

1. 다음 중 아래에서 설명하는 데이터모델의 개념으로 가장 적절한 것은?

학생이라는 엔터티에서 학년이라는 속성 값의 범위는 1~4 사이의 정수이며, 주민번호 속성은 13자리 이내 문자열로 정의할 수 있다.

- ① 도메인
- ② 릴레이션
- ③ 시스템카탈로그
- ④ 속성사전

2. 엔터티 - 인스턴스 - 속성 - 속성값에 대한 관계 설명 중 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 인스턴스는 다른 엔터티의 인스턴스간의 관계인 Paring을 가진다.

3. 속성에 대한 설명으로 가장 적절하지 않은 것은?

- ① 하나의 속성에 여러 개의 값이 있는 다중값일 경우 별도의 엔터티를 이용하여 분리한다.
- ② 정해진 주식별자에 함수적 종속성을 가져야 한다.
- ③ 업무상 인스턴스로 관리하고자 하는 더 이상 분리되지 않는 최소의 데이터 단위를 말한다.
- ④ 엔터티에 속한 속성은 엔터티에 대한 추상적인 값을 갖는다.

4. 엔터티간 1:1, 1:M 과 같이 관계의 기수성을 나타내는 것을 무엇이라 하는가?

- ① 관계명
- ② 관계차수
- ③ 관계선택성
- ④ 관계정의

5. 다음 주식별자에 대한 설명 중 가장 적절하지 않은 것은?

- ① 주식별자에 의해 엔터티 내의 모든 인스턴스들이 유일하게 구분되어야 한다.
- ② 주식별자로 지정되더라도 속성 값으로 NULL이 들어갈 수 있다.
- ③ 주식별자를 구성하는 속성의 수는 유일성을 만족하는 최소의 수가 되어야 한다.
- ④ 지정된 주식별자의 값은 자주 변하지 않는 것이어야 한다.

6. 다음이 설명하는 정규화로 가장 적절한 것은?

테이블의 컬럼이 원자성(한 속성이 하나의 값을 갖는 특성)을 갖도록 테이블을 분해하는 단계

- ① 제 1 정규화
- ② 제 2 정규화
- ③ 제 3 정규화
- ④ 제 4 정규화

7. 다음이 설명하는 관계로 가장 적절한 것은?

두 엔터티나 두 속성 간에 동시에 발생할 수 없는 관계를 의미합니다.
즉, 하나의 엔터티나 속성이 특정한 경우 다른 엔터티나 속성은 해당 경우가 될 수 없음을 나타냅니다.

- ① 상호종속적
- ② 상호포괄적
- ③ 상호배타적
- ④ 상호일관적

8. 다음 중 트랜잭션에 대한 설명으로 가장 적절하지 않은 것은?

- ① 두 엔터티의 관계가 서로 필수적일 때 하나의 트랜잭션을 형성한다.
- ② 두 엔터티가 서로 독립적 수행이 가능하다면 선택적 관계로 표현한다.
- ③ 하나의 트랜잭션은 부분 COMMIT이 가능하다.
- ④ 하나의 트랜잭션에는 여러 SELECT, INSERT, DELETE, UPDATE 등이 포함될 수 있다.

9. NULL에 대한 설명으로 틀린 것은?

- ① 값이 존재하지 않거나 확정되지 않은 값을 의미한다.
- ② NULL과의 비교연산은 FALSE(거짓)를 리턴한다.
- ③ NULL과의 수치연산은 NULL 값을 리턴한다.
- ④ 공백과 같은 ASCII 값을 가진다.

10. 다음이 설명하는 식별자로 가장 적절한 것은?

다른 엔터티 참조 없이 엔터티 내부에서 스스로 생성되는 식별자

- ① 보조식별자
- ② 인조식별자
- ③ 본질식별자
- ④ 내부식별자

11. 다음 중 DBMS 특징이 아닌 것은?

- ① DBMS에 저장된 데이터는 다른 사용자에게 공유될 수 없다.
- ② 데이터 무결성을 유지 할 수 있다.
- ③ 실시간 접근, 자료의 계속적인 변화의 적용에 유리하다.
- ④ 인증된 사용자만이 참조 할 수 있는 보안기능이 제공된다.

12. 테이블 생성 시 주의 할 사항으로 적절하지 않은 것은?

- ① 컬럼 뒤에 데이터 유형은 꼭 지정되어야 한다.
- ② 테이블명과 컬럼명은 숫자로 시작해도 무관하다.
- ③ 테이블 생성시 대소문자 구분은 하지 않는다.
- ④ 소유자가 다를 경우 같은 이름의 테이블을 생성할 수 있다.

13. 다음 설명 중 틀린 하나는?

- ① ORDER BY 절에 SELECT 절에 정의되지 않은 컬럼을 사용할 수 있다.
- ② 테이블 별칭을 선언하면 컬럼 앞의 구분자는 반드시 테이블명 대신 테이블 별칭을 사용한다.
- ③ GROUP BY 절을 사용하는 경우 ORDER BY 절에는 GROUP BY절에 정의되지 않은 컬럼을 사용할 수 있다.
- ④ SELECT 문은 ORDER BY 절이 가장 나중에 실행된다.

14. 다음 중 DISTINCT에 대한 설명으로 가장 적절하지 않은 것은?

- ① DISTINCT 뒤에 나열되는 컬럼들의 중복값을 한 번만 출력하기 위해 사용한다.
- ② SELECT 문에서만 사용 가능하다.
- ③ DISTINCT 뒤에 나열되는 컬럼의 순서에 따라 결과 집합의 수가 달라진다.
- ④ DISTINCT 뒤에 * 를 사용할 수 있다.

15. 다음 SQL 수행 결과로 가장 적절한 것은?

```
SELECT ROUND(TO_DATE('2024-02-20 14:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'MONTH') AS D1,
       TRUNC(TO_DATE('2024-09-12 09:00:00', 'YYYY-MM-DD HH24:MI:SS')) AS D2
FROM DUAL;
```

①

D1	D2
2024-03-01 00:00:00	2024-09-01 00:00:00

②

D1	D2
2025-01-01 00:00:00	2024-09-01 00:00:00

③

D1	D2
2024-03-01 00:00:00	2024-09-12 00:00:00

④

D1	D2
2024-03-20 00:00:00	2024-09-12 00:00:00

16. 다음 함수 사용시 결과값이 올바르지 않은 것은?

- ① CEIL(-12.345) = -13
- ② FLOOR(-12.345) = -13
- ③ MOD(8,3) = 2
- ④ SIGN(0) = 0

17. 다음 함수의 결과로 가장 적절한 것은?

```
SELECT LTRIM('ORACLE', 'A') AS C1,
       SUBSTR('SQL-SERVER', 3, 3) AS C2,
       LENGTH(REPLACE('SQL-SERVER', 'E')) AS C3
FROM DUAL;
```

①

C1	C2	C3
ORCLE	L-S	8

②

C1	C2	C3
ORACLE	L	8

③

C1	C2	C3
ORACLE	L-S	8

④

C1	C2	C3
ORACLE	L-S	2

18. 다음 SQL중 실행 결과가 다른 하나는?

① SELECT DECODE(DEPTNO, 10, DECODE(JOB, 'CLERK', 'A', 'B'), 20, 'C', 'D')
FROM EMP;

② SELECT CASE WHEN (DEPTNO = 10 AND JOB = 'CLERK') THEN 'A' ELSE 'B'
WHEN DEPTNO = 20 THEN 'C' ELSE 'D'
END

FROM EMP;

③ SELECT CASE WHEN DEPTNO = 10 THEN CASE WHEN JOB = 'CLERK' THEN 'A' ELSE 'B' END
WHEN DEPTNO = 20 THEN 'C' ELSE 'D'
END

FROM EMP;

④ SELECT CASE DEPTNO WHEN 10 THEN CASE WHEN JOB = 'CLERK' THEN 'A' ELSE 'B' END
WHEN 20 THEN 'C' ELSE 'D'
END

FROM EMP;

19. 다음 중 정상적으로 실행되지 않는 문장은? (단, DBMS는 오라클)

① SELECT 100 + '1' FROM DUAL;

② SELECT TO_DATE('20240101', 'YYYYMMDD') - 10 FROM DUAL;

③ SELECT TO_DATE('11', 'DD') + 10 FROM DUAL;

④ SELECT NVL(100, 'NULL') FROM DUAL;

20. 다음 SQL 수행 결과로 가장 적절한 것은?

<TAB1>

COL1	COL2	COL3
A	NULL	100
A	200	300
B	100	NULL
B	300	200
NULL	100	0

SELECT COUNT(COL1) RESULT FROM TAB1 WHERE COL3 < 100 GROUP BY COL1;
 SELECT COUNT(COL1) RESULT FROM TAB1 WHERE COL1 IS NOT NULL GROUP BY COL1
 HAVING SUM(COL2) > 500;

①

RESULT
NULL

RESULT
NULL

②

RESULT
0

RESULT
0

③

RESULT
0

RESULT

④

RESULT

RESULT

21. 아래 SQL의 실행 결과로 알맞은 것은?

<TAB1>

COL1	COL2	COL3
A	1	NULL
B	2	10
B	NULL	20
NULL	3	30

SELECT SUM(COL2 + COL3) FROM TAB1 WHERE COL1 IS NOT NULL;

① 0

② NULL

③ 12

④ 33

22. 다음 SQL 문장 중 COL1 값이 널(NULL)이 아닌 경우를 찾아내는 문장으로 가장 적절한 것은?

① SELECT * FROM TAB1 WHERE COL1 IS NOT NULL;

② SELECT * FROM TAB1 WHERE COL1 <> NULL;

③ SELECT * FROM TAB1 WHERE COL1 != NULL;

④ SELECT * FROM TAB1 WHERE COL1 NOT NULL;

23. 다음 중 에러가 나지 않는 문장은? (단, DBMS는 오라클)

- ① SELECT COL1 컬럼1, AVG(COL2) 평 균
FROM TAB1;
- ② SELECT COL1 컬럼1, AVG(COL2) 평균
FROM TAB1
GROUP BY COL2;
- ③ SELECT COL1 AS 컬럼1, AVG(COL2) AS 평균
FROM TAB1
WHERE AVG(COL2) >= 100
GROUP BY 컬럼1;
- ④ SELECT COL1 AS 컬럼1, AVG(COL2) AS 평균
FROM TAB1
GROUP BY COL1
HAVING AVG(COL2) >= 100;

24. 다음 수행 결과로 가장 적절한 것은?

<EMP>

EMPNO	NUMBER
ENAME	VARCHAR2(10)
SAL	NUMBER
JUMIN	CHAR(13)
DEPTNO	NUMBER

EMPNO	ENAME	SAL	JUMIN	DEPTNO
1000	SMITH	2500	8012011234567	10
1001	ALLEN	2000	9011072212345	10
1002	FORD	3400	9506232221234	20
1003	SCOTT	3800	9801181112345	20
1004	KING	4000	9908091234432	30

```
SELECT EMPNO
FROM EMP
ORDER BY TO_CHAR(TO_NUMBER(SUBSTR(JUMIN, 3, 2)));
```

①

EMPNO
1003
1002
1004
1001
1000

②

EMPNO
1003
1001
1000
1002
1004

③

EMPNO
1001
1003
1000
1002
1004

④

EMPNO
1000
1001
1002
1003
1004

25. 아래 SQL 수행 결과로 가장 적절한 것은? (단, DBMS는 오라클)

<EMP>

ENAME	DEPTNO	SAL
SMITH	10	2000
SCOTT	10	1800
FORD	10	3200
KING	20	4000
JAMES	20	5000
ADAMS	20	NULL

```
SELECT ENAME, DEPTNO, SAL
FROM EMP
ORDER BY DEPTNO, SAL DESC;
```

①

ENAME	DEPTNO	SAL
SCOTT	10	1800
SMITH	10	2000
FORD	10	3200
KING	20	4000
JAMES	20	5000
ADAMS	20	NULL

③

ENAME	DEPTNO	SAL
FORD	10	3200
SMITH	10	2000
SCOTT	10	1800
ADAMS	20	NULL
JAMES	20	5000
KING	20	4000

②

ENAME	DEPTNO	SAL	ENAME
SCOTT	10	1800	SCOTT
SMITH	10	2000	SMITH
FORD	10	3200	FORD
ADAMS	20	NULL	ADAMS
KING	20	4000	KING
JAMES	20	5000	JAMES

④

ENAME	DEPTNO	SAL	ENAME
JAMES	20	5000	JAMES
KING	20	4000	KING
ADAMS	20	NULL	ADAMS
FORD	10	3200	FORD
SMITH	10	2000	SMITH
SCOTT	10	1800	SCOTT

26. 아래 SQL 수행 결과로 가장 적절한 것은?

<TAB1>	
COL1	COL2
1	A
2	B
3	C
4	D

<TAB2>	
COL1	COL2
1	A
2	B
2	B
4	C
5	C


```

SELECT COUNT(TAB1.COL1) AS CNT
  FROM TAB1 LEFT OUTER JOIN TAB2
    ON TAB1.COL2 = TAB2.COL2
   AND TAB1.COL1 = TAB2.COL1;

```

- ① 3
- ② 4
- ③ 5
- ④ 6

27. 다음 FROM 절의 JOIN 형태에 대한 설명 중 옳바르지 못한 것은?

- ① INNER JOIN은 WHERE 절에서 사용하던 JOIN 조건을 FROM 절에서 정의하겠다는 표시이다.
- ② INNER JOIN 사용 시, USING 조건절이나 ON 조건절을 반드시 사용해야 한다.
- ③ RIGHT OUTER JOIN 결과와 LEFT OUTER JOIN 결과는 항상 다르다.
- ④ RIGHT OUTER JOIN, LEFT OUTER JOIN에서 OUTER는 생략 가능하다.

28. 다음 중 SELECT절에 사용하는 서브쿼리인 스칼라 서브쿼리에 대한 설명으로 가장 적절하지 않은 것은?

- ① 하나의 로우에 해당하는 스칼라 서브쿼리 결과 건수는 1건 이하여야 한다.
- ② 하나의 로우에 해당하는 스칼라 서브쿼리 결과가 0건이면 생략된다.
- ③ 스칼라 서브쿼리는 반드시 한 컬럼만 출력이 가능하다.
- ④ 메인쿼리와 스칼라 서브쿼리의 연결 조건이 필요하다면 반드시 스칼라 서브쿼리에 정의해야 한다.

29. 다음 서브쿼리 결과로 가장 적절한 것은?

<EMP>		
NAME	HIREDATE	SAL
TURNER	1981/09/08	1500
ADAMS	1987/05/23	1100
JAMES	1981/10/03	1000
FORD	1981/12/03	3000
MILLER	1982/01/23	1300


```

SELECT SUM(SAL)
  FROM EMP
 WHERE HIREDATE > (SELECT HIREDATE
                   FROM TAB1
                   WHERE NAME = 'JAMES');

```

- ① 3800
- ② 4100
- ③ 5400
- ④ 6900

30. 다음 서브쿼리 결과로 가장 적절한 것은?

<EMPLOYEES>			
NAME	HIREDATE	SAL	DEPTNO
TURNER	1981/09/08	1500	10
ADAMS	1987/05/23	1100	10
JAMES	1981/10/03	1000	20
FORD	1981/12/03	3000	20
MILLER	1982/01/23	3000	20

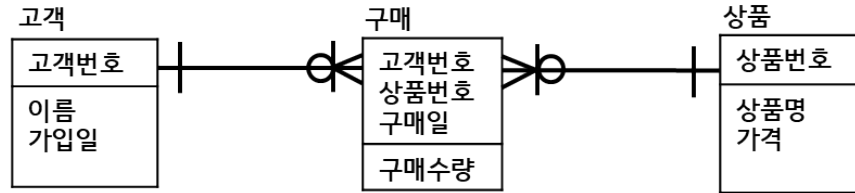

```

SELECT SUM(SAL)
  FROM EMPLOYEES
 WHERE (DEPTNO, SAL) IN (SELECT DEPTNO, MAX(SAL)
                        FROM EMPLOYEES
                        GROUP BY DEPTNO);

```

- ① 1500
- ② 3000
- ③ 4500
- ④ 7500

31. 다음 ERD를 보고 고객별로 가장 최근에 구매한 상품명(다수가능)을 출력하는 SQL로 가장 적절한 것은?



- ① SELECT G.고객번호, G.구매일, P.상품명
 FROM 구매 G, 상품 P, (SELECT 고객번호, MAX(구매일) 최근구매일
 FROM 구매
 GROUP BY 고객번호) I
 WHERE G.고객번호 = I.고객번호
 AND G.구매일 = I.최근구매일
 AND G.상품번호 = P.상품번호;
- ② SELECT G.고객번호, G.구매일, P.상품명
 FROM 고객 G, 상품 P, (SELECT 고객번호, MAX(구매일) 최근구매일
 FROM 구매
 GROUP BY 고객번호) I
 WHERE G.고객번호 = I.고객번호
 AND G.상품번호 = P.상품번호;
- ③ SELECT C.이름, G.구매일, P.상품명
 FROM 고객 C, 구매 G, 상품 P, (SELECT 고객번호, MIN(구매일) 최근구매일
 FROM 구매
 GROUP BY 고객번호) I
 WHERE G.고객번호 = I.고객번호
 AND G.구매일 = I.최근구매일
 AND G.상품번호 = P.상품번호
 AND C.고객번호 = G.고객번호;
- ④ SELECT C.이름, G.구매일, P.상품명
 FROM 고객 C, 구매 G, 상품 P
 WHERE G.상품번호 = P.상품번호
 AND C.고객번호 = G.고객번호;

32. 다음 쿼리의 수행 결과로 적절한 것은?

<TAB1>			<TAB2>	
NAME	CODE	FARE	CODE	STATUS
AAA	0001	500	0001	CLOSED
BBB	0001	100	0002	OPEN
CCC	0002	300	0003	CLOSED
DDD	0004	200	0004	OPEN


```

DELETE FROM TAB1
  WHERE CODE IN (SELECT CODE
                  FROM TAB2
                  WHERE STATUS = 'OPEN');

SELECT SUM(FARE)
  FROM TAB1;

```

- ① 100
- ② 400
- ③ 600
- ④ 800

33. 다음 수행 결과로 가장 적절한 것은?

<TAB1>		<TAB2>	
COL1	COL2	CODE	FARE
A	100	0001	250
B	200	0002	350
C	300		
D	400		


```

SELECT SUM(COL2)
  FROM TAB1
 WHERE COL2 < ANY(SELECT FARE
                  FROM TAB2);

```

- ① 100
- ② 300
- ③ 600
- ④ 1000

34. 다음 집합 연산자에 대한 설명 중 틀린 것은 무엇인가? (단, DBMS는 오라클)

- ① UNION 연산자는 조회 결과에 대한 합집합을 나타내며 정렬된 결과를 출력해준다.
- ② UNION ALL 연산자는 조회 결과를 정렬하고 중복되는 데이터를 한 번만 표현한다.
- ③ INTERSECT 연산자는 조회 결과에 대한 교집합을 의미한다.
- ④ MINUS 연산자는 조회 결과에 대한 차집합을 의미한다.

35. 아래 쿼리 결과와 같은 결과를 갖는 빈칸에 들어갈 문장으로 가장 적절한 것은?

```
SELECT DEPTNO, SUM(SAL) AS SUM_SAL
  FROM EMP
 GROUP BY DEPTNO
 UNION ALL
SELECT NULL DEPTNO, SUM(SAL) AS SUM_SAL
  FROM EMP;

SELECT DEPTNO, SUM(SAL) AS SUM_SAL
  FROM EMP
 GROUP BY _____;
```

- ① ROLLUP(DEPTNO)
- ② ROLLUP(SAL)
- ③ ROLLUP(DEPTNO, SAL)
- ④ ROLLUP(DEPTNO, ())

36. 순위관련 WINDOW 함수에 대한 설명 중 가장 적절하지 않은 것은?

- ① RANK함수는 동일한 값에 대해서는 동일한 순위를 부여한다.
- ② DENSE_RANK 함수는 RANK 함수처럼 동일한 값에 대해 동일한 순위를 부여하나, 동순위가 여럿 존재하더라도 다음 순위가 이어진다.
- ③ PERCENT_RANK 함수는 각 값의 누적된 순위를 부여할 수 있다.
- ④ RANK 함수가 동일한 값에 대해서는 동일한 순위를 부여하는데 반해, ROW_NUMBER 함수는 고유한 순위를 부여한다.

37. 다음 출력 결과를 갖도록 하는 빈칸의 문장으로 가장 적절한 것은?

<EMP>

EMPNO	ENAME	DEPTNO	SAL
7934	MILLER	10	1300
7782	CLARK	10	2450
7839	KING	10	5000
7369	SMITH	20	800
7876	ADAMS	20	1100
7566	JONES	20	2975
7788	SCOTT	20	3000
7902	FORD	20	3000

<RESULT>

EMPNO	ENAME	DEPTNO	SAL	RESULT
7934	MILLER	10	1300	3750
7782	CLARK	10	2450	8750
7839	KING	10	5000	7450
7369	SMITH	20	800	1900
7876	ADAMS	20	1100	4875
7566	JONES	20	2975	7075
7788	SCOTT	20	3000	8975
7902	FORD	20	3000	6000

```
SELECT EMPNO, ENAME, DEPTNO, SAL,
       SUM(SAL) OVER(PARTITION BY DEPTNO ORDER BY SAL
                     _____) AS RESULT
FROM EMP;
```

- ① RANGE BETWEEN UNBOUNDED PRECEDING AND 1 FOLLOWING
- ② RANGE BETWEEN 1 PRECEDING AND 1 FOLLOWING
- ③ ROWS BETWEEN UNBOUNDED PRECEDING AND 1 FOLLOWING
- ④ ROWS BETWEEN 1 PRECEDING AND 1 FOLLOWING

38. 다음 중 Top N Query에 대한 설명 중 틀린 것은?

- ① 윈도우 함수를 사용하여 상위 N개에 대한 값을 추출할 수 있으나 단일 Query로 표현 불가하다.
- ② ROWNUM을 사용한 방식은 ROWNUM 할당 전에 먼저 순서대로 데이터를 정렬한 뒤 ROWNUM을 부여 후 추출하는 것이 좋다.
- ③ FETCH 절을 사용하면 단일 Query로도 정렬 순서대로의 상위 N개에 대한 값을 추출할 수 있다.
- ④ SQL-Server의 TOP(N) 쿼리를 사용하면 정렬 순서대로 상위 N개 데이터만 출력 가능하다.

39. 아래 실행 결과를 출력하는 SQL로 가장 적절한 것은?

<사원>

사원번호	이름	상위관리자코드	지역
1000	홍길동	NULL	서울
1001	박길동	1000	경기
1002	최길동	1001	경기
1003	이길동	1001	인천
1004	구길동	1002	서울
1005	안길동	1003	서울
1006	송길동	1000	경기
1007	강길동	1006	경기
1008	공길동	1006	인천

```
SELECT 사원번호, 이름, LEVEL
      FROM 사원
     WHERE 지역 = '경기'
    START WITH 상위관리자코드 IS NULL
   CONNECT BY 상위관리자코드 = PRIOR 사원번호;
```

①

사원번호	이름	LEVEL
1001	박길동	2
1002	최길동	3
1006	송길동	2
1007	강길동	3

②

사원번호	이름	LEVEL
1000	홍길동	1
1001	박길동	2
1002	최길동	3
1006	송길동	2
1007	강길동	3

③ 공집합

④

사원번호	이름	LEVEL
1000	홍길동	1

40. 다음 수행 결과로 가장 적절한 것은?

<DEPARTMENT>

DEPTNO	DNAME	PART	BUILD
101	컴퓨터공학과	100	정보관
102	멀티미디어공학과	100	멀티미디어관
103	소프트웨어공학과	100	소프트웨어관
201	전자공학과	200	전자제어관
202	기계공학과	200	기계실험관
203	화학공학과	200	화학실습관
301	문헌정보학과	300	인문관
100	컴퓨터정보학부	10	
200	메카트로닉스학부	10	
300	인문사회학부	20	
10	공과대학		
20	인문대학		

```
SELECT DEPTNO, DNAME, LEVEL, CONNECT_BY_ROOT(DNAME) AS ROOT
  FROM DEPARTMENT
 START WITH PART IS NULL
CONNECT BY PRIOR PART = DEPTNO;
```

① 공집합

②

DEPTNO	DNAME	LEVEL	ROOT
10	공과대학	1	공과대학
20	인문대학	1	인문대학

③

DEPTNO	DNAME	LEVEL	ROOT
10	공과대학	1	공과대학
100	컴퓨터정보학부	2	공과대학
101	컴퓨터공학과	3	공과대학
102	멀티미디어공학과	3	공과대학
103	소프트웨어공학과	3	공과대학
200	메카트로닉스학부	2	공과대학
201	전자공학과	3	공과대학
202	기계공학과	3	공과대학
203	화학공학과	3	공과대학

④

DEPTNO	DNAME	LEVEL	ROOT
10	공과대학	1	공과대학
100	컴퓨터정보학부	2	공과대학
101	컴퓨터공학과	3	공과대학
102	멀티미디어공학과	3	공과대학
103	소프트웨어공학과	3	공과대학
200	메카트로닉스학부	2	공과대학
201	전자공학과	3	공과대학
202	기계공학과	3	공과대학
203	화학공학과	3	공과대학
20	인문대학	1	인문대학
300	인문사회학부	2	인문대학
301	문헌정보학과	3	인문대학

41. 다음 SQL의 실행 결과를 얻기 위한 빈칸에 들어갈 값으로 가장 적절한 것은?

<TAB1>		
성별	2023	2024
남자	10	20
여자	30	40
<결과>		
성별	연도	판매량
남자	2023	10
남자	2024	20
여자	2023	30
여자	2024	40
SELECT * FROM TAB1 UNPIVOT (_____ FOR _____ IN ("2023", "2024"));		

- ① 판매량, 연도
- ② 성별, 연도
- ③ 성별, 판매량
- ④ 판매량, 판매량

42. 다음 SQL 실행 결과로 알맞은 것은?

```
SELECT LENGTH(REGEXP_REPLACE('REGEXP \. ESCAPE CHARACTER, A|B : A OR B', '[A-z|0-9\.]'))
FROM DUAL;
```

- ① 2
- ② 3
- ③ 4
- ④ 5

43. DML에 대한 설명으로 가장 적절한 것은?

- ① DELETE 사용 시 FROM 문구는 생략이 불가능하다.
- ② 일부 데이터 DELETE 시 WHERE 절은 반드시 붙이지 않아도 된다.
- ③ 반드시 COMMIT 또는 ROLLBACK을 수행하여 TRANSACTION을 종료해야 한다.
- ④ UPDATE 사용 시 동시에 여러 컬럼 수정은 불가능하다.

44. COMMIT 이후의 데이터 상태로 옳지 않은 것은?

- ① 데이터에 대한 변경 사항이 데이터베이스에 영구 저장된다.
- ② 변경된 행에 대한 잠금이 풀리고 다른 사용자들이 행을 조작할 수 있다.
- ③ 이전 데이터는 영원히 되돌릴 수 없다.
- ④ COMMIT을 수행한 사용자만 결과를 볼 수 있다.

45. 컬럼 변경 시 주의 사항으로 옳지 않은 것은?

- ① 컬럼의 크기를 늘릴 수는 있지만 줄일 수는 없다.
- ② 컬럼이 NULL 값만 가지고 있으면 데이터 유형을 변경할 수 있다.
- ③ 컬럼에 NULL 값이 없을 경우에만 NOT NULL 제약조건을 추가할 수 있다.
- ④ 컬럼의 DEFAULT 값을 바꾸면 변경 작업 이후 발생하는 행 삽입에만 영향을 미친다.

46. 제약조건의 설명 중 가장 적절하지 않은 것은?

- ① 기본키는 테이블에 저장된 행 데이터를 고유하게 식별하기 위한 키이다.
- ② 테이블에 저장된 행 데이터를 고유하게 식별하기 위해 고유키를 정의한다.
- ③ 기본키는 고유키와 외래키 제약을 합쳐놓은 것이다.
- ④ NOT NULL 은 NULL 값의 삽입을 금지한다.

47. 아래와 같이 테이블 및 데이터가 생성된 경우 추가 실행이 불가능한 문장은?

(단, 보기 순서대로 실행됨을 가정)

```
CREATE TABLE TAB1(COL1 NUMBER, COL2 NUMBER);
INSERT INTO TAB1 VALUES(100, 100);
COMMIT;
```

- ① ALTER TABLE TAB1 ADD (COL3 NUMBER, COL4 VARCHAR2(10));
- ② ALTER TABLE TAB1 ADD COL5 NUMBER NOT NULL;
- ③ ALTER TABLE TAB1 MODIFY COL2 DEFAULT 100 NOT NULL;
- ④ ALTER TABLE TAB1 DROP COLUMN COL4;

48. 다음 문장의 수행 후 TAB2의 조회 결과로 가장 적절한 것은?

<TAB1>		<TAB2>		
NO	NAME	NO	NAME	CLASS_NO
1	A	1000	SMITH	1
2	B	1001	ALLEN	2
3	C	1002	FORD	3
4	D	1004	SCOTT	3

ALTER TABLE TAB2 ADD FOREIGN KEY(CLASS_NO) REFERENCES TAB1(NO) ON DELETE SET NULL;
DELETE FROM TAB1 WHERE NAME = 'C';

① DELETE 실행 오류

②

NO	NAME	CLASS_NO
1000	SMITH	1
1001	ALLEN	2

③

NO	NAME	CLASS_NO
1000	SMITH	1
1001	ALLEN	2
1002	FORD	NULL
1004	SCOTT	NULL

④

NO	NAME	CLASS_NO
1000	SMITH	1
1001	ALLEN	2
1002	FORD	3
1004	SCOTT	3

49. 다음 설명 중 가장 적절하지 않은 것은?

- ① 외래키를 생성 한 경우 부모 테이블의 참조키 컬럼을 삭제할 수 없다.
- ② 제약 조건 추가 시 제약조건 이름을 명시하지 않을 수 있다.
- ③ 이미 존재하는 컬럼에 대해 NOT NULL 제약조건 추가 시 반드시 MODIFY로 처리한다.
- ④ NULL값이 삽입되어 있는 경우 UNIQUE 제약조건을 추가할 수 없다.

50. 권한에 대한 설명으로 가장 적절한 것은?

- ① 권한은 테이블 소유자만이 부여할 수 있다.
- ② 테이블에 대한 조회 권한 부여 시 즉시 반영되지 않고 재접속을 해야 조회가 가능하다.
- ③ 롤에 있는 권한을 회수한 이후 롤을 부여받은 유저는 해당 권한을 갖지 않게 된다.
- ④ WITH ADMIN OPTION을 통해 부여받은 테이블 조회 권한을 다른 유저에게 부여할 수 있다.

답안 및 해설

1. ①

도메인은 속성값이 갖는 범위를 의미한다.

2. ③

하나의 속성은 한 개의 속성값을 가져야 한다.

3. ④

속성은 엔터티에 속한 엔터티에 대한 자세하고 구체적인 정보를 나타낸다.

4. ②

관계의 차수에 대한 설명이다.

5. ②

주식별자는 NULL값이 들어갈 수 없다.

6. ①

제 1 정규화는 한 속성이 하나의 값을 갖도록 이를 분해하는 단계를 말한다.

7. ③

두 엔터티나 두 속성 간에 동시에 발생할 수 없는 관계는 상호 배타적 관계이다.

8. ③

하나의 트랜잭션은 부분 COMMIT이 불가하다.

9. ④

공백과는 다른 개념이므로 공백과 다른 ASCII 값을 갖는다.

10. ④

다른 엔터티 참조 없이 엔터티 내부에서 스스로 생성되는 식별자는 내부식별자이다.

11. ①

DBMS는 데이터를 중앙 집중화하여 여러 응용프로그램이 데이터를 공유하고 사용할 수 있도록 하는 소프트웨어이다.

12. ②

테이블명과 컬럼명은 반드시 문자로 시작해야 한다. 또한, 소유자가 다른 경우 같은 이름의 테이블을 생성할 수 있다. 즉, SCOTT.TABLE1과 HR.TABLE1은 서로 다른 테이블로 존재할 수 있다.

13. ③

GROUP BY 절에 명시되지 않은 컬럼은 ORDER BY 절에 사용할 수 없다.

14. ③

DISTINCT COL1, COL2의 경우 두 컬럼의 값이 모두 같은 집합을 중복값으로 간주, 하나만 출력하기 때문에 DISTINCT COL2, COL1의 결과와 순서만 다른 집합의 수는 동일하다.

15. ③

ROUND와 TRUNC는 각각 날짜의 반올림과 버림을 수행할 수 있다. 두 번째 인수 생략 시 "일" 단위로의 반올림/버림이 진행되며, 'MONTH'의 경우 "월" 단위로의 반올림/버림이 진행된다. 즉, "일" 단위에서 반올림을 진행하게 된다.

16. ①

CEIL은 값보다 크면서 가장 작은 정수인 올림값을 리턴한다. 따라서 -12.345보다 값이 크면서 가장 작은 정수는 -12이다. 반대로 FLOOR는 값보다 작으면서 가장 큰 정수인 내림값을 리턴하므로 FLOOR(-12.345) 값은 -13이다.

17. ③

LTRIM은 왼쪽에서부터 특정 문자열을 지우며, 중간에 있는 문자열은 삭제되지 않는다. 따라서 LTRIM('ORACLE','A')은 왼쪽에 A가 없으므로 ORACLE 그대로 리턴된다. SUBSTR('SQL-SERVER', 3, 3)은 세 번째 위치에서 3개 문자열을 추출하기 때문에 L-S가 출력된다. 또한, REPLACE에 의해 E가 삭제된 문자열의 길이는 8이 된다.

18. ②

2번은 에러가 발생하는 문장으로 마지막 WHEN절에만 ELSE를 사용할 수 있다.

19. ④

오라클에서는 묵시적 형 변환으로 인해 숫자로 변환 가능한 문자값과 숫자값의 연산이 가능하다. 또한 "일"만 있는 경우 날짜로 변환하면, 현재 날짜의 연도와 월을 따른다. NVL의 경우 첫 번째와 두 번째 인수의 데이터 유형이 일치해야 한다.

20. ③

첫 번째 문장은 조건에 만족하는 COL1값이 NULL이므로 NULL 그룹이 리턴된다. 따라서 COUNT를 하면 0이 출력되며, 두 번째 문장은 HAVING 조건에 만족하는 그룹이 없으므로 공집합이 출력되어 COUNT 결과가 NULL이 된다.

21. ③

COL1이 NULL이 아닌 값은 위의 세 행인데, 이들의 COL2+COL3의 값은 순서대로 NULL, 12, NULL이 된다. 따라서 총 합은 12가 출력된다.

22. ①

NULL은 일반적인 비교연산을 수행할 수 없고 IS NULL, IS NOT NULL로 비교해야 한다.

23. ④

① 컬럼 별칭에 공백이 있는 경우 쌍따옴표로 묶어서 전달해야 한다.

② GROUP BY절에 있지 않은 컬럼을 SELECT절에 그룹함수 없이 전달할 수 없다.

③ WHERE절에는 그룹함수를 사용할 수 없고 그룹함수를 사용한 비교식은 HAVING절에 사용 가능하다.

24. ②

SUBSTR(JUMIN, 3, 2)은 태어난 월을 추출하기 때문에, 순서대로 12, 11, 06, 01, 08의 문자 유형으로 출력된다.

이를 TO_NUMBER를 사용하여 숫자 형태로 변환하면 12, 11, 6, 1, 8이 되는데, 이를 다시 TO_CHAR로 변환하게 되면 문자값의 비교 규칙에 따라 $1 < 11 < 12 < 6 < 8$ 순서대로 출력된다. (문자는 가장 왼쪽부터 비교하여 값이 작을수록 작은값이 된다)

25. ③

DEPTNO가 작은순서대로, DEPTNO가 같은 경우 SAL이 큰 순서대로 정렬하여 출력한다.

NULL이 마지막에 출력되는게 기본 순서이지만 DESC로 내림차순 정렬하면 NULL이 젤 먼저 출력된다.

26. ③

조인 결과는 아래와 같다

TAB1_COL1	TAB1_COL2	TAB2_COL1	TAB2_COL2
1	A	1	A
2	B	2	B
2	B	2	B
4	D		
3	C		

따라서 COUNT(TAB1.COL1)의 결과는 5 개이다

27. ③

두 테이블 조인 시, 조인 조건에 의해 생략되는 쪽이 둘 다 없을 경우 LEFT OUTER JOIN / RIGHT OUTER JOIN 결과는 같다.

28. ②

하나의 로우에 해당하는 스칼라 서브쿼리 결과 건수가 0건이더라도 메인쿼리절에서 생략을 하지 않는 한 NULL로 출력된다.

29. ③

JAMES의 입사일보다 늦게 입사한 직원들의 급여 총 합을 구하는 문제이다. JAMES의 입사일인 1981/10/03 보다 늦게 입사한 직원은 ADAMS, FORD, MILLER 이며 이들의 급여 총합은 5400이다.

30. ④

다중컬럼 서브쿼리를 사용하여 각 부서별로 최대급여를 받는 직원들의 급여 총합을 출력하고 있다.

10번 부서는 1500, 20번 부서는 3000이지만 FORD와 MILLER 둘 다 출력되므로 총 합은 7500이다.

31. ①

먼저 구매테이블을 통해 고객별 최근 구매일을 파악한 뒤(인라인뷰), 해당 고객과 구매일에 구매한 상품번호를 파악하기 위해 다시 구매테이블과 조인이 필요하다.

그 뒤 상품테이블과 조인하여 해당 상품의 상품명을 알아내면 된다.

32. ③

TAB2의 STATUS가 OPEN인 CODE는 0002와 0004이므로 TAB1에서 이들을 삭제하면 AAA,BBB만 남는다.

이들의 FARE 총합은 600이다.

33. ③

ANY는 작다와 만나면 값들 중 최댓값을 리턴한다.

따라서, 메인쿼리 WHERE절은 COL2 < 350 이 되므로 A, B, C들의 COL2의 총 합은 600이 된다.

34. ②

UNION ALL은 중복된 데이터를 모두 출력하며 정렬은 발생하지 않는다.

35. ①

ROLLUP은 전체 소계를 함께 출력한다. 즉, ROLLUP(A) => GROUP BY A 결과에 전체 소계 출력

36. ③

PERCENT_RANK는 값이 아닌 행의 상대 위치를 0~1 사이값으로 반환하는 함수이다.

37. ④

누적합의 범위가 각 행마다 이전행과 현재행, 다음행을 연산하고 있으므로(JONES 기준 $1100 + 2975 + 3000 = 7075$) 1 PRECEDING AND 1 FOLLOWING 이며, SAL이 같은 SCOTT과 FORD의 누적합이 각각 다르게 계산되었으므로 ROWS가 적절하다.

38. ④

TOP(N)은 WITH TIES 를 사용하면 N개보다 더 많은 데이터 추출이 가능하다.

39. ①

WHERE절은 출력 대상을 결정하기 때문에 서울 지역인 홍길동은 출력하지 않는다.

40. ②

PRIOR의 위치가 PART에 있으므로 가장 최상위 학과(PART IS NULL)를 먼저 출력하고, 두 행의 PART를 DEPTNO로 갖는 행을 찾지만 해당 행이 없으므로 최상위 학과인 공과대학과 인문대학만 출력된다.

41. ①

첫 번째 밑줄은 10, 20, 30, 40 값이 쌓여 하나의 컬럼을 이룰 때 컬럼명을 나타내는 자리이므로 판매량,

두 번째 밑줄은 2023, 2024를 넣을 컬럼명을 의미하므로 연도가 적절하다.

42. ①

[A-z|0-9\.\] 패턴은 영문 또는 | 또는 숫자 또는 \ 또는 . 그리고 공백을 모두 지칭하는 패턴이다.

따라서 이들을 모두 지우면 ,과 :만 남게 된다.

43. ③

UPDATE로 동시 여러 컬럼 수정 가능하다. DELETE 시 FROM은 생략이 가능하다.

DML 시 COMMIT 또는 ROLLBACK으로 트랜잭션을 종료하지 않으면 변경된 행의 잠금이 발생하여

다른 사용자의 사용에 제한이 생긴다.

44. ④

데이터베이스는 공유 저장 공간이므로 COMMIT 하여 영구 저장된 데이터는 다른 사용자에게 공유된다.

45. ①

컬럼 크기를 늘리는 것은 언제든지 가능하며 반대로 줄이는 것은 해당 컬럼의 최대 길이만큼 줄일 수 있다.

46. ③

기본키는 고유키와 NOT NULL 제약조건을 합쳐 놓은 것과 같다. 즉, 중복될 수 없으며 NULL이 삽입될 수 없다.

47. ②

새로운 컬럼 추가 시 기존 데이터의 새 컬럼 데이터는 NULL로 삽입된다. 따라서 DEFAULT 값 선언 없이 NOT NULL 속성을 갖는 컬럼 추가는 불가능하다.

48. ③

ON DELETE SET NULL 옵션에 의해 TAB1 데이터 삭제 시 자식 데이터의 외래키 컬럼은 NULL로 수정된다.

49. ④

NULL값이 있더라도 중복된 값만 없다면 UNIQUE 제약조건을 추가할 수 있다.

50. ③

롤에 있는 권한을 회수하는 경우 롤을 부여받은 유저도 해당 권한을 즉시 잃게 된다. 4번 문장은 WITH GRANT OPTION에 대한 설명이다.