

# LCP-기술인증테스트

## SQL개발

[2017년 #차]

사번 : \_\_\_\_\_

성명 : \_\_\_\_\_

1. 아래 SQL은 부서코드가 93445인 직원들의 자산 보유 내용을 조회하기 위한 구문이다. 입사하고 아직 자산을 보유하기 전이라도 결과는 추출되어야 할 때, 빈칸 (가) 에 알맞은 내용을 고르시오.

```
SELECT A.사원이름, B.자산번호,B.자산유형
FROM 사원 A (가) JOIN 자산이력 B
ON ( A.사원번호 = B.사원번호 AND B.부서코드 = 93445 )
```

- ① RIGHT OUTER
- ② LEFT OUTER
- ③ CROSS
- ④ INNER

(정답) 2

(배점) 3

(난이도) 하

(문제유형) 주관식

(해설) 사원 기준으로 만족하는 데이터가 없어도 추출되어야 하므로 LEFT OUTER JOIN 이 필요함

(대모듈 > 소모듈) SQL 활용 > 표준조인

2. 아래 주문과 제품 테이블의 DDL 문과 데이터를 참조하여 (1)~(7) 번 문장들을 순차적으로 실행하였을 때 (7) 번 SQL 구문의 결과 값을 쓰시오. 단, DML문 실행 이후에는 COMMIT 이 수행되며 문장 수행 시 오류가 발생할 경우 바로 다음 문장이 이어서 수행된다.

[주문, 배송 테이블 DDL 문]

```
CREATE TABLE 주문
(
    주문번호          VARCHAR2(20) NOT NULL ,
    주문일자          VARCHAR2(8)  NOT NULL ,
    고객번호          VARCHAR2(20) NOT NULL ,
    상품코드          VARCHAR2(10) NOT NULL ,
    주문수량          NUMBER       NOT NULL ,
    주문상태          VARCHAR2(20) NOT NULL
    PRIMARY KEY (주문번호)
)
CREATE TABLE 배송
(
    배송번호          VARCHAR2(20) NOT NULL ,
    배송일자          VARCHAR2(8)  NULL ,
    배송상태          VARCHAR2(20) NOT NULL ,
    주문번호          VARCHAR2(20) NOT NULL ,
    배송업체코드      VARCHAR2(20) NULL
    PRIMARY KEY (배송번호),
    FOREIGN KEY (주문번호) REFERENCES 주문(주문번호)
)
```

[주문]

주문번호	주문일자	고객번호	상품코드	주문수량	주문상태
------	------	------	------	------	------

O1234	20170225	C001	TV0032	1	배송중
-------	----------	------	--------	---	-----

[배송]

배송번호	배송일자	배송상태	주문번호	배송업체코드
D1230	20170227	배송중	O1234	T1234

1	UPDATE 주문 SET 주문상태 = '배송완료' WHERE 주문번호 = 'O1234'
2	UPDATE 배송 SET 배송상태 = '배송완료' WHERE 배송번호 = 'D1230' AND 주문번호 = 'O1234'
3	DELETE FROM 주문 WHERE 주문번호 = 'O1234'
4	INSERT 주문 (주문번호, 주문일자, 고객번호, 상품코드, 주문수량, 주문상태) VALUES ('O1234','20170301','C0001','TV0032',1,'반품배송중')
5	INSERT 배송 (배송번호, 배송일자, 배송상태, 주문번호, 배송업체코드) VALUES ('D1231','20170302','반품배송완료','O1234','V100')
6	UPDATE 주문 SET 배송상태 = '반품배송완료' WHERE 주문번호 = 'O1234' AND 배송상태 = '반품배송중'
7	SELECT 배송상태 FROM 배송 WHERE 주문번호 = 'O1234'

(정답) 배송완료, 반품배송완료

(배점) 3

(난이도) 중

(문제유형) 주관식

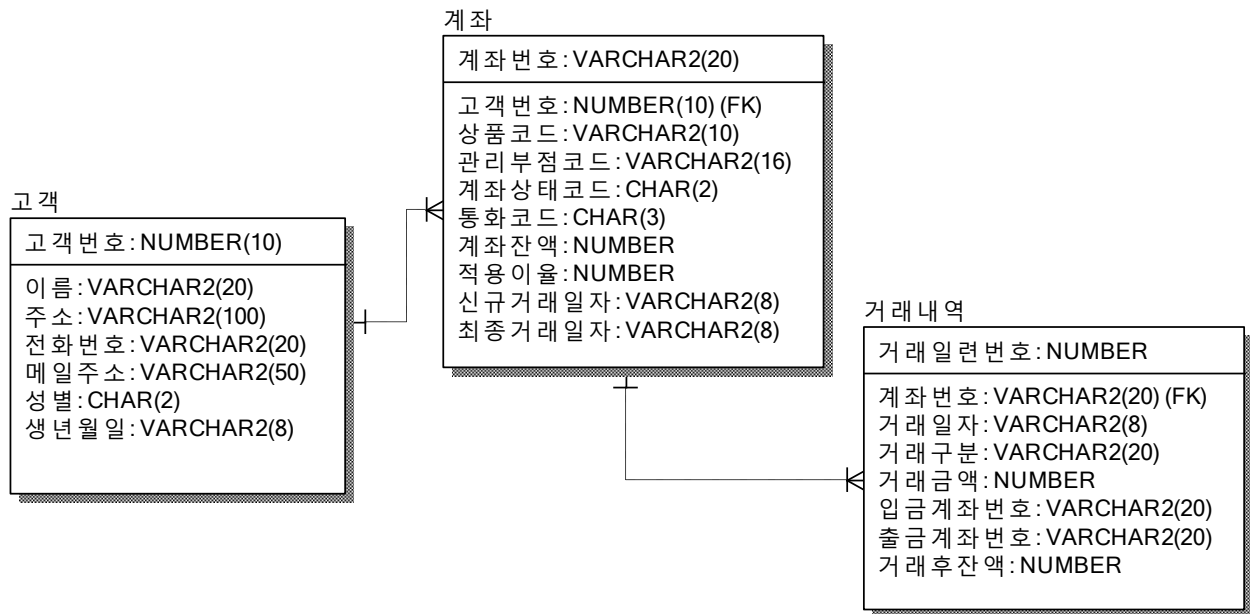
(해설) 3 번 라인 에러 발생 : 자식 테이블에 O1234 주문번호로 데이터가 존재하기 때문에 에러 발생

4 번 라인 에러 발생 : 3 번에서 O1234 로 데이터가 삭제되지 못하였기 때문에 PK Constraint 위반 에러 발생

6 번 라인 : 만족하는 데이터가 존재하지 않으므로 UPDATE 되지 않음

(대모듈 &gt; 소모듈) SQL 기본 &gt; DDL 문

\* 3 ~ 5번은 아래 ERD를 참고하여 물음에 답하시오.



3. 2017년 2월 결산을 수행하면서 'K100' 상품 계좌에 대하여 이자를 지급하고 현재 잔액에 이자금액을 더한 값으로 계좌잔액 칼럼을 변경하고자 한다. 위 ERD를 참조하여 아래의 보기 중 올바른 UPDATE 문을 고르시오.

① UPDATE 계좌 A

```

SET A.계좌잔액 = A.계좌잔액
+ ( SELECT AVG(거래후잔액)
      FROM 거래내역 B
      WHERE B.거래일자 BETWEEN '20170201' AND '20170228'
      GROUP BY B.계좌번호 , B.거래일자 ) * ((A.적용이율/100)/12 )
WHERE A.상품코드 = 'K100'
  
```

② UPDATE 계좌 A

```

SET A.계좌잔액 = A.계좌잔액
+ ( SELECT AVG(거래후잔액)
      FROM 거래내역 B
      WHERE B.거래일자 BETWEEN '20170201' AND '20170228'
      AND B.계좌번호 = A.계좌번호 ) * ((A.적용이율/100)/12 )
WHERE A.상품코드 = 'K100'
  
```

③ UPDATE 계좌 A

```

SET A.계좌잔액 = ( SELECT AVG(거래후잔액)
                    FROM 거래내역 B
                    WHERE B.거래일자 BETWEEN '20170201' AND '20170228'
                    AND B.계좌번호 = A.계좌번호
                    AND A.상품코드 = 'K100'
                  ) * ((A.적용이율/100)/12 )
  
```

④ UPDATE 계좌 A

```

SET A.계좌잔액 = AVG(거래후잔액) * ((A.적용이율/100)/12 )
INNER JOIN 거래내역 B
ON (A.계좌번호 = B.계좌번호 AND B.거래일자 BETWEEN '20170201' AND '20170228' )
WHERE A.상품코드 = 'K100'
AND A.계좌번호 = B.계좌번호
  
```

(정답) 2

(배점) 3

(난이도) 하

(문제유형) 객관식

(해설) 1.SET 절의 스칼라 서브 쿼리에서 GROUP BY B.계좌번호로 인해 다수개의 결과 건이 리턴되므로 에러 발생

3.기존 계좌잔액에 이자금액 만큼만 더해야 하므로 올바르지 않음

4.지원하지 않는 구문 형태이므로 에러 발생

(대모듈 &gt; 소모듈) SQL기본 &gt; DML

4. 위 ERD와 아래 데이터 를 참조하여 고객번호가 9877인 고객이 보유한 계좌잔액과 총잔액값을 아래의 실행결과 같이 추출하고자 할 때 빈칸 (가) 부분에 들어갈 올바른 내용을 쓰시오.

[계좌]

계좌번호	고객번호	상품코드	최종거래일자	계좌잔액
123-45-678	9877	D100	2016-03-05	500000
123-45-699	9877	K100	2016-03-26	200000

[SQL 구문]

```
SELECT A.고객번호,
       A.상품코드,
       A.계좌잔액,
       (가) AS "총잔액"
FROM 계좌 A
WHERE A.고객번호 = 9877
```

[실행 결과]

고객번호	상품코드	계좌잔액	총잔액
9877	D100	500000	700000
9877	K100	200000	700000

(정답) SUM(A.계좌잔액) OVER (),SUM( A.계좌잔액 ) OVER( PARTITIO BY A.고객번호 )

(배점) 3

(난이도) 상

(문제유형) 객관식

(해설) 고객번호로 PARTITION BY 를 하거나 PARTITION BY 절 없이 SUM(A.계좌번호)를 분석함수를 적용하여야 함

(대모듈 &gt; 소모듈) SQL활용 &gt; 분석함수

5. 성능개선을 위하여 아래와 같이 2개의 신규 인덱스(IX\_거래내역\_01, IX\_계좌\_01)를 생성하였을 경우 아래 보기 중 신규 인덱스를 가장 효율적으로 사용하는 것을 고르시오.

[테이블 정보]

테이블 명	테이블 건수	컬럼명	종류(Distinct) 건수
고객	500,000	고객번호	500,000
계좌	1,500,000	고객번호	500,000
거래내역	14,000,000	계좌번호	1,500,000
		거래일자	4,000

## [신규 인덱스 구성정보]

테이블 명	인덱스명	인덱스 컬럼 구성
계좌	IX_계좌_01	고객번호
거래내역	IX_거래내역_01	거래일자 , 계좌번호

- ① SELECT A.거래일자, B.고객번호, A.거래후잔액  
FROM 거래내역 A  
INNER JOIN 계좌 B ON (A.계좌번호 = B.계좌번호)  
WHERE SUBSTR(A.계좌번호,1,4) = :acc\_no  
AND A.거래일자 BETWEEN :from\_ymd AND :to\_ymd
- ② SELECT A.거래일자, C.이름, C.고객번호, A.거래후잔액  
FROM 거래내역 A  
INNER JOIN 계좌 B ON (A.계좌번호 = B.계좌번호)  
INNER JOIN 고객 C ON (B.고객번호 = C.고객번호)  
WHERE A.거래일자= CASE WHEN :gubun = 'ALL' THEN A.거래일자  
ELSE :yyyymmdd END  
AND C.고객번호 = :cust\_no  
AND A.계좌번호 = :acc\_no
- ③ SELECT A.고객번호,A.계좌잔액,A.최종거래일자  
FROM 계좌 A  
WHERE A.계좌번호 =:acc\_no  
AND A.최종거래일자 = (SELECT MAX(A.거래일자)  
FROM 거래내역 C  
WHERE A.거래일자 < :yyyymmdd  
AND C.계좌번호 = A.계좌번호)
- ④ SELECT C.이름, C.고객번호, A.거래후잔액  
FROM 거래내역 A  
INNER JOIN 계좌 B ON (A.계좌번호 = B.계좌번호)  
WHERE A.거래일자 = :yyyymmdd  
AND B.고객번호 IN (SELECT C.고객번호  
FROM 고객 C  
WHERE C.고객번호 = :cust\_no )

(정답) 4

(배점) 3

(난이도) 중

(문제유형) 객관식

(해설) 1 번:계좌번호 컬럼의 SUBSTR 사용으로 인덱스 효율 떨어짐 2 번: 거래일자 부분이 case 로 처리되어 올바르게 인덱스 액세스하기가 어려워짐 3 번 : 거래일자 부분이 < 로 되면서 인덱스 사용 효율 떨어짐  
(대모듈 > 소모듈) SQL성능 >인덱스

수고하셨습니다.



Copyright © 2017 by LG CNS All rights reserved.