

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

1번 풀이

사전 지식

* 비교연산자 ANY (서브쿼리)

- 서브쿼리의 결과에 존재하는 어느 하나의 값이라도 만족하는 조건을 의미.
비교연산자로 ">" > "<" 를 사용했다면 메인쿼리는 서브쿼리의 값들 중 어떤 값이라도 만족하면 되므로 서브쿼리의 결과의 최소값보다 큰 모든 것이 조건을 만족
- SOME 과 동일

1번 해답 : 1번

A.MANAGER_ID = B.EMPLOYEE_ID 이므로 A 테이블의 매니저 아이디가 B 테이블의 직원 아이디 이므로 A 가 직원, B 가 상사라고 볼수 있음

```
SELECT A.EMPLOYEE_ID AS "부하직원", A.LAST_NAME, A.MANAGER_ID "상사",  
A.SALARY AS "부하연봉",  
B.SALARY AS "상사연봉"  
FROM HR.EMPLOYEES A, -- EMPLOYEE SALARY  
      HR.EMPLOYEES B -- MANAGER SALARY  
WHERE 1=1  
AND A.MANAGER_ID = B.EMPLOYEE_ID  
AND B.SALARY >= ANY A.SALARY;
```

이므로 상사의 연봉이 어떤 부하직원보다도 큰 결과를 뽑는 sql임

2번 풀이

사전 지식

* 비교연산자 ALL (서브쿼리)

- 서브쿼리의 결과에 존재하는 모든값을 만족하는 조건을 의미.
- 비교연산자로 ">" > "<" 를 사용했다면 메인쿼리는 서브쿼리의 모든 값을
- 만족해야 하므로 서브쿼리의 결과의 최대값보다 큰 모든 것이 조건을 만족

2번 해답 : 2번

A.DEPARTMENT_ID <= ALL(30,50) 이므로 30 보다 작고, 50 보다 작아야 함
즉, 30보다 작거나 같은 값이 모두 해당됨

3번 풀이

사전 지식

NOT IN 의 경우 조건절이 아래와 같이 변형되어 수행됨

LNNVL(V1 <> :B1) -----> NULL 인 경우 Unkown 으로 True Return. --> 조건절 만족, Not Exists 를 만족하지 않음

HASH AN 으로 풀릴 경우 A.V1 = B.V1 으로 ACCESS 됨

즉, NOT 조건이므로 NULL 연산이 Unknow 으로 처리되 True 로 반환되는 현상이 나타남

3번 해답 : 2번

SQLD_21_02 의 V1 에 NULL 이 존재하므로 NOT IN 경우 NULL 로 인해 모든 조건이 참이 되어
NOT IN 결과 0건이 RETURN 됨

4번 풀이

4번 해답 : 1번

논리 모델링의 외래키는 물리 모델에서 반드시 구현되지 않음. (선택 사항임)

실제 데이터베이스 구축 시 참고되는 모델은 물리적 데이터 모델링이다

개념 모델링에서 물리 모델링으로 가면서 더 구체적이며 개념 모델링이 가장 추상적이다

데이터 모델링의 3요소는 Thing, Attributes, Relationship 이다

5번 풀이

5번 해답 : 1번

6번 풀이

6번 해답 : 3번

하나의 속성은 하나의 속성값을 가지며 하나 이상의 속성값을 가지는 경우 정규화가 필요함

7번 풀이

7번 해답 : 2번

학생 본인이 학번으로 주로 조회를 하기 때문에 학번이 선두 컬럼으로 있는 인덱스가 가장 적절함

학교명이 앞으로 올 경우 반드시 학교명을 적어야 하는 부담이 있음

해당 문제는 SQLD 보다는 SQLP 에 가까운 튜닝 문제임

8번 풀이

사전 지식

* FLOOR : 숫자보다 작거나 같은 최대 정수를 리턴

* CEIL/CEILING : 숫자보다 크거나 같은 최소 정수를 리턴

* TRUNC : 숫자를 소수 m 자리에서 잘라서 리턴 (m default : 0)

* ROUND : 숫자를 소수 m 자리에서 반올림하여 리턴 (m default : 0)

8번 해답 : 2번

TRUNC(3.46) = 3, FLOOR(3.46) = 3 , ROUND(3.46)=3

CEIL(3.46) = 4

9번 풀이

사전 지식

* 1차 정규형

- 모든 속성은 반드시 하나의 값을 가져야 한다. 즉, 반복 형태가 있어서는 안된다.
- 각 속성의 모든 값은 동일한 형식이어야 한다.
- 각 속성들은 유일한 이름을 가져야 한다.
- 레코드들은 서로 간에 식별 가능해야 한다.

* 2차 정규형

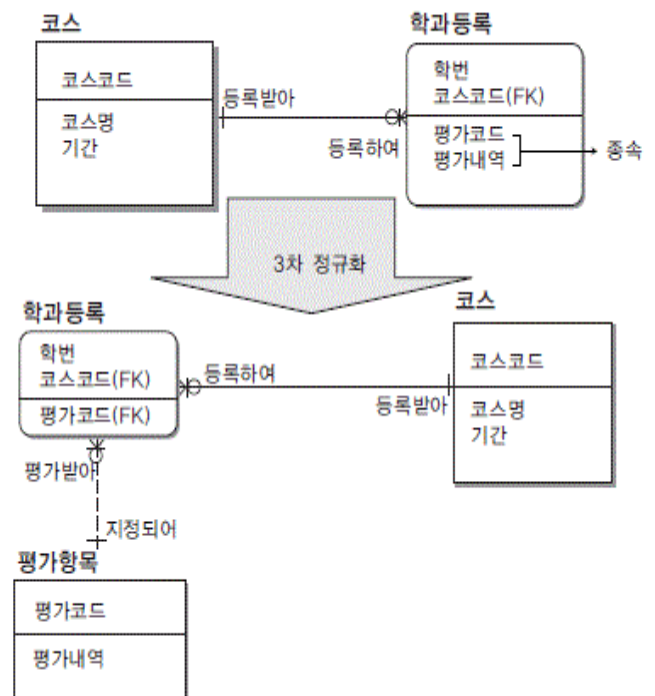
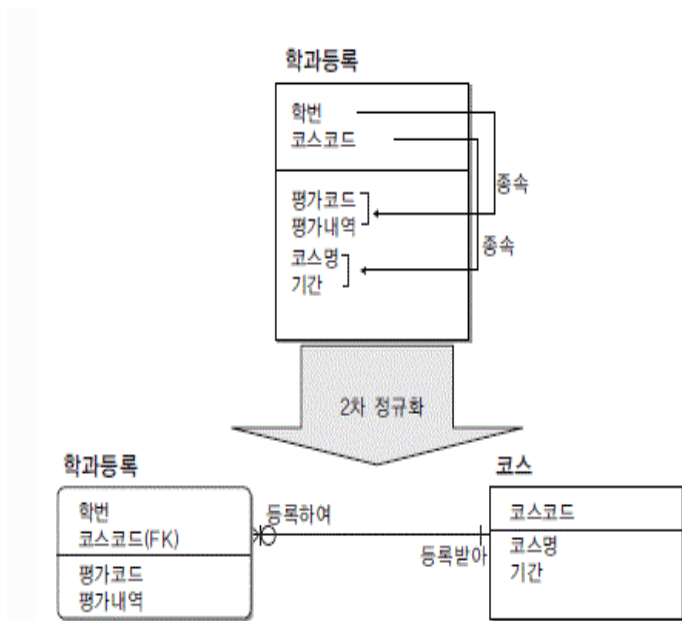
- 식별자가 아닌 모든 속성들은 식별자 전체 속성에 완전 종속되어야 한다.
- 이것을 물리 데이터 모델의 테이블로 말하면 기본키가 아닌 모든 칼럼들이 기본키에 종속적이어야 2차 정규형을 만족할 수 있다는 것이다

* 3차 정규형

- 2차 정규형을 만족하고 식별자를 제외한 나머지 속성들 간의 종속이 존재하면 안된다.
- 이것이 3차 정규형을 만족하는 것이다.

9번 해답 : 3번

왼쪽 2차 정규화, 3차정규화 참고



10번 풀이

사전 지식

GROUP BY ROLLUP(DNAME,JOB)

= GROUP BY DNAME,JOB

UNION ALL

GROUP BY DNAME

UNION ALL

모든 집합 그룹 결과

GROUP BY GROUPING SET(DNAME,JOB)

= GROUP BY DNAME

UNION ALL

GROUP BY JOB

GROUP BY CUBE(DNAME,JOB)

= GROUP BY DNAME,JOB

UNION ALL

GROUP BY DNAME

UNION ALL

GROUP BY JOB

UNION ALL

모든 집합 그룹 결과

10번 해답 : 4번

GROUPING SETS(A,B,C) = GROUP BY A UNION ALL GROUP BY B UNION ALL GROUP BY C 이므로

4번만 다른 결과를 도출함

11번 풀이

사전 지식

ANSI SQL 에서 조인 조건절 (ON 절) 에 사용된 조건절은 조인 전 조건으로 작용한다

ON 절 이후 WHERE 절에서 쓰인 조건절은 조인후 조건절로 사용된다

11번 해답 : 4번

조인 조건과 조회 조건이 분리되어야 하므로 조회 조건은 WHERE 절로 분리되어야 함

OUTER JOIN 이 아닌 INNER JOIN 이므로 INNER JOIN 에 조회 조건 분리가 된 4번이 맞음

12번 풀이

12번 해답 : 3번

View 는 논리적인 SELECT 를 포함하는 OBJECT 로 실제 데이터를 저장하지 않음

13번 풀이

* INSERT 구문은 IDENTITY 문제 또는 CHECK 문제가 주로 나옴

사전 지식

* SQL SERVER : IDENTITY [(seed , increment)]

- SEED : 첫번째 행이 입력될 때의 시작값
- 증가되는 값
- 해당 컬럼에 값을 넣을 경우 Error 발생 (Error 발생을 막기 위해서는 IDENTITY_INSERT 를 OFF 로 하면 되나 이런 경우 IDENTITY 를 쓴 의미가 없어짐)

13번 해답 : 1번

SQL SERVER 의 IDENTITY 로 지정된 컬럼에는 값을 넣으면 에러가 남

ORACLE 의 CHECK 조건을 만족하지 못할 경우 에러가 나나 NULL 은 무시됨

14번 풀이

14번 해답 : 3번, 4번

Hash Join 은 Non Equal Join 은 불가능함. Equal Join 만 가능함

사전 Sorting 작업이 필요한 Join 알고리즘은 Sort Merge Join 임

15번 풀이

15번 해답 : 1번

Outer Join 이라고 표시된 SQL Operation 이 존재하지 않음

Hash Join 의 선행 테이블은 SCOTT.DEPT 테이블임

Predicate 정보를 보면 DEPT 테이블에 대한 Full Scan 후 DNAME ='SALES' 로 필터됨

16번 풀이

16번 해답 : 1번

17번 풀이

사전 지식

* 문자를 숫자형으로 형변환함

* 예외적으로 LIKE 의 경우 COL 을 무조건 형변환함. 숫자를 문자로 변환함

17번 해답 : 2번

LIKE 의 경우 컬럼을 무조건 문자로 형변환함. 즉, 해당 조건절은

WHERE TO_CHAR(COL1) LIKE '2%' 로 변형되어 인덱스를 사용하지 못함

IS NOT NULL 은 해당 인덱스를 FULL SCAN 할 수 있으나 효율이 떨어짐

부정형 비교는 인덱스 사용이 불가함

18번 풀이

18번 해답 : 4번

SAPCE 를 추가하여 길이를 맞춰 비교하는 방법은 CHAR 타입인 경우임

19번 풀이

19번 해답 : 4번

Oracle 의 결합 함수 : CONCAT / ||

SQL Server : +

20번 풀이

사전 지식

* TOP (Expression) [PERCENT] [WITH TIES]

- WITH TIES : ORDER BY 조건 기준으로 TOP N 의 마지막 행으로 표시되는 추가 행의 데이터가 같을 경우 N+ 동일 정렬 순서 데이터를 추가 반환하도록 지정하는 옵션 (마지막 기준 공통일 경우 모두 출력)

20번 해답 : 2번

ROWNUM < 4 의 의미는 3건까지 출력이므로 TOP(3) 임

WITH TIES 를 사용할 경우 동일 데이터가 있을 때 추가 건수가 출력되는 현상이 가능함

21번 풀이

21번 해답 : 2번

NATURAL JOIN 에서 사용된 열은 식별자를 가질수 없음. 즉 EMP.DEPTNO 와 같이 OWNER 명을 사용하면 에러가 남

22번 풀이

사전 지식

* 엔터티 기준

- 엔터티는 사람, 장소, 물건, 사건, 개념 등의 명사에 해당한다.
- 엔터티는 업무상 관리가 필요한 관심사에 해당한다.
- 엔터티는 저장이 되기 위한 어떤 것(Thing)이다.

22번 해답 : 3번

교수,학생은 논란의 여지가 없으나, 수강과 청약자에서 논란이 일어남

청약자 라는 개념은 관심사라기 보다는 청약의 주체가 되는 속성에 가까운 곳으로 보임. 속성값을 가지기에 애매함

* 참고 : 대다수 수강이라고 했으나 청약자로 체크한 분이 만점으로 답으로 인정

23번 풀이

사전 지식

CONNECT_BY_ISLEAF : 전개 과정에서 해당 데이터가 리프 데이터면 1,아니면 0

CONNECT_BY_ISCYCLE : 전개 과정에서 자식을 갖는데, 해당 데이터가 조상으로 존재하면 1, 그렇지 않으면 0

여기서 조상이란 자신으로부터 루트까지의 경로에 존재하는 데이터를 말함

SYS_CONNECT_BY_PATH : 하위 레벨의 컬럼까지 모두 표시해줌 (구분자 지정 가능)

CONNECT_BY_ROOT : Root 노드의 정보를 표시

23번 해답 : 1번

24번 풀이

사전 지식

FULL OUTER JOIN = LEFT OUTER JOIN UNION RIGHT OUTER JOIN

24번 해답 : 1번

25번 풀이

사전 지식

RANGE BETWEEN start_point AND end_point

- start_point는 end_point와 같거나 작은 값이 들어감
- Default값은 RANGE BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW
- UNBOUNDED PRECEDING : start_point만 들어갈 수 있으며, 파티션의 first row
- UNBOUNDED FOLLOWING : end_point만 들어갈 수 있으며, 파티션의 last row
- CURRENT ROW : start, end_point 둘다 가능. 윈도우는 CUREENT ROW에서 start하거나 end 함

25번 해답 : 4번

UNBOUNDED PRECEDING 은 end point 에 사용될 수 없음

26번 풀이

26번 해답 : 1번

27번 풀이

사전 지식

Non Equal Join 의 경우는 조인 조건을 제외한 Cross 조인 후
조인 조건을 필터 조건으로 처리하는 것이 좋음

27번 해답 : 3번

Cross Join 결과 12건에 대한 조건 체크(조건 체크시 NULL 제외) 후 SUM

A	A	1
A		1
A	B	1
	A	2
		2
	B	2
B	A	3
B		3
B	B	3
C	A	4
C		4
C	B	4

SUM = 12

28번 풀이

28번 해답 : 3번

서브쿼리, 특히 INLINE VIEW 의 컬럼을 메인 쿼리에서도 사용 가능함

29번 풀이

29번 해답 : 2번

DBA 권한은 SYSTEM,SYS 등의 상위 유저와 그에 해당하는 권한을 가진 경우 부여 가능

단답형

1번 풀이

1번 해답 : UPPER

소문자 / 대문자로 되어 있는 경우 두개의 문자열이 같기 위해서 적용해야 하는 함수 UPPER 를 사용

유사 문제

만약 주어진 테이블의 데이터가 아래와 같을 경우는 빈칸 제거가 답임

TABLE SQLD_21_01

N1 V1

1 A (스페이스 뒤에 있음)

2

3 B (스페이스 뒤에 있음)

4 C (스페이스 뒤에 있음)

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

-> TRIM

2번 풀이

2번 해답 : MAX(LEVEL)

계층형 쿼리에서 최대 계층의 수를 구하기 위한 문제. MAX (LEVEL) 을 사용하여 최대 계층 수를 구함

3번 풀이

3번 해답 : P.사원번호 IS NULL

NOT EXISTS 의 OUTER JOIN 으로의 변형을 묻는 문제로 NOT EXISTS 는 OUTER JOIN 으로 변경 시 NOT NULL COLUMN 에 대한 IS NULL 체크로 NOT EXISTS 를 구현가능하다

유사 문제

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 P.사원번호 IS NULL

4번 풀이

사전 지식

** DNAME 의 그룹은 3, JOB 의 그룹은 5, DNAME,JOB 의 그룹은 9개가 형성됨

** GROUP BY CUBE(DNAME,JOB)

= GROUP BY DNAME,JOB

UNION ALL

GROUP BY DNAME

UNION ALL

GROUP BY JOB

UNION ALL

모든 집합

4번 해답 : 18건

<i>DNAME</i>	<i>JOB</i>	<i>Total Emp</i>	<i>Total Sal</i>
		14	29025
	CLERK	4	4150
	ANALYST	2	6000
	MANAGER	3	8275
	SALESMAN	4	5600
	PRESIDENT	1	5000
SALES		6	9400
SALES	CLERK	1	950
SALES	MANAGER	1	2850
SALES	SALESMAN	4	5600
RESEARCH		5	10875
RESEARCH	CLERK	2	1900
RESEARCH	ANALYST	2	6000
RESEARCH	MANAGER	1	2975
ACCOUNTING		3	8750
ACCOUNTING	CLERK	1	1300
ACCOUNTING	MANAGER	1	2450
ACCOUNTING	PRESIDENT	1	5000

5번 풀이

5번 해답 : WHERE

6번 풀이

객관식 27번과 동일함

6번 해답 : 7건

7번 풀이

7번 해답 : 0