DATA FILTERS/ MASKS TO SECURE PII & SPI DATA

VALLI RAMANATHAN

ORACLE CERTIFIED CLOUD ARCHITECT

INTRODUCTION

Valli Ramanathan

- Master's in Cloud Computing Architecture From University of Maryland
- Emphasis on Cloud Architecture, Oracle Architect & Database migration specialist
- Certified Oracle Cloud Architect Professional

DATA MASKING – WHY DO WE NEED IT ? WHAT IS IT?

VALLI RAMANATHAN

PII & SPI

- Data in the database is available for anyone with access to view
- Is this accurate to have Personal Identifiable Information like such?
- Is this accurate to have Sensitive Personal Information like such?
- Should there be measures to secure this data?

WHY DO WE NEED DATA MASKING?

- Organizations share data
- Production data is used to refresh lower environments
- Market research to stay competitive
- Healthcare organizations share patient data with medical researchers to assess the efficiency of clinical trials or medical treatments

DATA MASKING – WHAT IS THE DRIVING FACTOR

- Data breach is becoming very common
- Regulations:
  - HIPAA
  - PCI
  - DSS
- Security compliance – SOX, Audit regulations.

RECENT DATA BREACHES

- Social Media Profiles Data Leak – 4 Billion Records
- Capital One – 106 million records.
- State Farm – Unknown.
- Biometric Records – 27 million records.
- Quest Diagnostics/AMA – 24 million records.
- Ecuador Breach – 20 million records.
- Hostinger – 14 million records.
- DoorDash Breach – 5 million records.

HANDLING THE SENSITIVE DATA

- Identify the sensitive data
- Mask sensitive data
- Periodic Audit checks
- Compliance certifications.

DEFINE PII & SPI IN THE DATABASE

- To begin the process of masking data, the data elements that need to be masked in the application must be identified. The first step that any organization must take is to determine what is sensitive. This is because sensitive data is related to specific government regulations and industry standards that cover how the data can be used or shared. Thus, the first step is for the security administrator to publish what constitute sensitive data and get agreement from the company's compliance or risk officers

IDENTIFY THE EXACT PII & SPI DATA

Few examples:

- Credit card
- SSN
- Health industry – HIPPA related
- Financial industry -- SOX compliance
- Proprietary information

WHAT TO MASK AND WHERE

- On Site – always masked
- During transaction
- Replicated data
- Data in lower environments

- Trade off – encryption and masking can cause potential performance issues

GOAL OF THIS PRESENTATION

- Replicated data
- Restricted access
- Data in lower environments

- Trade off – encryption and masking can cause potential performance issues

SCENARIOS FOR DATA MASKING

- Data being replicated to downstream applications
- Data being copied to non prod environments
- Data being accessed over internet –public data feeds

MASKING METHODS AVAILABLE

Data masking can be performed by Golden Gate during replication

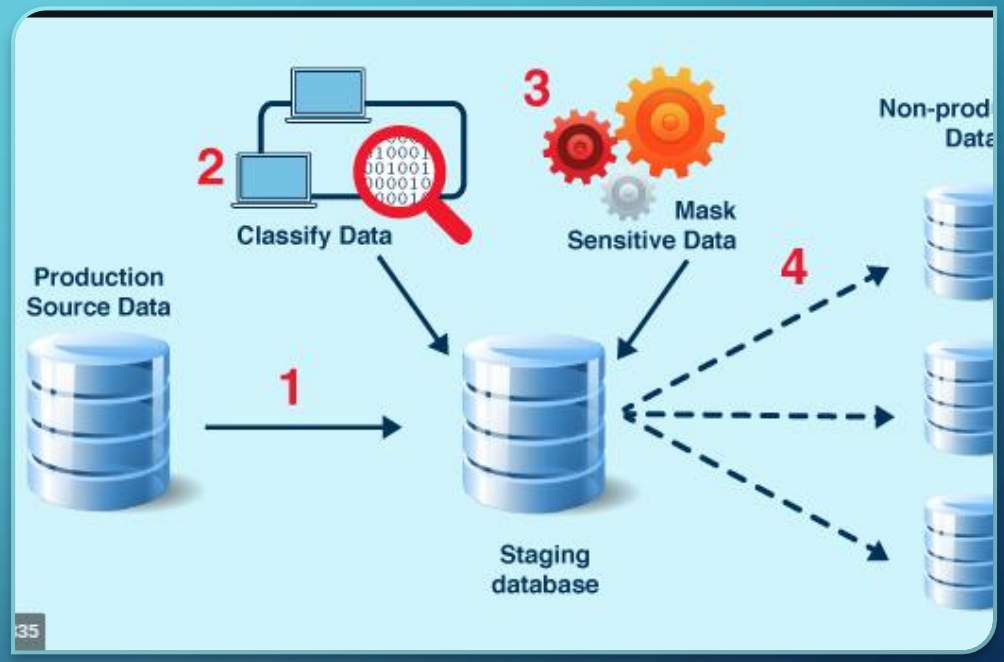
Materialized Views

Scripts

GOLDEN GATE DATA FILTERS

- Golden gate presents a variety of data filters and we can leverage them to ensure data masking
- We can filter in flight using
  - OGG Col.Conversion Functions

    Case/Eval/If



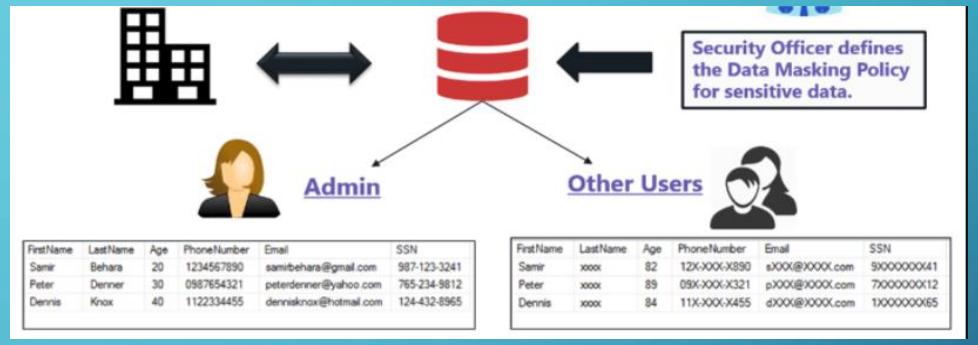
GOLDEN GATE MASKING AIDS --@CASE

- @Case function to select a value depending on a series of value tests. There is no limit to the number of cases you can test with the function.

- Examples:

Credit card number & SSN replaced with dummy value

- @CASE (SSN, ‘123445432', ‘111111111’)
- CC_NUMBER = @CASE (CC_TYPE,
  'AMEX', ‘3711112463111115',
  'Discover', '6011000111111119',
  'MasterCard', ‘5100005105101111',
  'Visa', ‘4033332222220002’)



VALLI RAMANATHAN

GOLDEN GATE MASKING AIDS --@EVAL

- @Eval : Use this function to select a value based on a series of independent tests. There is no limit to the number of conditions you can test. If the number of cases is large, list the most frequently encountered conditions first for best performance. This functions helps evaluate the column value and replace them with specific values so replicated instance do not have the original values.

- Flight_type= @EVAL (Flight=‘Fighter234’, ‘BlackHawk’, Flight=‘Fighter235’, ‘BlackEagle’)

- Insured_value= @EVAL (Value> 750000, ‘Cat1’, Value>10000000 =‘Cat5’)

GOLDEN GATE MASKING AIDS --@IF

- @IF : function to return one of two values, based on a condition. You can use the @IF function with other Oracle GoldenGate functions to begin a conditional argument that tests for one or more exception conditions. You can direct processing based on the results of the test. You can nest @IF statements, if needed.
- Example:

The following returns WEST if the STATE column is CA, AZ or NV; otherwise it returns EAST.

REGION = @IF (@VALONEOF (STATE, 'CA', 'AZ', 'NV'), 'WEST', 'EAST’)

Confidential masking with @IF

Security_Clearance = @IF (Level< 2, Level, Z)

GOLDEN GATE MASKING AIDS --@STRFIND

@STRFIND function to determine the position of a string within a string column or else return zero if the string is not found

- **Direct_Deposit= @IF (@STRFIND (Direct_deposit,'Wire ', 0) > 0,**
  @STRCAT ('Wire Code: 000000000 Account Number: 000000', ' ',
  @STREXT (Direct_Deposit, @STRFIND (Direct_Deposit, 'Wire Code', 0),
  @STRLEN(Direct_Deposit))),
  Direct_Deposit)

GOLDEN GATE MASKING AIDS --@SQLEXEC

When complicated aggregation/masking needs to happen – using a stored procedure to perform the task and applying its results to the filter is a very efficient way to employ

1. Create a Stored Procedure to execute the logic required to mask the data eg: new client information, Company profits
2. Use SQLEXEC to run the stored procedure via golden gate
3. Table mapping will run the procedure, obtain the results (masked/transformed data) and then use that in the replication

Contents of MAP statement:

```
MAP company.client, TARGET company.newacct, &
 SQLEXEC (SPNAME Asset, PARAMS (client_params = account_code)), &
  COLMAP (client_id = acct_id, newacct_val = Asset.desc_param);
```

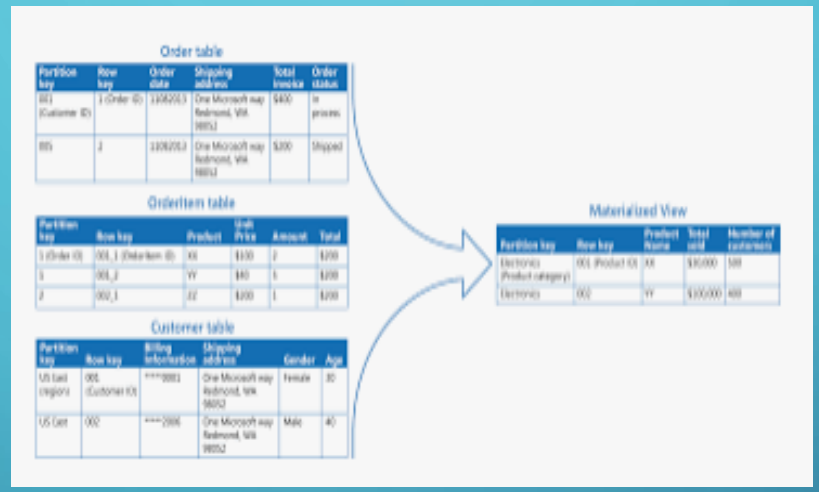
SQLEXEC executes the Asset stored procedure. Within the SQLEXEC clause, the PARAMS (client_params = account_code) statement identifies client_param as the procedure parameter to accept input from the account_code column in the account table.

Replicat executes the Asset stored procedure prior to executing the column map, so that the COLMAP clause can extract and map the results to the newacct_val colu

VALLI RAMANATHAN

MATERIALIZED VIEWS TO RESTRICT ACCESS

- When we have folks accessing Production data , we can restrict access to PII & SPI information using Materialized Views

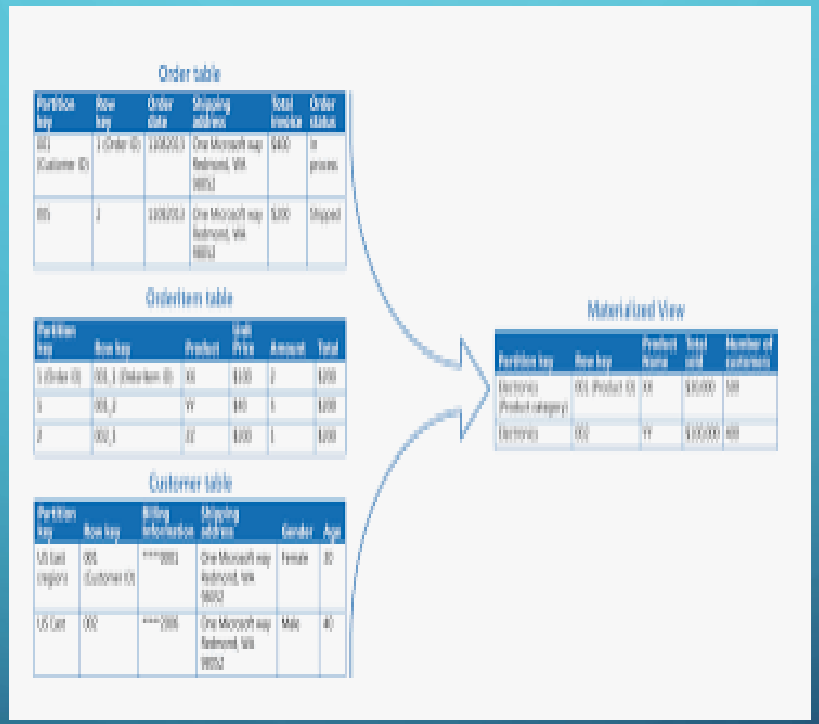


- Multiple tables with sensitive information
- Create Materialized Views that have the subset of data
- Date is refreshed and users access with no issues

VALLI RAMANATHAN

MATERIALIZED VIEWS – REPLICATION

- When replicating to downstream databases we can use just the materialized views and not the tables to secure PII info



How to configure ?
Will updates propagate ?
Truncate function?

SCRIPTS

- Scripts can be used for masking data when moving prod data to lower environments.
- Data cleanse & Data Transformation via bulk updates
- Reusability
- Transparency
- Maintenance

CLOSING THOUGHTS

- There are several different ways to handle sensitive data
- Need clear rules to identify them
- Need clear rules on handling them
- May be tedious but doable