1. **아래 Query를 Using조건 절과 Inner Join으로 각각 수정해보세요**

<Query>

SELELCT TEAM\_NAME, TEAM.STADIUM\_ID, STADIUM\_NAME

FROM TEAM JOIN STADIUM

ON TEAM.STADIUM\_ID = STADIUM.STADIUM\_ID

ORDER BY STADIUM\_ID;

<USING 조건 절>

< Inner JOIN>

1. **다음 SET 연산자에 대한 설명 중 틀린 것은 무엇인가?**  
   1) UNION 연산자는 조회 결과에 대한 합집합을 나타내며 자동으로 정렬을 해준다.   
   2) UNION ALL 연산자는 조회 결과를 정렬하고 중복되는 데이터를 한 번만 표현한다.  
   3) INTERSECT 연산자는 조회 결과에 대한 교집합을 의미한다.   
   4) EXCEPT 연산자는 조회 결과에 대한 차집합을 의미한다.
2. **일반적으로 FROM 절에 정의된 후 먼저 수행되어 SQL 문자 내에서 절차성을 주는 효과를 볼 수 있는 것은 어떤 유형의 서브쿼리 문장인가?**

1) SCALAR SUBQUERY

2) NESTED SUBQUERY

3) CORRELATED SUBQUERY

4) INLINE VIEW

1. **아래 ???? 은 누구일까요?**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SELECT LEVEL,**  **LPAD( ' ' , 4 \* (LEVEL-1) ) || e.ename emp,**  **( SELECT sq\_e.ename**  **FROM emp sq\_e**  **WHERE sq\_e.empno = e.mgr**  **) AS mgr ,**  **connect\_by\_isleaf isleaf**  **FROM emp e**  **START WITH e.empno = '7902'**  **CONNECT BY PRIOR e.mgr = e.empno**  **;**   |  |  |  |  | | --- | --- | --- | --- | | **LEVEL** | **EMP** | **MGR** | **ISLEAF** | | 1 | FORD | JONES | 0 | | 2 | JONES | KING | 0 | | 3 | ???? |  | 1 | |

1. **FORD**
2. **JONES**
3. **KING**
4. **(NULL)**
5. **2개의 테이블을 어떤 Join 으로 진행했을까요?**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **EMPNO** | **DEPTNO** | | 7902 | 20 | | 7934 | 10 | |  | 40 | | 8031 |  | |

1. Natural Join
2. Right Outer
3. Left Outer
4. Full Outer
5. **다음 중 옳지 않은 것은**
6. Union 과 Union All 은 성능차이가 없다.
7. Natural 조인에 해당하는 컬럼은 테이블의 컬럼명이 동일하다.
8. View 는 실제 데이터를 가지고 있지 않는다.
9. 서브쿼리는 Order by 절에 사용할 수 있다.
10. **아래 두 쿼리에 대한 설명으로 올바르지 못한 것은?**

**(단, ebom\_packing\_items는 각 사업장에 Active한 Item을 대상으로 포장 체적정보를 관리하는 table이다.)**

|  |
| --- |
| **SELECT msi.inventory\_item\_id, msi.organization\_id**  **FROM mtl\_system\_items\_b msi**  **WHERE 1=1**  **AND msi.organization\_id = 102**  **UNION**  **SELECT epi.inventory\_item\_id, epi.organization\_id**  **FROM ebom\_packing\_items epi**  **WHERE 1=1**  **AND epi.organization\_id = 102** |
| **SELECT msi.inventory\_item\_id, msi.organization\_id**  **FROM mtl\_system\_items\_b msi**  **WHERE 1=1**  **AND msi.organization\_id = 102**  **UNION ALL**  **SELECT epi.inventory\_item\_id, epi.organization\_id**  **FROM ebom\_packing\_items epi**  **WHERE 1=1**  **AND epi.organization\_id = 102** |

1. 두 쿼리의 실행 결과는 동일할 수도, 다를 수도 있다.
2. 만약 두 쿼리의 실행 결과가 동일하다면 아래 쿼리를 사용하는 것이 낫다.
3. 아래 쿼리는 중복된 행도 다 보여주기 때문에 위의 쿼리보다 수행 속도가 느리다
4. 위의 쿼리는 중복된 행은 하나의 행으로 보여준다.
5. 아래 쿼리는 단순히 2개 이상의 SQL문의 결과를 하나로 합친 것에 불과하다.
6. **다음 FROM 절의 JOIN 형태에 대한 설명 중 올바르지 못한 것은?**
7. INNER JOIN은 WHERE 절에서 사용하던 JOIN 조건을 FROM 절에서 정의하겠다는 표시이다.
8. INNER JOIN 사용 시, USING 조건절이나 ON 조건절을 반드시 사용해야 한다.
9. RIGHT OUTER JOIN, LEFT OUTER JOIN은 명칭의 차이일 뿐 수행 시 결과값은 동일하다.
10. RIGHT OUTER JOIN, LEFT OUTER JOIN에서 OUTER는 생략가능하다.
11. USING 조건절은 JOIN 하고자 하는 테이블 내의 컬럼이 동일해야만 사용할 수 있다.
12. **아래 쿼리는 수행 시 에러가 난다. 어떤 에러가 발생하며 이유는 무엇인가?**

|  |
| --- |
| **SELECT mp.organization\_id**  **,epi.inventory\_item\_id**  **,(SELECT msi.segment1**  **FROM mtl\_system\_items\_b msi**  **WHERE 1=1**  **AND msi.inventory\_item\_id =**  **epi.inventory\_item\_id)**  **,epi.volume**  **,epi.weight**  **,epi.length**  **,epi.width**  **,epi.height**  **FROM ebom\_packing\_items epi**  **,mtl\_parameters mp**  **WHERE 1=1**  **AND epi.organization\_id = mp.organization\_id** |

**( )**

1. **결과값을 적으시오.**

TAB1(key1) TAB2(key2)

B A

C B

D C

E

1. Select t1.key1, t2.key2

From tab1 t1 Left outer join tab2 t2

On (t1.key1 = t2.key2)

1. Select t1.key1, t2.key2

From tab1 t1 right outer join tab2 t2

On (t1.key1 = t2.key2)

1. **알맞은 집합 연산자는?**
2. 여러 개의 SQL 문의 결과에 대한 합집합으로 결과에서 모든 중복된 행은 하나의 행으로 만든다. ( )
3. 여러 개의 SQL 문의 결과에 대한 합집합으로 중복된 행도 그대로 결과로 표시된다. ( )
4. 앞의 SQL 결과에서 뒤의 SQL, 결과에 대한 차집합으로, 중복된 행은 하나의 행으로 만든다. ( )
5. **하기 결과가 나올 수 있는 쿼리를 작성하시오.**

직원 직속상사 직급 (테이블명: EMP)

철수 순희 대리

순희 영희 과장

길동 순희 대리

영희 개똥 부장

개똥 사장

(결과)

직원 직급

개똥 사장

영희 부장

순희 과장

길동 대리

철수 대리

1. **다음은 Scalar subquery를 이용한 질의 구문이다. OUTER JOIN을 이용하여 작성하시오.**

SELECT A.COURSE\_CODE, A.COURSE\_NAME,

(SELECT /\*+ INDEX\_DESC(B EC\_COURSE\_SQ\_PK) \*/ B.COURSE\_SQ\_NO

FROM EC\_COURSE\_SQ B

WHERE B.COURSE\_CODE = A.COURSE\_CODE

AND B.YEAR = ‘2000’

AND ROWNUM = 1) MAXSQ

FROM EC\_COURSE A

ORDER BY A.COURSE\_CODE, A.COURSE\_NAME;

1. **계층형 질의 구문에 대한 설명이다. 빈칸 안에 알맞은 개념을 작성하시오.**

( ) 은 계층 구조 전개의 시작 위치를 지정하는 구문이다. 즉, 루트 데이터를 저장한다.

( ) 은 다음에 전개될 자식 데이터를 지정하는 구문이다.

( ) 데이터를 전개하면서 이미 나타났던 동일한 데이터가 전개 중에 다시 나타남을 막기 위해 이 옵션을 추가한다.

1. **다음과 같은 질의에 쓰인 조인방법을 무엇이라고 하나요?**

|  |
| --- |
| EX) 인사정보 테이블- 사원코드, 사원이름, 조직코드  조직테이블-  조직코드, 조직명, 조직장번호  각 사원별 조직장 정보 출력하는 쿼리 |

SELECT A1.사원코드

,A1.사원이름

,A1.조직코드

,B.조직명

,B.조직장번호

,A2.사원이름 AS 조직장이름

FROM 인사정보 A1

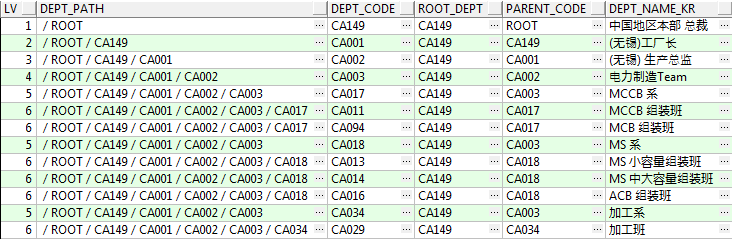
,인사정보 A2

       , 조직정보 B

WHERE A1.조직코드 = B.조직코드

          AND B.조직장번호 = A2.사원코드;

1. **다음 중 SELECT절에 사용하는 서브쿼리인 스칼라 서브쿼리(Scalar Subquery)에 대한 설명으로 가장 적절한 것을 2개 고르시오.**
2. 내부적으로 Unique 작업을 실행한다.
3. 하나의 로우에 해당하는 결과 건수는 1건 이하여야 한다.
4. Min 또는 Max 함수를 사용해야 한다.
5. 결과 칼럼의 개수는 1개여야 한다.
6. **다음 중 서브쿼리에 대한 설명으로 가장 부적절한 것은?**
7. 메인 쿼리는 조직 테이블(1)이고, 서브쿼리는 사원 테이블(M)을 사용할 경우, 질의 결과는 M레벨인 사원 레벨로 나온다.
8. 메인쿼리에서 서브쿼리 칼럼을 사용할 수 없다.
9. 서브쿼리에서 메인쿼리 칼럼은 사용할 수 있다.
10. 서브쿼리의 연결 연산자로서 ‘=’을 사용할 경우, 서브쿼리 파트의 결과는 1건 이하여야 한다.
11. 아래 쿼리 결과를 보고, 빈칸을 채우시오.



with t1 as

(select ffv.flex\_value as dept\_code,

decode(ffv.attribute32, '\*\*\*\*\*', 'ROOT', ffv.attribute32) as parent\_code,

ffv.attribute33,

rank() over(order by ffv.attribute33) as display\_order,

fft.description as dept\_name\_kr

from fnd\_flex\_value\_sets ffvs,

fnd\_flex\_values ffv,

fnd\_flex\_values\_tl fft

where ffvs.flex\_value\_set\_name in ('LSIS\_WX\_GL\_COSTCENTER', 'LSIS\_DL\_GL\_COSTCENTER')

and ffvs.flex\_value\_set\_id = ffv.flex\_value\_set\_id

and ffv.summary\_flag = 'N'

and ffv.flex\_value like 'C%'

and ffv.flex\_value\_id = fft.flex\_value\_id

and fft.language = 'US'

and fft.description not like '%不用%'

and fft.description not like '(WX)%'

and ffv.attribute32 is not null

)

select ( ) lv,

( ) as dept\_path,

t1.dept\_code,

( ) as root\_dept,

t1.parent\_code,

t1.dept\_name\_kr

from t1

where 1 = 1

start with t1.dept\_code = 'CA149'

( ) t1.dept\_code = ( ) t1.parent\_code

1. 1) USING조건 절

SELELCT TEAM\_NAME, STADIUM\_ID, STADIUM\_NAME

FROM TEAM JOIN STADIUM

USING STADIUM\_ID

ORDER BY STADIUM\_ID;

2) INNER JOIN

SELELCT TEAM\_NAME, TEAM.STADIUM\_ID, STADIUM\_NAME

FROM TEAM ,STADIUM

WHERE TEAM.STADIUM\_ID = STADIUM.STADIUM\_ID

ORDER BY STADIUM\_ID;

2. 2

3. 4

4. 3

5. 4

6. 1

7. 3

8. 3

9. single-row subquery returns more than one row. Organization\_id   
 조건을 걸지 않아 여러 org의 segment1 값이 반환된다.

10 B-B, C-C, D-null, E-null/

null-A, B-B, C-C

11. UNION, UNION ALL, EXCEPT(MINUS)

12. select EMP.직원, EMP.직급

from EMP

start with EMP.직원 = '개똥'

connect by EMP.직속상사 = prior EMP.직원

13. 생략.

14. 1) Start with  
 2) Connect\_by  
 3) no\_cycle

15. 셀프조인

16. 2, 4

17. 1

18. 1) LEVEL  
2) sys\_connect\_by\_path(t1.parent\_code, ' / ')

3) connect\_by\_root t1.dept\_code

4) connect by prior

5) N/A