

데이터베이스론

국가 전산 7급

2016년 8월 27일

♣ 합격선/최종합격인원(73.57점/27명) - 채용예정인원 26명 ♣

1. 다음은 데이터베이스 안에 있는 어떤 테이블을 보인 것이다. 파일처리 방식과 대비해서 <보기>와 같은 테이블을 이용하는 데이터베이스 방식의 주요 특성으로 옳은 것은? [2016년 국가 7급]

-----<보기>-----

TABLES

Table Name	No of Columns
STUDENT	4
....

COLUMNS

Column Name	Data Type	Table Name
StudentNumber	Character(4)	STUDENT
....

-
- ① 데이터 구조가 변경되면 프로그램도 수정해야 하는 성질
 - ② 데이터 중복이 증가해도 일관성을 유지하는 성질
 - ③ 데이터 정의가 응용프로그램의 일부로 표현되는 성질
 - ④ 데이터베이스 시스템의 스스로를 기술하는 성질

♣ 데이터 사전 = 시스템 카탈로그

-
- 보기 내용은 데이터 사전이다.(메타데이터)

// 데이터 사전 정의

데이터베이스에 저장된 모든 데이터 개체들에 대한 정의나 명세한 관한 정보를 유지하고, 관리하는 시스템이다.(데이터베이스에 저장된 정보 요약) - 자기정의(self definition)

- DBMS는 데이터 사전을 통하여 스스로 데이터베이스 시스템 구조와 내용을 정의한다.
- 자기 정의(self definition)는 데이터베이스 스스로 자신 구조와 내용을 정의하는 것이다.
- 데이터 사전은 DBMS에 의해 계속 갱신된다.
- 일반 사용자는 SQL을 이용하여 데이터 사전 검색(select)만 가능하다.(갱신은 불가)
- 데이터 사전에 저장되는 정보 : 릴레이션, 인덱스, 뷰, 사용자 정보, 통계 정보 등

정답 : ④

2. 하나의 데이터베이스 시스템 내에서 적절한 제어 없이 트랜잭션들을 동시에 실행하였을 경우 여러 문제가 발생할 수 있다. 이를 해결하기 위한 동시성 제어가 올바르게 동작하지 않을 경우 발생할 수 있는 문제점으로 옳지 않은 것은? [2016년 국가 7급]

- ① 갱신 손실 문제
- ② 부정확한 요약 문제
- ③ 반복할 수 없는 읽기 문제
- ④ 지역적 오류 문제

♣ 동시성 제어에서 발생할 수 있는 문제점

· 지역적 오류 문제는 분산 데이터베이스에서 발생할 수 있는 문제 유형이다.

// 동시성 제어 정의

- 다중 사용자 환경에서 데이터베이스에 여러 트랜잭션이 성공적으로 동시에 실행될 수 있도록 지원하는 기능이다.(데이터 무결성 및 일관성 보장)
- 트랜잭션들이 동시에 접근할 때 발행할 수 있는 문제점을 예방하고 해결하는 기능이다.

// 동시성 제어가 올바르게 동작하지 않을 경우 발생할 수 있는 문제점

① 갱신 손실 문제(lost update problem)

- 트랜잭션들이 하나의 데이터를 동시에 갱신할 때 발생하는 문제이다.
→ 어떤 트랜잭션이 갱신한 내용을 다른 트랜잭션이 덮어 쓴 경우(갱신 무효)
- 결과적으로, 특정 트랜잭션의 연산 결과가 분실된다.

② 불일치 분석 문제(inconsistent analysis problem)

- 데이터 모순성(data inconsistency), 부정확한 요약(incorrect summary), 반복할 수 없는 읽기(unrepeatable read) 문제라고도 한다.
- 두 트랜잭션이 동시에 실행할 때 데이터베이스가 일관성 없는 모순된 상태가 되는 문제
→ 데이터 모순성이 발생하여 요약을 해도 정확하지 않고, 반복적으로 읽으면 안 된다.

③ 비완료 의존성 문제(uncommitted dependency problem)

- 연쇄복귀(cascading rollback), 오손판독(dirty read), 임시 갱신 문제라고도 한다.
- 트랜잭션들이 동시에 같은 데이터에 접근하여 갱신할 때
→ 한 트랜잭션은 실행 성공하여 완료된 상태(복귀가 필요 없음)
→ 다른 트랜잭션은 실행 실패(갱신을 취소하고 원래 상태로 복귀해야 함)
- 다른 트랜잭션이 처리한 부분에 대해서는 취소가 불가능하다.(회복불능)

4. 두 릴레이션 R1(A, B, C), R2(B, C, D)를 오른쪽 외부조인(right outer join)을 한 결과에 나타나는 튜플의 수는? [2016년 국가 7급]

R1

A	B	C
1	2	3
4	2	3
7	8	9

R2

B	C	D
2	3	10
2	3	11
6	7	12

- ① 3개 ② 4개 ③ 5개 ④ 6개

☞ 외부조인(outer join)

// 외부조인 = 내부조인 + α

- 내부조인은 양쪽 테이블에서 조인 조건을 만족하는 행을 반환한다.
- 외부조인은 양쪽 테이블에서 조인 조건에 만족되지 않는 행까지도 포함시키는 것이다.
- 여기서, α 가 조인 조건을 만족하지 않는 행을 반환하는 것을 의미한다.

// 왼쪽 외부조인(left outer join) 결과 : 왼쪽 R1 우선 조인

A	B	C	B	C	D
1	2	3	2	3	10
1	2	3	2	3	11
4	2	3	2	3	10
4	2	3	2	3	11
7	8	9	null	null	null

→ 조인이 없는 부분은 null

- 왼쪽 테이블의 모든 데이터를 포함하는 결과가 생성된다.

// 오른쪽 외부조인(right outer join) 결과 : 오른쪽 R2 우선 조인

A	B	C	B	C	D
1	2	3	2	3	10
4	2	3	2	3	10
1	2	3	2	3	11
4	2	3	2	3	11
null	null	null	6	7	12

→ 조인이 없는 부분은 null

- 오른쪽 테이블의 모든 데이터를 포함하는 결과가 생성된다.

5. 다음 두 릴레이션 R, S에 대한 조인 결과로 옳지 않은 것은? (단, \bowtie_N 은 자연조인, \bowtie 은 왼쪽 외부조인, \bowtie_{\rightarrow} 은 오른쪽 외부조인을 의미한다) [2016년 국가 7급]

A	B	C
a1	b1	c1
a2	b2	c2
a3	b3	c3
a4	b4	c4

B	C	D
b1	c1	d1
b1	c2	d2
b3	c3	d3
b4	c4	d4

① $R \bowtie_{\neq} S$

A	B	C	D
a1	b1	c1	d1
a3	b3	c3	d3
a4	b4	c4	d4

② $R \bowtie_{b=b \wedge c=c} S$

A	B	C	B	C	D
a1	b1	c1	b1	c1	d1
a2	b1	c2	b1	c2	d2
a3	b3	c3	b3	c3	d3
a4	b4	c4	b4	c4	d4

③ $R \bowtie_{b=b \wedge c=c} S$

A	B	C	B	C	D
a1	b1	c1	b1	c1	d1
a2	b2	c2	null	null	null
a3	b3	c3	b3	c3	d3
a4	b4	c4	b4	c4	d4

④ $R \bowtie_{b=b \wedge c=c} S$

A	B	C	B	C	D
a1	b1	c1	b1	c1	d1
null	null	null	b1	c2	d2
a3	b3	c3	b3	c3	d3
a4	b4	c4	b4	c4	d4

☞ 릴레이션 R, S에 대한 조인 결과

• $R \bowtie_{b=b \wedge c=c} S$ (교집합)

A	B	C	B	C	D
a1	b1	c1	b1	c1	d1
a3	b3	c3	b3	c3	d3
a4	b4	c4	b4	c4	d4

→ 릴레이션 R의 B에는 b2가 있지만, S의 B에는 b2가 없으므로

// 다음 두 SQL문의 실행 결과는 같다.(위의 결과)

<pre>SELECT * FROM R Inner Join S ON (R.B = S.B) AND (R.C = S.C);</pre>	<pre>SELECT * FROM R, S Where (R.B = S.B) AND (R.C = S.C);</pre>
---	--

6. 도서(도서번호, 도서제목, 출판사명, 발행년도) 테이블에서, 2000년 이후에 10권 이상의 책을 발행한 출판사의 이름을 중복 없이 출력하는 SQL문으로 옳은 것은? (단, 출판사명이 동일한 출판사는 존재하지 않는 것으로 가정한다. 도서번호는 도서 테이블의 기본키이다) [2016년 국가 7급]

- ① Select 출판사명 From 도서 Where 발행년도>=2000 Order By Count(도서번호)>=10;
- ② Select 출판사명 From 도서 Where 발행년도>=2000 And Count(도서번호)>=10
Group By 출판사명;
- ③ Select 출판사명 From 도서 Where Count(도서번호)>=10
Group By 출판사명 Having 발행년도 >= 2000;
- ④ Select 출판사명 From 도서 Where 발행년도 >= 2000
Group By 출판사명 Having Count(도서번호) >= 10;

☞ SQL : Select ~ group by ~ having

Select [All | Distinct] 열_리스트

From 테이블_리스트

[Where 조건]

[Group By 열_리스트 [Having 조건]] //그룹화시킨 것에서 조건에 따라 추출

[Order By 열_리스트 [Asc | Desc]]; //오름 또는 내림차순으로 정렬

↓ 예제

Select 출판사명, Max(발행년도) //그룹화하지 않은 필드는 집계함수를 사용

From 도서

Where 발행년도 >= 2000 //조건, 집계함수를 사용한 조건 설정은 불가

Group By 출판사명 //출판사명으로 그룹화

Having Count(도서번호) >= 10 //조건, 반드시 집계함수를 사용하여 조건을 설정

Order By 출판사명 Asc; //출판사명으로 오름차순정렬

- group by절은 데이터들을 원하는 그룹으로 나눌 수 있다.
- 그룹하기 전에 where절을 사용하여 그룹 대상 집합을 먼저 선택 가능
- select 절에는 group by로 묶은 필드와 집계함수를 사용한 필드만을 선택적으로 사용될 수 있다.
- 즉, group by에 없는 필드도 집계함수를 사용하면 select절에 사용 가능하다.
- having절은 반드시 group by 뒤에서만 사용 가능하다.
- having절은 집계함수로만 조건을 지정할 수 있다.
- where절에서는 집계함수를 사용할 수 없다.

┌ Group By로 묶은 필드 지정

④ Select 출판사명 From 도서 Where 발행년도 >= 2000

Group By 출판사명 Having Count(도서번호) >= 10;

출판사명으로 그룹화

└, 반드시 집계함수로 조건 지정

7. 지연갱신을 통한 점진적 로깅(incremental logging with deferred updates) 기법을 사용하는 복구 시스템으로 옳은 것은? [2016년 국가 7급]

- ① 갱신된 데이터 아이템의 갱신 전 원래 값만을 로그에 저장한다.
- ② 갱신된 데이터 아이템의 갱신 후 수정된 값만을 로그에 저장한다.
- ③ 갱신된 데이터 아이템의 갱신 전, 후 값을 모두 로그에 저장해야 한다.
- ④ Begin Transaction과 Commit Transaction 레코드만을 로그에 저장하면 된다.

☞ 지연갱신을 통한 점진적 로깅

- 갱신된 데이터 아이템의 갱신 후 수정된 값만을 로그에 저장한다.(○)
 - 지연갱신은 수정된 값이 데이터베이스에 기록되지 않고, 로그에만 기록된다.
 - 해서, 이전 값을 가질 필요가 없다.
 - 복구 시에는 redo만 사용한다.(이 기법은 undo 연산이 필요 없다)

// 지연 갱신

- DBMS는 트랜잭션의 성공적인 실행(commit)이면 로그 파일의 내용을 참조하여 데이터베이스의 내용을 변경하고, 비성공적인 실행(rollback)이면 로그 파일을 무시한다.
- 트랜잭션 실행 중에 트랜잭션이 철회되거나 시스템이 붕괴되면 로그 파일의 정보만 버리면 된다. → 이 기법은 undo 연산이 필요 없다.
- 지연 갱신은 값이 데이터베이스에 기록되지 않기 때문에 이전 값을 가질 필요가 없으며 복구 시에는 redo만 사용한다.
- DB의 변경을 로그에 전부 기록해 두었다가 한꺼번에 실행시킴으로써 트랜잭션의 원자성을 보장하는 것이다.

// redo / undo

redo/undo는 데이터베이스에 장애가 발생했을 때, 회복을 위해 취할 수 있는 조치이다.

redo 재수행	<ul style="list-style-type: none"> • 완료된 트랜잭션이 변경한 데이터베이스 페이지는 유지(durability)되어야 한다. • 완료된 트랜잭션의 수정을 데이터베이스에 반영하는 복구 작업을 redo라 한다. • 로그레코드를 순차적으로 실행 (전진 회복)
undo 취소	<ul style="list-style-type: none"> • 미완료된 트랜잭션이 변경한 데이터베이스 페이지는 원상 복구되어야 한다. • 이러한 복구를 undo라 한다. • 시스템 장애가 발생하면 미완료된 트랜잭션이 변경한 내용은 취소(undo)한다. • 로그레코드를 역순으로 실행 (후진 회복)

- 회복 작업이 완료될 때까지 시스템은 새로운 트랜잭션을 받아들일 수 없다.

8. 데이터베이스에서 동시성 제어가 적절하게 이루어지지 않으면서 다음과 같이 두 트랜잭션 T1, T2가 동시에 실행될 때 문제점이 발생할 수 있다. 이 문제점이 발생하지 않을 수 있는 트랜잭션의 격리수준(isolation level)을 모두 고른 것은? [2016년 국가 7급]

T1	T2
update product set amount = amount + new where pName = "chair";	
	select productID, amount from product;
rollback;	
	commit;

- ① 미완료 읽기, 완료 읽기, 반복가능 읽기, 직렬 가능
- ② 완료 읽기, 반복가능 읽기, 직렬 가능
- ③ 반복가능 읽기, 직렬 가능
- ④ 직렬 가능

☞ 트랜잭션 격리수준(isolation level)과 문제점 발생 가능 여부

- 먼저, 주어진 문제에서 오손판독(dirty read) 문제점이 발생한다.
 - 완료되지 않은 트랜잭션이 수정한 데이터를 다른 트랜잭션이 읽을 때 발생 가능
 - 완료되지 않은 T1이 수정한 amount를 T2가 검색(dirty read)

// 트랜잭션 격리수준과 문제점 발생 가능 여부(ANSI SQL-92 표준)

수준	격리수준	dirty read	non-repeatable read	phantom read
L0	read uncommitted (미완료 읽기)	가능	가능	가능
L1	read committed (완료 읽기)	불가능	가능	가능
L2	repeatable read (반복 가능 읽기)	불가능	불가능	가능
L3	serializable read (직렬 가능 읽기)	불가능	불가능	불가능

- 오손판독(dirty read) 문제점이 발생되지 않도록 하려면
 - 격리수준 L1, L2, L3를 적용해야 한다. DBMS가 L0은 허용하지 않아야 한다.

9. 다음 ERD(Entity-Relationship Diagram)를 구성하는 두 릴레이션 Dept, Emp에 대해 참조무결성 제약조건을 정의하였다고 가정하자. 정의한 참조무결성 제약조건의 영향을 받는 연산들로만 모두 묶은 것은? [2016년 국가 7급]



- ㄱ. Emp 테이블에 새로운 레코드를 삽입하는 삽입 연산
- ㄴ. Emp 테이블에서 외래키 값을 수정하는 변경 연산
- ㄷ. Emp 테이블에서 기존 레코드를 삭제하는 삭제 연산
- ㄹ. Dept 테이블에 새로운 레코드를 삽입하는 삽입 연산
- ㅁ. Dept 테이블에서 기본키 값을 수정하는 변경 연산
- ㅂ. Dept 테이블에서 기존 레코드를 삭제하는 삭제 연산

- ① ㄱ, ㄷ, ㅁ ② ㄴ, ㄹ, ㅂ
- ③ ㄷ, ㄹ, ㅁ ④ ㄱ, ㄴ, ㅁ, ㅂ

☞ ERD - 주어진 문제에 대한 예제 테이블 구조 참조

[Dept]			[Emp]			
부서번호	부서명	위치	사원번호	사원명	연봉	부서번호
101	영업부	노량진	1	가	일천만원	101
102	생산부	신도림	2	나	사천만원	101
103	경리부	종로구	3	다	오천만원	101
			4	라	이천만원	102
			5	마	삼천만원	102
			6	바	이천만원	103
			7	사	오천만원	103

→ 부서번호(외래키)

- ERD에서 '일대다'인 경우에 '다' 쪽에 외래키를 넣으면 최소의 중복이 된다.
→ 즉, 외래키가 Emp 테이블에 존재해야 한다.(핵심 내용)
- Emp 테이블 : 삽입, 수정은 참조무결성 제약조건 영향 받음(외래키 제약조건)
→ 외래키 부서번호는 Dept 테이블의 부서번호 중에 어느 하나를 가져야 하므로
- Dept 테이블 : 수정, 삭제는 참조무결성 제약조건 영향 받음(외래키 제약조건)

10. 데이터베이스는 다양한 응용을 위해 사용되고 있으며 이를 위해 새로운 데이터베이스 기술이 등장하였다. 새로운 데이터베이스 기술에 대한 설명으로 옳지 않은 것은? [2016년 국가 7급]

- ① 공간 데이터베이스에서 다루는 데이터의 형태는 점, 선, 폴리곤 등을 포함하는 지도데이터, 지도의 특징을 기술하는 속성 데이터, 위성이나 항공사진과 같은 이미지 데이터가 주를 이룬다.
- ② 공간 데이터에 대한 색인을 위해서 그리드 파일, R-트리와 같은 새로운 색인 기법이 사용된다.
- ③ 공간조인은 주로 지도상의 점, 선, 폴리곤과 이에 대한 속성 데이터를 조인하는 데에 사용된다.
- ④ 공간 데이터 마이닝의 주요 기법은 공간 분류, 공간 연관성 분석, 공간 클러스터링이 있으며, 이때 공간 데이터에 대해서는 트랜잭션 개념을 가지고 분할하여 분석하지 않는다.

☞ 새로운 데이터베이스 기술 - 공간 데이터베이스(Spatial DBMS)

- 공간조인은 주로 지도상의 점, 선, 폴리곤과 이에 대한 속성 데이터를 조인하는 데에 사용된다.(x)
 - 지도상의 점, 선, 폴리곤과 이에 대한 속성 데이터 조인은 속성 조인이다.
 - 공간조인은 레이어 사이의 관계를 연결하는 조인이다.
 - 공간조인은 공간 관계를 바탕으로 한 레이어에서 다른 레이어로 속성을 전달한다.
 - 하나의 레이어 테이블이 되도록 공간 관계를 기준으로 속성을 추가하는 방법이다.

// 공간조인 예

- 서울시 자치구 경계 레이어와 서울시 지하철 위치 레이어 결합
- 서울시 자치구 경계 레이어와 서울시 종합병원 위치 레이어 결합

// 공간자료 / 속성자료

공간자료 (spatial data)	<ul style="list-style-type: none"> · 공간자료는 위치에 대한 정보이다. · 공간자료는 연속 또는 비연속일 수 있다. · 비연속적 특징 요소 : 점, 선, 면적으로 대변되고 · 연속적인 특징 요소 : 고도(높이), 강우량과 같이 끊기는 부분 · 예 : 건물이나 도로 위치(좌표 x, y)에 대한 정보
속성자료 (attribute data)	<ul style="list-style-type: none"> · 속성자료는 공간 특징 요소를 설명해 놓은 것을 말한다. · 예 : 각 도로의 속성 = (이름, 길이, 방향, 속도제한 등) · 예 : 각 건물의 속성 = (이름, 층수, 높이, 주소, 소유주 등) · 지도에 점이 있을 때, 이 점이 무엇인지? 데이터베이스에 연결되어야 한다.