**[문제1] PT5-14, dept 테이블에서 Dept에서 (deptno=10,20,30)를 만족하는 dept6(deptno, dname) 생성하는 SQL 작성.**

SQL> create table dept6 as

2 select deptno, dname

3 from dept

4 where deptno in (10,20,30);

Table created.

SQL> select \* from dept;

DEPTNO DNAME LOC

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

90 Hardware Urgench

50 ????

60 DBLAB SOFT212

10 ACCOUNTING BUSAN

20 RESEARCH DALLAS

30 SALES CHICAGO

40 OPERATIONS BOSTON

80 AAA dddd

8 rows selected.

SQL> select \* from dept6;

DEPTNO DNAME

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

10 ACCOUNTING

20 RESEARCH

30 SALES

**[실행결과]**

Create table as 하고 뒷절에 select 로 나온 결과를 집어넣으면 그 결과 그대로 테이블을 만들 수 있습니다.

**[문제2] PT5-16, dept6 테이블에서 컬럼 추가하면서 기본 값을 지정하는 SQL 문을 실습하라.**

SQL> alter table dept6

2 add (location varchar2(10) default 'SEOUL');

Table altered.

SQL> select \* from dept6;

DEPTNO DNAME LOCATION

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

10 ACCOUNTING SEOUL

20 RESEARCH SEOUL

30 SALES SEOUL

**[실행결과]**

Alter table 테이블 이름 add (컬럼이름 컬럼형식 [default ‘msg’]); 를 해주게 되면 해당 테이블에 컬럼이름에 해당하는 컬럼을 컬럼형식과 맞추어서 컬럼을 추가하게 되고 default 지정시 빈 값으로 지정이 되지 않고 값을 특정하게 지정해주지 않으면 msg 값으로 입력이 됩니다.

**[문제3] PT5-23, t\_readonly 테이블에 대해**

**테이블 생성->**

SQL> create table t\_readonly ( no NUMBER, name VARCHAR2(10));

**데이터 추가->**

SQL> insert into t\_readonly values(1,'AAA');

1 row created.

SQL> commit;

Commit complete.

**테이블검색->**

SQL> select \* from t\_readonly;

NO NAME

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

1 AAA

**read only로 변경 ->**

SQL> alter table t\_readonly **read only;**

Table altered.

**데이터입력(오류 확인)->**

insert into t\_readonly

\*

ERROR at line 1:

ORA-12081: update operation not allowed on table "USER23"."T\_READONLY"

**컬럼추가(오류 확인)->**

SQL> alter table t\_readonly

2 add(tel number default 111);

alter table t\_readonly

**read, write로 변경->**

SQL> alter table t\_readonly **read write;**

Table altered.

**데이터입력->**

SQL> alter table t\_readonly

2 add(tel number default 111);

Table altered.

**[실행결과]**

alter table t\_readonly **read only;** / alter table t\_readonly **read write;** 여기서 핵심은 두 쿼리입니다. Alter table 테이블 이름 read [**write , only**] 를 각각 적어주게 되면 , **읽기 전용인 테이블은 읽기/쓰기 상태일 때 아래의 명령으로 삭제 가능하고 , 읽기 전용인 테이블을 다시 읽기/쓰기 모드로 변경하다는 점이 핵심입니다.**

**[**

**[문제4] PT5-27, 가상 컬럼을 처리하는 Step1 ~ Step6 까지의 내용을 SQL 실습하고 결과를 확인하라.**

SQL> create table virtualcol(

2 col1 NUMBER,

3 col2 NUMBER,

4 col3 NUMBER GENERATED ALWAYS AS (col1+col2));

Table created.

**Virtualcol 이라는 이름으로 생성합니다.**

GENERATED ALWAYS AS (col1+col2) **컬럼 옆에 이 뜻은 이 col3의 열은 항상 co1+col2가 된다는 뜻입니다.**

SQL> insert into virtualcol values(1,2,3);

insert into virtualcol values(1,2,3)

\*

ERROR at line 1:

ORA-54013: INSERT operation disallowed on virtual columns

**후에 virtualcol 테이블에 전체 데이터를 삽입해주려고 했으니 col1 + col2 가상 컬럼이 포함이 되어있어 오류가 나오게 됩니다.**

SQL> insert into virtualcol (col1,col2) values(1,2);

1 row created.

SQL> select \* from virtualcol;

COL1 COL2 COL3

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

1 2 3

**가상 컬럼을 제외한 col1, col2에 삽입했더니 정상적으로 입력이 됩니다.**

SQL> update virtualcol set col1=5;

1 row updated.

SQL> select \* from virtualcol;

COL1 COL2 COL3

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

5 2 7

**Col1은 변경이 가능하기 때문에 update set 구무으로 1-> 5로 업데이트 하는 순간 col3에 걸린 조건으로 인해서 7이 되어버리는 것을 확인했습니다.**

SQL> alter table virtualcol add (col4 generated always as ((col1\*2)+col2));

Table altered.

SQL> select \* from virtualcol;

COL1 COL2 COL3 COL4

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

5 2 7 12

**컬럼을 추가해서 이번에 col4를 추가해서 공식을 col1\*2 +col2 를 해주었습니다. 예상대로 select을 쳐보니 결과가 col4를 12로 삽입했습니다.**

SQL> col column\_name for a10

SQL> col data\_type for a10

SQL> col data\_default for a25

SQL> SELECT COLUMN\_NAME,

2 DATA\_TYPE,

3 DATA\_DEFAULT

4 FROM USER\_TAB\_COLUMNS

5 WHERE TABLE\_NAME = 'VIRTUALCOL'

6 ORDER BY COLUMN\_ID;

COLUMN\_NAM DATA\_TYPE DATA\_DEFAULT

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

COL1 NUMBER

COL2 NUMBER

COL3 NUMBER "COL1"+"COL2"

COL4 NUMBER "COL1"\*2+"COL2"

**USER\_TAB\_COLUMNS 에서 컬럼의 이름, 컬럼의 타입, NULL인지 체크할 수 있고 테이블 열의 정보를 종합적으로 알 수 있는 시스템 테이블입니다.**

텍스트이(가) 표시된 사진

자동 생성된 설명

<https://goddaehee.tistory.com/119> : 시스템 테이블의 종류

**[실행결과]**

위의 쿼리 하나하나에 배운점을 다 적어놨습니다.

**[문제5] PT5-36, PT5-34~PT5-36 까지의 ST\_TABLE 테이블 적용사례를 SQL 실습하고 결과를 확인하라.**

SQL> CREATE TABLE STA\_TABLE

2 (NO NUMBER);

**STA \_ TABLE을 생성하고 컬럼은 NO 하나로 정의합니다.**

SQL> BEGIN

2 FOR i IN 1..1000 LOOP

3 INSERT INTO STA\_TABLE VALUES(i);

4 END LOOP;

5 COMMIT;

6 END;

7 /

PL/SQL procedure successfully completed.

**BEGIN FOR I IN [START .. END] LOOP / [QUERY] /END LOOP /COMMIT;/ END**

**BEGIN과 END 안에 FOR로 LOOP와 함께 루프돌린다는 표식을 적고 END LOOP로 끝낸다는 표식을 적습니다.**

**그리고 그안에 쿼릴 적게되면 그게 루프 횟수만큼 회전합니다.**

SQL> SELECT COUNT(\*) FROM STA\_TABLE ;

COUNT(\*)

----------

1000

**해당 테이블의 전체 행의 개수를 뽑아보면 1-1000까지 루프를 돌렸을 때의 결과와 같습니다.**

SQL> SELECT NUM\_ROWS, BLOCKS

2 FROM USER\_TABLES

3 WHERE TABLE\_NAME = 'STA\_TABLE';

NUM\_ROWS BLOCKS

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

**USER TABLE이라는 시스템 테이블 (딕셔너리)로 컬럼의 행개수와, 몇 개의 BLOCK 단위로 저장이 되었는 지확인합니다.**

SQL> ANALYZE TABLE STA\_TABLE COMPUTE STATISTICS;

Table analyzed.

**하지만 자세히 출력이 되지 않아서 ANALAYZE 구문을 실행합니다.**

텍스트이(가) 표시된 사진

자동 생성된 설명

<http://www.gurubee.net/lecture/1740>: ANALYZE USE

SQL> SELECT NUM\_ROWS, BLOCKS

2 FROM USER\_TABLES

3 WHERE TABLE\_NAME = 'STA\_TABLE';

NUM\_ROWS BLOCKS

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

1000 5

**[실행결과]**

* - COMPUTE : 각각의 값들을 정확하게 계산 한다. 가장 정확한 통계를 얻을 수 있지만 처리 속도가 가장 느리다
* - ESTIMATE : 자료사전의 값과 데이터 견본을 가지고 검사해서 통계를 예상 한다. COMPUTE보다 덜 정확 하지만 처리속도가 훨씬 빠르다.
* - DELETE : 테이블의 모든 통계 정보를 삭제 한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

**[문제6] PT5-36, 오라클에서 데이터사전의 대한 내용을 제시하고 인터넷에서 적용사례를 조사하라.**

다른파일에 작성해서 제출.

**[문제7] PT5-37, 연습문제 1~5까지의 SQL 사례를 실습하라.**

**1. 아래와 같은 구조의 일반 테이블을 생성하세요.(데이터는 3행만 입력)**

SQL> create table new\_emp (no NUMBER(5), name VARCHAR2(20), hiredate DATE, bonus NUMBER(6,2));

Table created.

SQL> insert into new\_emp values(1,'kim', TO\_DATE('21/01/01','YY-MM-DD'), 3.82);

1 row created.

SQL> insert into new\_emp values(2,'lee', TO\_DATE('21/01/02','YY-MM-DD'), 9.82);

1 row created.

SQL> insert into new\_emp values(3,'jang', TO\_DATE('21/01/03','YY-MM-DD'), 1122.82);

1 row created.

SQL> select \* from new\_emp;

NO NAME HIREDATE BONUS

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

1 kim 01-JAN-21 3.82

2 lee 02-JAN-21 9.82

3 jang 03-JAN-21 1122.82

**일단 TEST 데이터를 3개 넣은 EMP 테이블을 생성하였습니다.**

**2. 위 1번 문제에서 생성한 new\_emp 테이블에서 NO , NAME , HIREDATE 컬럼만 가져와서 아래 그림과 같이 new\_emp2 테이블을 생성하는 쿼리를 쓰세요.**

SQL> CREATE TABLE new\_emp2 AS SELECT no, name, hiredate FROM new\_emp;

Table created.

SQL> select \* from new\_emp2;

NO NAME HIREDATE

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

1 kim 01-JAN-21

2 lee 02-JAN-21

3 jang 03-JAN-21

**CREATE TABLE – AS 결과테이블 구문으로 첫번째 EMP에서 3개의 열을 (BOUNUS)제외 하고 가져왔습니다.**

**3. 위 2번 문제에서 생성한 new\_emp2 테이블과 동일한 구조의 테이블을 new\_emp3 이름으로 생성하되 테이블 구조만 가져오고 데이터는 가져오지 않도록 하는 쿼리를 쓰세요.**

SQL> create table new\_emp3 as select \* from new\_emp2 where 1=2;

Table created.

**여기서 핵심은 1=2라는 조건이 불일치하는 식이 있는데 이식을 넣어주게 되면 구조만 가져오고 테이블의 내용은 가져오지 않습니다.**

SQL> select \* from new\_emp3;

no rows selected

SQL> desc new\_emp;

Name Null? Type

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

NO NUMBER(5)

NAME VARCHAR2(20)

HIREDATE DATE

BONUS NUMBER(6,2)

**4. 위 2번 문제에서 생성한 new\_emp2 테이블에 DATE 타입을 가진 BIRTHDAY 컬럼을 추가하는 쿼리를 쓰세요. 단 해당 컬럼이 추가될 때 기본값으로 현재날짜 ( SYSDATE ) 가 자동으로 입력되도록 하세요.**

SQL> alter table new\_emp2 add (birthday date default sysdate);

Table altered.

**ALTER TABLE NEW\_EMP2 ADD를 통해서 BIRTHDAY 컬럼을 추가하면서 DEFAULT 값을 정했습니다.**

SQL> select \* from new\_emp2;

NO NAME HIREDATE BIRTHDAY

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

1 kim 01-JAN-21 16-NOV-21

2 lee 02-JAN-21 16-NOV-21

3 jang 03-JAN-21 16-NOV-21

**5. 위 4번 문제에서 생성한 new\_emp2 테이블의 BIRTHDAY 컬럼 이름을 BIRTH 로 변경하는 쿼리를 쓰세요.**

SQL> alter table new\_emp2 rename column birthday to birth;

Table altered.

SQL> desc new\_emp2

Name Null? Type

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

NO NUMBER(5)

NAME VARCHAR2(20)

HIREDATE DATE

BIRTH DATE

**[실행결과]**

위 쿼리의 사이에 실행결과를 첨부했습니다.

**[문제8] PT6-13, Professor 테이블의 예를 참조하고 EMP 테이블에 empno가 1-7800인 경우 emp3(empno, ename, sal)을 생성,**

**empno가 7801 ~ 10000인 경우는 emp4(empno, ename, sal) 를 생성한 후 2개의 테이블을 조회하는 SQL를 실습하라.**

SQL> insert all when empno between 1 and 7800 then

2 into emp3 values(empno, ename, sal)

3 when empno between 7801 and 10000 then

4 into emp4 values(empno, ename, sal)

5 select \* from emp;

16 rows created.

SQL> select \* from emp3;

EMPNO ENAME SAL

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

7839 KING 5000

7698 BLAKE 2850

7782 CLARK 2450

7566 JONES 2975

7788 SCOTT 3000

7902 FORD 3000

7369 SMITH 800

7499 ALLEN 1600

7521 WARD 1250

7654 MARTIN 1250

7844 TURNER 1500

7876 ADAMS 1100

7900 JAMES 950

7934 MILLER 1300

1000 Tiger 3600

2000 Cat 3000

7698 BLAKE 2850

7782 CLARK 2450

7566 JONES 2975

7788 SCOTT 3000

7369 SMITH 800

7499 ALLEN 1600

7521 WARD 1250

7654 MARTIN 1250

1000 Tiger 3600

2000 Cat 3000

7698 BLAKE 2850

7782 CLARK 2450

7566 JONES 2975

7788 SCOTT 3000

7369 SMITH 800

7499 ALLEN 1600

7521 WARD 1250

7654 MARTIN 1250

1000 Tiger 3600

2000 Cat 3000

36 rows selected.

SQL> select \* from emp4;

EMPNO ENAME SAL

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

7839 KING 5000

7698 BLAKE 2850

7782 CLARK 2450

7566 JONES 2975

7788 SCOTT 3000

7902 FORD 3000

7369 SMITH 800

7499 ALLEN 1600

7521 WARD 1250

7654 MARTIN 1250

7844 TURNER 1500

7876 ADAMS 1100

7900 JAMES 950

7934 MILLER 1300

1000 Tiger 3600

2000 Cat 3000

7839 KING 5000

7902 FORD 3000

7844 TURNER 1500

7876 ADAMS 1100

7900 JAMES 950

7934 MILLER 1300

7839 KING 5000

7902 FORD 3000

7844 TURNER 1500

7876 ADAMS 1100

7900 JAMES 950

7934 MILLER 1300

**[실행결과]**

SQL> insert all when empno between 1 and 7800 then

2 into emp3 values(empno, ename, sal)

3 when empno between 7801 and 10000 then

4 into emp4 values(empno, ename, sal)

5 select \* from emp;

INSERT ALL 구문을 사용해 INTO 와 그 사이에 조건을 넣어서 테이블에 추가하고, SELECT 로 뽑아주면 결과가 각각 INSERT 된 것으로 생깁니다.

**[문제9] PT6-14, EMP3, EMP4 테이블의 데이터를 삭제한 후 empno가 7500 에서 7900인 경우 emp3(empno, ename, sal),**

**emp4 (empno, ename, sal) 테이블에 데이터를 동시에 생성하고 2개의 테이블을 조회하는 SQL를 실습하라.**

SQL> INSERT ALL

2 INTO EMP3 VALUES (EMPNO, ENAME, SAL)

3 INTO EMP4 VALUES (EMPNO, ENAME, SAL)

4 SELECT \* FROM EMP

5 WHERE EMPNO BETWEEN 7500 AND 7900;

20 rows created.

SQL> select \* from emp3;

EMPNO ENAME SAL

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

7839 KING 5000

7698 BLAKE 2850

7782 CLARK 2450

7566 JONES 2975

7788 SCOTT 3000

7521 WARD 1250

7654 MARTIN 1250

7844 TURNER 1500

7876 ADAMS 1100

7900 JAMES 950

10 rows selected.

SQL> select \* from emp4;

EMPNO ENAME SAL

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

7839 KING 5000

7698 BLAKE 2850

7782 CLARK 2450

7566 JONES 2975

7788 SCOTT 3000

7521 WARD 1250

7654 MARTIN 1250

7844 TURNER 1500

7876 ADAMS 1100

7900 JAMES 950

10 rows selected.

**[실행결과]**

SQL> INSERT ALL

2 INTO EMP3 VALUES (EMPNO, ENAME, SAL)

3 INTO EMP4 VALUES (EMPNO, ENAME, SAL)

4 SELECT \* FROM EMP

5 WHERE EMPNO BETWEEN 7500 AND 7900;

각각에 대한 조건을 주기 위해서는 **WHERE EMPNO BETWEEN 7500 AND 7900;** 이렇게 적어줍니다.

**[문제1] PT3-67, emp 테이블에서 사원 중에 급여(sal)와 보너스(comm)를 합친 금액이 가장 많은 경우와 가장 적은 경우, 평균 금액을 구하도록  
   SQL 문을 실습하라. 단 보너스가 없을 경우는 보너스를 0 으로 계산하고 출력 금액은 모두 소수점 첫째 자리까지만 나오게 한다.**

**[실행결과]**

**[문제1] PT3-67, emp 테이블에서 사원 중에 급여(sal)와 보너스(comm)를 합친 금액이 가장 많은 경우와 가장 적은 경우, 평균 금액을 구하도록  
   SQL 문을 실습하라. 단 보너스가 없을 경우는 보너스를 0 으로 계산하고 출력 금액은 모두 소수점 첫째 자리까지만 나오게 한다.**

**[실행결과]**