DBMaker

TDE User's Guide

Version: 01.00

Document No: 54/DBM546-T01262024-01-TDEG
Author: Production Team
Syscom Computer Engineering CO.

Print Date: July 11, 2024

凌羣電腦
THE SYSCOM GROUP

Table of Content

1. Introduction

Welcome to the DBMaker TDE User's Guide. This guide discusses the details about the TDE (transparent data encryption) and guides the user how to use it to encrypt database and columns.

1.1. Preface of TDE

Transparent Data Encryption (often abbreviated to TDE) is a technology employed by Microsoft, IBM and Oracle to encrypted database files. TDE offers encryption at file level. TDE solves the problem of protecting data at rest, encrypt databases both on the hard drive and consequently on backup media. It does not protect data in transit nor data in use. Enterprises typically employ TDE to solve compliance issues such as PCI DSS which require the protection of data at rest.

After DBMaker version 5.4.6. Users can set the keyword DB_TDEMd=1 to create an encrypted database. In the encrypted database users can set TDE password and encrypt columns, refer to chapter below for more details.

2. Database Encryption

This chapter tells users how to activate database encryption.

2.1. Created Encrypted Database

To activate database encryption, users have to set keyword DB_TDEMd=1 in dmconfig.ini before database is created.

```
DB_TDEMd=1
```

After encrypted database is started, users can use the following command to check if database is encrypted:

```
dmsql> select * from SYSINFO where ID='0724';


ID          INFO                           VALUE
==== ================================ ================================
0724 TDE_DB                    ON
```

3. Column Encryption

Column encryption, sometimes known as column-level encryption or cell-level encryption, is used to prevent unauthorized access. If column is closed, users cannot insert/delete/update/select data from that column.

After encrypted database is created. SYSADM and SYSDBA can set TDE password and open/close column encryption.

3.1. TDE Password

Before activating column encryption, SYSDBA or SYSADM must set the TDE password, this password is used to open/close column encryption. "ERROR (6891): [DBMaker] please set new TDE password" will be returned if users try to open/close without TDE password is set.

3.1.1. SET/ALTER TDE PASSWORD

SYSADM/SYSDBA can use the following query to set and alter TDE password:

```
dmsql> alter tde password OLD_PASSWORD to NEW_PASSWORD;
```

First time set the TDE password, use NULL to represent OLD_PASSWORD

➲Example 1:

```
dmsql> alter tde password NULL to ABC;
```

➲Example 2:

```
dmSQL> ALTER TDE PASSWORD ABC TO new_password;
```

3.1.2. RESET TDE PASSWORD

SYSADM can use the following query to reset TDE password:

```
dmSQL> ALTER PASSWORD OF _DMTDE TO new_password;
```

3.2. Open/Close Column Encryption

After TDE password is set, SYSADM and SYSDBA can open/close column encryption. Open column means everyone with this column's privilege can access; Close column means no one can access the column.

3.2.1. OPEN/CLOSE COLUMN

SYSADM and SYSDBA can use the following query to open/close column:

```
dmSQL> CALL SETSYSTEMOPTION('tde_open','password');
dmSQL> CALL SETSYSTEMOPTION('tde_close','password');
```

Users can use the following query to check if column is opened or closed

```
dmSQL> SELECT * FROM SYSINFO WHERE ID='0725';

 ID                   INFO                                VALUE
==== ================================ ================================
0725 TDE_STATUS                       CLOSE
```

3.2.2. AUTO OPEN/CLOSE COLUMN

SYSADM and SYSDBA can use the following query to open/close column automatically when database is started.

```
dmSQL> CALL SETSYSTEMOPTION('tde_open_auto','password');
dmSQL> CALL SETSYSTEMOPTION('tde_close_auto','password');
```

Users can use the following query to check if column open auto is on or off

```
dmSQL>  SELECT * FROM SYSINFO WHERE ID='0726';

 ID                   INFO                                VALUE
==== ================================ ================================
0726 TDE_OPEN_AUTO                    ON
```

3.3. Encrypt/Decrypt Column

When column is opened, users can create encrypted columns. Keyword ENCRYPT can be used in CREATE TABLE and ALTER TABLE commands.

3.3.1. CREATE ENCRYPTED COLUMN

When column is opened, ENCRYPT keyword in CREATE TABLE command can be used to define encrypted column. Here are some examples:

➲Example 1

```
dmSQL> create table t1(c1 int encrypt);
```

➲Example 2

```
dmSQL> create table t2(c1 int primary key encrypt);
```

3.3.2. ENCRYPT/DECRYPT COLUMNS WITH ALTER TABLE

When column is opened, users can use ALTER TABLE command to encrypt/decrypt existed columns. Here are some examples:

➲Example 1

The following query shows how to add an encrypt column c1 into table T1

```
dmSQL> alter table t1 add column c1 int encrypt;
```

➲Example 2

The following two queries show how to encrypt column c1 in table T1

```
dmSQL> alter table t1 modify c1 encrypt;
dmSQL> alter table t1 modify c1 to c1 int encrypt;
```

➲Example 3

The following two queries show how to decrypt column c1 in table T1

```
dmSQL> alter table t1 modify c1 decrypt;
dmSQL> alter table t1 modify c1 to c1 int decrypt;
```

3.4. Working with Encrypted Columns

This chapter will teach users how to work with encrypted columns, include DML and other operations (domain/trigger/view/synonym).

3.4.1. DML WITH ENCRYPTED COLUMN

Every DML operations involves encrypted column will be protected when column is closed. "ERROR (6747): [DBMaker] TDE mode is not open" will be returned if users try to access encrypted columns when column is closed.

```
dmSQL> def table t1;
create table SYSADM.T1 (
 C1  INTEGER default null  encrypt  )
 in DEFTABLESPACE  lock mode row  fillfactor 80 ;


dmSQL> insert into t1 values(1);
ERROR (6747): [DBMaker] TDE mode is not open


dmSQL> update t1 set c1=2;
ERROR (6747): [DBMaker] TDE mode is not open


dmSQL> select * from t1;
ERROR (6747): [DBMaker] TDE mode is not open
```

```
dmSQL> delete from t1 where c1=1;
ERROR (6747): [DBMaker] TDE mode is not open
```

NOTE: If there's no clauses with encrypted column, DELETE statement will still success.

3.4.2. DOMAIN/TRIGGER/VIEW/SYNONYM

Domain, trigger, view, synonym with encrypted column can still be created properly when tde_status is closed. But column encryption must be opened when users want to use these elements to create table or select data.

Here are some examples (assume tde_status is closed)

Domain:

```
dmSQL> create domain d1 as int encrypt;

dmSQL> create table t1 (c1 d1);
ERROR (6747): [DBMaker] TDE mode is not open
```

Trigger:

```
dmSQL> def table t1;
create table SYSADM.T1 (
 C1  INTEGER default null  encrypt  )
 in DEFTABLESPACE  lock mode row  fillfactor 80 ;

dmSQL> def table t2;
create table SYSADM.T2 (
 C1  INTEGER default null  )
 in DEFTABLESPACE  lock mode row  fillfactor 80 ;

dmSQL> create trigger tr1 after insert on t2 for each row (insert into t1 values(new.c1));

dmSQL> insert into t2 values(1);
ERROR (6747): [DBMaker] TDE mode is not open
```

View:

```
dmSQL> create view v1 as select * from t1;

dmSQL> select * from v1;
ERROR (6747): [DBMaker] TDE mode is not open
```

Synonym:

```
dmSQL> create synonym s1 for t1;
```

```
dmSQL> select * from s1;
ERROR (6747): [DBMaker] TDE mode is not open
```

3.5. Constraints on TDE

1. DB_TDEMd=1 has to be set before database is created.
2. Column encryption can only be used in **Encrypted Database**.
3. Encrypted primary key cannot be referenced.

   Referenced primary key cannot be encrypted.
4. Foreign key cannot be encrypted.