1. **숫자형 함수 적용과 그 결과값이 아닌 것은 ?**
2. ABS(-20) = 20
3. SIGN(-50) = -1
4. MOD(7,3) = 2
5. CEIL(38,123) = 39
6. FLOOR(38,123) = 38
7. **다음 SQL 의 결과값은 무엇일까요?**

SELECT ltrim('Hello', 'e')

,substr('GREEN TEA', 6, 2)

,length(substr('GREEN TEA', 6, 2))

FROM dual;

( , , )

1. **SQL문장에서 집계함수의 조회조건을 제한하기 위하여 사용하는 절은 ?**

**( 절)**

1. **다음 함수 설명이 Oracle에서 사용 되는 문자형 함수이름은?**<설명>   
   문자열의 첫 문자 부터 확인해서 지정 문자가 나타나면 해당 문자를 제거한다.
2. **다음 숫자형 함수 사용시 결과값이 올바르지 않은 것은?**

**<양식> 함수 사용 - 함수 결과**

1)CEIL(-38.123) - -38  
2) FLOOR(-38.123) - -38  
3) MOD(8,3) - 2  
4) SIGN(0) - 0  
5) ROUND(38.5235) – 39

1. **해당 Query문은 정상적으로 조회가 되지 않는다.  
   조회가 되지 않는 이유는 무엇이며, 조회가 되기 위해선 어떻게 쿼리를 수정해야 하나요?**

<Query>  
select department\_id,  
       count(\*)  
from employees   
where department\_id is not null  
  and count(\*) <= 5  
group by department\_id  
order by department\_id

1. **아래 SQL 의 결과값은?**

|  |
| --- |
| WITH t1 AS  (  SELECT LEVEL lv  FROM dual  CONNECT BY LEVEL < 11  )  SELECT SUM(lv)  FROM t1  WHERE 1=1  AND ( CASE ROUND(lv/2)  WHEN lv/2  THEN 1  ELSE 2  END ) = 1 |

1. 24
2. 26
3. 28
4. 30
5. **아래 SQL 의 결과값은 ?**

|  |
| --- |
| SELECT  ROUND( to\_date( '2016-03-28 12:00:00', 'YYYY-MM-DD HH24:MI:SS' ) ) col1,  ROUND( to\_date( '2016-03-28 12:00:01', 'YYYY-MM-DD HH:MI:SS' ) ) col2  FROM dual |

**Col1 / Col2**

1. 2016-03-28 2016-03-28
2. 2016-03-28 2016-03-29
3. 2016-03-29 2016-03-28
4. 2016-03-29 2016-03-29
5. **SQL의 실행순서를 나열하세요.**

|  |
| --- |
| **ㄱ. select**  **ㄴ. from**  **ㄷ. where**  **ㄹ. group by**  **ㅁ. having**  **ㅂ. order by** |

1. **다음중 에러가 나지 않는 문장은?**
2. SELECT position 포지션, avg(height) 평균키

FROM player;

1. SELECT position 포지션, avg(height) 평균키

FROM player;

GROUP BY 포지션;.

1. SELECT position 포지션, round(avg(height),2) 평균키

FROM player

WHERE avg(height) >= 180

GROUP BY position;

1. SELECT position 포지션, round(avg(height),2) 평균키

FROM player

GROUP BY position

HAVING avg(height) >= 180;

1. **하기 두 쿼리의 결과가 같으면 O, 다르면 X를 선택하시오.**

SELECT player\_name 선수명, position 포지션, back\_no 백넘버

FROM player

ORDER BY 포지션 desc;

SELECT player\_name 선수명, position 포지션, back\_no 백넘버

FROM player

ORDER BY 2

1. **두 쿼리의 결과가 같으면 O, 다르면 X를 선택하시오**

SELECT ename, sal

FROM emp

WHERE rownum < 6

ORDER BY sal DESC;

SELECT ename, sal

FROM (SELECT ename, sal

FROM emp

ORDER BY sal DESC)

WHERE ROWNUM < 6;

1. **데이터가 아래와 같이 있을 때, 쿼리 결과값이 다음과 같이 출력이 되도록 쿼리를 작성하시오.**

N01 N02 N03 N04 N05 N06

1 2 3 4 5 6

2 10 20 30 40 50

3 300 400

4 4000 5000

5 5000 5

6 6000 3000

<결과> (Table 명은 편의상 Tab으로 한다.)

1 2

2 10

3 300

4 4000

5 5000

6 6000

1. **하기와 같은 결과가 나오도록 쿼리를 완성하시오.(단, 형변환(To\_char, To\_date)는 사용하지 않는다)**

COL1 CREATION\_DATE

1 6 2016-04-30 오후 11:59:59

2 50 2016-01-01 오전 12:00:01

3 2016-02-28 오전 11:59:59

4 5000 2016-01-28 오후 6:59:31

5 5 2016-03-21 오후 1:59:12

6 2016-03-31 오후 11:59:01

select x.col1, x.creation\_date

from tab x

where 1=1

and x.creation\_date between ( ) and ( )

1. **다음 중 아래 SQL에 대한 설명으로 가장 부적절한 것은?**

SELECT SUM(NVL(A.C2, 0))

FROM TAB1 A, TAB2 B

WHERE A.C1 = :V1

1. 조인 조건이 없어서 결과 건수가 여러 건이 된다.
2. 조인 조건이 없다고 문법 오류가 발생하지는 않는다.
3. SUM(NVL(A.C2, 0))의 처리에 비효율이 존재한다.
4. 조인 조건이 없어서 CARTESIAN PRODUCT이 발생한다.
5. **다음 설명 중 틀린 하나는?**
6. ORDER BY 절에서 쿼리 내 SELECT 절에 정의되지 않은 컬럼도 사용할 수 있다.
7. 권장하진 않지만, 복수의 테이블을 JOIN하여 WHERE 절을 작성할 시, 문장 내 유일하게 사용하는 컬럼이라면 ALIAS를 붙이지 않아도 된다.
8. “=” 연산자가 아닌 다른 연산자를 사용할 JOIN은 모두 NON EQUI JOIN이다.
9. SELECT 문은 SELECT 절이 가장 먼저 실행된다.
10. GROUP BY 절에서는 SELECT 절과는 달리 ALIAS 명을 사용할 수 없다.
11. **아래 두 SQL문의 결과가 같으면 O, 다르면 X. 이유도 간단하게 서술해 주세요.**
12. SELECT qlht.name,

qllv.product\_attr\_value,

qllv.product\_attr\_val\_disp,

qllv.product\_uom\_code,

COUNT(\*)

FROM qp\_list\_lines\_v qllv

,qp\_list\_headers\_tl qlht

WHERE product\_attribute = 'PRICING\_ATTRIBUTE1'

AND EXISTS (SELECT '1'

FROM mtl\_system\_items mtl

WHERE product\_attr\_value = mtl.inventory\_item\_id

AND mtl.organization\_id =

(SELECT qp\_util.get\_item\_validation\_org FROM dual)

AND product\_attribute\_context = 'ITEM'

AND product\_attribute = 'PRICING\_ATTRIBUTE1'

UNION

SELECT '1'

FROM dual

WHERE product\_attribute\_context = 'ITEM'

AND product\_attribute != 'PRICING\_ATTRIBUTE1')

AND (product\_attribute\_context = 'ITEM')

AND qllv.start\_date\_active < SYSDATE

AND NVL(qllv.end\_date\_active, SYSDATE) >= SYSDATE

AND qllv.list\_header\_id = qlht.list\_header\_id

HAVING COUNT(\*) > 1

GROUP BY qlht.name

,qllv.product\_attr\_value

,qllv.product\_attr\_val\_disp

,qllv.product\_uom\_code

Order by 1

1. SELECT qlht.name,

qllv.product\_attr\_value,

qllv.product\_attr\_val\_disp,

qllv.product\_uom\_code,

COUNT(\*)

FROM qp\_list\_lines\_v qllv

,qp\_list\_headers\_tl qlht

WHERE product\_attribute = 'PRICING\_ATTRIBUTE1'

AND EXISTS

(SELECT '1'

FROM mtl\_system\_items mtl

WHERE product\_attr\_value = mtl.inventory\_item\_id

AND mtl.organization\_id = (SELECT qp\_util.get\_item\_validation\_org

FROM dual) AND product\_attribute\_context = 'ITEM'

AND product\_attribute = 'PRICING\_ATTRIBUTE1' UNION

SELECT '1'

FROM dual

WHERE product\_attribute\_context = 'ITEM'

AND product\_attribute != 'PRICING\_ATTRIBUTE1')

AND (product\_attribute\_context = 'ITEM')

AND qllv.start\_date\_active < SYSDATE

AND NVL(qllv.end\_date\_active, SYSDATE) >= SYSDATE

AND qllv.list\_header\_id = qlht.list\_header\_id

GROUP BY qlht.name

,qllv.product\_attr\_value

,qllv.product\_attr\_val\_disp

,qllv.product\_uom\_code

HAVING COUNT(\*) > 1

Order by 1

**( )**

1. **오라클에서 제공하는 함수 중 단일행 함수 ; 숫자형 함수 중에서 ‘소수점 이하 지정 자리 수에서 올림’ 하는 함수가 있다? 없다?**

( )

1. **다음의 SQL문에 대해 NVL함수와 NVL2 함수를 사용한 결과이다. 틀린 것을 고르시오.**

(1) (2) (3) (4)

SQL> SELECT ename, comm FROM emp;

ENAME COMM

---------------- ----------

SMITH

ALLEN 300

WARD 500

JONES

MARTIN 1400

BLAKE

CLARK

SCOTT

KING

TURNER 0

ADAMS

1. NVL함수를 사용 했을 경우

SQL> SELECT ename, NVL(comm, 0) comm FROM emp;

ENAME COMM

---------------- ----------

SMITH 0

ALLEN 300 (1)

WARD 500

JONES 0

MARTIN 1400

BLAKE 0

CLARK 0

SCOTT 0

KING 0

TURNER 0 (2)

ADAMS 0

1. NVL2함수를 사용 했을 경우

SQL> SELECT ename, NVL2(comm, 1, 0) FROM emp;

ENAME COMM

--------------- ----------

SMITH 0

ALLEN 1

WARD 1

JONES 0

MARTIN 1 (3)

BLAKE 0

CLARK 0

SCOTT 0

KING 0

TURNER 0 (4)

ADAMS 0

1. **다음과 같은 sql문장이 있다. 예제와 같은 결과를 같는 구문을 EXIST를 이용하여 작성하시오.**

예제) SELECT A.COURSE\_CODE, A.COURSE\_NAME

FROM EC\_APPLY B, EC\_COURSE A

WHERE A.COURSE\_CODE = B.COURSE\_CODE

AND B.YEAR = '2000'

GROUP BY A.COURSE\_CODE, A.COURSE\_NAME;

1. **다음 SQL문장의 결과 출력되는 데이터는 무엇인가?**

SELECT NEXT\_DAY(ADD\_MONTHS(hire\_date,6),'월요일')

FROM employees;

(1) employees 테이블에 채용날짜(hire\_date)로 부터 6 일 후 첫번째 월요일에 해당하는 급여 협상 날짜

(2) employees 테이블에 채용날짜(hire\_date)로 부터 6개월 후 두번째 월요일에 해당하는 급여 협상 날짜

(3) employees 테이블에 채용날짜(hire\_date)로 부터 6 개월 후 첫번째 월요일에 해당하는 급여 협상 날짜

1. **다음 중 옳지 않은 것은?**
2. Length(‘SQL Expert’) = 9
3. (날짜1 – 날짜2)의 결과는 일수가 나온다.
4. CASE 표현은 If- Then – Else 논리와 유사한 방식으로 표현식을 작성해서 SQL의 비교 연산 기능을 보완하는 역할을 한다.
5. COUNT(표현식)은 표현식의 값이 Null값인 것을 제외한 행 수를 출력한다.
6. Having 절은 그룹을 나타내는 결과 집합의 행에 조건이 적용된다.
7. **다음 중 옳지 않은 것은?**
8. 기본적인 정렬 순서는 오름차순(ASC)이다.
9. Order by 절에는 Select 목록에 나타나지 않은 문자형 항목이 포함될 수 있다.
10. 4개의 테이블이 있다면 Join 조건은 적어도 3개 이상이 필요하다.
11. SQL 문장의 파싱 순서는 From – Where – Order by – Select 순이다.
12. Non Equi Join은 두 테이블 간 컬럼 값이 서로 정확하게 일치하지 않는 경우 사용한다.
13. **하기 두 쿼리의 결과값은 같은가?**

Select ename, sal

From emp

Where rownum < 4

Order by sal desc;

select top(3) ename, sal

from emp

order by sal desc;

1. 3

2. Hello, ‘ T’, 2

3. Having

4. LTRIM

5. 2

6. Count(\*) <= 5 조건을 Having절로 변경

7. 4

8. 4

9. ㄴ/ㄷ/ㄹ/ㅁ/ㄱ/ㅂ

10. 4

11. X

12. X

13. select coalesce(x.n01, x.n02, .n03, x.n04, x.n05, x.n06 )

from tab x

14. trunc(sysdate, 'Y') ,

last\_day(add\_months(trunc(sysdate), 1)) + 0.99999

15. 1

16. 4

17. O

18. X

19. 4

20. B에 해당하는 테이블을 Exists로 분리하여 작성.

21. 3

22. 1

23. 4

24. X