**[문제1] PT6-23, ch\_total 테이블에 2개의 행(141001 행, 141002 행)을 입력하고,**

**charge\_01 테이블에 2개의 행(141001 1000 2 1000 행과 141001 1001 2 1000 행)을 입력한 후**

**Merger 조건에 따른 처리시의 SQL 실행 결과를 확인할 것**

1. **테이블 생성**

SQL> create table charge\_01(

2 u\_date varchar2(6),

3 cust\_no number,

4 u\_time number,

5 charge number);

Table created.

SQL> create table charge\_02(

2 u\_date varchar2(6),

3 cust\_no number,

4 u\_time number,

5 charge number);

Table created.

SQL> create table ch\_total(

2 u\_date varchar2(6),

3 cust\_no number,

4 u\_time number,

5 charge number);

Table created.

1. **데이터 삽입**

SQL> insert into charge\_01 values('141001',1000,2,1000);

1 row created.

SQL> insert into charge\_01 values('141001',1001,2,1000);

1 row created.

SQL> insert into ch\_total values ('141001',null ,null ,null );

1 row created.

SQL> insert into ch\_total values ('141002',null ,null ,null );

1 row created.

1. **데이터 조회**

SQL> select \* from charge\_01;

U\_DATE CUST\_NO U\_TIME CHARGE

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

141001 1000 2 1000

141001 1001 2 1000

SQL> select \* from charge\_02;

no rows selected

SQL> select \* from ch\_total;

U\_DATE CUST\_NO U\_TIME CHARGE

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

141001

141002

1. **데이터 MERGE**

SQL> MERGE INTO ch\_total total

2 USING charge\_01 ch01

3 ON(total.u\_date = ch01.u\_date)

4 WHEN MATCHED THEN

5 UPDATE SET total.cust\_no = ch01.cust\_no

6 WHEN NOT MATCHED THEN

7 INSERT VALUES(ch01.u\_date, ch01.cust\_no, ch01.u\_time, ch01.charge);

MERGE INTO ch\_total total

\*

ERROR at line 1:

ORA-30926: **unable to get a stable set of rows in the source tables**

**[실행결과]**

**unable to get a stable set of rows in the source tables가 뜨는 이유는 두가지 임을 알았습니다.**

1) INTO 절에 사용되는 테이블에 Primary Key 를 사용하는 경우

- 즉 INSERT 구문에서 DUPLICATE가 발생하거나 UPDATE 에 MULTI ROW가 UPDATE되는 경우

2) ON 구문에서 UPDATE되는 ROW가 1개 이상일 경우

- 즉 ON 구문에서 맞는 테이블 값이 하나 이상일 경우

**[문제2] PT6-27,링크에 설정된 웹사이트에서 사용한 예제들을 실행하고 결과가 동일한 지를 확인 할 것.**

<https://meyouus.tistory.com/187>

**DBMS\_RANDOM** 은 임의의 숫자를 생성하는 기능을 제공하며

**NORMAL** 은 표준 정규 분포(가우스분포) 에서의 임의 값을 출력합니다.

SQL> SELECT LEVEL NO, DBMS\_RANDOM.NORMAL

2 FROM DUAL

3 CONNECT BY LEVEL <= 10;

NO NORMAL

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

1 -.39101091

2 -1.0796917

3 -.47428565

4 -.60143273

5 .771106149

6 -1.5874004

7 -.17201107

8 1.62484259

9 -.62251821

10 1.44787479

10 rows selected.

**RANDOM**

-(2의 31제곱) 보다 크거나 같고, 2의 31제곱보다 작은 임의의 정수를 생성한다.

SQL> SELECT LEVEL NO, DBMS\_RANDOM.RANDOM

2 FROM DUAL

3 CONNECT BY LEVEL <= 10;

NO RANDOM

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

1 1970907350

2 1859298744

3 -752713595

4 -1.469E+09

5 1445214441

6 -1.771E+09

7 -24301032

8 -568592244

9 788531519

10 -1.928E+09

10 rows selected.

**STRING** 임의의 문자열 생성

텍스트이(가) 표시된 사진

자동 생성된 설명

SQL> SELECT DBMS\_RANDOM.STRING('u',15) AS "u\_Val"

2 ,DBMS\_RANDOM.STRING('U',15) AS "U\_Val"

3 ,DBMS\_RANDOM.STRING('l',15) AS "l\_Val"

4 ,DBMS\_RANDOM.STRING('L',15) AS "L\_Val"

5 ,DBMS\_RANDOM.STRING('A',15) AS "A\_Val"

6 ,DBMS\_RANDOM.STRING('a',15) AS "a\_Val"

7 ,DBMS\_RANDOM.STRING('x',15) AS "x\_Val"

8 ,DBMS\_RANDOM.STRING('X',15) AS "X\_Val"

9 ,DBMS\_RANDOM.STRING('P',15) AS "P\_Val"

10 ,DBMS\_RANDOM.STRING('p',15) AS "p\_Val"

11 FROM DUAL

12 CONNECT BY LEVEL <= 10;

u\_Val U\_Val l\_Val L\_Val A\_Val a\_Val x\_Val X\_Val

P\_Val p\_Val

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

BMAIPAMFJAMRZDI NCBOVTMDOJOWVQD ufxqgtbfacdoahw uvdzxlttknkbxdg HNUMhmUCSRfNgcF fOxiQZQdUcPgxbT Y8HECSBEC74FDLH KC1C6QK7PMPK075 \*B;ofz#x{KmEE#v |twGuNBOg'Xw#GS

YLUQBWKKNLAELTF QNCQAPBJSXEYFWD mzjtcwgulkkxcvr saawsqncugwptvy AtTtIFBjciDwAlt NnTmouelxwlryYw S9NTAK04I7D2YVF ISU5ZNHBKVYL7UM #ny3LBO"pFK,Ja^ <)Pl(bMLjO\_39#,

FBYLKVLLXGXGZPR HUKCRAHSIKNMDMN nsbvwwzxdooatus juhfpjgotlqplpx SxIzEnFxZqtTWqY miZegFsTCMxQhNX RBUAJNRUQ76NNME TZ8808OBZXLWHMC 5q:?+G"3`3xp-58 P`B;L|U@VJawei1

ZASANHBRVCAXCJG NFZIFBWBZXROFBX pafkfldfxcpsgjv fboouxzpvexrgfp JijCYMSLdHepjff vEFaYYYKilnwuiW 1BVL42XMLECNC8E IWOXOD5QF9WFMX4 \_GTKddrY62>Jipb q~)@\H<"Fq<6=3$

FPADWCJFVQSFDSN BATNPOWXYONSUTY tcfgiendbnmsili yjadkxuedatdtyq AHOCuCVhrefZWsC tpPyjvfdvCnrSAO TJACLDRQ46IZIK8 XNYEZF1CIK2W4EW @62,w2s2W/%>9N8 .R4sLf9EBjDY3K}

HCZIILEKTVCSHRM SFWSLNRJUHDQUIJ eehomjtlwazqyde fypgnvdwvuofpdq rSdCUdECGRlJaie jsRUahXxJLymZIe 0LSF2D706AJOSR6 ETP662U22U1BAJK /\*\wjlTc`i|G"19 lqhuD.mA6]q\\*f[

YJDVFIMIMWEUDFJ IRYVTLMDXUUZHAN lmbvpetnntrboap neaempbpwnseccg svqcioytkuEQyTo AAOvxOcRfTxLMvk FLFCABUJWAF8C4U D2L4FOGNFU8U1SK NEL,7B]lv`<JQ8' ,wG3ODuaT-:m:C&

XIYJKRPZYTBJAQS TPOWBSGYMDQRHKQ lodvbsddgaxyugd crgdzionioayprv QBbDfEEIoxoZbhP tQemNgBrvwydEWl JF5R2I1VEX1GQNM NFUWA9KMHQ80351 l[Xwdq}WQakA#Nv 8\_7go8o\_&39`.q6

OWNZVTITGEIFEYU EVHTMCGYVCAXHCP cnlarqcjfdkzmyu aanjyttkubqwkgy RVNiJzsPPAQNOKW tPxRYxhCaTIJQWw KFRHTLZB821CRM5 6MQQE4T1AFPKFZL xGf%N-q9Q`z@;\*) \*o`5\*T<-L6}.>xk

FXTZDNMQXRWPBAN FYWZQVCTDBEKXJK ybegrcwutausghd bncifqylrxhcdwn xVkEwbjxpNLmdMw xINKVTCspEsROUb LT7G3P7X8XBQTXG KWLX4PG6PB191DC %9j b=Kk01tM~"@ n[hF!~$fX8usLNK

**VALUES**는 범위 내에서 임의의 숫자를 생성한다. 범위가 지정되지 않은 경우는 0보다 크거나 같고 1보다 작은 숫자를 생성한다.

SQL> SELECT DBMS\_RANDOM.VALUE AS VAL1

2 ,DBMS\_RANDOM.VALUE(1,10) AS VAL2

3 ,DBMS\_RANDOM.VALUE(100,1000) AS VAL3

4 ,TRUNC(DBMS\_RANDOM.VALUE(1000,10000)) AS VAL4

5 FROM DUAL

6 CONNECT BY LEVEL<=10;

VAL1 VAL2 VAL3 VAL4

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

.562013733 3.70614206 349.272776 9077

.059077989 7.68516952 493.784599 9642

.746245247 6.81438535 460.839502 8945

.976502586 4.4637212 793.347847 7783

.071692589 8.10240203 940.115291 3360

.572048632 2.50827561 607.144498 8345

.791759728 5.87289877 900.516244 3269

.849369617 4.90387376 655.653664 7195

.527743989 3.79898131 842.972222 3794

.117552042 2.02642319 415.290515 4836

**=================응용=================**

**100에서 5000 사이의 임의의 수 조회**

SELECT DBMS\_RANDOM.VALUE(100,5000) RANDOM FROM DUAL;

RANDOM

----------

1964.26578

**10에서 100 사이의 임의의 정수 조회**

SELECT TRUNC(DBMS\_RANDOM.VALUE(10,100)) RANDOM FROM DUAL;

RANDOM

----------

28

**TABLE 랜덤하게 쟁렬**

SELECT \* FROM EMP WHERE ROWNUM <= 10

ORDER BY DBMS\_RANDOM.RANDOM();

EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO

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

7782 CLARK MANAGER 7839 06-SEP-99 2450 10

7369 SMITH CLERK 7902 01-DEC-07 800 20

7839 KING PRESIDENT 17-NOV-96 5000 10

7566 JONES MANAGER 7839 04-FEB-01 2975 20

7654 MARTIN SALESMAN 7698 28-SEP-81 1250 1400 30

7698 BLAKE MANAGER 7839 05-JAN-91 2850 30

7521 WARD SALESMAN 7698 22-FEB-81 1250 500 30

7499 ALLEN SALESMAN 7698 20-FEB-81 1600 300 30

7788 SCOTT ANALYST 7566 17-JUN-03 3000 20

7902 FORD ANALYST 7566 12-MAR-81 3000 20

10 rows selected.

**랜덤문자 출력**

SELECT DBMS\_RANDOM.STRING('X',10) AS "BIG"

,DBMS\_RANDOM.STRING('U',10) AS "SMALL"

,DBMS\_RANDOM.STRING('P',10) AS "ENGLISH"

,DBMS\_RANDOM.STRING('L',10) AS "ENG+NUM"

,DBMS\_RANDOM.STRING('A',10) AS "CHAR"

FROM DUAL

CONNECT BY LEVEL <= 20;

BIG SMALL ENGLISH ENG+NUM CHAR

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

3KZASGCV4R BERMPGUFUU F4U5-8.-)\* sgsuklbonf GvBqiWxlue

87FMQ8T4FW PBDVOMPTAC Jv,mZ']Py# vjioslmnyv RhggCkTgdk

JEM0EU934O HRNTLHMQJN =xD]fOYPJ0 uexobgfqqv FoKXfnxFNW

7OKYC9W8IU ABGAGIYTKB ,YEW{wb?h~ yufadclvvp GMiutSFHKz

T2PN05L9A3 GVBSWDESVB v$~$z!zOe/ xnkmqyzvhk LlhsQzUqfl

F0671S3O5Q INURPXTANE SDIMw5$DsU tjnxjyindu jPRFEqgUtr

5X717C88KW IVBRRRMDFT aI<q9\2$sk bficpystdy tiRKWRzpLP

KAU6PFKE32 DHLXAYTTKV \?iRy+j;^M ytjszuqwmr AsDfrteEyr

5M4ZTIFMGD XMELUCCCTM [hZLGL3CZw zyxumtguil HcSiGpyHjf

D7ULBK3CDF KKOJEKFKFD S4\,.hVWX( bopmnhnzdv JjIupPxeES

XICBQIZF6W REXYXYRBJH 6F)E"?QWd6 dmcykucoxr JIHBpEfryH

BIG SMALL ENGLISH ENG+NUM CHAR

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

L6Z6RYUVWG TCYGWBYZNU Yo6O@"Nb5~ mdtskwlyvh MqiqwBLFAP

V7O7NX7HVZ YIZREEZJER B2l0[RsKXA ijxxedbzfh sUMqxUzhKF

PLATHIQJN8 HNWUXDLBQJ KojHO>Xkmo qpbwevxqlw TSaDEnbkte

RFGZJ7WVLA JJDSDMITLJ ~mH,GSX>,D xjghvevyuu FMBfdZrsqa

3JIBP6Z2AO PVGVPEOYRX 8J~JyU<8|5 bswqrqofjd sVnAgjVecM

D3ASY13TKO KAUDIHLUBY ;HBIvt,E#N vrjhaatymg PyMGOIkKqj

MO0WOAZF7Z OXNXZLBNCU Seg<$TX%\d otcbkmzisg AQsyINVOiT

Y8XKZN8LYB AOEHBYKMGR `$ozgc~/Wc yohohokirj qyuubDNNWV

X3BHBPYXQK JEEGWHCPQS =4I-U3\*(32 dygxfgggma GoargjMAYH

**[실행결과]**

여러가지 난수 생성 random을 생성하면서 값의 출력이 결과 값과 같았습니다.

Merge에 대해서 처음에 의아했습니다. Primary key로 조인할 때 select 결과가 2개이상 나오면 에러가 난다는 것입니다. 처음엔 그냥 참, 거짓으로 확실히 나뉘고 2개이상이면 update 쿼리나 insert 쿼리를 둘중에 하나 하면 되지 않나 라는 생각을 했습니다. 하지만 스펙상 그게 불가능 했고 위 1번문제의 결과에서 찾아낼 수 있었습니다.