

# SQLD 기출문제 39회 오답노트

다음의 정규화 단계에서 주식별자와 관련성이 가장 낮은 것은? 3

1. 제 1정규화
2. 제 2정규화
3. 제 3정규화
4. BCNF

제 3정규화는 주식별자를 제외한 칼럼 간에 종속성을 확인해서 종속성이 있으면 분할하는 과정임

문제 8. 다음 ERD(Entity Relationship Diagram) 작성 순서로 올바른 것을 고르시오.

1.1) 가) → 나) → 다) → 라) → 마) → 바)

가) 엔터티를 그린다.

나) 엔터티를 적절하게 배치한다.

다) 엔터티 간에 관계를 설정한다.

라) 관계명을 기술한다.

마) 관계의 참여도를 기술한다.

바) 관계의 필수 여부를 기술한다.

문제 9. 다음 ERD에서 식별자 분류로 올바른 것은?

2

2) ㄱ-주식별자, ㄴ-내부 식별자, ㄷ-외부 식별자

고객마스터 테이블의 고객번호는

대표성에 따른 식별자 분류로 ( ㄱ )이고

스스로 생성했는지 여부에 따라서는 ( ㄴ )이다.

또한 계좌마스터의 고객번호는 스스로 생성여부에 따라서

( ㄷ )이다.

문제 10. 식별자 중에서 비즈니스 프로세스에 의하여 만들어지는 식별자로 대체여부로 분리되는 식별자는 무엇인가?

1. 본질 식별자

해설 : 대체여부에 따라서 본질 식별자와 인조 식별자로 분류되고 → 본질 식별자는 비즈니스 프로세스에 의해서 만들어지는 식별자이다.

문제 20. 다음 주어진 SQL문에서 오류가 발생하지 않는 것은? 4

```
CREATE TABLE SQLD39_20(
  ID NUMBER PRIMARY KEY,
  AGE NUMBER NOT NULL,
  NAME VARCHAR2(1)
);
```

- - - - -

```
(1) INSERT INTO SQLD39_20 VALUES(10,20,SYSDATE);
(2) INSERT INTO SQLD39_20 VALUES(20,NULL,'A');
(3) INSERT INTO SQLD39_20(AGE, NAME) VALUES(20,'A');
(4) INSERT INTO SQLD39_20(ID, AGE, NAME) VALUES(20,10,NULL);
```

-----

(1) 번은 NAME에 DATE 타입의 데이터를 입력하여 에러  
 (2) 번은 AGE(NOT NULL)에 NULL을 입력하여 에러  
 (3) 번은 ID(PK) 값을 입력하지 않아서 에러

## 문제 21. 다음 계층형 쿼리문에 대한 설명으로 옳지 않은 것은? 4

```
SELECT ID, PARENT_ID, NAME, PARENT_NAME
FROM SQLD39_21
WHERE PARENT_ID NOT IN(3)
START WITH PARENT_ID = 0
CONNECT BY PRIOR ID = PARENT_ID
ORDER SIBLINGS BY PARENT_ID ASC, ID ASC;
```

1) PARENT\_ID가 0이라도 3이 포함되면 전개를 멈춘다.  
 2) 순방향 전개다.  
 3) 중복이 생겼을 때 루프를 돌지 않기 위해 NO CYCLE 옵션을 사용할 수 있다.  
 4) ORDER SIBLINGS BY를 하면 전체 테이블 기준으로 정렬한다.

-----

ORDER SIBLINGS BY 를 수행하면 전체 테이블이 아니라 계층형으로 된 데이터값(특정 칼럼) 기준으로 정렬된다.

(1) NOT IN (3) 이므로 3이 포함되면 전개를 멈춘다  
 (2) CONNECT BY PRIOR ID = PARENT\_ID (부모 ID 를 기준으로 자식 ID 를 검색) 순방향  
 (3) NO CYCLE 옵션을 사용할 수 있다.

## 문제 30. 다음의 SUB QUERY의 유형은 무엇인가? 3

```
SELECT A.EMPNO, A.ENAME
FROM EMP A
WHERE A.EMPNO = (SELECT 1 FROM
EMP_T B WHERE A.EMPNO = B.EMPNO);
```

1) SERVICE 서브쿼리

2) EARLY FILTER형 서브쿼리

3) CORRELATED 서브쿼리

4) LOOPING 서브쿼리

-----  
해설 : 메인쿼리의 값을 서브쿼리에서 주입을 받아서 비교를 하는것으로 상호연관 서브쿼리(CORRELATED SUB QUERY) 이다.

```
SELECT A.EMPNO, A.ENAME
FROM EMP A
WHERE A.EMPNO = (SELECT 1 FROM
EMP_T B WHERE A.EMPNO = B.EMPNO);
```

서브쿼리에 \*(A.EMPNO 값을 매번 가져와서 대입을 해야하므로 성능이 매우 좋지않다.)

### 문제 37. 다음 보기 중 서브쿼리에 대한 설명을 옳지 않은 것은? 2

- 1) 서브쿼리에서는 정렬을 수행하기 위해서 내부에 ORDER BY를 사용하지 못한다.
- 2) 메인 쿼리를 작성할 때 서브쿼리에 있는 칼럼을 자유롭게 사용할 수 있으면 편리하다.
- 3) 여러 개의 행을 되돌리는 서브쿼리는 다중행 연산자를 사용해야 한다.
- 4) EXIST는 TRUE와 FALSE만 되돌린다.

### 문제 39. PL/SQL에서 데이터베이스 CURSOR를 사용할 때 FETCH 전에 해야 하는 것은? 2

- 1) CURSOR DEFINE
- 2) CURSOR OPEN
- 3) CURSOR CLOSE
- 4) EXIT

해설 : FETCH(읽어오기) 위해 해야할 것은 CURSOR OPEN (\*CURSOR순서 : 선언 → OPEN → FETCH → CLOSE)

#### CURSOR(DB의 연결 포인트, 연결점)

- SQL 커서는 Oracle 서버에서 할당된 전용 메모리 영역에 대한 포인터이다.
- 질의의 결과로 얻어진 여러 행이 저장된 메모리상의 위치.
- 커서는 SELECT 문의 결과 집합을 처리하는데 사용된다.

명시적 커서란 사용자가 직접 정의해서 사용하는 커서이고 묵시적(암시적) 커서는 데이터베이스가 내부적으로 사용하는 커서이다.

모든 CURSOR는 사용하기 전에 반드시 선언을 해주어야 한다.

**문제 41. 다음은 분산 데이터베이스에 대한 설명이다. 옳바르지 않은 것은?**

- 1) 분산 데이터베이스는 네트워크를 경유하여 여러 개의 데이터베이스로 분리되어 있다.
- 2) 분산 데이터베이스는 시스템 가용성이 떨어진다.
- 3) 분산 데이터베이스는 여러 개의 데이터 베이스를 병렬적으로 실행하여 성능을 향상시킨다.
- 4) 사용자는 분산 데이터베이스를 인식하지 못하고 데이터베이스를 사용한다.

해설 : 분산 데이터베이스 신뢰성과 가용성이 높다.

장점	단점
데이터베이스 신뢰성과 가용성이 높다.	데이터베이스가 여러 네트워크를 통해서 분리되어 있기 때문에 관리와 통제가 어렵다.
분산 데이터베이스가 병렬 처리를 수행하기 때문에 빠른 응답이 가능	보안관리가 어렵다.
분산 데이터베이스를 추가하여 시스템 용량 확장이 쉽다.	데이터 무결성 관리가 어렵다.
	데이터베이스 설계가 복잡함