

A banner featuring a close-up of a person's face with a grid overlay. The word "SQL" is written in a large, white, stylized font.

SQL

Professional • Developer

SQL 개발자 · SQL 전문가 자격검정 시험 예제

2010. 10.



한국데이터베이스진흥원

1 과목

데이터 모델링의 이해

[예제 5문항]

1. 다음 중 아래에서 설명하는 데이터모델의 개념으로 가장 적절한 것은?

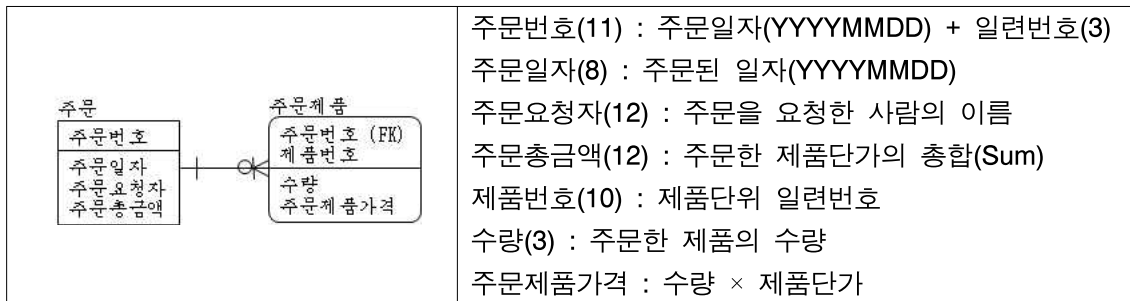
<아래>

학생이라는 엔터티가 있을 때 학점이라는 속성의 값의 범위는 0.0 에서 4.0 사이의 실수 값이며, 주소라는 속성은 길이가 20자리 이내의 문자열로 정의할 수 있다.

- ① 도메인(Domain)
- ② 용어사전(Word Dictionary)
- ③ 속성사전(Attribute Dictionary)
- ④ 시스템카탈로그(System Catalog)

2. 다음 중 아래 데이터 모델의 속성에 대한 설명으로 가장 부적절한 것은?

<아래>



- ① 주문일자는 날짜만을 입력하므로 기본속성(Basic Attribute)에 해당한다.
- ② 주문요청자는 주문한 사람의 정보를 입력하므로 기본속성에 해당한다.
- ③ 주문총금액은 주문한 제품의 가격을 모두 더한 값이므로 추출속성(Derived Attribute)에 해당한다.
- ④ 주문제품가격은 제품단가에 수량을 곱한 것으로 원래 값과 1:1 매핑되어 설계속성(Designed Attribute)에 해당한다.

3. 다음 중 아래 개념이 설명하는 관계로 가장 적절한 것은?

<아래>

부모엔터티의 주식별자를 상속받아 자식엔터티에서 외부식별자이면서 주식별자(Primary Key)로 사용하는 관계

- ① 식별자관계(Identifying Relationship)
- ② 일반속성관계(Attribute Relationship)

2과목

SQL 기본 및 활용

[예제 5문항]

6. 다음 중 ANSI/ISO 표준 SQL에서 표시하는 FROM 절의 JOIN 형태로 가장 부적절한 것은?

- ① NATURAL JOIN
- ② HASH JOIN
- ③ OUTER JOIN
- ④ CROSS JOIN

7. 다음 중 INSERT와 UPDATE를 동시에 수행할 수 있는 SQL 구문으로 적절한 것은?

- ① UPSERT 문
- ② MERGE 문
- ③ INSERT FIRST 문
- ④ INSERT ALL 문

8. 다음 중 관계 데이터베이스에서 뷰(view)를 사용하는 주요 장점으로 가장 적절한 것을 2개 고르시오.

- ① 데이터 처리의 절차성
- ② 보안 강화
- ③ 성능 향상
- ④ 복잡한 질의에 대한 단순화

9. [단답형]아래와 같은 테이블에서 다음 SQL 실행 결과의 □안에 들어갈 ㉠, ㉡을 작성하시오.

<아래>

[TAB]

C1	C2
1	1
	1
1	
1	1

SELECT COUNT(*) C1, COUNT(C1) C2, AVG(1) C3, AVG(C2) C4 FROM TAB;

실행 결과

C1	C2	C3	C4
4	<input type="text" value="Ⓜ"/>	1	<input type="text" value="Ⓛ"/>

10. [단답형]아래 테이블에서 사원들의 급여와, 매니저가 같은 사원들의 SALARY 합을 구하는 다음 SQL 문장의 □안에 들어갈 Ⓜ, Ⓛ을 작성하시오.
(※ 한글이나 영문으로 작성하고 영문은 대문자로, 띄어쓰기는 1칸만 작성할 것)

<아래>

[EMP]

<u>MGR</u>	<u>ENAME</u>	<u>SAL</u>	<u>MGR_SUM</u>
7566	FORD	3000	6000
7566	SCOTT	3000	6000
7698	JAMES	950	6550
7698	ALLEN	1600	6550
7698	WARD	1250	6550
7698	TURNER	1500	6550
7698	MARTIN	1250	6550
7782	MILLER	1300	1300
7788	ADAMS	1100	1100
7839	BLAKE	2850	8275
7839	JONES	2975	8275
7839	CLARK	2450	8275
7902	SMITH	800	800
	KING	5000	5000

```
SELECT MGR, ENAME, SAL,
SUM(SAL)  (  BY MGR) MGR_SUM
FROM EMP;
```

3과목

SQL 고급 활용 및 튜닝

[예제 5문항]

11. [단답형]아래 로그 파일에 대한 설명에서 □안에 공통적으로 들어갈 말을 작성하시오.

<아래>

대부분 DBMS는 버퍼 블록에 대한 변경사항을 건건이 데이터 파일에 기록하기보다 우선 로그 파일에 Append 방식으로 빠르게 기록하는 방식을 사용한다. 그리고 나서 버퍼 블록과 데이터 파일 간 동기화는 적절한 수단(DBWR, Checkpoint)을 이용해 나중에 배치 (Batch) 방식으로 일괄 처리한다.

사용자의 갱신내용이 메모리상의 버퍼 블록에만 기록된 채 아직 디스크에 기록되지 않았더라도 로그 파일을 믿고 빠르게 커밋을 완료한다는 의미에서, 이를 □ 메커니즘이라고 부른다. 인스턴스 장애가 발생하더라도 로그 파일을 이용해 언제든지 복구 가능하므로 안심하고 커밋을 완료할 수 있는 것이다. □은(는) 빠르게 트랜잭션을 처리해야 하는 모든 DBMS의 공통적인 메커니즘이다.

12. 다음 중 Lock에 대한 설명으로 가장 적절한 것은?

- ① DB를 Snapshot 모드로 운영하지 않는 한, SQL Server에서는 다른 트랜잭션이 갱신 중인 레코드를 동시에 읽고자 할 때 Lock에 의한 대기 현상이 발생한다.
- ② Oracle에서 테이블 Lock(=TM Lock)이 걸리면 그 순간부터 다른 트랜잭션의 동시 INSERT는 불가능해진다.
- ③ 교착상태가 발생했을 때 DB 관리자가 둘 중 어느 한 세션을 강제 종료시키지 않는 한 교착상태는 풀리지 않는다.
- ④ Lock Escalation이 발생하고 나면 Lock Resource 사용량도 증가한다.

13. 다음 중 옵티마이저에 대한 설명으로 가장 적절한 것을 2개 고르시오.

- ① Bind 변수를 사용해도 옵티마이저는 칼럼 히스토그램을 바탕으로 최적의 경로를 선택한다.
- ② CBO는 통계정보를 바탕으로 데이터 액세스 비용을 계산하여 최적의 경로를 선택한다.
- ③ RBO는 인덱스 구조나 사용 연산자 등의 우선순위를 근거해서 최적의 경로를 선택한다.
- ④ RBO도 경우에 따라서는 통계정보를 이용하여 최적의 경로를 선택한다.

14. 다음 중 아래 DDL 문장을 보고 정상적으로 인덱스 사용이 가능한 SQL 문장을 2개 고르시오. (“정상적인 인덱스 사용”이란, Oracle의 Index Unique Scan, Index Range Scan 또는 MS-SQL Server의 Index Seek 방식으로 인덱스를 스캔하는 것을 의미함)

<아래>

```
CREATE TABLE TAB1 (
    COL1 VARCHAR2(10),
    COL2 NUMBER,
    COL3 VARCHAR2(10)
);
ALTER TABLE TAB1 ADD CONSTRAINT PK_TAB1 PRIMARY KEY (COL1);
CREATE INDEX TAB1_IDX1 ON TAB1 (COL3, COL2);
```

- ① SELECT * FROM TAB1 WHERE COL1 NOT IN ('ABC', 'XYZ')
- ② SELECT * FROM TAB1 WHERE COL3 = 10 AND COL2 = 30
- ③ SELECT * FROM TAB1 WHERE COL3 LIKE 'A%' AND COL2 = 50
- ④ SELECT * FROM TAB1 WHERE COL3 BETWEEN 'A' AND 'Z'

15. 다음 중 아래 쿼리와 실행계획을 보고, 실행순서로 가장 적절한 것은?

<아래>

```
select *
from   Orders a, [Order Details] b
where  a.OrderDate <= '1996-07-05'
and    a.OrderID = b.OrderID
```

Rows	Executes	StmtText
5	1	select *
㉠ 5	1	--Nested Loops(Inner Join, OUTER REFERENCES:([a].[OrderID]))
㉡ 2	1	--Nested Loops(Inner Join, OUTER REFERENCES:([a].[Order
㉢ 2	1	--Index Seek(OBJECT:([Orders].[OrderDate] AS [a]),
㉣ 2	2	--Clustered Index Seek(OBJECT:([Orders].[PK_Orders
㉤ 5	2	--Clustered Index Seek(OBJECT:([Order Details].[PK_Orde

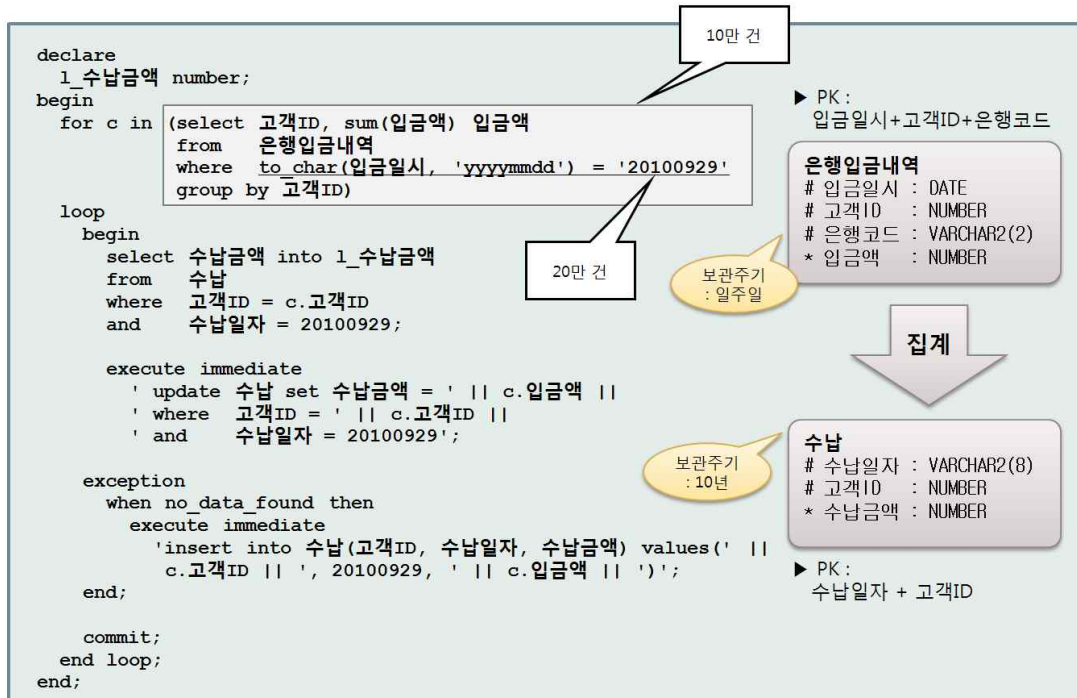
- ① ㉢ → ㉣ → ㉡ → ㉠ → ㉤
- ② ㉠ → ㉡ → ㉢ → ㉣ → ㉤
- ③ ㉤ → ㉣ → ㉢ → ㉡ → ㉠
- ④ ㉡ → ㉢ → ㉡ → ㉠ → ㉤

실기 예제

[예제 1문항]

16. 성능 측면에서 아래 프로그램이 가진 문제점을 나열하고, 각 문제점에 대한 개선 방안을 간단히 요약하시오. 또한, 최적의 성능을 내도록 하나의 SQL 문장으로 작성하시오.

<아래>



정답 및 해설

<1과목> 데이터 모델링의 이해

1. ①번

각 엔터티(테이블)의 속성에 대해서 어떤 유형의 값이 들어가는지를 정의하는 개념은 도메인(Domain)에 해당한다.

2. ④번

주문제품가격은 속성 정의된 내용에 보면 수량과 제품단가를 곱하여 나온 값이므로 계산된 값이라 할 수 있다. 따라서 추출속성에 해당한다.

3. ①번

부모엔터티와 자식엔터티의 관계에서 부모의 PK를 받아 자식엔터티의 주식별자관계로 사용하는 것은 식별자관계에 해당한다.

4. ②번

부분함수 종속성(Partial Dependency Attribute)을 가지고 있으므로 2차 정규화가 필요한 데이터 모델이다.

5. ④번

도메인 반정규화라는 용어는 존재하지 않는다.

<2과목> SQL 기본 및 활용

6. ㉔번

ANSI/ISO 표준 SQL에서 표시하는 FROM 절의 JOIN 형태는 다음과 같다.

- INNER JOIN
- NATURAL JOIN
- USING 조건절
- ON 조건절
- CROSS JOIN
- OUTER JOIN

7. ㉔번

INSERT와 UPDATE를 동시에 처리할 수 있는 SQL 구문은 MERGE 문이다.

8. ㉔번, ㉕번

뷰는 다음과 같은 장점을 갖는다.

- o 독립성 : 테이블 구조가 변경되어 뷰를 사용하는 응용 프로그램은 변경하지 않아도 된다.
- o 편리성 : 복잡한 질의를 뷰로 생성함으로써 관련 질의를 단순하게 작성할 수 있다. 또한 해당 형태의 SQL문을 자주 사용할 때 뷰를 이용하면 편리하게 사용할 수 있다.
- o 보안성 : 직원의 급여정보와 같이 숨기고 싶은 정보가 존재한다면, 뷰 생성할 때 해당 칼럼을 빼고 생성함으로써 사용자에게 정보를 감출 수 있다.

9. ㉖ 3, ㉗ 1

집합 함수에서 컬럼명을 사용할 경우에는 NULL은 계산에서 제외된다.

10. ㉖ OVER, ㉗ PARTITION

```
SELECT MGR, ENAME, SAL,  
SUM(SAL) OVER (PARTITION BY MGR) MGR_SUM  
FROM EMP;
```

OVER는 윈도우 함수의 핵심 용어이며, PARTITION BY MGR 구문을 통해 매니저별로 데이터를 파티션화 한다.

<3과목> SQL 고급 활용 및 튜닝

11. Fast Commit

12. ①번

가장 일반적으로 사용되는 Row Exclusive 모드 테이블 Lock은 물론, Row Share, Share, Share Row Exclusive 모드에서도 동시 DML을 블로킹하지 않는다. (물론 같은 로우를 액세스하지 않는 경우이다.)

Exclusive 모드 테이블 Lock일 때만 동시 DML을 블로킹한다.

교착상태가 발생하면 DBMS가 둘 중 한 세션을 강제 종료시킨다.

SQL Server 등에서 Lock Escalation 메커니즘을 사용하는 이유는, Lock Resource 사용량을 줄이기 위해서다.

13. ②번, ③번

Bind 변수를 사용하면 옵티마이저가 칼럼 히스토그램을 사용할 수 없게 된다.

RBO는 통계정보를 전혀 활용하지 못한다.

14. ③번, ④번

①번은 NOT 연산자이므로 인덱스 사용이 불가하고, ②번은 COL3 칼럼에 내부적인 형변환이 발생한다.

15. ①번

실기 예제

16.

문제점	개선방안
루프 내에서 반복 액세스되는 수납 테이블 조건절에 내부적인 형변환 발생 → Full Table Scan 반복	수납일자 상수 조건을 문자형으로 기술 → 수납일자 = '20100929'
루프 내에서 반복 액세스되는 수납 테이블 update, insert 문에 바인드 변수 미사용 → 하드파싱 부하	바인드 변수 사용
절차형(Loop Query)으로 개발 → 과도한 데이터베이스 Call 발생	1. Update & Insert 방식으로 구현 2. Merge문 활용 3. Array Processing 활용

```
merge into 수납 r
using (select 고객ID, sum(입금액) 입금액
      from   은행입금내역
      where  입금일시 between to_date('20100929', 'yyyymmdd')
                           and   to_date('20100930', 'yyyymmdd') - 1/24/60/60
      group by 고객ID) i
on (r.고객ID = i.고객ID and r.수납일자 = '20100929')
when matched then update
    set r.수납금액 = i.입금액
when not matched then
    insert (고객ID, 수납일자, 수납금액) values (i.고객ID, '20100929', i.입금액);
```