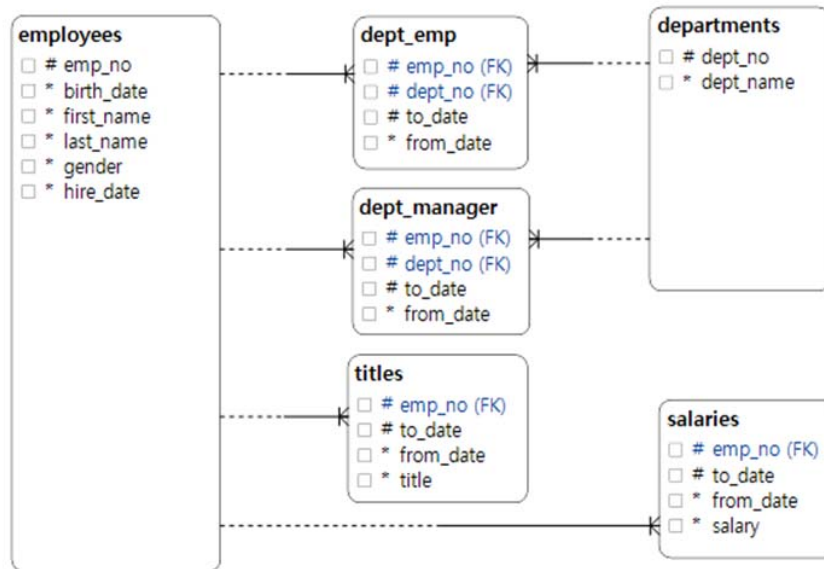


DA 과정 : 고급 SQL 특강 Quiz

1회차)

1-1. (ERD) 다음 ERD에 대한 해석 중 틀린 것은?



- ① 사원(employee)의 소속 부서는 변경될 수 있다
- ② 역대 부서장(department manager)을 관리하고자 한다
- ③ 부서(department)에 따라 직급(title) 체계가 다르다
- ④ 직급별로 급여(salary)가 책정되는지 여부는 알 수 없다
- ⑤ 구조조정에 의해서 부서 자체가 변경될 경우, 부서변경이력은 관리하지 않는다

1-2. (제약조건) DB 무결성(integrity)을 보장하기 위한 제약조건(constraint)에 대한 설명 중 틀린 것은?

- ① 기본키(PK) 속성은 NULL값 허용이 되지 않으며 유일성(uniqueness)을 보장해야 한다
- ② 체크(check) 제약조건이 걸릴 경우, 속성의 도메인(domain)은 좁아진다
- ③ 외래키(foreign key) 속성이 참조하는 기본키 속성값들 중 하나의 값을 가져야 한다
- ④ 외래키에 ON UPDATE CASCADE 옵션이 걸릴 경우, 외래키 값을 수정하면 해당 외래키가 참조하는 기본키 값도 연쇄적으로 수정된다
- ⑤ NOT NULL 제약조건이 걸린 속성에 빈 값을 입력하려고 하면, 튜플 전체의 입력이 거부된다

1-3. (SQL#1) 실습용 DB를 대상으로 다음을 조회하는 SQL을 작성하시오.

부서별 2000년 1월 1일 당시 부서장의 사번(emp_no), 이름(first_name+last_name), 성별(gender), 입사일(hire_date), 직급(title), 급여(salary)

```
select /*+ ordered use_nl(a b c d e) */ b.dept_name, c.emp_no, c.first_name||' '||c.last_name, c.gender, c.hire_date, d.title, e.salary
```

```

from dept_manager a, departments b, employees c, titles d, salaries e
where '20000101' between a.from_date and a.to_date
    and a.dept_no = b.dept_no
    and a.emp_no = c.emp_no
    and a.emp_no = d.emp_no and '20000101' between d.from_date and d.to_date
    and a.emp_no = e.emp_no and '20000101' between e.from_date and e.to_date ;

```

2회차)

2-1. (실행계획) 옵티마이저가 SQL문을 실행하기 위해 수행한 일련의 동작(OPERATIONS)을 트리형식으로 표현한 것으로 사용자로 하여금 어떻게 SQL을 수행할 것인지를 보여주고 SQL 성능 판단을 위한 정보 제공하는 것을 무엇이라고 하는가?

실행계획(explain plan, execution plan)

2-2. (조인) 다음 조인(join)에 대한 설명 중 틀린 것은?

- ① 예약번호로 검색 등 작은 범위의 검색이 많은 온라인 조회에서는 sort-merge 조인 방식이 좋다
- ② Nested loop 조인에서는 선행 테이블의 개념이 있으나 sort-merge에서는 없다
- ③ Nested loop 조인은 인덱스와 함께 활용할 때 성능이 극대화될 수 있다
- ④ hash 조인에서는 조인 대상 로우수가 작은 쪽을 가지고 hash table을 만든다
- ⑤ parallel processing을 이용한 hash 조인은 대용량 데이터를 처리하기 위한 최적의 솔루션이다

2-3. (힌트/인덱스) 다음 SQL 수행을 위해 테이블 A, B에서 읽은 로우 수의 합은?

```

select /*+ ordered use_nl(A B) index(A A_UK1) */ A.COL3, B.COL4
from A, B
where A.COL1 = 'aaa'
    and A.COL2 = B.COL2
    and B.COL3 = 'bbb' ;
-- A_UK1 : COL1을 인덱스 컬럼으로 하는 unique index
-- B 테이블의 PK : COL2

```

2

2-4. (SQL#2) 실습용 DB를 대상으로 다음을 조회하는 SQL을 작성하시오.

1958년 5월생 사원의 사번(emp_no), 생년월일(birth_date), 성별(gender), 고용일(hire_date), 소속부서명(dept_name), 직급(title), 급여(salary) 조회

```

select a.emp_no

```

```

, a.birth_date, a.gender, a.hire_date
, c.dept_name, d.title, e.salary
from employees a, dept_emp b, departments c, titles d, salaries e
where a.birth_date between to_date('19580501', 'yyyymmdd') and to_date('19580531', 'yyyymmdd')
and a.emp_no = b.emp_no and b.to_date = '99991231'
and b.dept_no = c.dept_no
and a.emp_no = d.emp_no and d.to_date = '99991231'
and a.emp_no = e.emp_no and e.to_date = '99991231' ;

```

3회차)

3-1. (실행계획분리) 다음 중 실행계획 분리에 사용되는 SQL 연산자는?

- ① UNION
- ② **UNION ALL**
- ③ INTERSECT
- ④ MINUS
- ⑤ OUTER JOIN

3-2. (ROLLUP/CUBE) 다음 3개의 SQL 각각의 수행결과로 나오는 그룹핑 레벨 개수의 합은?

```

select a, b, c, count(*)
from tab1
group by rollup(a, b, c) ;

```

```

select a, b, c, count(*)
from tab1
group by cube(a, b, c) ;

```

```

select a, b, c, count(*)
from tab1
group by grouping sets(a, b, c) ;

```

4 + 8 + 3 = 15

3-3. (데이터복제) 다음 SQL문의 실행결과는?

```

select sum(cnt)
from ( select decode(b.no, 1, floor(a.no/10)), count(*) cnt
      from copy_t a, copy_t b
      where a.no <= 20 and b.no <= 2
      group by decode(b.no, 1, floor(a.no/10)) ) ;

```

3-4. (SQL#3) 실습용 DB를 대상으로 다음을 조회하는 SQL을 작성하시오.

현재 기준 부서별 직급별 사원수 (그룹핑 레벨 : 부서별, 직급별, 부서별/직급별, 전체)

```
select nvl(y.dept_name, '합계') 부서명
      , x.title 직급명
      , x.cnt_emps 사원수
from ( select decode(grouping(b.dept_no), 1, '합계', b.dept_no) dept_no
      , decode(grouping(c.title), 1, '합계', c.title) title
      , count(*) cnt_emps
  from salaries a, dept_emp b, titles c
 where a.emp_no = b.emp_no
       and b.emp_no = c.emp_no
       and a.to_date = '99991231' and b.to_date = '99991231' and c.to_date = '99991231'
 group by cube(b.dept_no, c.title) ) x, departments y
where x.dept_no = y.dept_no(+);
```

4회차)

4-1. (분석함수#1) 다음 중 오라클 분석함수에 대한 설명 중 틀린 것은?

- ① SELECT 절에서 수행되며, FROM, WHERE, GROUP BY 절에서 사용 불가하다
- ② 분석함수는 그룹마다가 아니라 결과Set의 각 행마다 그룹별 계산결과를 보여준다
- ③ OVER 절에 ORDER BY를 넣을 경우, window의 범위는 unbounded preceding에서 current row까지이다
- ④ 분석함수를 쓰더라도 결과 집합의 레벨은 바뀌지 않는다
- ⑤ 분석함수 중 LEAD()는 ORDER BY 컬럼 기준으로 current row의 이전 로우를 LAG()는 이후 로우를 반환한다

4-2. (데이터랭글링) 다음 데이터처리에 관한 설명 중 틀린 것은?

- ① 데이터랭글링은 복잡하고 지저분한 상태의 데이터를 간단한 분석과 접근을 위해 통합하는 과정으로 데이터를 분석할 수 있는 상태로 조직하는 필수 과정이다
- ② 데이터 분석을 위한 전처리에는 데이터 정제, 데이터 통합, 데이터 변환, 데이터 축소 등의 기법이 있다
- ③ ETL이 분석가의 관점이라면, Data wrangling은 최종사용자 관점에서 대부분 루틴한 과정으로 자동화가 용이하다
- ④ 데이터 랭글링은 데이터 먼징(data munging)이라고도 한다
- ⑤ 데이터 행/열 전환(pivoting)도 데이터 랭글링의 일종이다

4-3. (분석함수#2) 다음 SQL문의 실행결과?

```

select sum(col1) + sum(col2)
from ( select no
        , sum(no) over (partition by mod(no, 2)) col1
        , sum(no) over (partition by mod(no, 2) order by rownum) col2
  from copy_t
 where no <= 5 ) ;

```

61

4-4. (SQL#4) 다음 SQL을 실행하여 copy_ymd 테이블과 데이터를 생성하고 년월 조건을 주면 해당 월의 달력을 아래 그림과 같은 포맷으로 출력하는 SQL을 작성하시오.

일	월	화	수	목	금	토
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

```

create table copy_ymd (
  ymd varchar2(8) primary key,
  ymd_date date not null ) ;

```

```

insert into copy_ymd
select to_char(to_date('19000101','yyyymmdd')+rownum-1, 'yyyymmdd')
      , to_date('19000101','yyyymmdd')+rownum-1
from employees
where rownum <= 100000 ;

```

```

commit ;

```

```

select ceil((mum+start_yoil-1)/7) 주차
      , min(sun) 일
      , min(mon) 월

```

```

, min(tue) 화
, min(wed) 수
, min(thu) 목
, min(fri) 금
, min(sat) 토
from (
select decode(mod(yoil, 7), 1, day) sun
      , decode(mod(yoil, 7), 2, day) mon
      , decode(mod(yoil, 7), 3, day) tue
      , decode(mod(yoil, 7), 4, day) wed
      , decode(mod(yoil, 7), 5, day) thu
      , decode(mod(yoil, 7), 6, day) fri
      , decode(mod(yoil, 7), 0, day) sat
      , rownum rnum
from (
select trunc(substr(ymd, 7), '0') day
      , to_char(ymd_date, 'd') yoil
from copy_ymd
where ymd like :v_month||'%' )) a, ( select to_char(ymd_date, 'd') start_yoil
                                     from copy_ymd
                                     where ymd = :v_month||'01') b

group by ceil((rnum+start_yoil-1)/7)
order by 1 ;

```