

2024 신유형 예상 문제

1. 다음 중 아래 그림에서처럼 데이터를 변환하는 SQL 구문으로 적절하지 않은 것은?

JOB	10	20	30
CLERK	1	2	1
SALESMAN	0	0	4
ANALYST	0	2	0
MANAGER	1	1	1
PRESIDENT	1	0	0

- ① SELECT * FROM (SELECT EMPNO, JOB, DEPTNO FROM EMP)
PIVOT (COUNT(EMPNO) FOR DEPTNO IN (10,20,30));
- ② SELECT * FROM (SELECT EMPNO, JOB, DEPTNO FROM EMP)
PIVOT (COUNT(JOB) FOR DEPTNO IN (10,20,30));
- ③ SELECT * FROM (SELECT JOB, DEPTNO FROM EMP)
PIVOT (COUNT(DEPTNO) FOR DEPTNO IN (10,20,30));
- ④ SELECT * FROM (SELECT ENAME, JOB, DEPTNO FROM EMP)
PIVOT (COUNT(ENAME) FOR DEPTNO IN (10,20,30));

2. 다음 SQL 의 실행 결과로 올바른 것은?

```

23 SELECT DEPTNO,
24        "'CLERK'" AS CLERK,
25        "'SALESMAN'" AS SALESMAN,
26        "'ANALYST'" AS ANALYST,
27        "'MANAGER'" AS MANAGER,
28        "'PRESIDENT'" AS PRESIDENT
29 FROM (SELECT SAL, JOB, DEPTNO FROM EMP)
30 PIVOT (SUM(SAL) FOR JOB IN ('CLERK', 'SALESMAN', 'ANALYST', 'MANAGER', 'PRESIDENT'))
31 ORDER BY 1;
  
```

①

JOB	10	20	30
ANALYST		6000	
CLERK	1300	1900	950
MANAGER	2450	2975	2850
PRESIDENT	5000		
SALESMAN			5600

②

JOB	10	20	30
ANALYST	0	6000	0
CLERK	1300	1900	950
MANAGER	2450	2975	2850
PRESIDENT	5000	0	0
SALESMAN	0	0	5600

③


DEPTNO	CLERK	SALESMAN	ANALYST	MANAGER	PRESIDENT
10	1300			2450	5000
20	1900		6000	2975	
30	950	5600		2850	

④

DEPTNO	CLERK	SALESMAN	ANALYST	MANAGER	PRESIDENT
10	1300	0	0	2450	5000
20	1900	0	6000	2975	0
30	950	5600	0	2850	0

3. 다음과 같은 데이터 처리를 하고자 한다. 올바른 SQL 구문을 고르시오.

JOB	10	20	30
CLERK	1	2	1
SALESMAN	0	0	4
ANALYST	0	2	0
MANAGER	1	1	1
PRESIDENT	1	0	0



JOB	DEPTNO	CNT
CLERK	10	1
CLERK	20	2
CLERK	30	1
SALESMAN	10	0
SALESMAN	20	0
SALESMAN	30	4
ANALYST	10	0
ANALYST	20	2
ANALYST	30	0
MANAGER	10	1
MANAGER	20	1
MANAGER	30	1
PRESIDENT	10	1
PRESIDENT	20	0
PRESIDENT	30	0

- ① SELECT * FROM TEST1 UNPIVOT(CNT FOR DEPTNO IN ("10","20","30"));
- ② SELECT * FROM TEST1 UNPIVOT(DEPTNO FOR CNT IN ("10","20","30"));
- ③ SELECT * FROM TEST1 UNPIVOT(CNT FOR DEPTNO IN (10,20,30));
- ④ SELECT * FROM TEST1 UNPIVOT(DEPTNO FOR CNT IN (10,20,30));

4. TEST2 테이블의 데이터 결과가 아래와 같을 때 다음의 SQL 문장의 수행 결과로 알맞은 것은?

< TEST2 TABLE >

DEPTNO	CLERK	SALESMAN	ANALYST	MANAGER	PRESIDENT
10	1300			2450	5000
20	1900		6000	2975	
30	950	5600		2850	

< 결과 >

DEPTNO	JOB	SUM_SAL
10	CLERK	1300
10	MANAGER	2450
10	PRESIDENT	5000
20	CLERK	1900
20	ANALYST	6000
20	MANAGER	2975
30	CLERK	950
30	SALESMAN	5600
30	MANAGER	2850

- ① SELECT * FROM TEST2
UNPIVOT (SUM(SAL) FOR JOB IN ('CLERK','SALESMAN','ANALYST','MANAGER','PRESIDENT'));

② SELECT * FROM TEST2

UNPIVOT (SUM_SAL FOR JOB IN ('CLERK','SALESMAN','ANALYST','MANAGER','PRESIDENT'));

③ SELECT * FROM TEST2

UNPIVOT (SUM(SAL) FOR JOB IN (CLERK,SALESMAN,ANALYST,MANAGER,PRESIDENT));

④ SELECT * FROM TEST2

UNPIVOT (SUM_SAL FOR JOB IN (CLERK,SALESMAN,ANALYST,MANAGER,PRESIDENT));

5. 다음 중 급여가 높은 순서대로 상위 5 명을 추출하는 SQL 문으로 적절하지 않은 것은?

① SELECT * FROM EMP ORDER BY SAL DESC FETCH FIRST 5 ROWS ONLY;

② SELECT * FROM EMP ORDER BY SAL DESC FETCH NEXT 5 ROW ONLY;

③ SELECT * FROM (SELECT EMPNO, ENAME, DEPTNO, SAL,
RANK() OVER(ORDER BY SAL DESC) AS SAL_RANK FROM EMP) WHERE SAL_RANK <= 5;

④ SELECT * FROM (SELECT E.*, ROWNUM AS SAL_RANK FROM EMP E ORDER BY SAL DESC)
WHERE SAL_RANK <= 5;

6. 다음 중 급여가 낮은 순서대로 6~8 번째 값을 출력하는 SQL 문으로 적절하지 않은 것은?

< EMP TABLE >

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	1980/12/17 00:00:00	800		20
7900	JAMES	CLERK	7698	1981/12/03 00:00:00	950		30
7876	ADAMS	CLERK	7788	1987/05/23 00:00:00	1100		20
7521	WARD	SALESMAN	7698	1981/02/22 00:00:00	1250	500	30
7654	MARTIN	SALESMAN	7698	1981/09/28 00:00:00	1250	1400	30
7934	MILLER	CLERK	7782	1982/01/23 00:00:00	1300		10
7844	TURNER	SALESMAN	7698	1981/09/08 00:00:00	1500	0	30
7499	ALLEN	SALESMAN	7698	1981/02/20 00:00:00	1600	300	30
7782	CLARK	MANAGER	7839	1981/06/09 00:00:00	2450		10
7698	BLAKE	MANAGER	7839	1981/05/01 00:00:00	2850		30
7566	JONES	MANAGER	7839	1981/04/02 00:00:00	2975		20
7788	SCOTT	ANALYST	7566	1987/04/19 00:00:00	3000		20
7902	FORD	ANALYST	7566	1981/12/03 00:00:00	3000		20
7839	KING	PRESIDENT		1981/11/17 00:00:00	5000		10

< 결과 >

EMPNO	ENAME	DEPTNO	SAL
7934	MILLER	10	1300
7844	TURNER	30	1500
7499	ALLEN	30	1600

① SELECT EMPNO, ENAME, DEPTNO, SAL

FROM (SELECT E.*, RANK() OVER(ORDER BY SAL) AS SAL_RANK FROM EMP E)
WHERE SAL_RANK BETWEEN 6 AND 8;

② SELECT EMPNO, ENAME, DEPTNO, SAL

FROM (SELECT E.*, DENSE_RANK() OVER(ORDER BY SAL) AS SAL_RANK
FROM EMP E)

WHERE SAL_RANK BETWEEN 5 AND 7;

③ SELECT EMPNO, ENAME, DEPTNO, SAL

FROM (SELECT E.*, ROWNUM AS SAL_RANK FROM EMP E ORDER BY SAL)
WHERE SAL_RANK BETWEEN 6 AND 8;

④ SELECT EMPNO, ENAME, DEPTNO, SAL

FROM EMP

ORDER BY SAL OFFSET 5 ROWS FETCH NEXT 3 ROWS ONLY;

7. 다음 중 Top N Query 에 대한 설명 중 틀린 것은?

- ① 윈도우 함수를 사용하여 상위 N 개에 대한 값을 추출할 수 있으나 단일 Query 로 표현 불가하다.
- ② ROWNUM 을 사용한 방식은 ROWNUM 할당 전에 먼저 순서대로 데이터를 정렬한 뒤 ROWNUM 을 부여 후 추출하는 것이 좋다.
- ③ FETCH 절을 사용하면 단일 Query 로도 정렬 순서대로의 상위 N 개에 대한 값을 추출할 수 있다.
- ④ OFFSET 절을 사용하면 FETCH 절에서 반드시 NEXT 함수를 사용하여 그 다음 n 개를 추출할 수 있다.

8. 다음 중 빈칸에 들어갈 함수식으로 알맞지 않은 것은?

SQL* 1 SELECT JOB_ID, _____ AS RESULT 2 FROM EMPLOYEES;	
Result	
Grid Result	Server Output
Text Output	Explain Plan
Statistics	
JOB_ID	RESULT
1 AC_ACCOUNT	ACCOUNT
2 AC_MGR	MGR
3 AD_ASST	ASST
4 AD_PRES	PRES
5 AD_VP	VP
6 AD_VP	VP
7 FI_ACCOUNT	ACCOUNT
8 FI_ACCOUNT	ACCOUNT
9 FI_ACCOUNT	ACCOUNT
10 FI_ACCOUNT	ACCOUNT
11 FI_ACCOUNT	ACCOUNT

- ① REGEXP_SUBSTR(JOB_ID, '\D+', 4)
- ② REGEXP_SUBSTR(JOB_ID, '(\D+)(\D+)', 1, 1, NULL, 1)
- ③ REGEXP_REPLACE(JOB_ID, '[A-Z]_+')
- ④ REGEXP_REPLACE(JOB_ID, '._+', '')

9. 다음 SQL 의 실행 결과로 알맞은 것을 고르시오.

SELECT PHONE_NUMBER, REGEXP_REPLACE (PHONE_NUMBER, '\d+', 'xxx', 1, 2) AS RESULT FROM EMPLOYEES;	
--	--

- ①

PHONE_NUMBER	RESULT
650.507.9833	XXX.507.9833
650.507.9844	XXX.507.9844
515.123.4444	XXX.123.4444
515.123.5555	XXX.123.5555
603.123.6666	XXX.123.6666
- ②

PHONE_NUMBER	RESULT
650.507.9833	650.XXX.9833
650.507.9844	650.XXX.9844
515.123.4444	515.XXX.4444
515.123.5555	515.XXX.5555
603.123.6666	603.XXX.6666
- ③

PHONE_NUMBER	RESULT
650.507.9833	650.507.XXX
650.507.9844	650.507.XXX
515.123.4444	515.123.XXX
515.123.5555	515.123.XXX
603.123.6666	603.123.XXX
- ④

PHONE_NUMBER	RESULT
650.507.9833	XXX.XXX.XXX
650.507.9844	XXX.XXX.XXX
515.123.4444	XXX.XXX.XXX
515.123.5555	XXX.XXX.XXX
603.123.6666	XXX.XXX.XXX

10. 다음의 SQL 실행 결과로 알맞은 것을 고르시오.

SQL* *	
1	SELECT LAST_NAME,
2	REGEXP_COUNT(LAST_NAME, '[ko]', 1, 'i') AS RESULT
3	FROM EMPLOYEES;
4	

Result	
Grid Result	Server Output
Text Output	Explain Plan
Statistics	

LAST_NAME	RESULT
1 King	
2 Kochhar	
3 Mavris	
4 OConnell	

- ①

LAST_NAME	RESULT
King	1
Kochhar	2
Mavris	0
OConnell	2
- ②

LAST_NAME	RESULT
King	0
Kochhar	1
Mavris	0
OConnell	0
- ③

LAST_NAME	RESULT
King	0
Kochhar	0
Mavris	0
OConnell	0
- ④

LAST_NAME	RESULT
King	1
Kochhar	1
Mavris	0
OConnell	2

11. 다음 결과 값으로 알맞은 것을 고르시오.

SQL* *	
1	SELECT REGEXP_INSTR('SELECT * FROM EMP WHERE DEPTNO = 10 ;', '^[^]+', 7, 3)
2	FROM DUAL;
3	

- ① 0
- ② 10
- ③ 15
- ④ 32

정답)

1. ②

해설 : COUNT(JOB)을 하면서 이미 JOB 컬럼을 사용하였으므로 남아 있는 EMPNO 컬럼에 대해 STACK 처리되고, DEPTNO 컬럼이 UNSTACK 처리 되어 아래와 같이 출력된다.

EMPNO	10	20	30
7782	1	0	0
7788	0	1	0
7698	0	0	1
7876	0	1	0
7521	0	0	1
7566	0	1	0
7902	0	1	0
7499	0	0	1
7900	0	0	1
7369	0	1	0
7654	0	0	1
7934	1	0	0
7839	1	0	0
7844	0	0	1

2. ③

해설 : FOR 뒤에 오는 컬럼의 값이 IN 뒤에 나열된 순서대로 컬럼으로 UNSTACK 처리되며(10,20,30 컬럼 생성됨), 일반적으로 SUM 은 연산할 값이 없는 경우 NULL 로 리턴된다.

3. ①

해설 : 10, 20, 30 이름의 컬럼을 STACK 처리 하는 과정에서 숫자 컬럼명에 쌍따옴표를 전달해야 하며, 10,20,30 데이터가 저장되는 컬럼명을 FOR 뒤에 작성해야 한다.

4. ④

해설 : 현재 컬럼 값으로 되어있는 CLERK, SALESMAN 등을 모두 선택하여 STACK 처리 하는 과정에서 컬럼명이 홑따옴표 없이 IN으로 전달되어야 하며, FOR 앞에 있는 값은 VALUE 값들이 하나의 컬럼으로 STACK 처리되면서 필요 시 되는 컬럼명이므로 실제 연산함수가 들어갈 수 없다.

5. ④

해설 : 서브쿼리의 ROWNUM 이 먼저 부여되고 ORDER BY 로 정렬하기 때문에 정렬된 순서를 보장한 ROWNUM 값을 확보할 수 없다. 따라서 4 번의 결과는 다음과 같이 원래 출력 결과(SELECT * FROM EMP) 중 5 명의 값이 출력하게 된다)

EMPNO	ENAME	DEPTNO	SAL
7566	JONES	20	2975
7499	ALLEN	30	1600
7521	WARD	30	1250
7654	MARTIN	30	1250
7369	SMITH	20	800

6. ③

해설 : 서브쿼리의 ROWNUM 은 정렬 전에 먼저 행 번호를 부여하므로 ROWNUM 을 사용하여 SAL 순서대로의 행을 추출할 수 없다. 따라서 아래와 같이 의미 없는 3 명이 출력된다.

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	SAL_RANK
7782	CLARK	MANAGER	7839	1981/06/09 00:00:00	2450		10	7
7698	BLAKE	MANAGER	7839	1981/05/01 00:00:00	2850		30	6
7788	SCOTT	ANALYST	7566	1987/04/19 00:00:00	3000		20	8

7. ④

해설 : FETCH 절의 FIRST 와 NEXT 는 사실상 구분하지 않아도 된다.

8. ②

해설 : REGEXP_SUBSTR(JOB_ID, '(\D+)(\D+)', 1, 1, NULL, 1)는 JOB_ID 에서 숫자가 아닌 값(\D)의 연속 글자를 JOB_ID 의 처음부터(1) 찾아 처음으로(1) 발견되는 패턴 중 첫 번째(마지막 1) 그룹을 추출하라는 의미이므로 아래와 같이 출력된다.

JOB_ID	RESULT
AC_ACCOUNT	AC
AC_MGR	AC
AD_ASST	AD
AD_PRES	AD
AD_VP	AD
AD_VP	AD
FI_ACCOUNT	FI
FI_ACCOUNT	FI
FI_ACCOUNT	FI
FI_ACCOUNT	FI
FI_ACCOUNT	FI

REGEXP_REPLACE(대상, 찾을문자열, [바꿀문자열], [검색위치], [발견횟수], [옵션])

REGEXP_SUBSTR(대상, 패턴, [검색위치], [발견횟수], [옵션], [추출그룹])

9. ②

해설 : REGEXP_REPLACE(PHONE_NUMBER, '\d+', 'XXX', 1, 2)의 의미는 PHONE_NUMBER 의 처음부터 숫자의 연속을 찾아 두 번째로 발견되는 값을 'XXX'로 치환하라는 의미이므로 .과 . 사이에 있는 가운데 숫자 전체가 바뀐다.

10. ①

해설 : '[ko]'는 k 또는 o 를 나타내는 한 글자의 문자를 의미하므로 처음부터 발견되는 k 또는 o 를 대소문자 구분 없이 세면, 각각 1,2,0,2 가 된다.

11. ③

해설 : [^] 는 공백이 아닌 한 글자를 의미하며 뒤에 있는 +에 의해 해당 문자가 연속적으로 이어지는 경우를 나타낸다. 즉, [^]+는 공백 없이 이어지는 한 단어를 표현하므로 왼쪽에서부터 7 번째 글자부터 오른쪽으로 단어를 찾아 세 번째 발견되는 단어인 EMP 의 위치를 리턴하게 된다.