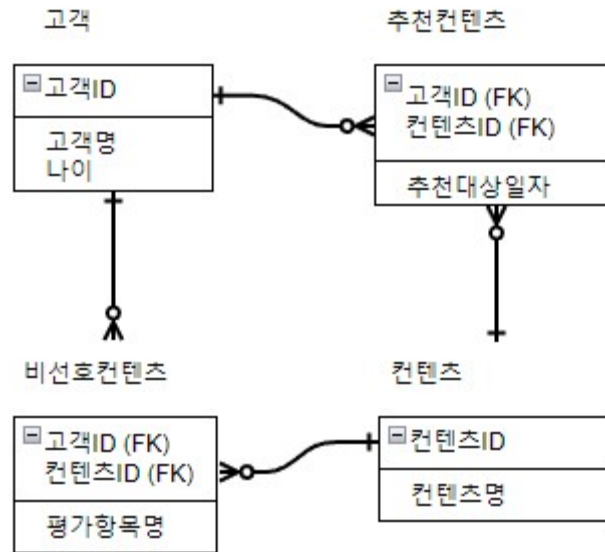


[66] 다음 중 아래 데이터 모델을 참고하여 설명에 맞게 올바르게 작성한 SQL문장을 2개 고르시오

[데이터 모델]



[설명]

우리는 매일 배치작업을 통하여 고객에게 추천할 컨텐츠를 생성하고 고객에게 추천서비스를 제공한다.

추천 컨텐츠가 엔티티에서 언제 추천을 해야 하는지를 정의하는 추천대상일자가 있어 해당일자에만 컨텐츠를 추천해야 한다.

또한 고객이 컨텐츠를 추천 받았을 때 선호하는 컨텐츠가 아닌 경우에는 고객이 비선호 컨텐츠로 분류하여 더 이상 추천받기를 원하지 않는다. 그러므로 우리는 비선호 컨텐츠 엔티티에 등록된 데이터에 대해서는 추천을 수행하지 않아야 한다.

※ 배치 작업이란? 어떤 처리를 연속적으로 하는 것이 아니고 일정량씩 나누어 처리하는 경우 그 일정량을 배치(batch) 라고 한다.

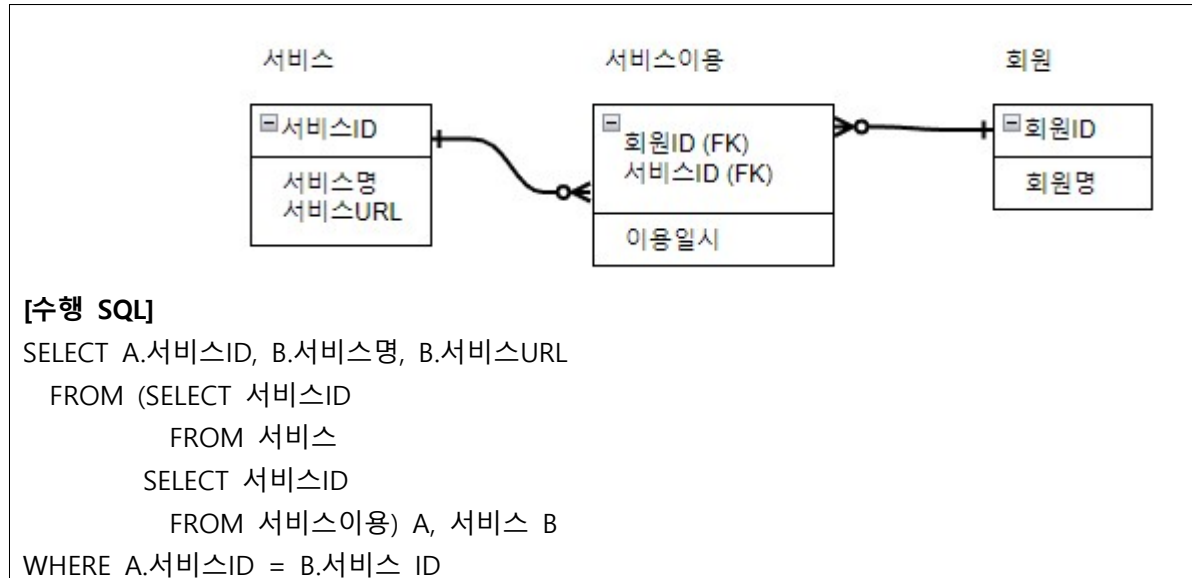
배치의 원뜻은 한 묶음이라는 의미다. [기계공학용어사전]

예) 상품을 주문하는 로직은 그 당시에 발생하는 트랜잭션에 대한 처리이므로 배치작업이라 표현하지는 않는다.

하지만 상품별 주문량을 집계하는 로직의 경우 특정 조건(기간등)으로 일괄처리를 해야함으로 배치작업이라 표현할 수 있다.

- ① SELECT C.컨텐츠ID, C.컨텐츠명
 FROM 고객 A
 INNER JOIN 추천콘텐츠 B
 ON (A.고객ID = B.고객ID)
 INNER JOIN 컨텐츠 C
 ON (B.컨텐츠ID = C.컨텐츠ID)
 WHERE A.고객ID = #custID#
 AND B.추천대상일자 = TO_CHAR(SYSDATE, 'YYYY.MM.DD')
 AND NOT EXISTS (SELECT X.컨텐츠ID
 FROM 비선호컨텐츠 X
 WHERE X.고객ID = B.고객ID);
- ② SELECT C.컨텐츠ID, C.컨텐츠명
 FROM 고객 A
 INNER JOIN 추천콘텐츠 B
 ON (A.고객ID = #custID# AND A.고객ID = B.고객ID)
 INNER JOIN 컨텐츠 C
 ON (B.컨텐츠ID = C.컨텐츠ID)
 RIGHT OUTER JOIN 비선호컨텐츠 D
 ON (B.고객ID = D.고객ID AND B.컨텐츠ID = D.컨텐츠ID)
 WHERE B.추천대상일자 = TO_CHAR(SYSDATE, 'YYYY.MM.DD')
 AND B.컨텐츠ID IS NOT NULL;
- ③ SELECT C.컨텐츠ID, C.컨텐츠명
 FROM 고객 A
 INNER JOIN 추천콘텐츠 B
 ON (A.고객ID = B.고객ID)
 INNER JOIN 컨텐츠 C
 ON (B.컨텐츠ID = C.컨텐츠ID)
 RIGHT OUTER JOIN 비선호컨텐츠 D
 ON (B.고객ID = D.고객ID AND B.컨텐츠ID = D.컨텐츠ID)
 WHERE A.고객ID = #custID#
 AND B.추천대상일자 = TO_CHAR(SYSDATE, 'YYYY.MM.DD')
 AND D.컨텐츠ID IS NULL;
- ④ SELECT C.컨텐츠ID, C.컨텐츠명
 FROM 고객 A
 INNER JOIN 추천콘텐츠 B
 ON (A.고객ID = #custID# AND A.고객ID = B.고객ID)
 INNER JOIN 추천콘텐츠 B
 ON (B.컨텐츠ID = C.컨텐츠ID)
 WHERE B.추천대상일자 = TO_CHAR(SYSDATE, 'YYYY.MM.DD')
 AND NOT EXISTS (SELECT X.컨텐츠ID
 FROM 비선호컨텐츠 X
 WHERE X.고객ID = B.고객ID
 AND X.컨텐츠ID = B.컨텐츠ID);

[80] 아래와 같은 데이터 모델에 대해 SQL을 수행하였다. 다음 중 수행된 SQL과 동일한 결과를 도출하는 SQL은?



- ① SELECT B.서비스ID, A.서비스명, A.서비스URL
FROM 서비스 A, 서비스이용B
WHERE A.서비스ID = B.서비스ID;
- ② SELECT X.서비스ID, X.서비스명, X.서비스URL
FROM 서비스X
WHERE NOT EXISTS (SELECT 1
FROM (SELECT 서비스ID
FROM 서비스
MINUS
SELECT 서비스ID
FROM 서비스이용) Y
WHERE X.서비스ID = Y.서비스ID)
- ③ SELECT B.서비스ID, A.서비스명, A.서비스URL
FROM 서비스 A
LEFT OUTER JOIN 서비스이용 B
ON(A.서비스ID = B.서비스ID)
WHERE B.서비스ID IS NULL
GROUP BY B.서비스ID, A.서비스명, A.서비스URL;
- ④ SELECT A.서비스ID, A.서비스명, A.서비스URL
FROM 서비스 A
WHERE 서비스ID IN (SELECT 서비스ID
FROM 서비스이용
MINUS
SELECT 서비스ID
FROM 서비스);

[93] 아래와 같이 일자별매출 테이블이 존재할 때 아래 결과처럼 일자별 누적매출액을 SQL로 구하려고 한다. WINDOW FUNCTION을 사용하지 않고 일자별 누적매출액을 구하는 SQL로 옳은 것은?

[테이블 : 일자별매출]		[결과 : 일자별 누적매출액]	
일자	매출액	일자	누적매출액
2015.11.01	1000	2015.11.01	1000
2015.11.02	1000	2015.11.02	2000
2015.11.03	1000	2015.11.03	3000
2015.11.04	1000	2015.11.04	4000
2015.11.05	1000	2015.11.05	5000
2015.11.06	1000	2015.11.06	6000
2015.11.07	1000	2015.11.07	7000
2015.11.08	1000	2015.11.08	8000
2015.11.09	1000	2015.11.09	9000
2015.11.10	1000	2015.11.10	10000

- ① SELECT A.일자, SUM(A.매출액) AS 누적매출액
FROM 일자별매출 A
GROUP BY A.일자
ORDER BY A.일자;
- ② SELECT B.일자, SUM(B.매출액) AS 누적매출액
FROM 일자별매출 A JOIN 일자별매출 B ON (A.일자 >= B.일자)
GROUP BY B.일자
ORDER BY B.일자;
- ③ SELECT A.일자, SUM(B.매출액) AS 누적매출액
FROM 일자별매출 A JOIN 일자별매출 B ON (A.일자 >= B.일자)
GROUP BY A.일자
ORDER BY A.일자;
- ④ SELECT A.일자
 , SELECT
 SUM(B.매출액)
 FROM 일자별매출 A JOIN 일자별매출 B ON (A.일자 >= B.일자) AS 누적매출액
FROM 일자별매출 A
GROUP BY A.일자
ORDER BY A.일자;

[101] 아래와 같은 데이터 모델에서 평가대상상품에 대한 품질평가항목별 최종 평가 결과를 추출하는 SQL문장으로 옳은 것은? (단, 평가항목에 대한 평가(평가등급)가 기대수준에 미치지 못할 경우 해당 평가항목에 대해서만 재평가를 수행한다.)



- ① SELECT B.상품ID, B.상품명, C.평가항목ID, C.평가항목명, A.평가회차, A.평가등급, A.평가일자
FROM 평가결과 A, 평가대상상품 B, 품질평가항목 C
 , (SELECT MAX(평가회차) AS 평가회차 FROM 평가결과) D
WHERE A.평가항목ID = C.평가항목ID
 AND A.상품ID = B.상품ID
 AND A.평가회차 = D.평가회차;
- ② SELECT B.상품ID, B.상품명, C.평가항목ID, C.평가항목명, A.평가회차, A.평가등급, A.평가일자
FROM 평가결과 A, 평가대상상품 B, 품질평가항목 C
WHERE A.상품ID = B.상품ID
 AND A.평가항목ID = C.평가항목ID
 AND A.평가회차 = (SELECT MAX(X.평가회차)
 FROM 평가결과 X
 WHERE X.상품ID = B.상품ID
 AND X.평가항목ID = C.평가항목ID);
- ③ SELECT B.상품ID, B.상품명, C.평가항목ID, C.평가항목명
 , MAX(A.평가회차) AS 평가회차
 , MAX(A.평가등급) AS 평가등급
 , MAX(A.평가일자) AS 평가일자
FROM 평가결과 A, 평가대상상품 B, 품질평가항목 C
WHERE A.상품ID = B.상품ID
 AND A.평가항목ID = C.평가항목ID
GROUP BY B.상품ID, B.상품명, C.평가항목ID, C.평가항목명;
- ④ SELECT B.상품ID, B.상품명, C.평가항목ID, C.평가항목명, A.평가회차, A.평가등급, A.평가일자
FROM (SELECT 상품ID, 평가항목ID
 , MAX(A.평가회차) AS 평가회차
 , MAX(A.평가등급) AS 평가등급
 , MAX(A.평가일자) AS 평가일자
FROM 평가결과
GROUP BY 상품ID, 평가항목ID) A, 평가대상상품 B, 품질평가항목 C
WHERE A.상품ID = B.상품ID
 AND A.평가항목ID = C.평가항목ID;

[102] 아래 부서 테이블의 담당자 변경을 위해 부서임시 테이블에 입력된 데이터를 활용하여 주기적으로 부서 테이블을 아래 결과와 같이 반영하기 위한 SQL으로 가장 적절한 것은? (단, 부서임시 테이블에서 변경일자를 기준으로 가장 최근에 변경된 데이터를 기준으로 부서 테이블에 반영되어야 한다.)

[테이블 : 부서]

부서코드(PK)	부서명	상위부서코드	담당자
A001	대표이사	NULL	김대표
A002	영업본부	A001	홍길동
A003	경영지원본부	A001	이순신
A004	마케팅본부	A001	강감찬
A005	해외영업팀	A002	이청용
A006	국내영업팀	A002	박지성
A007	총무팀	A003	차두리
A008	인사팀	A003	이민정
A009	해외마케팅팀	A004	이병헌
A010	국내마케팅팀	A004	차승원

[테이블 : 부서임시]

변경일자(PK)	부서코드(FK)	담당자
2014.01.23	A007	이달자
2015.01.25	A007	홍경민
2015.01.25	A008	유재석

[결과]

부서코드(PK)	부서명	상위부서코드	담당자
A001	대표이사	NULL	김대표
A002	영업본부	A001	홍길동
A003	경영지원본부	A001	이순신
A004	마케팅본부	A001	강감찬
A005	해외영업팀	A002	이청용
A006	국내영업팀	A002	박지성
A007	총무팀	A003	홍경민
A008	인사팀	A003	유재석
A009	해외마케팅팀	A004	이병헌
A010	국내마케팅팀	A004	차승원

① UPDATE 부서 A SET 담당자 = (SELECT C.부서코드
 FROM 부서임시 (SELECT 부서코드, MAX(변경일자 AS 변경일자
 FROM 부서임시
 GROUP BY 부서코드) B, 부서임시 C
 WHERE B.부서코드 = C.부서코드
 AND B.변경일자 = C.변경일자
 AND A.부서코드 = C.부서코드);

② UPDATE 부서 A SET 담당자 = (SELECT C.부서코드
 FROM 부서임시 (SELECT 부서코드, MAX(변경일자 AS 변경일자
 FROM 부서임시
 GROUP BY 부서코드) B, 부서임시 C
 WHERE B.부서코드 = C.부서코드
 AND B.변경일자 = C.변경일자
 AND A.부서코드 = C.부서코드)
 WHERE EXISTS (SELECT 1 FROM 부서 X WHERE A.부서코드 = X.부서코드);

③ UPDATE 부서 A SET 담당자 = (SELECT B.담당자
 FROM 부서임시 B, 부서임시 C
 WHERE B.부서코드 = C.부서코드
 AND B.변경일자 = (SELECT MAX(C.변경일자)
 FROM 부서임시 C
 WHERE C.부서코드 = B.부서코드))
 WHERE 부서코드 IN (SELECT 부서코드 FROM 부서임시);

④ UPDATE 부서 A SET 담당자 = (SELECT C.부서코드
 FROM 부서임시
 WHERE B.부서코드 = C.부서코드
 AND B.변경일자 = '2015.01.25');