Data Security using Real Time Data Masking in Relational Database Management System

[1]Ravishankar Belkunde
Senior Database Administrator & Researcher
Jefferson City, Missouri, 65109

Abstract : Data security is vital for any organization. Data can be secured in different ways many IT security experts would agree that the best way to protect data it to enforce security layers. Real Time Data Masking is best way to enforce security layers over sensitive live data. This paper presents research on different real time data masking implementation and its limitations**.**

***IndexTerms* - Data Security, Data Masking, Relational Database Management System, Real Time Data Security in Database, Oracle Data Redaction, Dynamic Data Masking.**

I. INTRODUCTION

Data Masking is defined as a method of creating structurally similar but inauthentic version of original data, it is the process of hiding original data with modified content. Data Masking is used to secure classified data from unauthorized access, however inauthentic version of data remains usable and visible. Data Masking can be broadly classified in two types as Static Data Masking and Real Time Data Masking. In Static Data Masking original data is permanently replaced with masked inauthentic data. This gives the user feel of real data as structure of the masked data remains same, this data can be used for testing demo purpose. The Real Time Data Masking process masks sensitive data on the fly actual data is not changed, data is dynamically changes in-transit, Real Time Data Masking is required during production live real production data testing or during partial data visibility access authorization. Following fig 1. & fig 2. depict block diagram of the Static Data Masking and Real Time Data Masking.

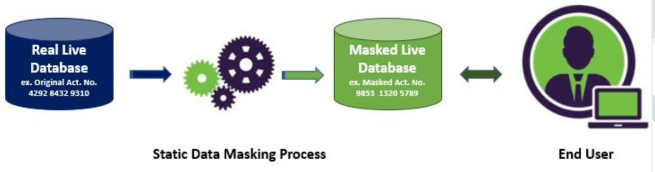

Real Live Database
ex. Original Act. No. 4292 8432 9310
Masked Live Database
ex. Masked Act. No. 9853 1320 5789
Static Data Masking Process
End User

Fig 1. Static Data Masking

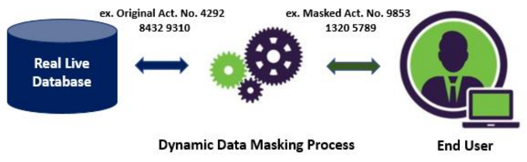

ex. Original Act. No. 4292 8432 9310
ex. Masked Act. No. 9853 1320 5789
Real Live Database
Dynamic Data Masking Process
End User

Fig 2. Real Time Data Masking

II. REAL TIME DATA MASKING IMPLEMENTATION

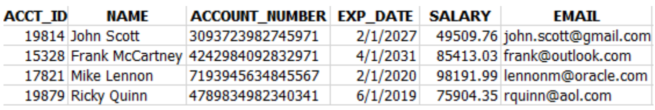
A. FULL REAL TIME DATA MASKING: In full real time data masking original data is completely replaced with void values as per the data type. For example (char, nchar, varchar, nvarchar, text, ntext) type values are replaced by Xs, (number, bigint, bit, decimal, int, money, numeric, smallint, smallmoney, tinyint, float, real) type values are replaced by zero and binary types are replaced by single byte of ASCII value 0. Following code illustrates full real time data masking in Oracle and MS SQL database. In the code we give the details of table owner, table name and sensitive data column name. In oracle we define masking policy and associate it with *DBMS_REDACT.FULL* function and in MS SQL database the data is masked with *default()* function. In below example we have masked all values of the account number column.

Oracle snippet:

```
BEGIN
 DBMS_REDACT.ADD_POLICY (
   object_schema       => 'BANKADMIN',
   object_name         => 'ACCOUNT_DETAILS',
   column_name         => 'ACCOUNT_NUMBER',
   policy_name         => 'ACCOUNT_NUMBER_POLICY',
   function_type       => DBMS_REDACT.FULL,
   expression          => '1=1'
 );
END;
/
```

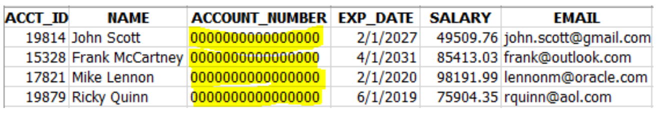
MS SQL snippet:

```
ALTER TABLE BANKADMIN.ACCOUNT_DETAILS
ALTER COLUMN ACCOUNT_NUMBER NUMBER(16)
MASKED WITH (FUNCTION = 'default()');
```



ACCT_ID	NAME	ACCOUNT_NUMBER	EXP_DATE	SALARY	EMAIL
19814	John Scott	3093723982745971	2/1/2027	49509.76	john.scott@gmail.com
15328	Frank McCartney	4242984092832971	4/1/2031	85413.03	frank@outlook.com
17821	Mike Lennon	7193945634845567	2/1/2020	98191.99	lennonm@oracle.com
19879	Ricky Quinn	4789834982340341	6/1/2019	75904.35	rquinn@aol.com

Fig 3. Original Data before masking



ACCT_ID	NAME	ACCOUNT_NUMBER	EXP_DATE	SALARY	EMAIL
19814	John Scott	0000000000000000	2/1/2027	49509.76	john.scott@gmail.com
15328	Frank McCartney	0000000000000000	4/1/2031	85413.03	frank@outlook.com
17821	Mike Lennon	0000000000000000	2/1/2020	98191.99	lennonm@oracle.com
19879	Ricky Quinn	0000000000000000	6/1/2019	75904.35	rquinn@aol.com

Fig 4. Data after Full Real Time Data Masking

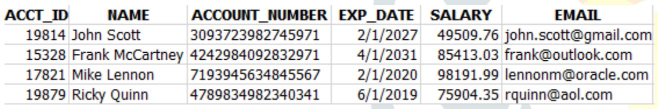
B. PARTIAL REAL TIME DATA MASKING: In partial data masking original data gets partially masked as per the business need and data sensitivity. The data is masked with user defined values at user defined position/s. Following code illustrates partial real time data masking in Oracle and MS SQL database. In the code we give the details of table owner, table name and sensitive data column name. In Oracle we define masking policy & associate *DBMS_REDACT.PARTIAL* function to it with input parameters as per business needs ex. *'0,1,12'* i.e. mask data value, start position and number of values. In MS SQL database data is masked using *partial()* function and input parameters as start position, mask value and count of unmask data. In this example we have masked position 1 to 12 of the account number with a zero.

Oracle snippet:

```
BEGIN
 DBMS_REDACT.ADD_POLICY (
   object_schema      => 'BANKADMIN',
   object_name        => 'ACCOUNT_DETAILS',
   column_name        => 'ACCOUNT_NUMBER',
   policy_name        => 'ACCOUNT_NUMBER_POLICY',
   function_type      => DBMS_REDACT.PARTIAL,
   function_parameters => '0,1,12',
   expression         => '1=1'
 );
END;
/
```

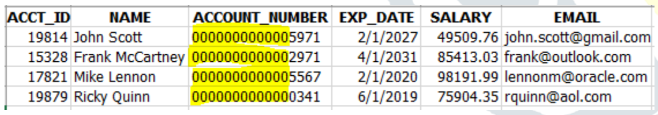
MS SQL snippet:

```
ALTER TABLE BANKADMIN.ACCOUNT_DETAILS
ALTER COLUMN ACCOUNT_NUMBER NUMBER (16)
MASKED WITH (FUNCTION = 'partial ( 0,
"XXXXXXXXXXXX" , 4)' );
```



ACCT_ID	NAME	ACCOUNT_NUMBER	EXP_DATE	SALARY	EMAIL
19814	John Scott	3093723982745971	2/1/2027	49509.76	john.scott@gmail.com
15328	Frank McCartney	4242984092832971	4/1/2031	85413.03	frank@outlook.com
17821	Mike Lennon	7193945634845567	2/1/2020	98191.99	lennonm@oracle.com
19879	Ricky Quinn	4789834982340341	6/1/2019	75904.35	rquinn@aol.com

Fig 3. Original Data before masking



ACCT_ID	NAME	ACCOUNT_NUMBER	EXP_DATE	SALARY	EMAIL
19814	John Scott	0000000000005971	2/1/2027	49509.76	john.scott@gmail.com
15328	Frank McCartney	0000000000002971	4/1/2031	85413.03	frank@outlook.com
17821	Mike Lennon	0000000000005567	2/1/2020	98191.99	lennonm@oracle.com
19879	Ricky Quinn	0000000000000341	6/1/2019	75904.35	rquinn@aol.com

Fig 5. Data after Partial Real Time Data masking

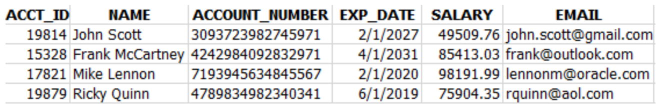
C. RANDOM REAL TIME DATA MASKING: In random data masking all original data values are masked with some random values as per the business need and data sensitivity. Following code illustrates random real time data masking in Oracle and MS SQL database. In the code we give details of table owner, table name and sensitive data column name. In Oracle we define the policy and associate *DBMS_REDACT.RANDOM* function to it. In MS SQL database data is masked using *random()* function and provide input parameters as a range of lowest random value and max random value. In this MS SQL example we have replaced original account numbers with random numbers within 4200000000000000 to 9999999999999999 range.

Oracle snippet:

```
BEGIN
 DBMS_REDACT.ADD_POLICY (
   object_schema      => 'BANKADMIN',
   object_name        => 'ACCOUNT_DETAILS',
   column_name        => 'ACCOUNT_NUMBER',
   policy_name        => 'ACCOUNT_NUMBER_POLICY',
   function_type      => DBMS_REDACT.RANDOM,
    expression        => '1=1'
 );
END;
/
```

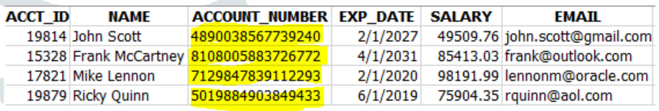
MS SQL snippet:

```
ALTER TABLE BANKADMIN.ACCOUNT_DETAILS
ALTER COLUMN ACCOUNT_NUMBER NUMBER (16)
MASKED WITH (FUNCTION = 'random
(4200000000000000,9999999999999999)');
```



ACCT_ID	NAME	ACCOUNT_NUMBER	EXP_DATE	SALARY	EMAIL
19814	John Scott	3093723982745971	2/1/2027	49509.76	john.scott@gmail.com
15328	Frank McCartney	4242984092832971	4/1/2031	85413.03	frank@outlook.com
17821	Mike Lennon	7193945634845567	2/1/2020	98191.99	lennonm@oracle.com
19879	Ricky Quinn	4789834982340341	6/1/2019	75904.35	rquinn@aol.com

Fig 3. Original Data before masking



ACCT_ID	NAME	ACCOUNT_NUMBER	EXP_DATE	SALARY	EMAIL
19814	John Scott	4890038567739240	2/1/2027	49509.76	john.scott@gmail.com
15328	Frank McCartney	8108005883726772	4/1/2031	85413.03	frank@outlook.com
17821	Mike Lennon	7129847839112293	2/1/2020	98191.99	lennonm@oracle.com
19879	Ricky Quinn	5019884903849433	6/1/2019	75904.35	rquinn@aol.com

Fig 6. Data after Random Real Time Data masking

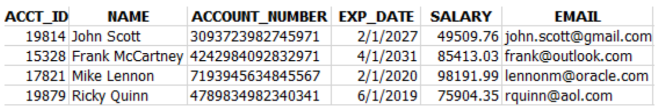
D. EXPRESSION REAL TIME DATA MASKING: In expression data masking original data values are masked based on pattern matching. We could define different expressions using conditions and variables to identify and mask sensitive data. Predefined patterns can also be used. Following code illustrates expression real time data masking in Oracle and MS SQL database. In the code we give details of table owner, table name and sensitive data column name. In Oracle we define the policy and associate *DBMS_REDACT.REGEXP* function and give expression pattern, replace string value, position and number of occurrence. In MS SQL database data is masked using *email()* function. In the example account holders email address first part is masked.

Oracle snippet:

```
BEGIN
 DBMS_REDACT.ADD_POLICY (
   object_schema      => 'BANKADMIN',
   object_name        => 'ACCOUNT_DETAILS',
   column_name        => 'EMAIL',
   policy_name        => 'ACCOUNT_NUMBER_POLICY',
   function_type      => DBMS_REDACT.REGEXP,
   regexp_pattern     =>  DBMS_REDACT.
   RE_PATTERN_EMAIL_ADDRESS,
   regexp_replace_string =>
   DBMS_REDACT.RE_REDACT_EMAIL_NAME,
   regexp_position    => DBMS_REDACT.RE_BEGINNING,
   regexp_occurrence  => DBMS_REDACT.RE_ALL,
   expression         => '1=1'
 );
END;
/
```

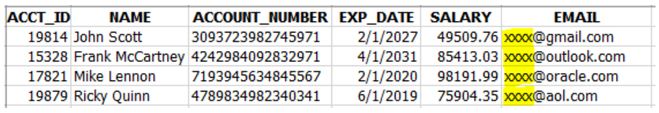
MS SQL snippet:

```
ALTER TABLE BANKADMIN.ACCOUNT_DETAILS
ALTER COLUMN ACCOUNT_NUMBER NUMBER(16)
MASKED WITH (FUNCTION = 'Email()');
```



ACCT_ID	NAME	ACCOUNT_NUMBER	EXP_DATE	SALARY	EMAIL
19814	John Scott	3093723982745971	2/1/2027	49509.76	john.scott@gmail.com
15328	Frank McCartney	4242984092832971	4/1/2031	85413.03	frank@outlook.com
17821	Mike Lennon	7193945634845567	2/1/2020	98191.99	lennonm@oracle.com
19879	Ricky Quinn	4789834982340341	6/1/2019	75904.35	rquinn@aol.com

Fig 1. Original Data before masking



ACCT_ID	NAME	ACCOUNT_NUMBER	EXP_DATE	SALARY	EMAIL
19814	John Scott	3093723982745971	2/1/2027	49509.76	xxxx@gmail.com
15328	Frank McCartney	4242984092832971	4/1/2031	85413.03	xxxx@outlook.com
17821	Mike Lennon	7193945634845567	2/1/2020	98191.99	xxxx@oracle.com
19879	Ricky Quinn	4789834982340341	6/1/2019	75904.35	xxxx@aol.com

Fig 7. Data after Expression Real Time Data masking

III. REAL TIME DATA MASKING LIMITATIONS

A. Oracle database do not support real time data masking for the following

   a. SYS/SYSTEM owned database objects cannot be masked.
   b. At data type level data masking not allowed.
   c. Real time data masking can’t be used on virtual columns.
   d. More than one real time data masking policy for a table or view not allowed.

B. MS SQL database do not support real time data masking for the following:

   a. FILESTREAM
   b. COLUMN_SET or a sparse column that is part of a column set.
   c. Computed columns and Non mask dependent computed column.
   d. Masked columns cannot be used as a key for FULLTEXT index.

C. Real Time Data Masking could be by passed using inference or brute-force cyber-attack techniques, proxy is a single point of failure and can be bypassed by users connecting directly to the database potentially exposing the original data stored in the database.

D. Realtime read/write enterprise application could cause data corruption if limited authorized user have database write privilege, masked data could be written back to the database, corrupting original data.

E. In real time data masking actual data is not changed super users and database administrators can have access to original data.

F. The data will be masked on the fly all database traffic will be inspected for masking there will be performance overhead associated with it.

G. Real time data masking is a less mature technology compared to static data masking, currently limited masking functionalities could be used.

H. In real time data masking, masked data values are not persistent.

IV. CONCLUSION

Real time data masking is powerful data security technology. It is still less mature with few functionalities and have long way to go. Data masking empowers organizations to enforce extra security layer on top of existing security over live data with ease. If implemented cautiously it could resolve many complex real time data security problems.

V. REFERENCES

[1] Scott Gaetjen, David Knox, William Maroulis Oracle Database 12c Security V1.0.