데이터베이스론	국가 전산 7급	2021년 9월 11일

♠ 합격선(84점) - 선발예정인원 42명 ♠

1. SQL 언어에 대한 설명으로 옳지 않은 것은? [2021년 국가 7급]

① SQL은 관계 데이터베이스 시스템의 표준 언어이다.
② SQL은 포괄적인 데이터베이스 언어로서 데이터 정의, 질의, 갱신을 위한 문들을 가지고 있다.
③ 트랜잭션의 시작, 철회, 완료 등을 표현하기 위해 SQL에서는 COMMIT, ROLLBACK 등을 사용한다.
④ 데이터 조작어는 데이터베이스에 데이터를 검색하여 추가하고 삭제하는 데 사용하며 SELECT, REVOKE가 이에 해당된다.

♤ SQL 언어

• **데이터 조작어**는 데이터베이스에 데이터를 검색하여 추가하고 삭제하는 데 사용하며 SELECT, REVOKE가 이에 해당된다.**(×)**
→ REVOKE는 **데이터 제어어**에 속한다.

정의 기능 (DDL)	• **Create(생성), Drop(삭제), Alter(구조 변경)** 등의 연산이 있다. • 데이터베이스 구조를 정의하는 기능이다. • 데이터베이스를 디스크와 같은 물리적 매체에 저장시킬 수 있어야 한다. • 데이터의 논리적 구조와 물리적 구조 사이에 상호 변환이 가능하도록 하는 사상(mapping)이 포함되어야 한다. • 데이터베이스와 다양한 응용프로그램이 서로 인터페이스할 수 있는 방법을 제공한다.
조작 기능 (DML)	• **Select(검색), Insert(삽입), Update(갱신), Delete(삭제)** 연산이 있다. • 사용자와 데이터베이스 사이의 인터페이스 기능이다. • DBMS는 사용자의 요구에 따라 데이터베이스를 처리할 수 있어야 한다. • 데이터베이스는 동시에 여러 사용자들이 접근할 수 있다.
제어 기능 (DCL)	• **Commit(정상적인 완료), Rollback(비정상적인 종료시 원래 상태 복귀),** • **Grant(사용 권한 부여), Revoke(사용 권한 취소)** 등이 DCL로 분류될 수 있다. • 데이터베이스를 정확하고, 안전하게 유지하는 기능이다.(무결성 유지) • 정당하게 허가된 사용자만 허가된 자료에 접근할 수 있도록 권한을 검사하고, 보안이 유지되어야 한다. • 데이터베이스는 동시에 여러 사용자들이 접근하여 데이터를 처리하여도 결과는 항상 무결성이 유지되도록 병행제어할 수 있어야 한다.

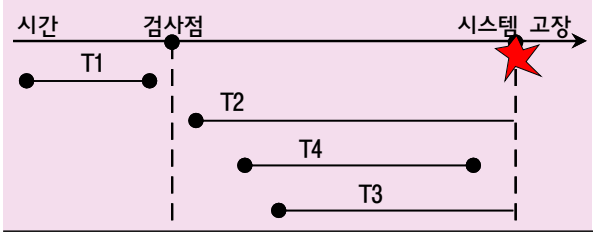
정답 : ④

2. 다음은 4개의 트랜잭션 T1, T2, T3, T4에 대하여 시스템 고장(crash) 시점에 특정 스케줄에 대응하는 로그를 나타낸 것이다. 이 시스템은 로그를 이용한 회복기법으로 검사점(checkpoint)을 가진 즉시갱신규약(immediate update protocol)을 사용한다고 가정한다. 시스템 고장으로부터 회복(recovery)하는 과정에서 undo와 redo 연산들을 수행하게 된다. 회복 후 데이터 항목 A, B, C, D 값들을 바르게 연결한 것은? (단, 로그 레코드 구조는 〈트랜잭션, 데이터 항목, 현재 값, 변경 값〉이다) [2021년 국가 7급]

로그 번호	로그 레코드
1	<T1 start>
2	<T1, D, 20, 25>
3	<T1 commit>
4	<checkpoint {}>
5	<T2 start>
6	<T2, B, 12, 18>
7	<T4 start>
8	<T4, D, 25, 15>
9	<T3 start>
10	<T3, C, 30, 40>
11	<T4, A, 30, 20>
12	<T4 commit>
13	<T2, D, 15, 25>
	시스템 고장

	A	B	C	D
①	30	12	30	20
②	20	18	40	25
③	30	18	40	15
④	20	12	30	15

♤ 검사점을 가진 즉시갱신 규약



- T1 : 검사점 이전에 완료(no action)
- T2 : 실행 중에 시스템 붕괴(undo)
- T3 : 실행 중에 시스템 붕괴(undo)
- T4 : 검사점 이후에 완료(redo)

- A=20, B=12, C=30, D=15

정답 : ④

3. 뷰(view)에 대한 설명으로 옳지 않은 것은? [2021년 국가 7급]

① 생성된 뷰는 새로운 독립적인 테이블로 물리 저장소에 저장된다.
② 데이터베이스 질의 시 사용자 편의성 및 재사용성, 보안성을 높이기 위한 기술이다.
③ 뷰의 질의연산은 제한을 받지 않지만 갱신연산은 제한을 받는다.
④ 하나의 테이블로 여러 개의 상이한 뷰를 정의하여 사용자의 요구에 따라 활용할 수 있다.

♤ **뷰(view)**

- 생성된 뷰는 새로운 독립적인 테이블로 **물리 저장소에 저장**된다.(×)
  → 뷰는 물리적으로 구현되지 않는다. 그 **정의만 시스템에 저장**된다.
- **뷰**는 하나 이상의 다른 테이블로부터 유도되어 만들어진 이름을 가진 '**가상테이블**'이다.
- DBMS는 뷰의 정의만을 **시스템 카탈로그**에 저장해 두었다가 실행시간에 참조한 테이블과 연동되어 테이블을 구축하여 보여 준다.
- **가상**이라고 하는 이유는 실제로 튜플을 갖지 않는 테이블이기 때문이다.

정답 : ①

4. 관계모델에 대한 설명으로 옳지 않은 것은? [2021년 국가 7급]

① 관계모델에서 행은 투플(tuple), 열은 애트리뷰트(attribute)로 부른다.
② 투플 내의 각 값은 더 이상 나누어지지 않는 원자값이어야 한다.
③ 투플 내 어떤 애트리뷰트의 값을 알 수 없거나 값이 지정되지 않을 때는 NULL이라는 특수한 값을 사용한다.
④ 릴레이션은 투플들의 집합이기 때문에 릴레이션에서 투플이 순서대로만 나타나야 한다.

♤ **관계모델**

- 릴레이션은 투플들의 집합이기 때문에 릴레이션에서 투플이 **순서대로만** 나타나야 한다.(×)
  → 릴레이션에서 투플의 순서는 의미가 없다.

// 릴레이션 특성

① 속성의 원자성 : 릴레이션을 구성하는 모든 속성값은 원자값이다.
② 속성의 무순서성 : 릴레이션을 구성하는 속성 사이에는 순서가 없다.
③ 튜플의 유일성 : 릴레이션을 구성하는 모든 튜플은 서로 다르다.
④ 튜플의 무순서성 : 릴레이션의 튜플 사이에는 순서가 없다.

정답 : ④

5. 다음 사원 테이블에 대해 뷰_사원1과 같은 뷰를 생성하는 SQL 질의어를 작성할 때, ㉠, ㉡에 들어갈 내용을 바르게 연결한 것은? [2021년 국가 7급]

사원

사번	이름	직급	부서
108002	김진수	부장	인사
105123	이수진	대리	연구
128372	박지훈	과장	영업
126721	김지수	부장	영업
132122	홍성대	대리	인사

뷰_사원1

사번	이름
105123	이수진

```
Create View 뷰_사원1(사번, 이름)
(     ㉠     ) Select 사번, 이름
              From 사원
              Where (     ㉡     );
```

㉠	㉡
① IN	직급 = '대리'
② AS	부서 = '연구'
③ AS	직급 = '대리'
④ IN	부서 = '연구'

♤ 뷰(view)

• 부서 = '연구'인 사원의 사번과 이름을 구하는 뷰를 생성하는 문제이다.

```
Create View 뷰_사원1(사번, 이름)
( ㉠ As ) Select 사번, 이름
          From 사원
          Where ( ㉡ 부서 = '연구' );
```

// 뷰 생성 구문은 다음과 같다.

```
Create View 뷰_이름[(열_이름_리스트)]                    //대괄호 []는 생략 가능한 부분
       As Select 열_이름_리스트 From 테이블_이름
       [Where 조건]
       [Group By 열_이름] [Having 조건]
       [With Check Option];
```

정답 : ②

6. 트랜잭션 T1과 T2가 아래와 같이 수행될 때, T1의 ㉠, ㉡의 출력 값은? (단, 팬텀(phantom)을 초래하는 위반들을 허용하지 않는다) [2021년 국가 7급]

sale

no	name	price
1	a1	200
2	a2	150
3	b1	100

T1	T2
set transaction isolation level repeatable read;	
	set transaction isolation level repeatable read;
start transaction;	
	start transaction;
	insert into sale values(4, 'b2', 50);
select ㉠ sum(price) from sale;	
	commit
update sale set price = price + 100 where no = 3;	
select ㉡ sum(price) from sale;	
commit	

	㉠	㉡
①	450	550
②	450	600
③	500	600
④	500	500

♤ **트랜잭션 격리수준이 반복가능읽기(repeatable read) 일 때 – 팬텀을 초래하는 위반은 없음**

- 반복가능읽기 : 트랜잭션에서 한번 조회한 데이터를 반복 조회해도 같은 데이터가 조회된다.
- 트랜잭션 T2의 수행 결과는 T1에 영향을 주지 않는다.

```
㉠ select  sum(price) from sale;                  // 출력 : 450 (200+150+100)
update sale set price = price + 100 where no = 3; // no=3의 price=price+100=100+100=200
㉡ select sum(price) from sale;                   // 출력 : 550 (200+150+200)
```

정답 : ①

7. 테이블 R과 S에 대해 아래의 SQL 질의어를 수행한 결과로 나오는 RSID 값은? [2021년 국가 7급]

R

RSID	CName
1001	국어
1002	영어
1003	국어
1003	수학
1004	국어
1004	영어
1004	수학
1005	과학

S

CName
영어
국어

```
SELECT distinct(RSID) FROM R as x
WHERE NOT EXISTS (
        (SELECT p.CName FROM S as p)
         EXCEPT
        (SELECT y.CName FROM R as y WHERE x.RSID = y.RSID));
```

---

① 1001 ② 1002
③ 1003 ④ 1004

♤ Not Exists / Except(차집합)

• 먼저, Not Exists / Except 조합은 테이블 S에 있는 **과목**을 모두 수강한 학생을 구한다.**(1004)**

```
SELECT distinct(RSID) FROM R as x      -- 첫번째, R 정보를 차례로 검색 RSID = {1001, 1002, …}
WHERE NOT EXISTS (                     -- 부정, (S에 있는 모든 과목을 수강한 경우는 공집합 : 참)
        (SELECT p.CName FROM S as p)       — CName = {영어, 국어}
         EXCEPT                            -- 차집합 = S - R
        (SELECT y.CName FROM R as y WHERE x.RSID = y.RSID));— 셀프조인
                        -- RSID = {1001}일 때, {영어, 국어} - {국어} = {영어} ← 거짓
                        -- RSID = {1002}일 때, {영어, 국어} - {영어} = {국어} ← 거짓
                        -- RSID = {1003}일 때, {영어, 국어} - {국어, 수학} = {영어} ← 거짓
                        -- RSID = {1004}일 때, {영어, 국어} - {국어, 영어, 수학} = ∅ ← 참
                        -- RSID = {1005}일 때, {영어, 국어} - {과학} = {영어, 국어} ← 거짓
```

정답 : ④

8. 다음과 같은 student 테이블의 스키마가 주어졌을 때, 주어진 요구에 대한 SQL 질의어로 옳지 않은 것은? [2021년 국가 7급]

---

student(studno, name, grade, score, deptno)
(단, studno는 학번, name은 학생 이름, grade는 학년, score는 성적,
deptno는 학생이 속한 학과 번호를 의미하며, studno는 기본키이다)

---

① student 테이블에서 각 학과별 평균 성적, 최고 성적, 최저 성적 검색
```
SELECT deptno, AVG(score), MAX(score), MIN(score)
FROM student GROUP BY deptno
```
② student 테이블에서 각 학과의 각 학년별 인원수와 평균 성적 검색
```
SELECT deptno, grade, COUNT(*), AVG(score)
FROM student GROUP BY deptno, grade
```
③ student 테이블에서 학과 번호가 100번 이상인 학과들의 평균 성적 검색
```
SELECT deptno, AVG(score)
FROM student
WHERE deptno >= 100
GROUP BY deptno
```
④ student 테이블에서 각 학년별로 학생 수가 10명 이상인 학년, 학생 수, 평균 성적 검색
```
SELECT grade, COUNT(*), AVG(score)
FROM student
WHERE COUNT(*) >= 10
GROUP BY grade
```

♤ **SQL 질의어**

---

④ student 테이블에서 각 학년별로 학생 수가 10명 이상인 학년, 학생 수, 평균 성적 검색
```
SELECT grade, COUNT(*), AVG(score)      -- 학년, 학생수, 평균
FROM student
WHERE COUNT(*) >= 10        -- Where에는 집계함수 사용 불가(오류)
GROUP BY grade              -- grade(학년) 기준으로 그룹화
        ↓올바르게 고치면
SELECT grade, COUNT(*), AVG(score)      -- 학년, 학생수, 평균
FROM student
GROUP BY grade              -- grade(학년) 기준으로 그룹화
Having Count(*) >= 10;      -- 해당 학년의 학생 수가 10명 이상인 경우
```

정답 : ④

9. 정규화(normalization)에 대한 설명 중 옳은 것만을 모두 고르면? [2021년 국가 7급]

ㄱ. 데이터의 정규화는 중복을 최소화하고 삽입, 삭제, 수정 이상을 최소화하기 위해서 함수적 종속성과 기본키를 기반으로, 주어진 릴레이션 스키마를 분석하는 과정이다.

ㄴ. 릴레이션 스키마 R의 모든 원소들의 도메인(domain)이 나눌 수 있는 단위로 되어있을 때, R이 제1정규형에 속한다.

ㄷ. 제2정규형이 되기 위해서는 릴레이션 R이 제1정규형이고 기본키가 아닌 속성이 기본키에 부분함수종속이어야 한다.

ㄹ. 제3정규형이 되기 위해서는 릴레이션 R이 제2정규형이고, 릴레이션 R의 함수종속 관계에서 이행적함수종속을 제거해야 한다.

ㅁ. 제4정규형이 되기 위해서는 릴레이션 R이 제3정규형이고, 함수종속 관계에서 모든 결정자가 후보키이면 된다.

---

① ㄱ, ㄹ　　② ㄴ, ㄷ

③ ㄱ, ㄹ, ㅁ　　④ ㄷ, ㄹ, ㅁ

♤ **정규화**

ㄴ. 릴레이션 스키마 R의 모든 원소들의 도메인(domain)이 **나눌 수 있는 단위**로 되어있을 때, R이 제1정규형에 속한다.**(×)**

→ R의 모든 원소들의 도메인이 **나눌 수 없는 단위**로 되어있을 때, 제1정규형에 속한다.

ㄷ. 제2정규형이 되기 위해서는 릴레이션 R이 제1정규형이고 기본키가 아닌 속성이 기본키에 **부분함수종속**이어야 한다.**(×)**

→ 제2정규형이 되기 위해서는 제1정규형이고 기본키가 아닌 속성이 기본키에 **완전함수종속**이어야 한다.

ㅁ. **제4정규형**이 되기 위해서는 릴레이션 R이 제3정규형이고, 함수종속 관계에서 모든 결정자가 후보키이면 된다.**(×)**

→ **보이스/코드 정규형**(BCNF; boyce/codd normal form)에 대한 설명이다.

// **제4정규형**을 간단하게 정의하면 다음과 같다.

함수종속이 아닌 다치종속(MVD)이 제거되면 4NF에 속한다.

- **다치종속**은 하나의 릴레이션에 **다가속성**이 **2개이상** 존재할 때 발생한다.
- 원래, 관계 데이터베이스에서는 속성 값으로 다중값(다가속성)을 허용하지 않는다.(제1NF 제약)
- 다치종속은 머리가 두 개인 이중 화살표 기호 →→로 나타낸다.
- 예 : 과목 →→ 교수, 교수 = {P1, P2}

정답 : ①

10. NOSQL에 대한 설명으로 옳지 않은 것은? [2021년 국가 7급]

① NOSQL은 샤딩(sharding)을 지원한다.
② BigTable, Cassandra 등이 대표적인 NOSQL이다.
③ NOSQL은 RDBMS와 같이 스키마(schema)를 필요로 한다.
④ NOSQL은 가용성(availability)과 확장성(scalability)을 중요시 한다.

♤ NOSQL

• NOSQL은 RDBMS와 같이 **스키마**(schema)를 필요로 한다.**(×)**
→ NOSQL은 **스키마**를 강제 적용하지 않는다. 데이터 관계와 정해진 규격(table 정의)이 없다.

// 샤딩(sharding)
• 샤딩은 하나의 거대한 데이터베이스나 네트워크 시스템을 여러 개의 작은 조각으로 나누어 분산 저장하여 관리하는 것을 말한다.
• 영어 **shard**는 **조각**이라는 의미이다.

정답 : ③