Question:

Does DBMaker have the solution for Oracle syntax like “SELECT id FROM groups START WITH id=2 CONNECT BY PRIOR parent_id = id”, if not is there some other method to get the same result? (DBMR1954)

Answer:

DBMaker doesn't support recursive SQL, but it can get same result by splitting several SQL.

For example, Customer can via temp table and run following steps:

```
drop table t1;

drop table t2;

create table t1 (c1 int, c2 int);

create table t2 (c1 int, c2 int);



insert into t1 values (1, 11);

insert into t1 values (1, 12);

insert into t1 values (2, 21);

insert into t1 values (11, 111);

insert into t1 values (11, 112);

insert into t1 values (12, 121);

insert into t1 values (12, 122);

insert into t1 values (12, 123);

insert into t1 values (13, 131);

insert into t1 values (14, 141);

insert into t1 values (121, 1211);

insert into t1 values (123, 1231);
```

```
insert into t1 values (131, 1311);

insert into t1 values (1231, 12311);

insert into t1 values (12311, 123111);


/*

SELECT c1 FROM t1

   START WITH c1=1 CONNECT BY

    PRIOR c1 = c2;


RecursiveQuery(sqlcmd, startwith, prior)


RecursiveQuery("SELECT c1 FROM t1", "c1=1", "c1=c2") */


drop table tt1;

drop table tt2;

create table tt1 (c1 int, c2 int);

create table tt2 (c1 int, c2 int);


delete from tt1;

select c1, c2 from t1 where c1=1 into tt1; select * from tt1 into t2;


// loop begin


delete from tt2;
```

```
select t1.c1, t1.c2 from t1, tt1 where tt1.c2=t1.c1 into tt2; select * from tt2 into t2;

delete from tt1;
select t1.c1, t1.c2 from t1, tt2 where tt2.c2=t1.c1 into tt1; select * from tt1 into t2;

delete from tt2;
select t1.c1, t1.c2 from t1, tt1 where tt1.c2=t1.c1 into tt2; select * from tt2 into t2;

delete from tt1;
select t1.c1, t1.c2 from t1, tt2 where tt2.c2=t1.c1 into tt1; select * from tt1 into t2;

delete from tt2;
select t1.c1, t1.c2 from t1, tt1 where tt1.c2=t1.c1 into tt2; select * from tt2 into t2;

// loop end until select into no rows

drop table tt1;
drop table tt2;

select distinct c1 from t2;
```

Also, DBMaker SP can help:

First, create the sp **connby.ec**

```
/*
SELECT id FROM groups
  START WITH id=2 CONNECT BY
    PRIOR parent_id = id;
*/

exec sql create procedure connectby_groups (integer hid) returns int outid;
{
exec sql begin declare section;
int i, cnt;
char sqlstr[256];
exec sql end declare section;

exec sql begin code section;

$ whenever sqlerror continue;
exec sql drop table temp_connectby;
$ whenever sqlerror goto dmSP_EXIT_LABEL;

exec sql create temp table temp_t1 (id int, parent_id int);
exec sql create temp table temp_t2 (id int, parent_id int);
exec sql create temp table temp_connectby (id int, parent_id int);

exec sql select id, parent_id from groups where id=:hid into temp_t1;
```

```
exec sql select id, parent_id from temp_t1 into temp_connectby;



do {

   exec sql delete from temp_t2;

   exec sql select a.id, a.parent_id from groups a, temp_t1 b where a.id=b.parent_id
into temp_t2;

   exec sql select count(*) from temp_t2 into :cnt;

   if (cnt > 0)

      exec sql select id, parent_id from temp_t2 into temp_connectby;

   else

      break;



   exec sql delete from temp_t1;

   exec sql select a.id, a.parent_id from groups a, temp_t2 b where a.id=b.parent_id
into temp_t1;

   exec sql select count(*) from temp_t1 into :cnt;

   if (cnt > 0)

      exec sql select id, parent_id from temp_t1 into temp_connectby;

   else

      break;

   } while (1);



exec sql RETURNS select distinct id from temp_connectby into :outid;



exec sql drop table temp_t1;
```

```
exec sql drop table temp_t2;

exec sql end code section;

}
```

Then run the sample SQLs:

```
create db sample5;


create table groups (id int, parent_id int);


insert into groups values (1, null);

insert into groups values (11, 1);

insert into groups values (12, 1);

insert into groups values (13, 1);

insert into groups values (111, 11);

insert into groups values (112, 11);

insert into groups values (131, 13);

insert into groups values (1121, 112);

insert into groups values (11211, 1121);

insert into groups values (112111, 11211);


terminate db;


start db sample5 sysadm;


create proc from 'connby.ec';
```

```
call connectby_groups(112111);

call connectby_groups(1);

call connectby_groups(131);

call connectby_groups(168);

call connectby_groups(11211);


terminate db;
```