Study Room Doc.02 : SQLD 예상문제

네이버 Cafe : 데이터베이스 전문가 포럼 Study Room

http://cafe.naver.com/sqlpd

SQLD 21 회 기출문제를 바탕으로 작성

작성자 : 월야루

2016-09-04

객관식

1. 아래의 쿼리를 만족하는 결과를 가장 잘 설명한 것은?

```
SELECT A.*
FROM HR.EMPLOYEES A,
        HR.EMPLOYEES B
WHERE 1=1
AND A.MANAGER_ID = B.EMPLOYEE_ID
AND B.SALARY >= ANY A.SALARY;
```

1) 어떤 부하 직원보다도 연봉이 높은 상사
2) 어떤 부하 직원보다도 연봉이 낮은 상사
3) 어떤 상사 보다도 연봉이 높은 부하 직원
4) 어떤 상사 보다도 연봉이 낮은 부하 직원

2. 아래의 쿼리의 결과를 만족하는 결과로 가장 알맞은 것은?

```
DEPARTMENT_ID : NULL, 10,20,30,40,50,~90,100,110

SELECT DISTINCT DEPARTMENT_ID
FROM HR.EMPLOYEES A
WHERE A.DEPARTMENT_ID <= ALL (30,50);
```

1) 10,20
2) 10,20,30
3) 10,20,30,40
4) 10,20,30,40,50

3. 아래와 같은 테이블에 데이터가 있다. 각 SQL 에 대한 결과값이 잘못된 것은?

TABLE SQLD_21_01

```
N1         V1
---------- ----------
         1 A
         2
         3 B
         4 C
```

TABLE SQLD_21_02

```
N1         V1
---------- ----------
         1 A
         2
         3 B
```

1) SELECT * FROM SQLD_21_01
   WHERE V1 IN (SELECT V1 FROM SQLD_21_02);

```
N1         V1
---------- ----------
         1 A
         3 B
```

2) SELECT * FROM SQLD_21_01
   WHERE V1 NOT IN (SELECT V1 FROM SQLD_21_02);

```
N1         V1
---------- ----------
         4 C
```

3) SELECT * FROM SQLD_21_01 A
   WHERE EXISTS (SELECT 'X' FROM SQLD_21_02 B
                 WHERE A.V1 = B.V1);

```
N1         V1
---------- ----------
         1 A
         3 B
```

4) SELECT * FROM SQLD_21_01 A
   WHERE NOT EXISTS (SELECT 'X' FROM SQLD_21_02 B
                     WHERE A.V1 = B.V1);

```
N1         V1
---------- ----------
         2
         4 C
```

4. 데이터 모델링에 대한 아래 보기 설명 중 알맞은 것은?

1) 논리 모델링의 외래키는 물리 모델에서 반드시 구현되지는 않는다
2) 실제로 데이터베이스를 구축할 때 참고되는 모델은 개념적 데이터 모델링이다
3) 물리 모델링 -> 논리 모델링 -> 개념 모델링 단계로 갈수록 구체적이다
4) 데이터 모델링의 3가지 요소는 Process, Attributes, Relationship 이다

5. 데이터 모델링에 대한 단계 중 아래에서 설명하는 단계는 어떤 단계의 모델링인가?

추상화 수준이 높고 업무중심적이고 포괄적인 수준의 모델링 진행. 전사적 데이터 모델링,
EA 수립 시 많이 이용됨

1) 개념적 데이터 모델링
2) 논리적 데이터 모델링
3) 물리적 데이터 모델링
4) 추상적 데이터 모델링

6. 엔터티 - 인스턴스 - 속성 - 속성값에 대한 관계 설명중 틀린 것을 고르시오

1) 한 개의 엔터티는 두 개 이상의 인스턴스의 집합이어야 한다
2) 한 개의 엔터티는 두 개 이상의 속성을 갖는다
3) 하나의 속성은 하나 이상의 속성값을 가진다
4) 하나의 엔터티의 인스터스는 다른 엔터티의 인스턴스간의 관계인 Paring 을 가진다

7. 학생관련 정보를 조회하는 SQL 을 작성할려고 한다. 조회하는 사람은 주로 학생 본인이 학번으로 조회를 주로 한다. 이런 SQL 일 때 성능을 개선하는 방법으로 가장 알맞은 것은?

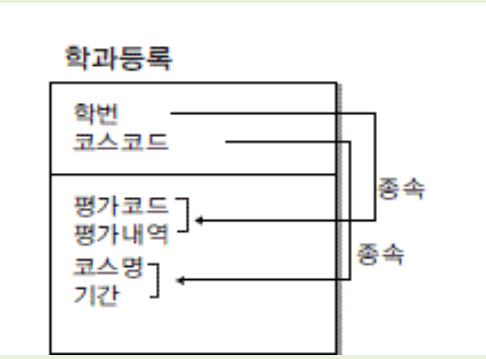
1) 학교명을 선두컬럼으로 하는 INDEX 를 생성한다
2) 학번을 선두컬럼으로 하는 INDEX 를 생성한다
3) 학교명 + 학번순으로 구성된 INDEX 를 생성한다
4) 학교명 + 이름 + 학번으로 구성된 INDEX 를 생성한다

8. 아래의 SQL 에서 FUNCTION 자리에 쓰인 함수에 의한 결과값이 다른 하나는?

SELECT function(3.46) FROM DUAL;

1) TRUNC
2) CEIL
3) FLOOR
4) ROUND

9. 아래의 ERD 에서 3차 정규형을 만족하게 할 때 엔터티의 개수는 몇개가 되는가?



ㄱ. 평가코드,평가내역은 학번에 종속적
ㄴ. 코스명,기간은 코스코드에 종속적
ㄷ. 평가코드 평가내역은 속성간 종속적 관계

-> 1차 정규형 : 모든 속성은 반드시 하나의 값. 속성값의 중복 제거
-> 2차 정규형 : 식별자에 종속되지 않는 속성의 중복 제거
-> 3차 정규형 : 2차 정규형 만족 + 식별자 외 일반 컬럼간의 종속 존재 제거

1) 1개
2) 2개
3) 3개
4) 4개

10. 아래 쿼리중 결과값이 다른 하나는?

1)
```
SELECT DNAME,JOB,
       COUNT(*) "Total Empl",
       SUM(SAL) "Total Sal"
FROM SCOTT.EMP  A,
     SCOTT.DEPT B
WHERE A.DEPTNO = B.DEPTNO
GROUP BY ROLLUP(DNAME,JOB)
ORDER BY DNAME,JOB;
```

2)
```
SELECT DNAME,JOB,
       COUNT(*) "Total Empl",
       SUM(SAL) "Total Sal"
FROM SCOTT.EMP  A,
     SCOTT.DEPT B
WHERE A.DEPTNO = B.DEPTNO
GROUP BY GROUPING SETS( (DNAME,JOB),DNAME,NULL)
ORDER BY DNAME,JOB;
```

3)
```
SELECT DNAME,JOB,
       COUNT(*) "Total Empl",
       SUM(SAL) "Total Sal"
FROM SCOTT.EMP  A,
     SCOTT.DEPT B
WHERE A.DEPTNO = B.DEPTNO
GROUP BY DNAME,JOB
UNION ALL
SELECT DNAME,'' AS JOB,
       COUNT(*) "Total Empl",
       SUM(SAL) "Total Sal"
FROM SCOTT.EMP  A,
     SCOTT.DEPT B
WHERE A.DEPTNO = B.DEPTNO
GROUP BY DNAME
UNION ALL
SELECT '' AS DNAME,'' AS JOB,
       COUNT(*) "Total Empl",
       SUM(SAL) "Total Sal"
FROM SCOTT.EMP  A,
     SCOTT.DEPT B
WHERE A.DEPTNO = B.DEPTNO
ORDER BY 1,2;
```

4)
```
SELECT DNAME,JOB,
       COUNT(*) "Total Empl",
       SUM(SAL) "Total Sal"
FROM SCOTT.EMP  A,
     SCOTT.DEPT B
WHERE A.DEPTNO = B.DEPTNO
GROUP BY CUBE(DNAME,JOB)
ORDER BY DNAME,JOB;
```

11. 다음의 SQL 을 표준 ANSI SQL 로 알맞게 바꾼것은?
단, 조인 조건과 조회 조건은 분리한다.

```
SELECT *
FROM SCOTT.EMP A,
       SCOTT.DEPT B
WHERE A.DEPTNO = B.DEPTNO
AND B.DNAME = 'SALES'
```

1)
```
SELECT *
FROM SCOTT.EMP A LEFT OUTER JOIN SCOTT.DEPT B
    ON ( A.DEPTNO = B.DEPTNO
        AND B.DNAME ='SALES')
WHERE 1=1;
```

2)
```
SELECT *
FROM SCOTT.EMP A RIGHT OUTER JOIN SCOTT.DEPT B
    ON ( A.DEPTNO = B.DEPTNO
        AND B.DNAME ='SALES')
WHERE 1=1;
```

3)
```
SELECT *
FROM SCOTT.EMP A INNER JOIN SCOTT.DEPT B
    ON ( A.DEPTNO = B.DEPTNO
        AND B.DNAME ='SALES')
WHERE 1=1;
```

4)
```
SELECT *
FROM SCOTT.EMP A INNER JOIN SCOTT.DEPT B
    ON A.DEPTNO = B.DEPTNO
WHERE 1=1
AND B.DNAME ='SALES';
```

12. 아래 VIEW 에 대한 설명 중 가장 올바르지 않은 것은?

1) 독립성 : 테이블 구조가 변경되어도 뷰를 사용하는 응용 프로그램은 변경하지 않아도 된다.
2) 편리성 : 복잡한 질의를 뷰로 생성함으로써 관련 질의를 단순하게 작성할 수 있다.
또한, 해당 형태의 SQL문을 자주 사용할 때 뷰를 이용하면 편리하게 사용할 수 있다.
3) 물리성 : 실제 데이터를 가지고 있어서 물리적인 관리가 가능하다
4) 보안성 : 직원의 급여정보와 같이 숨기고 싶은 정보가 존재한다면 , 뷰를 생성할 때 해당 칼럼을 빼고 생성함으로써 사용자에게 정보를 감출 수 있다

13. 보기의 테이블 TAB_A, TAB_B 에 INSERT 를 한 결과로 알맞은 것은?

TAB_A	A IDENTITY (1,1)	***TAB_B***	A CHECK (A < 5)
(Sql Server)	B VARCHAR2(10)	(Oracle)	B VARCHAR2(10)

```
INSERT INTO TAB_A(A,B) VALUES(1,'A');
INSERT INTO TAB_A(B) VALUES('B');
INSERT INTO TAB_A(B) VALUES('D');
```

```
INSERT INTO TAB_B VALUES(1,'A');
INSERT INTO TAB_B VALUES(2,'B');
INSERT INTO TAB_B VALUES(6,'D');
INSERT INTO TAB_B VALUES(NULL,'X');
```

1) TAB_A	TAB_B
1,B	1,A
2,D	2,B
	NULL,X

2) TAB_A	TAB_B
1,A	1,A
2,B	2,B
3,D	6,D

3) TAB_A	TAB_B
1,A	1,A
	2,B

4) TAB_A	TAB_B
1,B	1,A
2,D	2,B

14. 조인에 대한 설명 중 Hash Join 에 대한 특성으로 부적절한 것 2개를 고르시오

1) 각 테이블에 INDEX 가 반드시 필요한 것은 아니다
2) 일반적으로 작은 테이블을 MEMORY 에 올리는 선행 테이블로 사용한다
3) Non Equal Join 이 가능하다 (비동등)
4) 조인을 위해 사전 소트 작업이 필요하다

15. 아래의 실행 계획을 올바르게 설명한 것은?

```
------------------------------------------------------------
  0      SELECT STATEMENT Optimizer=ALL_ROWS (Cost=7 Card=9 Bytes=1K)
  1    0   HASH JOIN (Cost=7 Card=9 Bytes=1K)
  2    1     TABLE ACCESS (FULL) OF 'SCOTT.DEPT' (TABLE) (Cost=3 Card=1 Bytes=30)
  3    1     VIEW (Cost=3 Card=9 Bytes=783)
  4    3       COUNT (STOPKEY)
  5    4         TABLE ACCESS (FULL) OF 'SCOTT.EMP' (TABLE) (Cost=3 Card=14 Bytes=1K)
------------------------------------------------------------

Predicate information (identified by operation id):
------------------------------------------------------------
  1 - access("A"."DEPTNO"="B"."DEPTNO")
  2 - filter("B"."DNAME"='SALES')
  4 - filter(ROWNUM<10)
------------------------------------------------------------
```

1) EMP TABLE 에 대한 행제한 구문이 있다
2) EMP TABLE 과 DEPT TABLE 은 OUTER JOIN 으로 수행되고 있다
3) EMP TABLE 과 DEPT TABLE 에서 선행 테이블은 EMP TABLE 이다
4) DEPT TABLE 은 별도의 조건이 없어 FULL SCAN 을 하고 있다

16. 아래의 SQL 에 대해서 실행 순서를 올바르게 나열한 것은?

```
SELECT DEPTNO, COUNT(EMPNO)
FROM SCOTT.EMP
WHERE SAL >= 500
GROUP BY DEPTNO
HAVING COUNT(EMPNO) > 2
ORDER BY DEPTNO;
```

1) FROM -> WHERE -> GROUP BY -> HAVING -> SELECT -> ORDER BY
2) FROM -> WHERE -> GROUP BY -> HAVING -> ORDER BY -> SELECT
3) FROM -> WHERE -> HAVING -> GROUP BY -> SELECT -> ORDER BY
4) FROM -> WHERE -> GROUP BY -> SELECT -> HAVING -> ORDER BY

17. 아래와 같은 컬럼으로 구성된 테이블에 COL1 을 구성컬럼으로 가지는 인덱스가 있다.
가장 효율적으로 해당 인덱스를 사용할 수 있는 조건절은?

TAB_A	COL1 NUMBER COL2 VARCHAR2(10)	***INDEX***	COL1

1) WHERE COL1 LIKE '2%'
2) WHERE COL1 = 10
3) WHERE COL1 IS NOT NULL
4) WHERE COL1 <> 10

18. 비교연산자의 어느 한쪽이 VARCHAR 유형 타입인 경우 문자 우형 비교에 대한 설명 중 가장 알맞지 않은 것은?

1) 서로 다른 문자가 나올 때까지 비교한다
2) 길이가 다르다면 짧은 것이 끝날 때까지만 비교한 후에 길이가 긴 것이 크다고 판단한다
3) 길이가 같고 다른 것이 없다면 같다고 판단한다
4) 길이가 다르다면 작은 쪽에 SPACE 를 추가하여 길이를 같게 한 후에 비교한다

19. 아래의 SQL 에 대해서 결과값이 다른 것은?

1) SELECT CONCAT ('RDBMS', ' SQL') FROM DUAL;
2) SELECT 'RDMBS' || ' SQL' FROM DUAL;
3) SELECT 'RDBMS' + ' SQL' FROM DUAL;
4) SELECT 'RDBMS' & ' SQL' FROM DUAL;

20. 아래의 ORACLE SQL 을 SQL SERVER SQL 로 전환한 것중 가장 알맞은 것은?

```
SELECT ENAME, SAL
FROM (SELECT ENAME, SAL
        FROM SCOTT.EMP
        ORDER BY SAL DESC)
WHERE ROWNUM < 4 ;
```

1)
```
SELECT TOP(4) ENAME,SAL
FROM SCOTT.EMP
ORDER BY SAL DESC
```

2)
```
SELECT TOP(3) ENAME,SAL
FROM SCOTT.EMP
ORDER BY SAL DESC
```

3)
```
SELECT TOP(4) WITH TIES  ENAME,SAL
FROM SCOTT.EMP
ORDER BY SAL DESC
```

4)
```
SELECT TOP(3) WITH TIES  ENAME,SAL
FROM SCOTT.EMP
ORDER BY SAL DESC
```

21. 아래의 ANSI JOIN SQL 에서 가장 올바르지 않은 것은?

1)
```
SELECT EMP.DEPTNO, EMPNO, ENAME, DNAME
FROM EMP INNER JOIN DEPT
        ON EMP.DEPTNO = DEPT.DEPTNO;
```

2)
```
SELECT EMP.DEPTNO, EMPNO, ENAME, DNAME
FROM EMP NATURAL JOIN DEPT;
```

3)
```
SELECT *
FROM DEPT JOIN DEPT_TEMP
USING (DEPTNO);
```

4)
```
SELECT E.EMPNO, E.ENAME, E.DEPTNO, D.DNAME
FROM EMP E INNER JOIN DEPT D
        ON (E.DEPTNO = D.DEPTNO);
```

22. 다음주 엔터티의 종류가 아닌 것은?

1) 교수
2) 학생
3) 청약자
4) 수강

23. 아래의 계층형 SQL 에서 리프 데이터이면 1, 그렇지 않으면 0 을 출력하고 싶을 때 사용하는 키워드로 알맞은 것은?

```
SELECT LEVEL, LPAD(' ',4 * (LEVEL -1) ) || EMPNO,
        MGR, (          ) AS ISLEAF
FROM SCOTT.EMP
START WITH MGR IS NULL
CONNECT BY PRIOR EMPNO = MGR;
```

1) CONNECT_BY_ISLEAF
2) CONNECT_BY_ISCYCLE
3) SYS_CONNECT_BY_PATH
4) CONNECT_BY_ROOT

24. 아래와 같은 테이블 TAB1, TAB2 가 있을 때 아래의 SQL의 결과 건수를 알맞게 나열한 것은?

TAB1	COL1	COL2	KEY1
	BBB	123	B
	DDD	222	C
	EEE	233	D
	FFF	143	E

TAB2	KEY2	COL1	COL2
	A	10	BC
	B	10	CD
	C	10	DE

```
SELECT * FROM TAB1 A INNER JOIN TAB2 B ON (A.KEY1 = B.KEY2)
SELECT * FROM TAB1 A LEFT OUTER JOIN TAB2 B ON (A.KEY1 = B.KEY2)
SELECT * FROM TAB1 A RIGHT OUTER JOIN TAB2 B ON (A.KEY1 = B.KEY2)
SELECT * FROM TAB1 A FULL OUTER JOIN TAB2 B ON (A.KEY1 = B.KEY2)
SELECT * FROM TAB1 A CROSS JOIN TAB2 B
```

1) 2, 4, 3, 5, 12
2) 2, 4, 5, 3, 12
3) 2, 3, 4, 5, 12
4) 2, 4, 3, 7, 12

25. 아래의 WINDOW FUNCTION 을 사용한 SQL 중 가장 올바르지 않은 것은?

1) SUM(SAL) OVER()
2) SUM(SAL) OVER(PARTITION BY JOB ORDER BY EMPNO
   RANGE BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING) SAL1
3) SUM(SAL) OVER(PARTITION BY JOB ORDER BY JOB
   RANGE BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) SAL2
4) SUM(SAL) OVER(PARTITION BY JOB ORDER BY EMPNO
   RANGE BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED PRECEDING) SAL3

26. SQL 구문에서 FROM 절에 대한 설명 중 가장 올바르지 않은 것은?

1) FROM 절에 ALIAS 를 쓰기 위해서 AS 키워드 사용이 가능하다
2) FROM 은 가장 먼저 수행된다
3) FROM 절에 사용되는 subquery 를 보통 inline view 라고 한다
4) FROM 절은 SELECT 와 항상 짝을 이룬다

27. 아래의 SQL 의 결과로 알맞은 것은?

TABLE SQLD_21_01

```
N1         V1
---------- ----------
         1 A
         2
         3 B
         4 C
```

TABLE SQLD_21_02

```
N1         V1
---------- ----------
         1 A
         2
         3 B
```

```
SELECT SUM(A.N1)
FROM SQLD_21_01 A,
        SQLD_21_02 B
WHERE A.V1 <> B.V1;
```

1) 10
2) 30
3) 12
4) 8

28. 서브쿼리에 대한 설명 중 가장 올바르지 않은 것은?

1) 서브쿼리는 괄호로 감싸서 사용한다
2) 서브쿼리는 비교 연산자와 함께 사용 가능하다
3) 메인쿼리는 서브쿼리의 컬럼을 쓸수 없다
4) 서브쿼리는 SELECT 절, FROM 절, WHERE 절등에서 사용 가능하다

29. 유저와 권한 중 권한에 대한 설명 중 가장 올바르지 않은 것은?

1) 사용자가 실행하는 모든 DDL 문장은 그에 해당하는 적절한 권한이 있어야만 문장을 실행 할 수 있다.
2) DBA 권한을 가진 유저만이 권한을 부여 할 수 있다
3) 테이블의 소유자는 해당 테이블의 DML 권한을 다른 유저에게 부여 할 수 있다.
4) 권한 부여를 편리하게 관리하기 위해 만들어진 권한의 집합인 ROLE 이 있다

단답형

1. 아래와 같은 테이블이 있을때 아래와 같은 결과가 나오기 위한 주어진 SQL 구문을 완성하시오

TABLE SQLD_21_01

N1	V1
1	a
2	
3	b
4	c

TABLE SQLD_21_02

N1	V1
1	A
2	
3	B

```
SELECT A.*
FROM SQLD_21_01 A,
     SQLD_21_02 B
WHERE (    ) (A.V1) LIKE B.V1||'%'
```

RESULT>

N1	V1
1	a
3	b

2. EMP 테이블은 사원과 메니저의 정보를 담은 계층형 데이터를 포함한 테이블이다.
메니저부터 사원까지 결제 단계가 가장 많은 레벨을 구할려고 할때 빈칸을 완성하시오

```
SELECT (      )
FROM SCOTT.EMP
START WITH MGR IS NULL
CONNECT BY PRIOR EMPNO = MGR;
```

3. 아래의 NOT EXISTS 구문을 동일한 결과를 출력하는 SQL 로 변경할 때 빈칸을 완성하시오

```
SELECT ...
FROM 급여이력 S
WHERE NOT EXISTS (SELECT 'X'
                  FROM 사원 P
                  WHERE P.사원번호 = S.사원번호)

SELECT ....
FROM 급여이력 S LEFT OUTER JOIN 사원 P
    ON (S.사원번호 = P.사원번호)
WHERE (      )
```

4. 아래 SQL 의 출력되는 ROWS 의 개수를 구하시오

EMP TABLE	DEPTNO	JOB	SAL
	20	CLERK	800
	30	SALESMAN	1600
	30	SALESMAN	1250
	20	MANAGER	2975
	30	SALESMAN	1250
	30	MANAGER	2850
	10	MANAGER	2450
	20	ANALYST	3000
	10	PRESIDENT	5000
	30	SALESMAN	1500
	20	CLERK	1100
	30	CLERK	950
	20	ANALYST	3000
	10	CLERK	1300

DEPT TABLE	DEPTNO	DNAME
	10	ACCOUNTING
	20	RESEARCH
	30	SALES
	40	OPERATIONS

```
SELECT DNAME,JOB,
       COUNT(*) "Total Emp",
       SUM(SAL) "Total Sal"
FROM SCOTT.EMP  A,
     SCOTT.DEPT B
WHERE A.DEPTNO = B.DEPTNO
GROUP BY CUBE(DNAME,JOB)
```

5. 아래와 같은 SQL 이 있을 때 조건절을 넣기 위한 키워드는 무엇인지 작성하시오

```
SELECT *
FROM EMP
(      ) EMPID = 10;
```

6. 아래의 SQL 의 결과로 나오는 ROWS 의 수는?

TAB1	COL1
	1
	2
	3

TAB2	COL2
	1
	2
	4

```
SELECT *
FROM TAB1 A, TAB2 B
WHERE A.COL1 <> B.COL1;
```

7. 아래의 SQL 의 출력 결과를 작성하시오.

TAB1	COL1	COL2	*TAB2*	COL1	COL2
	Z	10		Y	1
	Y	20		Y	2
	X	30		Y	3

```
SELECT COUNT(*)
FROM TAB1
WHERE EXISTS (SELECT 1 FROM TAB2 WHERE TAB2.COL1 = 'X');
```