

S.O.S Network - Second Project : SQLD 기출문제집 작성

30회 기출문제 복원 SQLD 예상문제



Team : *S.O.S Network*
월야루

Creation Date : 2019.03.10

Last Updated : 2019.03.14

Version : 1.0

Special Thanks to C.Y.J

About S.O.S Network

뭐하는 모임인가요?

데이터에 대해 공부하고, 탐구하여 열린 마음으로 모든 것을 공유하는 집단을 꿈꾸는 모임
SQL 과 NOSQL 모두를 아우르는 열린 마음의 데이터를 다루는 사람들의 모임 이길 ..
주요 거주지 카페 : <http://cafe.naver.com/sqlpd> & email 연락처 : ibutu@naver.com

누구누구 있나요?

데이터를 다루는 사람들이 있음. 현재는 주로 Oracle , Cloud 영역의 사람들.
앞으로는 더 많은 분야의 사람들이 있을 예정(?)

앞으로의 활동은?

영문 기술 문서에 대한 번역 Project 2탄 및 DBA 를 위한 운영 DB 관리 매뉴얼 작성 준비
SQL Coding 교육 커리큘럼 개발 및 강의 준비 및 기타 교육 커리큘럼 개발 준비

앞의 그림은 로고 인가요?

S.O.S Network 의 로고. 이 로고가 표시된 문서는 S.O.S Network의 공식(?) 문서



30회 기출 문제

1. 아래 내용의 특징은 무엇을 설명하고 있는지 고르시오.

[설명] 엔터티 내에서 속성에 대한 데이터 타입과 크기 그리고 제약사항을 지정하는 것

- 1) 제약조건
- 2) 도메인
- 3) 정규화
- 4) 식별자

2. 아래의 반정규화에 대한 설명 중 부적절한 것은 ?

- 1) 데이터를 조회할 때 디스크 I/O 량이 많아서 성능이 저하되거나 경로가 너무 멀어 조인으로 인한 성능저하가 예상될 때 수행한다.
- 2) 컬럼을 계산하여 읽을 때 성능이 저하될 것이 예상되는 경우 수행한다.
- 3) 반정규화를 적용할 때는 기본적으로 데이터 무결성이 깨질 가능성이 많이 있으므로 반드시 데이터 무결성을 보장할 수 있는 방법을 고려해야한다.
- 4) 반정규화 전에 테이블 추가(통계 테이블, 중복 테이블, 이력 테이블 추가) 를 통해 반정규화를 회피한다.

3. 발생시점에 따른 엔터티 분류에 의한 중심 엔터티가 아닌 것은?

- 1) 매출
- 2) 계약
- 3) 사원
- 4) 주문

4. 발생시점에 따른 엔터티 분류에 대해서 아래의 내용 중 잘못 짝지워진 것은?

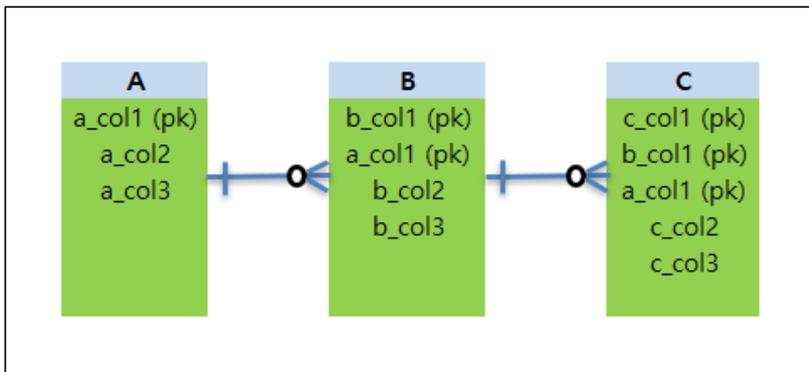
A : 기본/키 엔터티 : 조직, 사원
B : 기본/키 엔터티 : 부서
C : 중심 엔터티 : 상품, 주문상품
D : 행위 엔터티 : 주문내역, 계약진행

- 1) A
- 2) B
- 3) C
- 4) D

5. 데이터 모델링이 최종적으로 완료된 상태라고 정의할 수 있는, 즉 물리적인 스키마 설계를 하기 전 단계를 가리키는 말은 무엇인가?

- 1) 개괄적 모델링
- 2) 개념적 모델링
- 3) 논리적 모델링
- 4) 물리적 모델링

6. 아래의 ERD 에 대한 설명중 가장 부적절한 것을 고르시오



- 1) SQL 작성 시 B 를 제외할 경우 A 와 C 는 카테시안 조인이 된다
- 2) 3개 테이블을 모두 조인할 경우 조인 최소 조건은 3개이다.
- 3) B, C 에서 C 의 데이터는 모두 B 에 존재하므로 Outer Join 을 안해도 된다.
- 4) B 는 A 의 데이터를 모두 포함하지 않는다.

7. 아래의 설명 중 반정규화 대상이 아닌 것은?

- 1) 자주 사용되는 테이블에 접근하는 프로세스의 수가 많고 항상 일정한 범위만을 조회하는 경우
- 2) 테이블의 대량의 데이터가 있고 대량의 데이터 범위를 자주 처리하는 경우에 처리범위를 일정하게 줄이지 않으면 성능을 보장할 수 없을 경우
- 3) 통계성 프로세스에 의해 통계 정보를 필요로 할 때 별도의 통계 테이블을 생성해야 하는 경우
- 4) 테이블에 지나치게 많은 조인과 Sorting, Order by 프로세스가 많은 경우

8. 아래의 식별자의 분류체계에 대한 설명 중 가장 부적절한 것은?

- 1) 대표성 여부에 따라 분류되는 주식별자와 보조식별자에서 주식별자는 대표성을 가지지 못해 참조관계 연결을 못한다.
- 2) 스스로 생성여부에 따라 분류되는 식별자는 내부 식별자와 외부 식별자이다.
- 3) 둘 이상의 속성으로 구성된 식별자를 복합식별자라 하며 속성의 수에 따른 식별자 분류이다.
- 4) 업무적으로 만들어지지 않는지만 필요에 따라 인위적으로 만든 식별자를 인조 식별자라 한다.

9. 분산 데이터베이스의 특징 중 사용하려는 데이터의 저장 장소 명시가 불필요하다는 특징은 무엇인가?

- 1) 분할 투명성
- 2) 위치 투명성
- 3) 지역사상 투명성
- 4) 중복 투명성

10. Row migration과 Row Chaining 에 대한 아래의 설명 중 바른 것은?

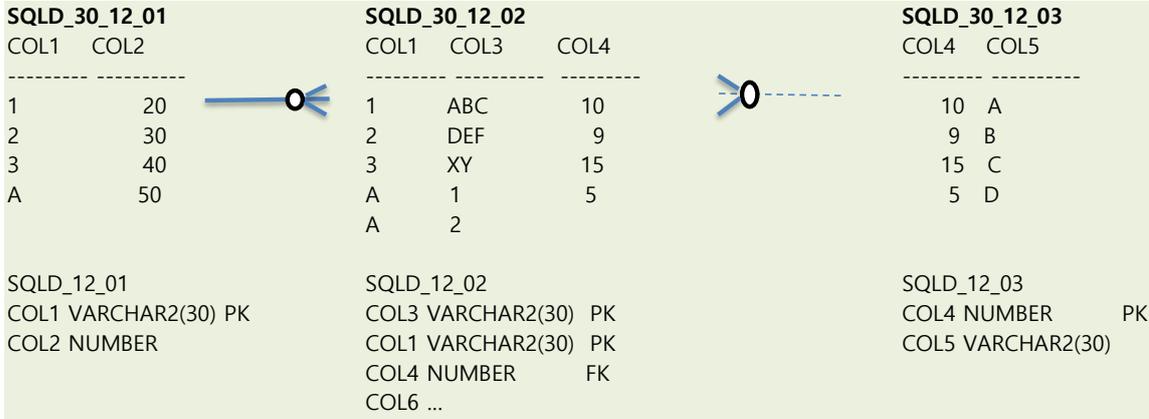
- 1) Row Chaining 과 Row Migration 이 많아지게 되더라도 성능 저하는 일어나지 않는다.
- 2) 로우 길이가 너무 길어서 데이터 블록 하나에 데이터가 모두 저장되지 않고 두 개 이상의 블록에 걸쳐 하나의 로우가 저장되는 현상을 Row Chaining 이라고 한다.
- 3) Row Migration 은 신규 데이터의 입력이 발생할 때 발생하는 현상이다.
- 4) Row Chaining 과 Row Migration 이 발생되면 디스크 I/O 가 발생 될 때 I/O 가 많아지게 되어 성능이 향상될 수 있다.

11. 아래의 ERD 를 참고하여 보기의 SQL 중 오류가 날 수 있는 SQL 을 고르시오

TABLE SQLD_30_11_01			TABLE SQLD_30_11_02			SQLD_30_11_01	SQLD_30_11_02
COL1	COL2		COL1	COL3	COL4	COL1 VARCHAR2(30)	COL1 VARCHAR2(30)
1	20		1	ABC	10	COL2 NUMBER	COL3 VARCHAR2(30)
2	30		2	DEF	9		COL4 NUMBER
3	40		3	XY	15		
A	50		A	1	5		

- | | |
|---|---|
| <ol style="list-style-type: none"> 1) SELECT A.COL1, A.COL2 FROM SQLD_30_11_01 A WHERE A.COL1 > 0; 3) SELECT COUNT(*) FROM SQLD_30_11_01 A WHERE EXISTS (SELECT 'X' FROM SQLD_30_11_02 B WHERE A.COL2 = B.COL4); | <ol style="list-style-type: none"> 2) SELECT A.COL1, B.COL4 FROM SQLD_30_11_01 A INNER JOIN SQLD_30_11_02 B ON (A.COL1 = B.COL1) WHERE B.COL3 > 'X'; 4) SELECT SUM(A.COL2) FROM SQLD_30_11_01 A INNER JOIN SQLD_30_11_02 B ON (A.COL1 = B.COL1) WHERE B.COL4 > '1'; |
|---|---|

12. 아래의 ERD 를 참고하여 보기의 SQL 중 오류가 날 수 있는 SQL 을 고르시오



1)
 SELECT A.COL1,
 (SELECT COL3 FROM SQLD_30_12_02 B
 WHERE A.COL1 = B.COL1) AS B_COL3
 FROM SQLD_30_12_01 A
 WHERE A.COL2 > 10;

2)
 SELECT A.COL1,
 (SELECT COL5 FROM SQLD_30_12_03 B
 WHERE A.COL4 = B.COL4) AS B_COL3
 FROM SQLD_30_12_02 A
 WHERE A.COL4 > 0;

3)
 SELECT A.COL1, B.COL3
 FROM SQLD_30_12_01 A
 INNER JOIN SQLD_30_12_02 B
 ON (A.COL1 = B.COL1)
 INNER JOIN SQLD_30_12_03 C
 ON (B.COL4 = C.COL4)
 WHERE C.COL4 > 5
 AND A.COL2 > 10;

4)
 SELECT A.COL1, B.COL3
 FROM SQLD_30_12_01 A
 INNER JOIN SQLD_30_12_02 B
 ON (A.COL1 = B.COL1)
 WHERE EXISTS (SELECT 'X'
 FROM SQLD_30_12_03 C
 WHERE B.COL4 = C.COL4)

13. 아래의 SQL 결과를 보고 Group by 뒤 빈칸에 들어가는 내용을 고르시오.

```
SELECT JOB_ID, MANAGER_ID, SUM(SALARY)
FROM HR.EMPLOYEES
WHERE JOB_ID IN ('SH_CLERK','ST_CLERK','SA_REP')
GROUP BY (      );
```

- 1) ROLLUP (JOB_ID, MANAGER_ID)
- 2) CUBE(JOB_ID, MANAGER_ID)
- 3) GROUPING SET(JOB_ID, MANAGER_ID)
- 4) ROLLUP (MANAGER_ID, JOB_ID)

JOB_ID	MANAGER_ID	SUM(SALARY)
SA_REP	145	51000
SA_REP	146	51000
SA_REP	147	46600
SA_REP	148	51900
SA_REP	149	50000
SA_REP	-	250500
SH_CLERK	120	11600
SH_CLERK	121	14700
SH_CLERK	122	12800
SH_CLERK	123	13900
SH_CLERK	124	11300
SH_CLERK	-	64300
ST_CLERK	120	10500
ST_CLERK	121	10700
ST_CLERK	122	10800
ST_CLERK	123	12000
ST_CLERK	124	11700
ST_CLERK	-	55700
-	-	370500

14. 아래의 SQL 결과로 올바른 것은?

<pre>SQL> SELECT COUNT(*) FROM SQLD_30_14_01 WHERE (COL1, COL2) IN (('A', 50));</pre>	<pre>SQLD_30_14_01 COL1 COL2 ----- Y 20 X 30 A 40 A 50</pre>	<pre>SQLD_30_14_01 COL1 VARCHAR2(30) COL2 NUMBER</pre>
--	---	--

- 1) 0
- 2) 1
- 3) 2
- 4) 3

15. 아래의 데이터를 가진 테이블에서 아래와 같은 SQL 이 수행되었을때의 결과는?

<pre>SQL> SELECT SUM(COL2) FROM SQLD_30_15_01 WHERE COL1 IN ('A','X',NULL);</pre>	<pre>SQLD_30_15_01 COL1 COL2 ----- Y 20 X 30 A 40 A 50 10 80</pre>	<pre>SQLD_30_15_01 COL1 VARCHAR2(30) COL2 NUMBER</pre>
--	---	--

- 1) 210
- 2) 120
- 3) 230
- 4) 90

16. 아래의 실행계획에 대한 SQL 수행 순서를 올바르게 적은 것은?

- [보기]
- 0- SELECT ..
 - 1- NESTED LOOP JOIN
 - 2- NESTED LOOP JOIN
 - 3- TABLE ACCESS (FULL)
 - 4- TABLE ACCESS (BY INDEX ROWID)
 - 5- INDEX (RANGE SCAN)
 - 6- TABLE ACCESS (BY INDEX ROWID)
 - 7- INDEX (RANGE SCAN)

- 1) 3-5-4-2-7-6-1-0
- 2) 5-3-4-2-7-6-1-0
- 3) 3-4-5-2-7-6-1-0
- 4) 3-5-4-2-6-7-1-0

17. 아래 SQL 의 수행결과로 올바른 것은?

<pre>SQL> SELECT nvl(count(*), 9999) FROM SQLD_30_17_01 WHERE 1=2</pre>	<pre>SQLD_30_17_01 COL1 COL2 ----- Y 20 X 30 A 40 A 50 10 80</pre>	<pre>SQLD_30_17_01 COL1 VARCHAR2(30) COL2 NUMBER</pre>
--	---	--

- 1) 0
- 2) 9999
- 3) 1
- 4) ERROR

18. 아래 데이터를 참고하여 아래 SQL 의 수행결과로 올바른 것은?

SQL>	SQLD_30_18_01	SQLD_30_18_01
SELECT COL1, SUM(COL2)	COL1 COL2	COL1 VARCHAR2(30)
FROM SQLD_30_18_01	-----	COL2 NUMBER
GROUP BY COL1;	1 100	
	1 NULL	
	2 100	
	2 200	

- | | | | | | |
|----|------|-----------|----|------|-----------|
| 1) | COL1 | SUM(COL2) | 2) | COL1 | SUM(COL2) |
| | 1 | NULL | | 1 | 100 |
| | 2 | 300 | | 2 | 300 |
| 3) | COL1 | SUM(COL2) | 4) | COL1 | SUM(COL2) |
| | 2 | 300 | | 1 | 100 |
| | | | | 1 | NULL |
| | | | | 2 | 300 |

19. 아래와 같은 SQL 이 순서대로 수행되었을 때 최종 결과값으로 알맞은 것은?

[SQL]

(1) INSERT INTO SQLD_30_19 VALUES (1);	SQLD_30_19
(2) INSERT INTO SQLD_30_19 VALUES (2);	COL1 NUMBER
(3) SAVEPOINT SV1;	
(4) UPDATE SQLD_30_19 SET COL1=4 WHERE COL1=2;	
(5) SAVEPOINT SV1;	
(6) DELETE SQLD_30_19 WHERE COL1 >=2;	
(7) ROLLBACK TO SV1;	
(8) INSERT INTO SQLD_30_19 VALUES (3);	
(9) SELECT MAX(COL1) FROM SQLD_30_19;	

- 1) 1
- 2) 2
- 3) 3
- 4) 4

20. 아래의 Trigger 에 대한 설명 중 가장 부적절한 것은 ? ***

- 1) DELETE ON TRIGGER 의 경우 :OLD 는 삭제 전 데이터를 , :NEW 는 삭제 후 데이터를 나타낸다.
- 2) 특정 테이블에 DML 문이 수행되었을 때 자동으로 동작하도록 작성된 프로그램이다.
- 3) 테이블, 뷰에만 트리거를 작성할 수 있으며 데이터베이스 작업에는 정의할 수 없다.
- 4) UPDATE TRIGGER 에서 :OLD 에는 수정 전, :NEW 에는 수정 후 값이 들어간다.

21. 아래의 SELECT 결과가 NULL 이 아닌 경우는?

- 1) SELECT COALESCE ('AB','BC','CD') FROM DUAL;
- 2) SELECT CASE 'AB' WHEN 'BC' THEN 'CD' END FROM DUAL;
- 3) SELECT DECODE ('AB','CD','DE') FROM DUAL;
- 4) SELECT NULLIF ('AB','AB') FROM DUAL;

22. 아래와 같은 테이블이 있을때 아래와 같은 SQL 의 수행결과에 대해 작성하시오

SQLD_30_22	COL1	COL2
	100	100
	NULL	60
`	NULL	NULL

SELECT COALESCE(COL1, COL2 *50 , 50) FROM SQLD_30_22;

- 1) 100, 3000, 50
- 2) 100, 60, NULL
- 3) 100, NULL,NULL
- 4) 100, 3000, NULL

23. 아래의 DML, DCL, DDL 이 잘못 짝지워진 것은?

- 1) DDL : CREATE
- 2) DML : UPDATE
- 3) DCL : ROLLBACK
- 4) DCL : SELECT

24. 보기의 SQL 명령어중 TCL 인 것은 어떤 것인가?

[보기] UPDATE GRANT SELECT COMMIT

- 1) UPDATE
- 2) GRANT
- 3) SELECT
- 4) COMMIT

25. 아래의 특징을 가지는 데이터 타입은 무엇인가?

[보기] 고정 길이 문자열 정보로 S만큼 최대 길이를 갖고 고정 길이를 가지고 있으므로 할당된 변수 값의 길이가 S보다 작을 경우에는 그 차이 길이 만큼 공간으로 채워진다.

- 1) CHARACTER
- 2) VARCHAR
- 3) NUMERIC
- 4) DATETIME

26. 아래의 SQL 과 항상 동일한 결과를 나타내는 SQL 을 고르시오

[SQL] SELECT TOP(10) FIRST_NAME, JOB_ID
FROM HR.EMPLOYEES
ORDER BY SALARY;

1)
SELECT FIRST_NAME, JOB_ID
FROM HR.EMPLOYEES
WHERE ROWNUM <= 10
ORDER BY SALARY;

3)
SELECT FRIST_NAME, JOB_ID
FROM (
SELECT FIRST_NAME, JOB_ID, ROWNUM RN
FROM HR.EMPLOYEES
ORDER BY SALARY
)
WHERE RN <= 10;

2)
SELECT TOP(10) WITH TIES FIRST_NAME, JOB_ID
FROM HR.EMPLOYEES
ORDER BY SALARY;

4)
SELECT FRIST_NAME, JOB_ID
FROM (
SELECT FIRST_NAME, JOB_ID
FROM HR.EMPLOYEES
ORDER BY SALARY
)
WHERE ROWNUM <= 10;

27. 아래의 계층형 SQL 수행하여 아래와 같은 결과가 나올 때 () 에 맞는 것을 고르시오

```
[SQL]
SELECT LEVEL,
       LPAD('** ', (LEVEL -1 ) *2 , '**') || EMPLOYEE_ID AS EMP_TREE,
       MANAGER_ID,
       EMPLOYEE_ID
FROM HR.EMPLOYEES
WHERE 1=1
START WITH (      )
CONNECT BY PRIOR (      );
```

```
[RESULT]
LEVEL      EMP_TREE      MANAGER_ID      EMPLOYEE_ID
-----
1 100
2 **101              100              101
3 *** 108              101              108
4 **** 109              108              109
4 **** 110              108              110
4 **** 111              108              111
4 **** 112              108              112
4 **** 113              108              113
3 *** 200              101              200
3 *** 203              101              203
3 *** 204              101              204
3 *** 205              101              205
4 **** 206              205              206
2 **102              100              102
3 *** 103              102              103
4 **** 104              103              104
4 **** 105              103              105
4 **** 106              103              106
4 **** 107              103              107
```

뒷 결과 생략

- 1) MANAGER_ID IS NULL, MANAGER_ID = EMPLOYEE_ID
- 2) MANAGER_ID IS NOT NULL, MANAGER_ID = EMPLOYEE_ID
- 3) MANAGER_ID IS NULL, EMPLOYEE_ID = MANAGER_ID
- 4) MANAGER_ID IS NOT NULL, EMPLOYEE_ID = MANAGER_ID

28. SQL Set Operation 에서 중복 제거를 위해 정렬 작업을 하지 않는 집합 연산자는?

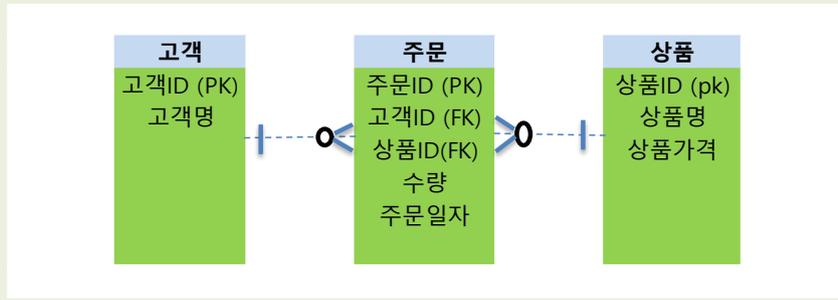
- 1) UNION
- 2) UNION ALL
- 3) INTERSECT
- 4) MINUS

29. SQL Set Operation 에서 중복 제거를 위해 정렬 작업을 하지 않는 집합 연산자는?

[보기] 대용량 데이터를 정렬하여 조인한다
동등 조인, 비동등 조인에서 모두 사용 가능하다
각 테이블을 정렬한 후 조인한다.

- 1) Sort Merge Join
- 2) Hash Join
- 3) Nested Loop Join
- 4) Cartesian Join

30. 아래 ERD 를 참고하여 모든 회원의 총 주문금액을 구하는 SQL 중 잘못된 SQL 은?



```
1)
SELECT A.고객명,
       NVL(SUM(C.수량*B.상품가격),0) AS 총주문금액
FROM SQLD_30_30_고객 A
     INNER JOIN SQLD_30_30_주문 C ON A.고객ID = C.고객ID
     INNER JOIN SQLD_30_30_상품 B ON C.상품ID = B.상품ID
WHERE 1=1
GROUP BY A.고객명
ORDER BY 1,2;
```

```
2)
SELECT A.고객명,
       NVL(SUM(총가격),0) AS 총주문금액
FROM SQLD_30_30_고객 A
     LEFT OUTER JOIN
(
  SELECT C.고객ID, C.수량 * B.상품가격 AS 총가격
  FROM SQLD_30_30_주문 C
       INNER JOIN SQLD_30_30_상품 B ON C.상품ID = B.상품ID) D
  ON A.고객ID = D.고객ID
WHERE 1=1
GROUP BY A.고객명
ORDER BY 1,2;
```

```
3)
SELECT A.고객명,
       NVL(
         SELECT SUM(C.수량 * B.상품가격)
         FROM SQLD_30_30_주문 C
              LEFT JOIN SQLD_30_30_상품 B ON C.상품ID = B.상품ID
              WHERE C.고객ID = A.고객ID),0) AS 총주문금액
FROM SQLD_30_30_고객 A
WHERE 1=1
ORDER BY 1,2;
```

```
4)
SELECT A.고객명,
       NVL(SUM(C.수량*B.상품가격),0) AS 총주문금액
FROM SQLD_30_30_고객 A
     LEFT OUTER JOIN SQLD_30_30_주문 C ON A.고객ID = C.고객ID
     LEFT OUTER JOIN SQLD_30_30_상품 B ON C.상품ID = B.상품ID
WHERE 1=1
GROUP BY A.고객명
ORDER BY 1,2;
```

31. 아래의 SQL 의 로우의 건수로 올바른 것은?

SQLD_30_31_01	COL1	COL2	SQLD_30_31_02	COL1	COL2
	1	2		1	2
	1	2		1	4
	1	3		1	5

```
[SQL]
SELECT DISTINCT COL1, COL2
FROM SQLD_30_31_01
UNION ALL
SELECT COL1, COL2
FROM SQLD_30_31_02;
```

- 1) 3
- 2) 4
- 3) 5
- 4) 6

32. 아래의 데이터를 참고하여 다음 SQL 의 로우의 건수를 맞는 것은?

SQLD_30_32_01	COL1	COL2	COL3
	1	2	null
	2	null	1
	4	null	null

SQLD_30_32_02	COL1	COL2	COL3
	1	null	-1
	2	null	1
	3	5	2

[SQL]
 SELECT COUNT(*)
 FROM SQLD_30_32_01 A, SQLD_30_32_02 B
 WHERE A.COL1 =B.COL1
 AND NVL(A.COL2, -1) = NVL(B.COL2, -1)
 AND NVL(A.COL3, -1) = NVL(B.COL3, -1)

- 1) 1
- 2) 2
- 3) 3
- 4) 4

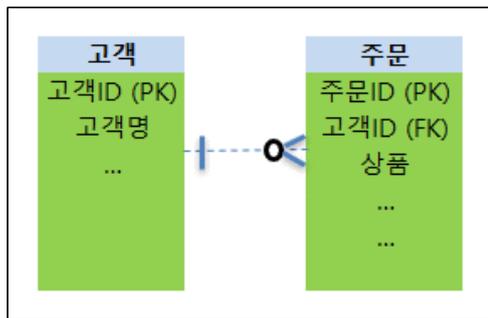
33. 아래와 같은 구조를 가진 테이블에서 Unique Index Scan 을 할 수 없는 것은?

[Table] SLQD_30_33

KEY1 (PK)	NUMBER
KEY2 (PK)	NUMBER
COL1	VARCHAR2
COL2	VARCHAR2
COL3	VARCHAR2

- 1) 모두 사용할 수 없다
- 2) WHERE KEY1 =1 AND KEY2 = 2
- 3) WHERE (KEY1, KEY2) IN ((1,2))
- 4) WHERE KEY1 =1

34. 아래의 ERD 에서 아래 SQL 문을 순서대로 수행 할 경우 오류가 발생하는 구간은?



[테이블 정보] 주문 (고객ID) REFERENCES 고객 (고객 ID)

[SQL]
 (1) INSERT INTO 고객 VALUES ('C001','AAA');
 (2) INSERT INTO 주문 VALUES ('O001','C001','XXX');
 (3) UPDATE 주문 SET 고객ID = NULL WHERE 주문ID = 'O001';
 (4) INSERT INTO 주문 VALUES ('O002','C002','YYY');

- 1) 1번 SQL
- 2) 2번 SQL
- 3) 3번 SQL
- 4) 4번 SQL

35. Cross Join 와 Natural Join 에 대한 차이점에 대해서 가장 부적절한 것은?

- 1) Cross Join 과 Natural Join 은 Where 절에서 조건을 걸 수 없다.
- 2) Natural Join 에서는 특정 Join 커럼을 명시적으로 적을 수 없다.
- 3) Cross Join 은 Join 에 참여하는 테이블의 Join Key 가 없을 경우 발생 한다.
- 4) Natural Join 에서 Join Key 는 컬럼명으로 결정된다.

36. 아래와 같이 PK 컬럼이 구성되어 있을 때 가장 적합한 파티셔닝 방법은 무엇인가?

[테이블 정보] 특징 : 데이터의 양이 매우 많은 대용량 테이블이다
데이터의 생성일자를 구분짓는 특정 컬럼이 없는 형태이다

PK : 지점, 코드

- 1) Range
- 2) List
- 3) Hash
- 4) Interval

37. 아래 2개 테이블에 대한 Join 후 나타난 결과이다. 해당 Join 의 방법을 고르시오

SQLD_30_37_01	COL1	COL2	SQLD_30_37_02	COL1	COL2
	1	2		1	2
	2	2		2	4
	3	3		4	5

[SQL]
SELECT *
FROM SQLD_30_37_01 A
 () SQLD_30_37_02 B
 ON (A.COL1 = B.COL1)
WHERE 1=1;

[RESULT]	A.COL1	A.COL2	B.COL1	B.COL2
	1	2	1	2
	2	2	2	4
	NULL	NULL	4	5

- 1) LEFT OUTER JOIN
- 2) RIGHT OUTER JOIN
- 3) FULL OUTER JOIN
- 4) INNER JOIN

38. 특정 그룹에서 특정 컬럼으로 정렬된 결과에서 첫번째 값을 구하는 Window Function 을 고르시오.

- 1) LAST_VALUE () OVER
- 2) FIRST_VALUE () OVER
- 3) MAX() OVER
- 4) 없다

39. 아래의 SQL 구문이 순서대로 수행된 후의 결과로 올바른 것은?

[SQL]
CREATE TABLE SQLD_30_39 (N1 NUMBER);
INSERT INTO SQLD_30_39 VALUES(1);
INSERT INTO SQLD_30_39 VALUES(2);
CREATE TABLE TMP_SQLD_30_39 (N1 NUMBER);
INSERT INTO TMP_SQLD_30_39 VALUES(1);
TRUNCATE TABLE TMP_SQLD_30_39;
ROLLBACK;
COMMIT;
SELECT SUM(N1) FROM SQLD_30_39;

- 1) 1
- 2) 2
- 3) 3
- 4) 4

40. 아래와 같은 데이터가 존재할 때 SQL 의 실행 결과로 올바른 것은?

SQLD_30_40	COL1	COL2
	1	10
	2	20
	3	null

```
[SQL] SELECT AVG ( NVL (COL2, 0 ) ) AS AVG_COL
FROM SQLD_30_40;
```

- 1) 10
- 2) 15
- 3) NULL
- 4) 20

41. 아래의 SQL 에서 ORDER BY 로 사용 할 수 없는 것은?

```
[SQL] SELECT JOB, COUNT(*) AS CNT
FROM TAB_A
GROUP BY JOB;
```

- 1) ORDER BY JOB
- 2) ORDER BY CNT DESC
- 3) ORDER BY COUNT(*)
- 4) ORDER BY 3

42. 아래의 데이터가 존재할 때 보기의 SQL 을 수행한 후 SQLD_30_42_01 테이블의 건수는 ?

SQLD_30_42_01	COL1	COL2	COL3
	A	X	1
	B	Y	2
	C	Z	3
	X	T	1

SQLD_30_42_02	COL1	COL2	COL3
	A	X	1
	B	Y	2
	C	Z	3
	D	가	4
	E	나	5

```
[SQL] MERGE INTO SQLD_30_42_01 A
USING SQLD_30_42_02 B
ON (A.COL1 = B.COL1)
WHEN MATCHED THEN
UPDATE SET A.COL3 = 4
WHERE A.COL3 = 2
DELETE WHERE A.COL3 <= 2
WHEN NOT MATCHED THEN
INSERT (A.COL1,A.COL2,A.COL3) VALUES(B.COL1,B.COL2,B.COL3);
```

- 1) 3
- 2) 4
- 3) 5
- 4) 6

43. SELECT ABS(-3.8), FLOOR(3.8), TRUNC(3.8), ROUND(3.8) FROM DUAL; 에 대한 값을 구하시오.

44. 아래에서 설명하는 것은 무엇인가?

[보기] 복잡한 질의를 단순하게 작성할 수 있다.또한 해당 형태의 SQL문을 자주 사용할 때 이용하면 편리하게 사용할 수 있다.
사용자에게 정보를 감출 수 있다
실제 데이터 를 가지고 있지 않다

45. 아래와 같은 데이터가 들어있는 테이블에서 아래 sql 이 수행되었을 때의 결과 건수는?

Table : SQLD_30_45

[SQL]

계층번호 상위계층번호

```
SELECT LEVEL,
       LPAD('** ', (LEVEL - 1)*2, ' ') || 계층번호 AS 계층트리,
       계층번호,
       상위계층번호
FROM SQLD_30_45
START WITH 상위계층번호 IS NULL
CONNECT BY 계층번호 = PRIOR 상위계층번호;
```

계층번호	상위계층번호
1	null
2	null
4	1
5	1
6	2
7	2
8	4
9	5
10	6
11	7

46. 아래의 SQL 구문은 컬럼의 데이터 타입을 변형하는 SQL 구문이다. SQL 구문을 완성하시오. (SQL SERVER 기준임)

[SQL] () DEPT () VARCHAR(30) NOT NULL;

47. 아래와 같은 테이블이 있을 때 아래의 SQL 구문이 순서대로 수행되었을 때 마지막 SQL 의 수행 결과를 적으시오.

[Table] SLQD_30_47

COL1	COL2
VARCHAR2(30)	NUMBER

[SQL]

```
INSERT INTO SQLD_47(COL1, COL2) VALUES('ABCD',NULL);
INSERT INTO SQLD_47(COL1, COL2) VALUES('BC',NULL);
ALTER TABLE SQLD_47 MODIFY COL2 DEFAULT 10;
INSERT INTO SQLD_47(COL1, COL2) VALUES('XY',NULL);
INSERT INTO SQLD_47(COL1) VALUES('EXD');
SELECT SUM(COL2) FROM SQLD_47;
```

48. 테이블 생성을 위한 권한을 주기 위해 아래의 SQL 을 완성하시오

[SQL] () CEATE TABLE TO USER01;

49. 아래와 같은 결과가 나오도록 SQL 을 작성하시오

[RESULT]

TABLE : SQLD_30_49

회원ID	RANK	주문금액
B	1	450
G	2	255
F	2	255
H	3	100

[SQL]

```
SELECT 회원ID,
       DENSE_RANK() OVER(ORDER BY ( ) ),
       주문금액
FROM SQLD_30_49;
```

50. 아래의 SQL 결과를 작성하시오

[SQL] SELECT UPPER("Sqldeveloper") FROM DUAL;