Oracle Data Masking in der Praxis

Frank Hilgendorf
Berenberg
Hamburg

Schlüsselworte

Datenbanken, Data Masking, Testdatenmanagement, Anonymisierung, Enterprise Manager, Oracle Data Masking Pack, emcli

Einleitung

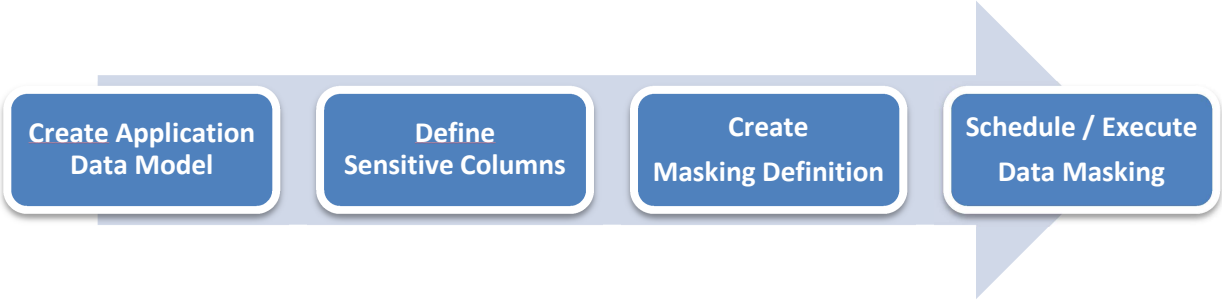
Das Anonymisieren von Daten in nicht-produktiven Umgebungen ist ein wichtiger Bestandteil des Testdatenmanagements. Es soll den unautorisierten Zugriff auf reale Daten verhindern und die Erfüllung von externen und internen Datenschutzrichtlinien sowie die Vermeidung von wirtschaftlichen Schäden und möglichen Reputationsschäden gewährleisten. Um diese Anforderungen möglichst flexibel und mit geringem Zeit- und Administrationsaufwand zu erfüllen, kommt bei Berenberg das Tool Oracle Data Masking zum Einsatz. Dieser Vortrag gibt eine Einführung in Data Masking und beschreibt den praktischen Einsatz bei der Berenberg Bank.

Oracle Data Masking

Das “Oracle Data Masking Pack for Oracle and non-Oracle Databases” ist im Bundle mit dem Oracle Enterprise Manager erhältlich. Mit dem Einsatz dieses Packages stehen folgende Komponenten für das Testdatenmanagement zur Verfügung:

- Masking Templates für E-Business Suite, Fusion Applications
- Data Discovery and Modeling
- Data Subsetting
- Data Masking (Daten und Workloads)

Für das Maskieren von Testdatenbanken spielen Data Subsetting und das Maskieren von Workloads keine Rolle und werden hier nicht weiter betrachtet. Die verbleibenden Komponenten ergeben folgenden Workflow für einen Data Masking Prozess:



Application Data Model

Seit dem Release des Enterprise Manager 12c ist beim Data Masking der Einsatz eines Application Data Models (ADM) notwendig. Erzeugt wird das ADM über die GUI des Enterprise Managers.

Ein ADM überprüft die Schemata einer Datenbank um Beziehungen zwischen Tabellen und Spalten zu beschreiben. Es erschließt dabei automatisch Datenbeziehungen und Charakteristiken für sensible Spalten. Die Informationen werden in einem zentralen Repository abgespeichert. Oracle bietet für die E-Business Suite und für Fusion Applications vordefinierte ADMs an. Alle anderen ADMs werden als Custom Applications Suites erstellt. Ein DB-Schema wird dann jeweils als separate Application angelegt.

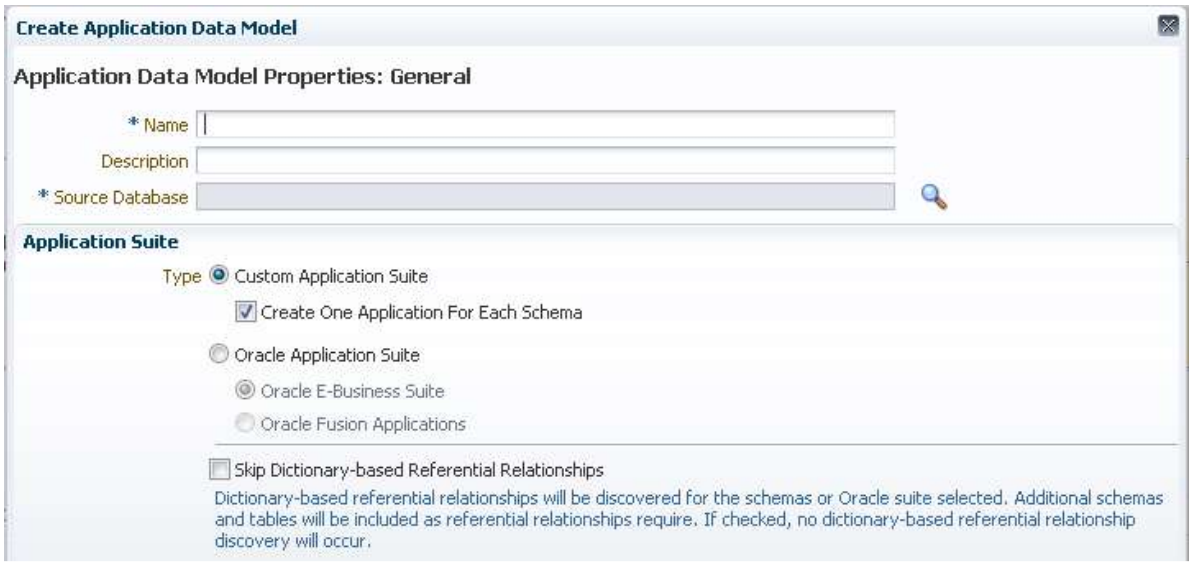


Abbildung 1: Erstellen eines ADM

Sensitive Columns

Damit Tabellenspalten beim Erstellen von Data Masking Definitions sichtbar sind, ist es zwingend notwendig, dass diese Spalten im ADM als „Sensitive Columns“ definiert sind. Die Definition erfolgt im ADM entweder über ein automatisches Discovery oder eine manuelle Definition.

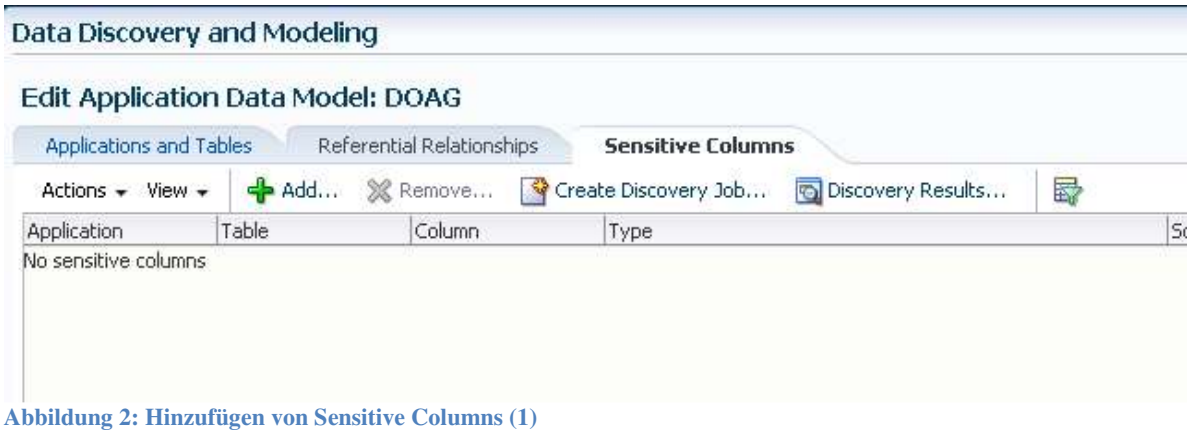


Abbildung 2: Hinzufügen von Sensitive Columns (1)

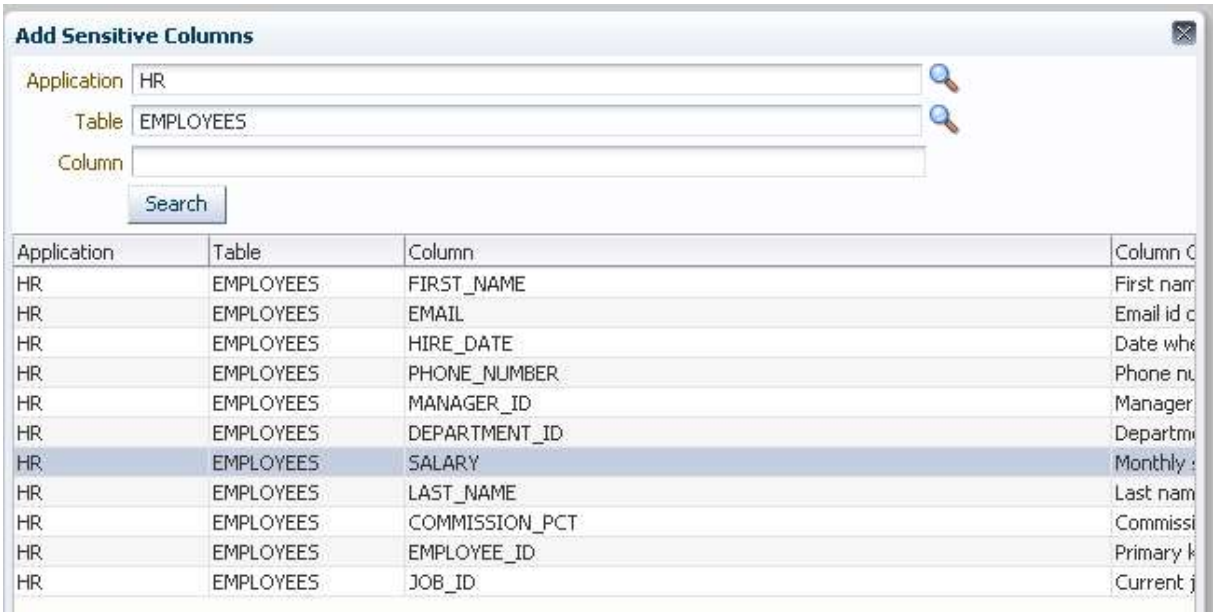


Abbildung 3: Hinzufügen von Sensitive Columns (2)

Data Masking Definition

Das Erzeugen einer Data Masking Definition beginnt mit der Auswahl eines ADM. Die im ADM enthaltenen Spalten und Tabellen stehen zum Maskieren zur Verfügung. Nach der Auswahl des ADM werden alle relevanten Tabellenspalten zur neu erstellten Masking Definition hinzugefügt. Die Referenzielle Integrität der Daten wird durch das automatische Hinzufügen von Foreign Key Columns gewährleistet. (Abbildung 4Abbildung 1)

Anschließend gilt es, für alle Spalten das Masking Format festzulegen. Grundsätzlich wird zwischen zwei Ansätzen beim Masking unterschieden. Das Ersetzen von Daten oder das Mischen (Shuffle) von vorhandenen Daten. Zur Auswahl stehen die in Abbildung 6 gelisteten Optionen. Ausschlaggebend sind hier das originale Format und der gewählte Maskierungsansatz. Neu in EM 12c ist ein Schlüssel-basiertes reversibles Masking (encrypt-decrypt). Das heißt, die ursprünglichen Werte können bei Zugriff auf den Schlüssel wieder hergestellt werden.

Beim Mischen (Shuffle) der Daten entsprechen die Eigenschaften der neuen Daten dem Original. Allerdings ist die Maskierung sehr schwach und kann oftmals zurückgerechnet werden. Anders sieht es beim Ersetzen der Originaldaten durch Random Daten aus. Die Maskierung ist sehr gut, aber die Daten weichen stark vom Original ab. Beispielhaft zu sehen am Sample eines Random Strings in Abbildung 7.

Das später erzeugte Masking Script kann mit verschieden Optionen an die jeweiligen Bedürfnisse hinsichtlich Performance und Security angepasst werden. (Abbildung 5)

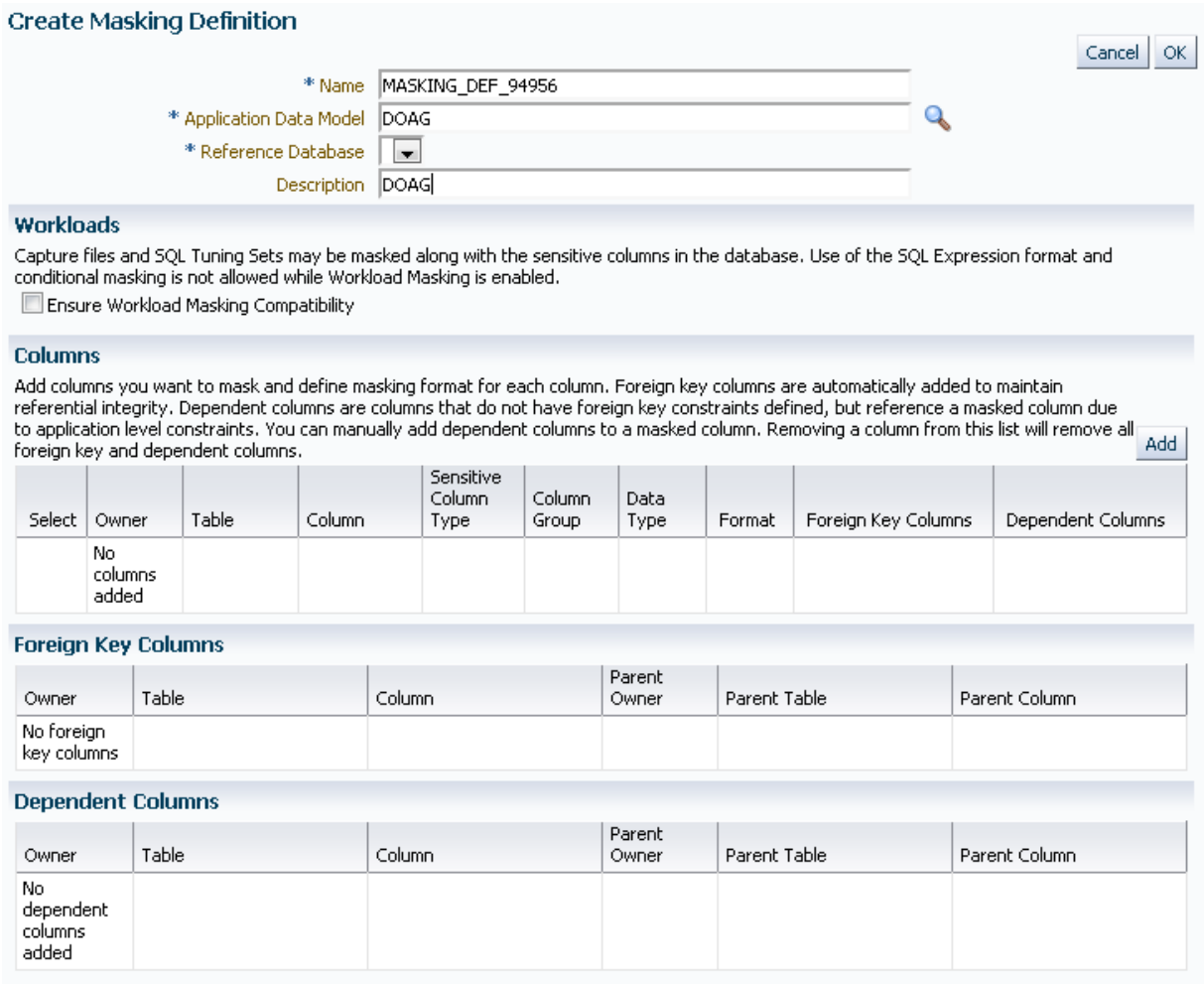


Abbildung 4: Erstellen einer Data Masking Definition

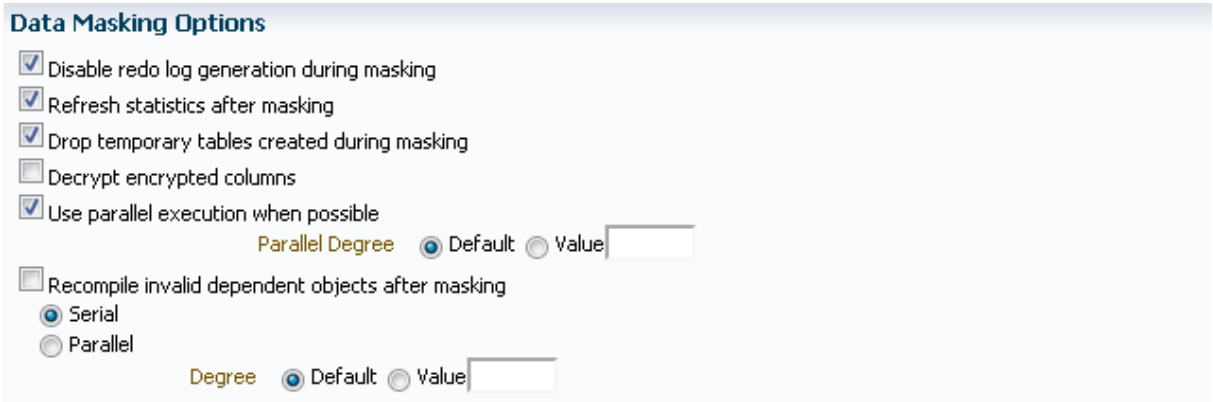


Abbildung 5: Data Masking Options

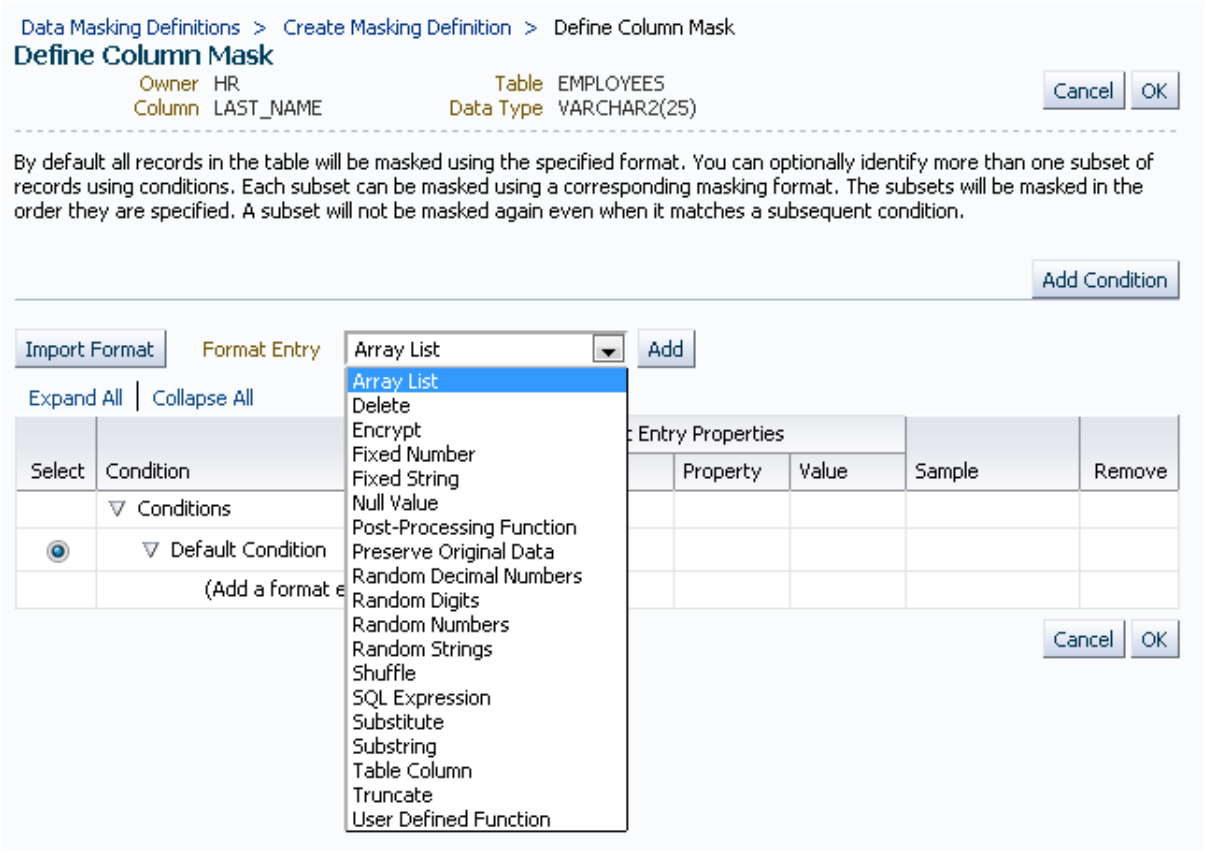


Abbildung 6: Format Entry Options

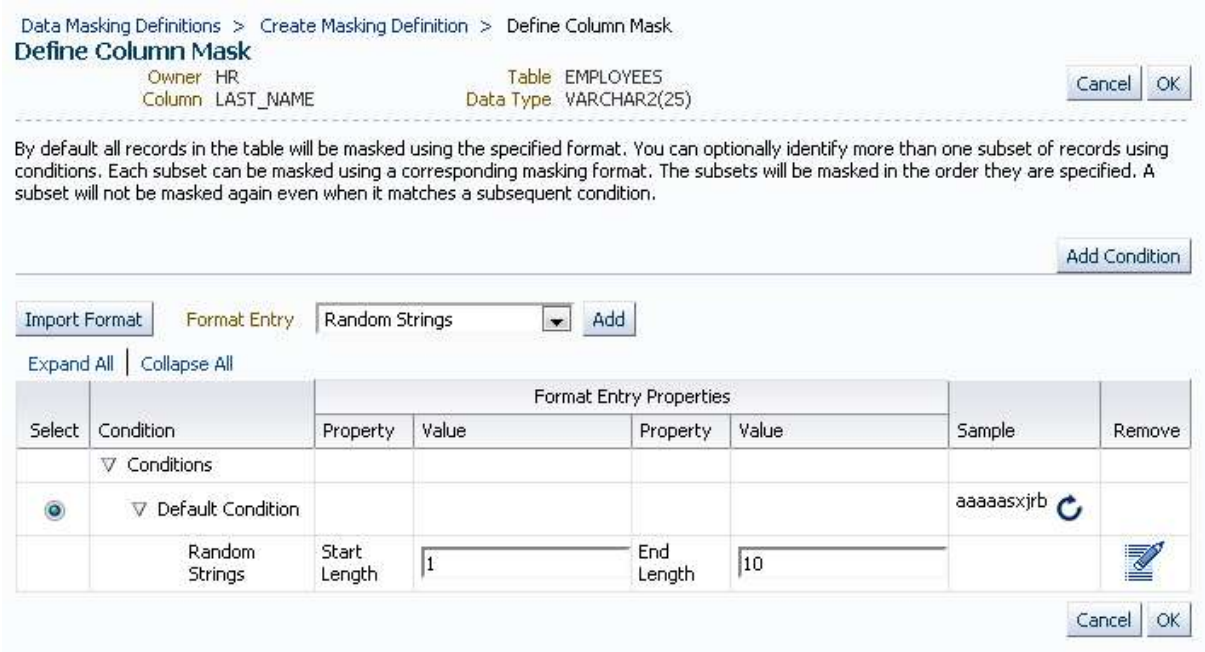


Abbildung 7: Column Masking mit Random Strings

Abschließend wird die Data Masking Definition gespeichert und das Masking Script kann per EM GUI oder emcli erzeugt werden. Die EM GUI bietet auch die Möglichkeit das Data Masking als Job einzuplanen. (Abbildung 8)

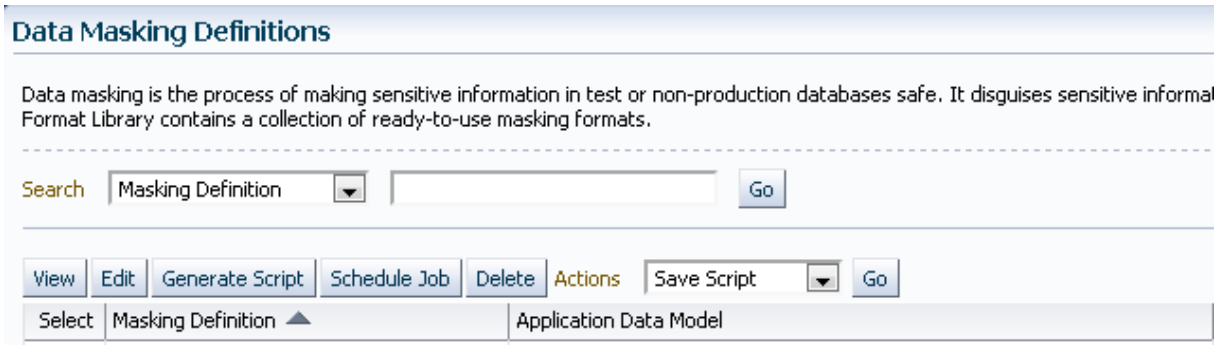


Abbildung 8: Skript Erstellung / Job Scheduling per EM

Data Masking in der Praxis

Es gibt diverse Möglichkeiten Data Masking Prozesse zu etablieren. Beispielhaft sind einige der folgenden initialen Fragen zu beantworten:

- Wer definiert was sensible Daten sind?
- Wer bewertet neu erzeugte Tabellen?
- Wo und wie wird das Masking Script generiert?

Ein daraus resultierender Workflow eines Data Masking Prozesses könnte wie folgt aussehen:

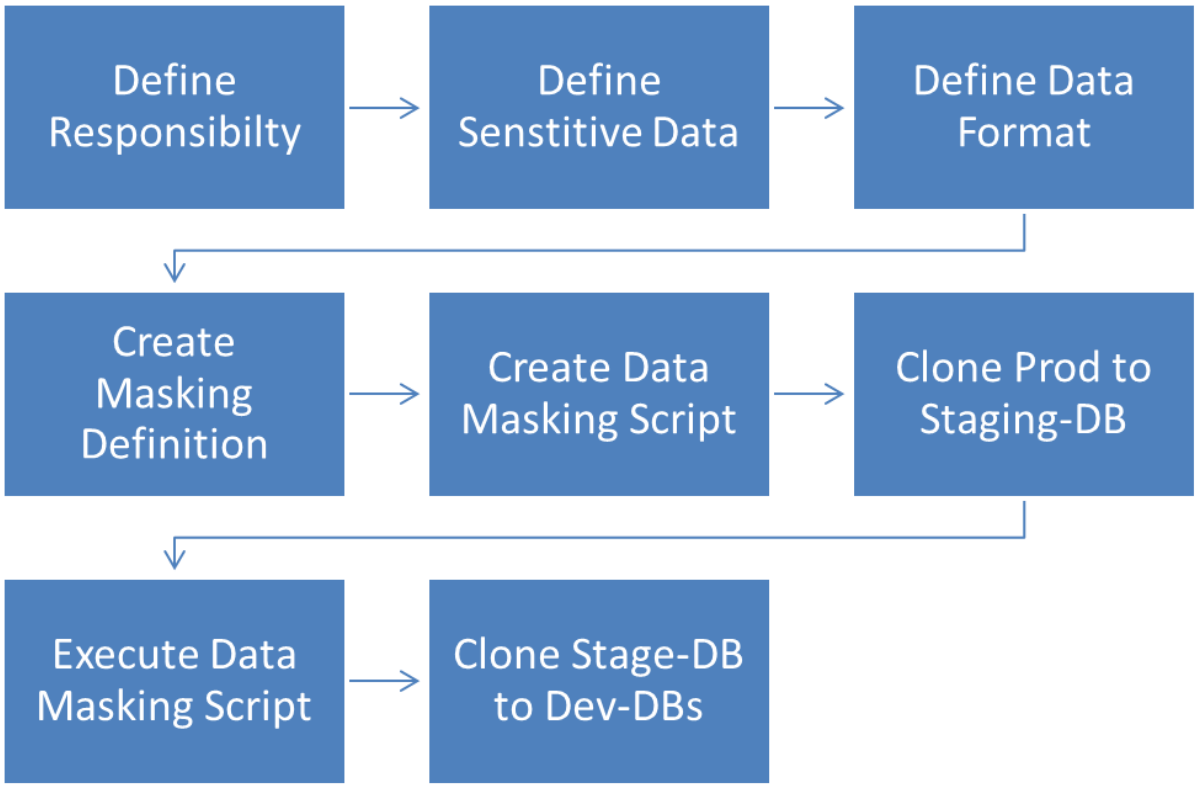


Abbildung 9: Data Masking Workflow

Wie oft in der Praxis ein Data Masking Prozess angepasst werden muss, hängt maßgeblich vom Erzeugen neuer Objekte mit sensiblen Daten ab. Dann muss die Data Masking Definition um die neuen Spalten ergänzt werden. Bei Berenberg hat sich gezeigt, dass es selbst in hoch dynamischen Datenbank-Umgebungen nur sehr selten zu Änderungen an relevanten Objekten kommt.

Das bedeutet, dass der Teil eines Data Masking Prozesses der das Editieren/Anpassen der Masking Definition beschreibt, nur sporadisch ausgeführt wird. Somit kann ein Data Masking Prozess nach dem initialen Aufsetzen fast komplett ohne manuellen Eingriff ablaufen. Der bei Berenberg etablierte Prozess sieht wie folgt aus:

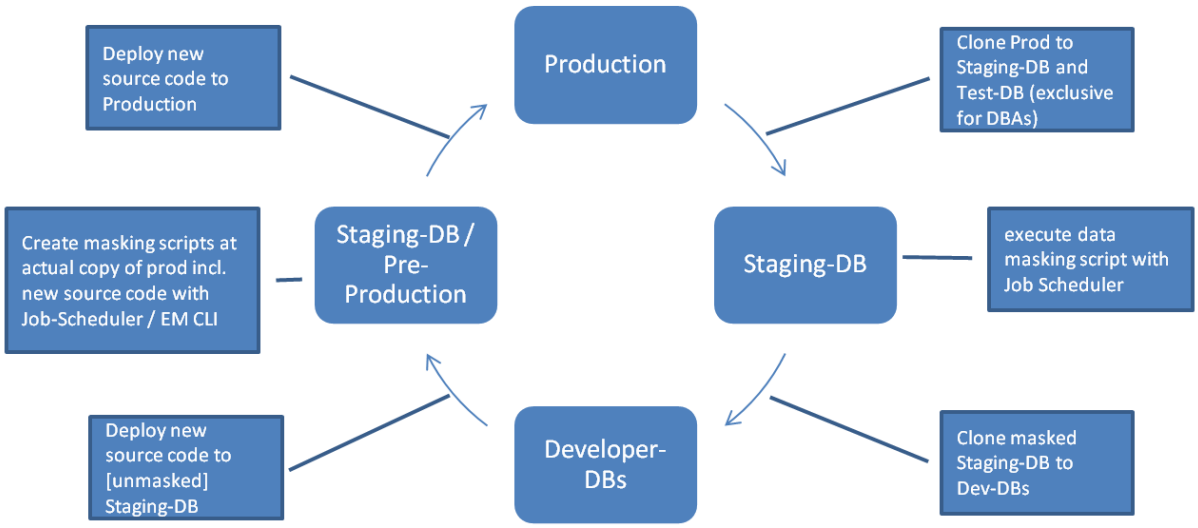


Abbildung 10: Data Masking Prozess Berenberg

Manuelle Eingriffe durch DBAs erfolgen nur dann, wenn es neue Tabellen mit sensiblen Daten gibt oder in Masking Definitionen enthaltene Tabellen oder Spalten gelöscht werden.

Damit neue Tabellen mit sensiblen Daten auch als solche erkannt werden, ist es notwendig alle daran beteiligten Personen für das Thema zu sensibilisieren. Es ist Aufgabe der Entwickler und SW-Architekten neue Tabellen auf DM-Relevanz zu bewerten und an die DBAs zu kommunizieren. In der Praxis hat sich das Vier-Augen Prinzip aus Entwickler und SW-Architekten bewährt. Die interne Revision und DBAs führen zusätzlich Stichproben durch. Das automatische Discovery von sensiblen Spalten hilft hier nur bedingt, zum Beispiel würde eine neue Spalte „Konto.Inhaber“ nicht als sensitive Column erkannt werden. Das ADM muss dann manuell um diese Tabellenspalte ergänzt werden.

Die wenigen Schwächen des Tools wie zum Beispiel die fehlende Fehlertoleranz des erzeugten Masking Scripts oder die Beschränkung des automatischen Discovery auf größtenteils amerikanische Datenformate wie zum Beispiel „National Insurance Number“ lassen sich mit wenig Aufwand korrigieren oder können vernachlässigt werden.

Fazit

Der Einsatz von Oracle Data Masking für das Testdatenmanagements bei Berenberg hat sich bewährt. Eine komplette Prozesskette benötigt aber noch weitere Richtlinien und die Definition von Zuständigkeiten. Die Vorteile von Data Masking zeigen sich nach dem Aufsetzen des Prozesses. Manuelle Eingriffe sind danach auf ein Minimum beschränkt und auch neue Mitarbeiter können nach kurzer Zeit selbständig mit dem Tool arbeiten. Ein selbstprogrammierter Prozess könnte das gleiche Ergebnis erzielen, aber der Aufwand für Entwicklung und Wartung wäre um ein Vielfaches höher.

Kontaktadresse:

Frank Hilgendorf

Org/IT

BERENBERG

Joh. Berenberg, Gossler & Co. KG

Neuer Jungfernstieg 20

20354 Hamburg

Telefon +49 40 350 60-192

Telefax +49 40 350 60-398

E-Mail frank.hilgendorf@berenberg.de

www.berenberg.de

Sitz: Hamburg - Amtsgericht Hamburg HRA 42659