

이름	일분기	이분기	삼분기	사분기
루피	90	70	80	80
조로	(null)	77	(null)	100
상디	88	(null)	78	98
나미	(null)	95	(null)	100
쵸파	(null)	(null)	(null)	50

[실적 : 테이블]

다음 SQL의 결과셋을 완성하시오

1. select 이름, nvl(일분기, 0) from 실적;

루피	
조로	
상디	
나미	
쵸파	

2. select 이름, nvl2(일분기, 'success', 'fail') from 실적;

루피	
조로	
상디	
나미	
쵸파	

3. select 이름, nullif(사분기, 100) from 실적;

루피	
조로	
상디	
나미	
쵸파	

4. select 이름, coalesce(일분기, 이분기, 삼분기, 사분기) from 실적;

루피	
조로	
상디	
나미	
쵸파	

[학생 : 테이블]

이름	나이	성별	주소	문의	상담자
강승권	27	남	경기도	2023-04-28	김지한
황윤제	29	남	경기도	2023-04-25	박상용
김찬혁	29	남	경기도	2023-04-20	김지한
심희수	29	여	인천시	2023-04-20	김지한
전원영	32	남	서울시	2023-04-15	박상용
왕유빈	26	여	서울시	2023-04-11	이명경
남궁수민	32	남	인천시	2023-04-11	이명경
송윤호	26	남	경기도	2023-04-06	박상용
이승환	29	남	대전시	2023-04-01	이명경
허이	22	여	다낭	2023-03-03	한국

[사원]

이름	파트	입사년도	파트장
김지한	모집	2020	한국
박상용	모집	2021	한국
이명경	모집	2021	한국
최지영	교육	2021	null
이희인	취업	2019	null
한국	모집	2019	null
김선민	취업	2019	이희인
이에인	취업	2021	이희인

## 5. 에러가 발생하는 SQL은?

- 1) select \* from 학생 order by 나이
- 2) select \* from 학생 order by 2
- 3) select 이름 from 학생 order by 나이
- 4) select 이름 from 학생 order by 2

## 6. 학생 성별 오름차순, 성별이 같으면 주소 오름차순, 주소까지 같으면 나이 내림차순 SQL은?

- 1) SELECT 성별, 주소, 나이 FROM 학생 ORDER BY 성별 DESC, 주소 DESC, 나이 ASC
- 2) SELECT 성별, 주소, 나이 FROM 학생 ORDER BY 성별 ASC, 주소 ASC, 나이
- 3) SELECT 성별, 주소, 나이 FROM 학생 ORDER BY 성별 ASC, 주소, 3 DESC
- 4) SELECT 성별, 주소, 나이 FROM 학생 ORDER BY 성별, 주소, DESC 나이

## 7. 테이블이 다음과 같을 때 아래의 SQL연산결과로 알맞은 것은?

과목

국어	영어	수학	과학
1	1	1	1
null	1	null	null
3	null	3	3
null	4	null	4

SELECT SUM(국어+영어+수학+과학) FROM 과목

SELECT SUM(국어)+SUM(영어)+SUM(수학)+SUM(과학) FROM 과목

- 1) 4, NULL
- 2) NULL, 22
- 3) 4, 22
- 4) NULL, NULL

### 8. 다음 SQL의 결과는?

select 성별, count(\*) from 학생 group by 성별 having count(\*)>7

1) 

성별	count(*)
----	----------

2) 

성별	count(*)
여	0
남	0

3) 

성별	count(*)
여	null
남	null

4) 

성별	count(*)
여	3
남	7

### 9. 다음 보기 중 COL1이 NULL이 아닌 데이터를 찾는 SQL로 올바른 것은?

- 1) SELECT COL1 FROM WHERE T1 WHERE COL1 <> "
- 2) SELECT COL1 FROM WHERE T1 WHERE COL1 != "
- 3) SELECT COL1 FROM WHERE T1 WHERE COL1 IS NOT NULL
- 4) SELECT COL1 FROM WHERE T1 WHERE COL1 NOT IN (NULL)

## 10. 보기의 SQL 명령어중 TCL 인 것은 어떤 것인가?

[보기]

UPDATE / GRANT / SELECT / COMMIT

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

## 11. 다음 중 SELECT 문장의 실행 순서를 올바르게 나열한 것은?

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

## 12. 다음 중 아직 COMMIT 되지 않은 데이터에 대한 설명으로 잘못된 것은?

- 1) ROLLBACK 명령어로 바로 직전에 COMMIT한 지점까지 데이터를 복구할 수 있다.
- 2) 나 자신이 볼 수 있다.
- 3) 다른 사용자가 볼 수 없다.
- 4) 다른 사용자가 COMMIT되지 않은 변경된 데이터를 고칠 수 있다.

## 13. 다음 주어진 테이블에서 수행한 SQL문의 결과값으로 잘못된 것은?

EMP 테이블

EMPNO    ENAME    SAL

-----

1	'NOAH'	1000
2	'LIAM'	2000
3	'AIDEN'	3000
4	'JAMES'	4000
5	'ETHAN'	5000
6	'OLIVER'	6000

- 1)  
SELECT ENAME, SAL

FROM (SELECT ENAME, SAL FROM EMP ORDER BY SAL DESC)  
 WHERE ROWNUM = 1;  
 => 맨 끝에 데이터가 추출된다.

2)  
 SELECT ENAME, SAL  
 FROM (SELECT ENAME, SAL FROM EMP ORDER BY SAL DESC)  
 WHERE ROWNUM = 2;  
 => 끝에서 2건의 데이터가 추출된다.

3)  
 SELECT ENAME, SAL  
 FROM (SELECT ENAME, SAL FROM EMP ORDER BY SAL DESC)  
 WHERE ROWNUM > 0;  
 => 밑에서부터 위까지 모든 데이터가 추출된다.

4)  
 SELECT ENAME, SAL  
 FROM (SELECT ENAME, SAL FROM EMP ORDER BY SAL DESC)  
 WHERE ROWNUM <= 3;  
 => 끝에서 3건의 데이터가 추출된다.

#### 14. 다음 SQL의 결과값은

select 5+null from dual;

#### 15. 다음 보기 중 주어진 테이블에서 아래의 SQL문의 실행결과로 가장 적절한 것은?

[훈련생]

ID
1000
1000
1000
3000
3000
4000
9999
9999

[SQL]

SELECT ID FROM 훈련생  
 GROUP BY ID

HAVING COUNT(\*) = 2  
ORDER BY (CASE WHEN ID = 1000 THEN 0 ELSE ID END);

(1)	(2)
ID	ID
3000	9999
9999	3000

(3)	(4)
ID	ID
1000	999
3000	3000
9999	1000

### 16. 사원테이블에서 다음 SQL의 결과는?

```
select count(*), count(이름), count(파트장)
from 사원;
```

- 1) 8 8 5
- 2) 5 5 5
- 3) 8 8 8
4. 8 8 null

### 17. 다음 주어진 SQL문에서 오류가 발생하지 않는 것은?

```
CREATE TABLE 훈련생 (
    ID NUMBER PRIMARY KEY,
    AGE NUMBER NOT NULL,
    NAME VARCHAR2(1)
);
```

[emp : 테이블]

- (1) INSERT INTO 훈련생 VALUES(10,20,SYSDATE);
- (2) INSERT INTO 훈련생 VALUES(20,NULL,'A');
- (3) INSERT INTO 훈련생(AGE, NAME) VALUES(20,'A');
- (4) INSERT INTO 훈련생(ID, AGE, NAME) VALUES(20,10,NULL);

### 18. 아래의 테이블에 대해 다음의 SQL을 수행한 결과로 알맞은 것은?

ID	DEPT_NM	SALARY
1	A	1000
1	A	100
2	B	500
2	B	4000

2	B	10
3	C	150
3	C	10

```
SELECT ID, DEPT_NM, SUM(SALARY)
FROM emp
GROUP BY ROLLUP(ID,DEPT_NM);
```

1)

ID	DEPT_NM	SUM(SALARY)
1	A	1100
2	B	4510
3	C	160

2)

ID	DEPT_NM	SUM(SALARY)
1	A	1100
1	NULL	1100
2	B	4510
2	NULL	4510
3	C	160
3	NULL	160
NULL	NULL	5770

3)

ID	DEPT_NM	SUM(SALARY)
1	NULL	1100
2	NULL	1100
3	NULL	160
NULL	A	1100
NULL	B	4510
NULL	C	160

4)

ID	DEPT_NM	SUM(SALARY)
NULL	NULL	5770
NULL	A	110
NULL	B	4510
NULL	C	160
1	NULL	1100
1	A	1100
2	NULL	4510
2	B	4510
3	NULL	160

19. SQL 를 수행한 결과가 다음과 같을 때 수행한 SQL의 빈칸에 넣을 알맞은 그룹함수는?

```
SELECT ID, DEPT_NM, SUM(AMT)
FROM 테이블이름
GROUP BY ( )
```

ID	DEPT_NM	"SUM(AMT)"
<null>	<null>	195
<null>	가	25
<null>	나	100
<null>	다	70
1	<null>	25
1	가	25
2	<null>	100
2	나	100
3	<null>	70
3	다	70

1) CUBE (ID, DEPT\_NM)

- 2) ROLLUP (ID, DEPT\_NM)
- 3) GROUPING SETS (ID, DEPT\_NM)
- 4) CUBE (ID)

**20. 아래의 GROUP 함수에 대한 설명으로 가장 적절한 것은?**

- 1) CUBE는 결합 가능한 모든 값에 대하여 다차원 집계를 생성한다.
- 2) ROLLUP은 계층구조가 평등한 관계이므로 인수의 순서가 바뀌어도 결과는 같다.
- 3) ROLLUP, CUBE, GROUPING SETS 은 특정 컬럼에 대한 정렬은 가능하나 계층간 정렬은 불가능하다.
- 4) ROLLUP은 CUBE에 비해 시스템에 많은 부담을 주므로 사용에 주의해야 한다.