How to accomplish row-level security

Version: 01.00

Document No: 54/DBM54-T11262018-1-RLSE
Author: DBMaster Support Team, SYSCOM Computer Engineering CO.
Publication Date: November 26, 2018

凌羣電腦
THE SYSCOM GROUP

Table of Content

1. Introduction

DBMaker can let user accomplish row-level security. Row-level security can be used to restrict data access for given users. Filters restrict data access at the row level, and you can define filters within roles. DBMaker uses select from stored procedure feature to implement it.

2. An example to implement row-level security

2.1 Create table/data

```
create table sales(orderid int,salesrep varchar(20),product varchar(10), qty int);
insert into sales values(1, 'sales1', 'valve', 5);
insert into sales values(2, 'sales1', 'wheel', 2);
insert into sales values(3, 'sales1', 'valve', 4);
insert into sales values(4, 'sales2', 'bracket', 2);
insert into sales values(5, 'sales2', 'wheel', 5);
insert into sales values(6, 'sales2', 'seat', 5);
```

2.2 Create new user

```
grant connect to "sales1" "sales1";
grant resource to "sales1";
grant connect to "sales2" "sales2";
grant resource to "sales2";
grant connect to "manager" "manager";
grant dba to "manager";
```

2.3 Ceate SQLSP for table “sales” to filter records with column “salesrep”

```
create or replace procedure sp salesfilter returns int orderid, varchar(20) salesrep, varchar(10)
 product, int qty
language sql
begin
    declare cur scroll cursor with return for select * from sales where upper(salesrep)=user() or user()=
 'MANAGER' or user()= 'SYSADM';
    open cur;
end;
```

2.4 Grand authority to SQLSP for all users

```
grant execute on procedure "sp_salesfilter"."sysadm"."sp_salesfilter" to "sales1";
grant execute on procedure "sp_salesfilter"."sysadm"."sp_salesfilter" to "sales2";
grant execute on procedure "sp_salesfilter"."sysadm"."sp_salesfilter" to "manager";
```

2.5 Select table with different user gets different result

```
connect to dbsample5 sales1 sales1;
call sp_salesfilter.sysadm.sp_salesfilter;
```

or

```
select * from (call sp salesfilter.sysadm.sp salesfilter);

  ORDERID          SALESREP          PRODUCT       QTY
=========== ==================== ========== ===========
       1           sales1             valve         5
       2           sales1             wheel         2
       3           sales1             valve         4
```

```
connect to dbsample5 sales2 sales2;
call sp salesfilter.sysadm.sp salesfilter;

  ORDERID       SALESREP              PRODUCT     QTY
=========== ==================== ========== ===========
       4         sales2             bracket        2
       5         sales2             wheel          5
       6         sales2             seat           5
```

```
connect to dbsample5 manager manager;
call sp salesfilter.sysadm.sp salesfilter;

  ORDERID           SALESREP         PRODUCT       QTY
=========== ==================== ========== ===========
       1            sales1           valve          5
       2            sales1           wheel          2
       3            sales1           valve          4
       4            sales2           bracket         2
       5            sales2           wheel            5
       6            sales2           seat            5
```