 나타낸 것이다.



2. 물리 데이터베이스 설계-SEC_05(뷰(View) 설계)

2) 뷰(View)의 특징

- 뷰는 기본 테이블로부터 유도된 테이블이기 때문에 기본 테이블과 같은 형태의 구조를 사용하며, 조작도 기본 테이블과 거의 같다.
- 뷰는 가상 테이블이기 때문에 물리적으로 구현되어 있지 않다.
- 데이터의 논리적 독립성을 제공할 수 있다.
- 필요한 데이터만 뷰로 정의해서 처리할 수 있기 때문에 관리가 용이하고 명령문이 간단해진다.
- 뷰를 통해서만 데이터에 접근하게 하면 뷰에 나타나지 않는 데이터를 안전하게 보호하는 효율적인 기법으로 사용할 수 있다.
- 기본 테이블의 기본키를 포함한 속성(열) 집합으로 뷰를 구성해야만 삽입, 삭제, 갱신 연산이 가능하다.
- 일단 정의된 뷰는 다른 뷰의 정의에 기초가 될 수 있다.
- 뷰가 정의된 기본 테이블이나 뷰를 삭제하면 그 테이블이나 뷰를 기초로 정의된 다른 뷰도 자동으로 삭제된다.
- 뷰를 정의할 때는 CREATE문, 제거할 때는 DROP문을 사용한다.

정의 테이블 : 뷰는 하나 이상의 다른 테이블로부터 유도된 하나의 가상 테이블이며, 뷰를 만들기 위해 유도된 기본 테이블을 정의 테이블이라 한다.

2. 물리 데이터베이스 설계-SEC_05(뷰(View) 설계)

3) 뷰(View)의 장, 단점

장점	단점
<ul style="list-style-type: none">•논리적 데이터 독립성을 제공한다.•동일 데이터에 대해 동시에 여러 사용자의 상이한 응용이나 요구를 지원해 준다.•사용자의 데이터 관리를 간단하게 해준다.•접근 제어를 통한 자동 보안이 제공된다.	<ul style="list-style-type: none">•독립적인 인덱스를 가질 수 없다.•뷰의 정의를 변경할 수 없다.•뷰로 구성된 내용에 대한 삽입, 삭제, 갱신 연산에 제약이 따른다.

2. 물리 데이터베이스 설계-SEC_05(뷰(View) 설계)

4) 뷰 설계 순서

① 대상 테이블을 선정한다.

- 외부 시스템과 인터페이스에 관여하는 테이블
- CRUD 매트릭스를 통해 여러 테이블이 동시에 자주 조인되어 접근되는 테이블
- SQL문 작성 시 거의 모든 문장에서 인라인 뷰 방식으로 접근되는 테이블

② 대상 컬럼을 선정한다.

- 보안을 유지해야 하는 컬럼은 주의하여 선별한다.

③ 정의서를 작성한다.

- 뷰 정의서 예시

뷰명	뷰 설명	관련 테이블	관련 컬럼명	데이터 타입
V_SELLITEM	판매와 판매목록 처리	SELL	SELLNO SELLNAME SELLDATE	VARCHAR(5) VARCHAR(40) DATE
		SELLITEM	ITEMNO PRICE	VARCHAR(5) NUMBER(10)

서브쿼리가 FROM 절에서 안에서 사용되는 경우, 해당 서브쿼리를 '인라인 뷰'라고 한다. FROM 절에서 사용된 서브쿼리의 결과가 하나의 테이블에 대한 뷰(View)처럼 사용된다.
서브쿼리란 SELECT 문 안에 다시 SELECT 문이 기술된 형태의 쿼리(QUERY)이다.

2. 물리 데이터베이스 설계-SEC_05(뷰(View) 설계)

5) 뷰 설계 시 고려사항

- 테이블 구조가 단순화 될 수 있도록 반복적으로 조인을 설정하여 사용하거나 동일한 조건절을 사용하는 테이블을 뷰로 생성한다.

예) <주문> 테이블과 <거래처> 테이블을 조인하여 사용하는 경우가 많다면 <주문> 테이블과 <거래처> 테이블에서 필요한 필드로 구성된 뷰를 생성한다.

- 동일한 테이블이라도 업무에 따라 테이블을 이용하는 부분이 달라질 수 있으므로 사용할 데이터를 다양한 관점에서 제시해야 한다.

예) <회원> 테이블은 '회원번호', '이름', '주소', '전화번호', '총구매횟수', '총구매금액' 필드로 구성되어 있는데, 제품 발송 업무를 처리할 때는 '이름', 주소', '전화번호' 필드만 필요하므로 이 3개의 필드로 구성된 뷰를 설계한다.

- 데이터의 보안 유지를 고려하여 설계한다.

예) <회원> 테이블의 '총구매횟수', '총구매금액' 등은 회사 차원에서 중요한 자료일 수 있으므로 발송 담당자가 볼 수 없도록 뷰를 설계한다.

물리 데이터베이스 설계-SEC_05(뷰(View) 설계) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(뷰(View) 설계)

1. 뷰(View)에 대한 설명으로 옳지 않은 것은?

- ① 뷰는 CREATE문을 사용하여 정의한다.
- ② 뷰는 데이터의 논리적 독립성을 제공한다.
- ③ 뷰를 제거할 때에는 DROP문을 사용한다.
- ④ 뷰는 저장장치 내에 물리적으로 존재한다.

뷰의 개념

뷰는 사용자에게 접근이 허용된 자료만을 제한적으로 보여주기 위해 하나 이상의 테이블로부터 유도된, 이름을 가지는 가상 테이블이다.

- 뷰는 저장장치 내에 물리적으로 존재하지 않지만, 사용자에게는 있는 것처럼 간주된다.
- 뷰는 데이터 보정 작업, 처리 과정 시험 등 임시적인 작업을 위한 용도로 활용된다.
- 뷰는 조인문의 사용 최소화로 사용상의 편의성을 최대화한다.
- 뷰를 생성하면 뷰 정의가 시스템 내에 저장되었다가 생성된 뷰 이름을 질의어에서 사용할 경우 질의어가 실행될 때 뷰에 정의된 기본 테이블로 대체되어 기본 테이블에 대해 실행된다.

2. 뷰(VIEW)에 대한 설명으로 틀린 것은?

3. 뷰(View)의 장점이 아닌 것은?

- ① 뷰 자체로 인덱스를 가짐
- ② 데이터 보안 용이
- ③ 논리적 독립성 제공
- ④ 사용자 데이터 관리 용이

뷰의 장점

- 논리적 데이터 독립성을 제공한다.
- 동일 데이터에 대해 동시에 여러 사용자의 상이한 응용이나 요구를 지원해준다.
- 사용자의 데이터 관리를 간단하게 해준다.
- 접근 제어를 통한 자동 보안이 제공된다.

뷰의 단점

- 독립적인 인덱스를 가질 수 없다.
- 뷰의 정의를 변경할 수 없다.
- 뷰로 구성된 내용에 대한 삽입, 삭제, 갱신 연산에 제약이 따른다.

4. 데이터베이스에서의 뷰(View)에 대한 설명으로 틀린 것은?

- ① 뷰는 다른 뷰를 기반으로 새로운 뷰를 만들 수 있다.
- ② 뷰는 일종의 가상 테이블이며, UPDATE에는 제약이 따른다.

물리 데이터베이스 설계-SEC_05(뷰(View) 설계) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(뷰(View) 설계)

5. 뷰(View)의 이점으로 거리가 먼 것은?

- ① 독자적인 인덱스를 가질 수 있으므로 관리가 편하다.
 - ② 여러 사용자의 상이한 응용이나 요구를 지원해 준다.
 - ③ 사용자의 데이터 관리를 간단하게 해준다.
 - ④ 숨겨진 데이터를 위한 자동 보안이 제공된다.
- 뷰는 독립적인 인덱스를 가질 수 없다.

6. 뷰(View)에 대한 설명으로 틀린 것은?

- ① 뷰 위에 또 다른 뷰를 정의할 수 있다.
- ② DBA는 보안성 측면에서 뷰를 활용할 수 있다.
- ③ 사용자가 필요한 정보를 요구에 맞게 가공하여 뷰로 만들 수 있다.
- ④ SQL을 사용하면 뷰에 대한 삽입, 갱신, 삭제 연산 시 제약사항이 없다.

뷰는 기본 테이블의 기본키를 포함한 속성(열) 집합으로 뷰를 구성해야만 삽입, 삭제, 갱신 연산이 가능하므로, SQL을 사용하더라도 기본 테이블과 비교할 때 삽입, 삭제, 갱신 연산에 제약이 따른다.

7. 뷰(VIEW)의 특징으로 옳지 않은 것은?

- ① 뷰는 기본 테이블 검색 연산과 비교하여 제약이 따른다.
- ② DBA는 보안 측면에서 뷰를 활용할 수 있다.
- ③ 뷰 위에 또 다른 뷰를 정의할 수 있다.
- ④ 물리적으로 존재하지 않는 가상 테이블이다.

뷰는 삽입, 삭제, 갱신 연산에는 제한이 따르지만, 검색은 기본 테이블 검색 연산과 동일하다.

뷰의 특징

- 뷰는 기본 테이블로부터 유도된 테이블이기 때문에 기본 테이블과 같은 형태의 구조를 사용하며, 조작도 기본 테이블과 거의 같다.
- 뷰는 가상 테이블이기 때문에 물리적으로 구현되어 있지 않다.
- 데이터의 논리적 독립성을 제공할 수 있다.
- 필요한 데이터만 뷰로 정의해서 처리할 수 있기 때문에 관리가 용이하고 명령문이 간단해진다.
- 뷰를 통해서 데이터에 접근하게 하면 뷰에 나타나지 않는 데이터를 안전하게 보호하는 효율적인 기법으로 사용할 수 있다.
- 기본 테이블의 기본키를 포함한 속성 집합으로 뷰를 구성해야만 삽입, 삭제, 갱신 연산이 가능하다.

2. 물리 데이터베이스 설계-SEC_06(클러스터 설계)

1) 클러스터(Cluster)의 개요

; 클러스터는 데이터 저장 시 데이터 액세스 효율을 향상시키기 위해 동일한 성격의 데이터를 동일한 데이터 블록에 저장하는 물리적 저장 방법이다.

- 클러스터링 키로 지정된 컬럼 값의 순서대로 저장되고, 여러 개의 테이블이 하나의 클러스터에 저장된다.

예) 클러스터링 되지 않은 테이블

부서번호	부서
b001	기획부
b002	총무부
b003	영업부

<부서> 테이블

사원번호	이름	부서번호
1	홍길동	b001
2	장발장	b001
3	이예슬	b002
4	김강단	b003
5	정숙한	b002

<사원> 테이블

예) 클러스터링된 테이블

부서번호		부서	
b001		기획부	
	사원번호		이름
	1		홍길동
	2		장발장
b002		총무부	
	사원번호		이름
	3		이예슬
	5		정숙한
b003		영업부	
	사원번호		이름
	4		김강단

클러스터링은 비슷한 종류끼리 묶어준다는 의미로 테이블에서는 동일한 성격의 데이터를 동일한 데이터 블록에 저장하는 방법이고, 서버에서는 두 대 이상의 서버를 하나의 서버처럼 운영하기 위한 방법을 의미한다.

클러스터링 키

- 클러스터링된 테이블에서 각각의 행을 접근할 때 기준이 되는 열로 데이터를 조회하면 클러스터링 키로 지정된 필드에서 시작하여 클러스터링된 테이블의 데이터를 조회한다.

<부서> 테이블과 <사원> 테이블이 '부서번호' 필드를 기준으로 클러스터링 되었다. 이런 경우 '부서번호'를 클러스터링 키라고 한다.

2. 물리 데이터베이스 설계-SEC_06(클러스터 설계)

2) 클러스터(Cluster)의 특징

- 클러스터링 된 테이블은 데이터 조회 속도는 향상시키지만 데이터 입력, 수정, 삭제에 대한 성능은 저하시킨다.
- 클러스터는 데이터의 분포도가 넓을수록 유리하다.
- 데이터 분포도가 넓은 테이블을 클러스터링 하면 저장 공간을 절약할 수 있다.
- 클러스터링된 테이블은 클러스터링 키 열을 공유하므로 저장 공간이 줄어든다.
- 대용량을 처리하는 트랜잭션은 전체 테이블을 스캔 하는 일이 자주 발생하므로 클러스터링을 하지 않는 것이 좋다.
- 처리 범위가 넓은 경우에는 단일 테이블 클러스터링을 조인이 많이 발생하는 경우에는 다중 테이블 클러스터링을 사용한다.
- 파티셔닝 된 테이블에는 클러스터링을 할 수 없다.
- 클러스터링을 하면 비슷한 데이터가 동일한 데이터 블록에 저장되기 때문에 디스크 I/O가 줄어든다.
- 클러스터링된 테이블에 클러스터드 인덱스를 생성하면 접근 성능이 향상된다.

분포도는 인덱스에서는 분포도가 좁은 테이블이 좋지만 클러스터링은 분포도가 넓은 테이블에 유리하다.

단일 테이블 클러스터링이란 여러 개의 테이블 뿐만 아니라 한 개의 테이블에 대해서도 클러스터링을 수행할 수 있다.

특정 컬럼의 동일한 값을 동일 블록이나 연속된 블록에 저장하므로 데이터 조회 성능이 향상된다.

파티셔닝 : 데이터베이스에서의 파티셔닝은 대용량의 테이블이나 인덱스를 작은 논리적 단위인 파티션으로 나누는 것이다.

2. 물리 데이터베이스 설계-SEC_06(클러스터 설계)

3) 클러스터 대상 테이블

- 분포도가 넓은 테이블
- 대량의 범위를 자주 조회하는 테이블
- 입력, 수정, 삭제가 자주 발생하지 않는 테이블
- 자주 조인되어 사용되는 테이블
- ORDER BY, GROUP BY, UNION이 빈번한 테이블

ORDER BY 정렬, GROUP BY는 그룹 지정, UNION은 테이블 통합 지정 시 사용하는 SQL 명령어이다.

물리 데이터베이스 설계-SEC_06(클러스터 설계) 출제 예상 문제

출제 예상 문제(클러스터 설계)

1. 다음 중 데이터베이스 테이블에서의 클러스터에 대한 설명으로 가장 옳지 않은 것은?

- ① 클러스터는 동일한 성격의 데이터를 물리적으로 동일한 데이터 블록에 저장하는 것이다.
- ② 클러스터는 데이터 조회 속도는 물론이고 데이터 입력, 수정, 삭제에 대한 성능도 향상시킨다.
- ③ 분포도가 넓은 테이블을 클러스터링 하면 저장 공간을 절약할 수 있다.
- ④ 파티셔닝 된 테이블은 클러스터링을 할 수 없다.

클러스터(Cluster)

클러스터는 데이터 저장 시 데이터 액세스 효율을 향상시키기 위해 동일한 성격의 데이터를 동일한 데이터 블록에 저장하는 물리적 저장 방법이다.

- 클러스터링 키로 지정된 컬럼 값의 순서대로 저장되고, 여러 개의 테이블이 하나의 클러스터에 저장된다.
- 클러스터링 된 테이블은 데이터 조회 속도는 향상시키지만 데이터의 입력, 수정, 삭제에 대한 성능은 저하시킨다.

3. 다음 중 클러스터에 대한 설명으로 틀린 것은?

- ① 클러스터링을 하면 비슷한 데이터를 동일한 데이터 블록에 저장하기 때문에 디스크 I/O가 줄어든다.
 - ② 단일 테이블 클러스터링은 조인이 많이 발생하는 경우, 다중 테이블 클러스터링은 처리 범위가 넓은 경우 주로 사용된다.
 - ③ 클러스터링 된 테이블은 클러스터 키 열을 공유하므로 저장공간이 줄어든다.
 - ④ 클러스터는 데이터의 액세스 효율을 향상시킨다.
- 단일 테이블 클러스터링은 처리 범위가 넓은 경우, 다중 테이블 클러스터링은 조인이 많이 발생하는 경우 주로 사용된다.

2. 물리 데이터베이스 설계-SEC_07(파티션 설계)

1) 파티션(Partition)의 개요

; 데이터베이스에서 파티션은 대용량의 테이블이나 인덱스를 작은 논리적 단위인 파티션으로 나누는 것을 말한다.

- 대용량 DB의 경우 중요한 몇 개의 테이블에만 집중되어 데이터가 증가되므로, 이런 테이블들을 작은 단위로 나눠 분산시키면 성능 저하를 방지할 뿐만 아니라 데이터 관리도 쉬워진다.
- 테이블이나 인덱스를 파티셔닝 하면 파티션 키 또는 인덱스 키에 따라 물리적으로 별도의 공간에 데이터가 저장된다.
- 데이터 처리는 테이블 단위로 이뤄지고, 데이터 저장은 파티션 별로 수행된다.

파티션은 용량이 큰 테이블을 관리하기 쉽도록 작은 단위로 나눈 것을 말한다.

파티션 키 : 파티션을 나누는 기준이 되는 열로 파티션 키로 지정된 열의 데이터 값에 따라 파티션이 나뉘어진다.

테이블 단위로 처리 : 하나의 테이블이 여러 개의 파티션으로 나뉘어져 있어도 DB에 접근하는 애플리케이션은 테이블 단위로 데이터를 처리하기 때문에 파티션을 인식하지 못한다.

2. 물리 데이터베이스 설계-SEC_07(파티션 설계)

2) 파티션의 장·단점

장점	<ul style="list-style-type: none">•데이터 접근 시 액세스 범위를 줄여 쿼리 성능이 향상된다.•파티션 별로 데이터가 분산되어 저장되므로 디스크의 성능이 향상된다.•파티션 별로 백업 및 복구를 수행하므로 속도가 빠르다.•시스템 장애 시 데이터 손상 정도를 최소화할 수 있다.•데이터 가용성이 향상된다.•파티션 단위로 입·출력을 분산시킬 수 있다.
단점	<ul style="list-style-type: none">•하나의 테이블을 세분화 하여 관리하므로 세심한 관리가 요구된다.•테이블간 조인에 대한 비용이 증가한다.•용량이 작은 테이블에 파티셔닝을 수행하면 오히려 성능이 저하된다.

2. 물리 데이터베이스 설계-SEC_07(파티션 설계)

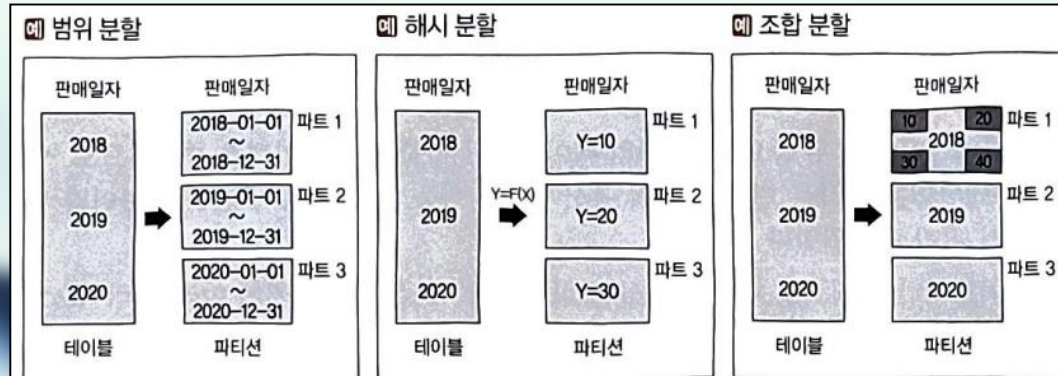
3) 파티션의 종류

; 파티션의 종류는 파티셔닝 방식에 따라 범위 분할, 해시 분할, 조합 분할, 목록 분할, 라운드 로빈 분할 등으로 나뉜다.

범위 분할 (Range Partitioning)	지정한 열의 값을 기준으로 범위를 지정하여 분할한다. 예) 일별, 월별, 분기별 등
해시 분할 (Hash Partitioning)	<ul style="list-style-type: none">•해시 함수를 적용한 결과 값에 따라 데이터를 분할한다.•특정 파티션에 데이터가 집중되는 범위 분할의 단점을 보완한 것으로 데이터를 고르게 분산할 때 유용하다.•특정 데이터가 어디에 있는지 판단할 수 없다.•고객번호, 주민번호 등과 같이 데이터가 고른 컬럼에 효과적이다.
조합 분할 (Composite Partitioning)	<ul style="list-style-type: none">•범위 분할로 분할한 다음 해시 함수를 적용하여 다시 분할하는 방식이다.•범위 분할한 파티션이 너무 커서 관리가 어려울 때 유용하다.
목록 분할 (List Partitioning)	지정한 열 값에 대한 목록을 만들어 이를 기준으로 분할한다. 예) '국가'라는 열에 '한국', '미국', '일본'이 있는 경우 '미국'을 제외할 목적으로 '아시아'라는 목록을 만들어 분할함
라운드 로빈 분할 (Round Robin Partitioning)	<ul style="list-style-type: none">•레코드를 균일하게 분배하는 방식이다.•각 레코드가 순차적으로 분배되며, 기본키가 필요 없다.

2. 물리 데이터베이스 설계-SEC_07(파티션 설계)

3) 파티션의 종류



- 범위 분할 : '판매일자' 필드를 기준으로 연도별로 파티션을 나눈다.
- 해시 분할 : 해시 함수($F(X)$)에 '판매일자' 필드를 인수로 입력하여 그 결과값에 따라 파티션을 나눈다.
- 조합 분할 : '판매일자' 필드를 기준으로 연도별로 파티션을 나눈 다음 하나의 파티션 안에서 다시 해시 함수를 적용하여 다시 파티션을 나눈다.

2. 물리 데이터베이스 설계-SEC_07(파티션 설계)

4) 파티션 키 선정 시 고려 사항

- 파티션 키는 테이블 접근 유형에 따라 파티셔닝이 이뤄지도록 선정한다.
- 데이터 관리의 용이성을 위해 이력성 데이터는 파티션 생성주기와 소멸주기를 일치시켜야 한다.
- 매일 생성되는 날짜 컬럼, 백업의 기준이 되는 날짜 컬럼, 파티션 간 이동이 없는 컬럼, I/O 병목을 줄일 수 있는 데이터 분포가 양호한 컬럼 등을 파티션 키로 선정한다.

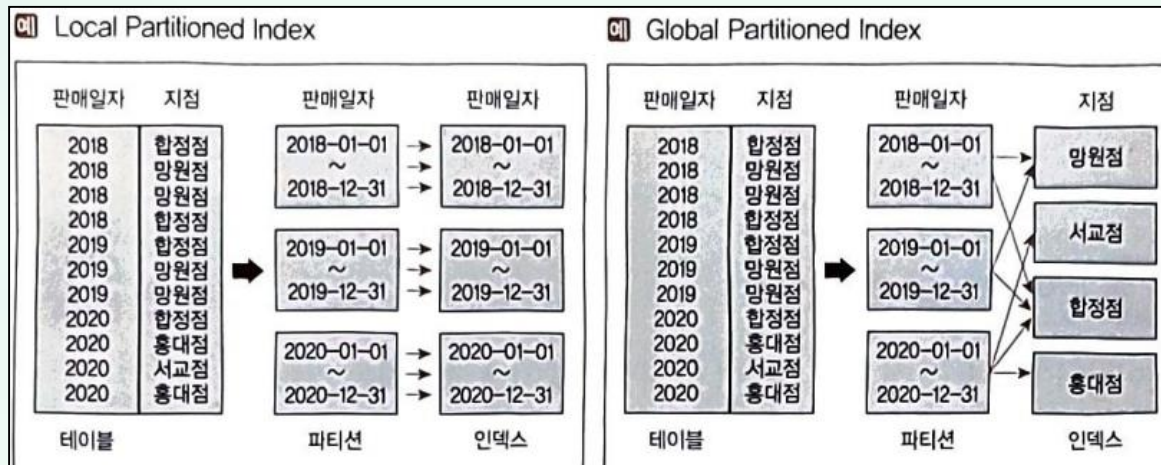
2. 물리 데이터베이스 설계-SEC_07(파티션 설계)

5) 인덱스 파티션

; 인덱스 파티션은 파티션된 테이블의 데이터를 관리하기 위해 인덱스를 나눈 것이다.

- 인덱스 파티션은 파티션된 테이블의 종속 여부에 따라 Local Partitioned Index와 Global Partitioned Index로 나뉜다.

- Local Partitioned Index : 테이블 파티션과 인덱스 파티션이 1:1 대응되도록 파티셔닝 한다.
- Global Partitioned Index : 테이블 파티션과 인덱스 파티션이 독립적으로 구성되도록 파티셔닝 한다.
- Local Partitioned Index가 Global Partitioned Index에 비해 데이터 관리가 쉽다.



- Local Partitioned Index : 파티션과 인덱스를 모두 '판매일자' 필드를 기준으로 수행한다.
- Global Partitioned Index : 파티션은 '판매일자' 필드를 기준으로 수행하고, 인덱스는 '지점' 필드를 기준으로 수행한다.

2. 물리 데이터베이스 설계-SEC_07(파티션 설계)

5) 인덱스 파티션

- 인덱스 파티션은 인덱스 파티션 키 컬럼의 위치에 따라 Prefixed Partitioned Index와 Non-prefixed Partitioned Index로 나뉜다.
 - Prefixed Partitioned Index: 인덱스 파티션 키와 인덱스 첫 번째 컬럼이 같다.
 - Non-Prefixed Partitioned Index: 인덱스 파티션 키와 인덱스 첫 번째 컬럼이 다르다.

파티션 인덱스 유형	인덱스 파티션키	인덱스키 컬럼
Prefixed Partitioned Index	판매일자	판매일자(+품목코드)
Non-Prefixed Partitioned Index	판매일자	품목코드(+거래일자)

- Local과 Global, Prefixed와 Non-prefixed를 조합하여 Local Prefixed Partitioned Index, Local Non-Prefixed Partitioned Index, Global Prefixed Partitioned Index 등으로 구성하여 사용한다. 단, Global Non-Prefixed Partitioned Index는 허용되지 않는다.

물리 데이터베이스 설계-SEC_07(파티션 설계) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(파티션 설계)

1. 물리 데이터 저장소의 파티션 설계에서 파티션 유형으로 옳지 않은 것은?

- ① 범위 분할(Range Partitioning)
- ② 해시 분할(Hash Partitioning)
- ③ 조합 분할(Composite Partitioning)
- ④ 유닛 분할(Unit Partitioning)

1. 범위 분할(Range Partitioning)

지정한 열의 값을 기준으로 범위를 지정하여 분할한다.

예) 일별, 월별, 분기별 등

2. 해시 분할(Hash Partitioning)

- 해시 함수를 적용한 결과 값에 따라 데이터를 분할한다.
- 특정 파티션에 데이터가 집중되는 범위 분할의 단점을 보완한 것으로 데이터를 고르게 분산할 때 유용하다.
- 고객 번호, 주민 번호 등과 같이 데이터가 고른 컬럼에 효과적이다.

3. 조합 분할(Composite Partitioning)

- 범위 분할로 분할한 다음 해시 함수를 적용하여 다시 분할하는 방식이다.

3. 파티셔닝 방식 중 '월별, 분기별'과 같이 지정한 열의 값을 기준으로 범위를 지정하여 분할하는 방식은?

- ① Range Partitioning
- ② Hash Partitioning
- ③ Composite Partitioning
- ④ List Partitioning

범위(Range)를 지정하는 분할은 범위 분할이다.

4. 다음 중 파티셔닝의 장점으로 틀린 것은?

- ① 데이터 관리가 수월해진다.
- ② 시스템 장애 시 데이터 손상 정도를 최소화할 수 있다.
- ③ 파티션 별로 백업 및 복구를 수행하므로 속도가 빠르다.
- ④ 테이블간 조인에 대한 비용이 감소한다.

파티셔닝의 장점

- 데이터 접근 시 액세스 범위를 줄여 쿼리 성능이 향상된다.
- 파티션 별로 데이터가 분산되어 저장되므로 디스크의 성능이 향상된다.
- 파티션 별로 백업 및 복구를 수행하므로 속도가 빠르다.
- 시스템 장애 시 데이터 손상 정도를 최소화 할 수 있다.
- 데이터 가용성이 향상된다.

물리 데이터베이스 설계-SEC_07(파티션 설계) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(파티션 설계)

5. 다음 중 테이블 파티셔닝의 종류에 대한 설명으로 틀린 것은?

- ① 범위 분할은 일별, 월별, 분기별과 같이 열의 값을 기준으로 분할한다.
- ② 해시 분할은 해시 함수를 적용한 결과 값에 따라 데이터를 분할한다.
- ③ 해시 분할은 특정 파티션에 데이터가 집중되는 단점이 있다.
- ④ 조합 분할은 범위 분할한 파티션이 너무 커서 관리가 어려울 때 용이하다.

2. 물리 데이터베이스 설계-SEC_08(데이터베이스 용량 설계)

1) 데이터베이스 용량 설계

; 데이터베이스 용량 설계는 데이터가 저장될 공간을 정의하는 것이다.

- 데이터베이스 용량을 설계할 때는 테이블에 저장할 데이터 양과 인덱스, 클러스터 등이 차지하는 공간 등을 예측하여 반영해야 한다.

데이터베이스의 용량 설계는 데이터베이스 설계만큼이나 중요하다. 용량이 필요할 때마다 용량을 증설한다면 데이터베이스의 전체적인 성능이 저하되고 비용이 계속 증가할 수 있다.

2. 물리 데이터베이스 설계-SEC_08(데이터베이스 용량 설계)

2) 데이터베이스 용량 설계의 목적

- 데이터베이스의 용량을 정확히 산정하여 디스크의 저장 공간을 효과적으로 사용하고 확장성 및 가용성을 높인다.
- 디스크의 특성을 고려하여 설계함으로써 디스크의 입·출력 부하를 분산시키고 채널의 병목 현상을 최소화한다.
- 디스크에 대한 입·출력 경합이 최소화되도록 설계함으로써 데이터 접근성이 향상된다.
- 데이터 접근성을 향상시키는 설계 방법
 - 테이블의 테이블스페이스와 인덱스의 테이블스페이스를 분리하여 구성한다.
 - 테이블스페이스와 임시 테이블스페이스를 분리하여 구성한다.
 - 테이블을 마스터 테이블과 트랜잭션 테이블로 분류한다.
- 데이터베이스에 생성되는 오브젝트의 익스텐트 발생을 최소화하여 성능을 향상시킨다.
- 데이터베이스 용량을 정확히 분석하여 테이블과 인덱스에 적합한 저장 옵션을 지정한다.

익스텐트(범위, Extent) : 익스텐트는 기본적인 용량이 모두 찼을 경우 추가적으로 할당되는 공간을 의미한다.

2. 물리 데이터베이스 설계-SEC_08(데이터베이스 용량 설계)

3) 데이터베이스 용량 분석 절차

- ① 데이터 예상 건수, 로우(Row) 길이, 보존 기간, 증가율 등 기초 자료를 수집하여 용량을 분석한다.

예) 기초 자료 수집

NO	엔티티	테이블	Row 길이	보존 기간	초기건수	주기별 발생건수	발생 주기	년증가율(건)
001	주문	ODDER	1,000	2	19,250	80	일	500
002	상품	ITEM	430	2	34,502	200	일	300
003	판매자	SELLER	310	2	12,520	15	일	150

- ② 분석된 자료를 바탕으로 DBMS에 이용될 테이블, 인덱스 등 오브젝트 별 용량을 산정한다.
- ③ 테이블과 인덱스의 테이블스페이스 용량을 산정한다.
- 테이블스페이스 용량은 테이블스페이스에 생성되는 테이블 용량을 모두 더한 값에 약 40% 정도를 추가하여 산정한다.
- 예) 테이블 용량이 100M라면 테이블스페이스의 용량은 $100 + (100 \times 40\%) = 140\text{M}$ 정도로 산정한다.
- ④ 데이터베이스에 저장될 모든 데이터 용량과 데이터베이스 설치 및 관리를 위한 시스템 용량을 합해 디스크 용량을 산정한다.

약 40%라는 값은 절대적인 값이 아니다. 업무에 따라 확장 가능성이 높은 것은 40% 정도 확장, 가능성이 낮은 것은 20% 정도로 지정하면 된다.

2. 물리 데이터베이스 설계-SEC_08(데이터베이스 용량 설계)

3) 데이터베이스 용량 분석 절차

예) 디스크 용량 계산

NO	이름	용량(G)	사용용량(M)	사용비율	파일명	파일 용량
1	디스크1	3	2,700	90%	AB001,ABC01	1,700M
					AB002,ABC01	1,000M
2	디스크2	5	2,500	50%	AB001,ABC02	1,500M
					AB002,ABC02	1,000M

- 디스크1 : 디스크에 저장될 전체 파일의 용량이 2,700M이고 추후에 추가될 데이터가 많지 않아 300M의 여유 공간이 있는 3G 디스크를 선택하였다.
- 디스크2 : 디스크에 저장될 전체 파일의 용량이 2,500M 이나 추후에 추가될 데이터 양이 많아 2,500M의 여유 공간이 있는 5G 디스크를 선택하였다.

물리 데이터베이스 설계-SEC_08(데이터베이스 용량 설계) 출제 예상 문제

출제 예상 문제(데이터베이스 용량 설계)

1. 다음은 데이터베이스 용량 설계에 대한 내용이다. 가장 옳지 않은 것은?

- ① 데이터베이스 용량을 분석하려면 먼저 데이터 예상 건수, 로우 길이, 보존 기간, 증가율 등 기초 자료를 수집하여 분석해야 한다.
- ② 용량 설계를 정확히 산정하면 데이터의 확장성 및 가용성을 높일 수 있다.
- ③ 입·출력 경합을 최소화하여 데이터 접근성을 높이려면 테이블과 인덱스의 테이블스페이스를 같은 곳에 위치하도록 구성한다.
- ④ 데이터의 특성을 고려한 용량 설계를 통해 디스크 채널의 병목 현상을 최소화할 수 있다.

데이터베이스 용량 설계의 목적

- 데이터베이스의 용량을 정확히 산정하여 디스크의 저장 공간을 효과적으로 사용하고 확장성 및 가용성을 높인다.
- 디스크의 특성을 고려하여 설계함으로써 디스크의 입·출력 부하를 분산시키고 채널의 병목 현상을 최소화한다.
- 디스크에 대한 입·출력 경합이 최소화되도록 설계함으로써 데이터 접근성이 향상된다.

3. 데이터베이스 용량 분석에서 테이블스페이스 용량은 테이블스페이스에 생성되는 테이블 용량을 모두 더한 값에 약 몇 프로를 추가 산정하는가?

- ① 10%
- ② 20%
- ③ 30%
- ④ 40%

약 40%라는 값은 절대적인 값이 아니다. 업무에 따라 확장 가능성이 높은 것은 40% 정도 확장, 가능성이 낮은 것은 20% 정도로 지정하면 된다.



감사합니다.