Testdatenanonymisierung mit dem Data Masking Pack

Bernhard Eichhorn
MIC Management Consulting GmbH
München

Schlüsselworte:
Testdatengenerierung, Datenanonymisierung, Oracle Enterprise Manager, Data Masking Pack

Einleitung
Für die Entwicklung und Tests großer Softwaresysteme ist es von erheblichem Vorteil, wenn in den Entwicklungs- und Testumgebungen produktionsnahe Testdaten zur Verfügung stehen. Nach den vielen Skandalen der letzten Jahre im Bereich Datenklau kann es sich aber heute keine Firma mehr leisten, produktive Daten in seinen Testsystemen unverändert einzuspielen oder an ihre Softwarehersteller weiterzugeben.

Seit einigen Jahren bieten mehrere Software-Firmen Tools zur Anonymisierung von Daten an so auch Oracle mit seinem Data Masking Pack.

Als grundlegenden Anforderungen bei der Datenanonymisierung sind zu nennen:
1. Bereitstellung von realistischen aber nicht realen Daten, z.B. echte Telefonnummern, gültige Kreditkartennummern, inkl. Prüfziffer und Checksumme, echte Namen statt zufälliger Zeichenketten, etc.
2. Unumkehrbarkeit, d.h. unter keinen Umständen darf es auf der Zielumgebung möglich sein, auf die originalen Daten zurück zu schließen. Ein wesentliches Kriterium dafür ist die:
3. Vollständigkeit, d.h. die Identifikation und Maskierung redundanter und zusammengesetzter Spalten.
4. Erhalt der Datenqualität
    - Maskierung abgeleiteter Spalten, z.B. Sozialversicherungsnummern bzw. Matchcode-Spalten
    - Erhalt der referentiellen Integrität, d.h. Beziehungen zwischen den Datenbanktabellen müssen auch nach der Maskierung identisch aufgelöst werden können.
5. Erhalt der Datenverteilung, d.h. durch die Datenanonymisierung darf sich das Verhalten des Optimizers bei der Abarbeitung der SQL-Befehle nicht ändern.

Das Data Masking Pack ist eine lizenzpflichtige Option für den Oracle Enterprise Manager und bietet:
1. Graphische Benutzeroberfläche.
2. Vordefinierte Maskierungsformate, z.B. Kreditkartennummern.
3. Vordefinierte Maskierungsalgorithmen, z.B. Mischen, Löschen, Leeren, etc.
4. Berücksichtigung redundanter Spalten
5. Generierung ablauffähiger Anonymisierungsskripte.

Darüber hinaus ist eine mehr als brauchbare bildschirmspezifische Online-Hilfe verfügbar.

Maskierungsstrategien

Im Data Masking Pack werden u.a. folgende Maskierungsstrategien standardmäßig angeboten:

1. Nullwert:
   Die Maskierungsstrategie "Nullwert" setzt alle Werte in der damit zu anonymisierenden Spalte auf NULL. Sie ist damit nur für optionale Spalten möglich.
2. Löschen:
   Die Maskierungsstrategie "Löschen" löscht alle Sätze aus einer Tabelle, die eine bestimmte Bedingung erfüllen. "Löschen" ohne Bedingen hinterlässt eine leere Tabelle.
3. Leeren:
   Die Maskierungsstrategie "Leeren " löscht alle Sätze aus einer Tabelle.
4. Festwerte:
   Festwerte können für numerische, alphanumerische und Datumspalten individuell vorgegeben werden, wobei bei der Maskierung nur die Datensätze einer Tabelle betroffen sind, deren Spaltenwert ungleich NULL ist.
   Feste Zahlen, feste Zeichenketten und feste Datumswerte sollten aber nur für Tabellenspalten benutzt werden, die nicht indiziert sind. Da nach der Anonymisierung alle Sätze den gleichen Spaltenwert haben, wird auf die Tabelle nicht mehr über den Spaltenindex sondern über einen "FULL TABLE SCAN" zugegriffen, was sich entsprechend negativ auf die Performance auswirkt.
5. SQL-Ausdruck
   Ein SQL-Ausdruck ist eine sehr elegante Methode, eine Spalte zu anonymisieren.
   Beispiel:
   Setze den Tagesanteil des Geburtsdatums (Spalte "pa00gebdat") auf "1", wenn das Geburtsdatum <> 0 ist:
   DECODE(%pa00gebdat%,0,0,TO_NUMBER(TO_CHAR(SUBSTR(%pa00gebdat%,1,6))||'01'))
6. Mischen:
   Mischen funktioniert nach dem Round-Robin-Prinzip, d.h. alle Vorkommen eines bestimmten Wertes in einer Tabellenspalte werden durchgängig durch einen anderen zufällig ausgewählten Wert aus dieser Tabellenspalte ersetzt.
   Beispiel Vorname:
   Hubert => Peter, Peter => Christian, Christian => Anita, Anita => Hubert
   Es werden nur Sätze maskiert, deren Spaltenwert ungleich NULL ist.
   Vorteile:
   - lesbare, da reelle Daten
   - Erhalt die Datenverteilung (wichtig bei indizierten Spalten)
7. Zufallszeichenkette:
   Es werden nur Sätze maskiert, deren Spaltenwert ungleich NULL ist.
   Vorteile:
   - Erhalt die Datenverteilung (wichtig bei indizierten Spalten)

   Nachteil:
   - unlesbare Daten

Die generierten Skripte
Die generierten Skripte können entweder direkt über den im Data Masking Pack integrierten Oracle Scheduler (Button "Job planen") ausgeführt werden oder auf das Dateisystem gespeichert werden und über ein Batch-Steuerungssystem, z.B. UC4 ausgeführt werden. Im 2. Fall ist es aber u.U. notwendig, den im Skript hinterlegten Pfad und Namen der Spool-Datei "spool /pkg/i25lifed/oracle/dbs/masking<lfd_nr>.log" manuell anzupassen, da das Batch-Steuerungssystem i.d.R. keine Schreibberechtigung auf das angegebene Directory hat. Im Data Masking Pack selbst besteht keine Möglichkeit, Pfad und Namen der Spool-Datei zu bestimmen.

Um die generierten Skripte fehlerfrei ausführen zu können, braucht der DB-User die Zuweisung der DBA-Rolle und Zugriff auf die System-Packages DBMS_RANDOM und DMBS_CRYPTO.

Nach Ausführung eines Skriptes bleiben im Schema des DB-Users folgende Datenbankobjekte persistent erhalten:

1. PROCEDURE mgmt$mask_sendMsg:
   Die Prozedur dient zur Ausgabe von Meldungen in die Log-Datei bei Ausführung des Anonymisierungsskriptes. Über die System-Package-Prozedur "DBMS_OUTPUT.PUT_LINE" werden die Meldungen in Portionen zu 255 Zeichen geschrieben, wodurch die auszugebenden Meldungen beliebig lang sein können.
2. PROCEDURE mgmt$mask_errorExit:
   Die Prozedur dient zur Ausgabe von Fehlermeldungen in die Log-Datei bei Ausführung des Anonymisierungsskriptes.
3. PROCEDURE mgmt$mask_errorExitOraError:
   Die Prozedur dient zur Ausgabe von Oracle-Fehlermeldungen in die Log-Datei bei Ausführung des Anonymisierungsskriptes.
4. PROCEDURE mgmt$mask_checkDBAPrivs:
   Die Prozedur prüft, ob der DB-User, unter dem das Anonymisierungsskript gestartet wird, DBA-Rechte hat, d.h. ob ihm die Rolle "DBA" zugewiesen wurde.
   Die Ausführung des Anonymisierungsskriptes wird merkwürdigerweise nicht abgebrochen, wenn festgestellt wurde, dass der ausführende DB-User nicht über die Rolle "DBA" verfügt. Stattdessen wird nur eine Warnung ausgegeben, dass "das Skript nicht erfolgreich laufen kann, wenn der DB-User keine DBA-Rechte hat".
5. PROCEDURE mgmt$mask_setUpJobTable:
   Die Prozedur erzeugt, falls nicht vorhanden, im Schema des ausführenden DB-Users die Tabelle MGMT$MASK_CHECKPOINT, die persistent erhalten bleibt und die für die Wiederholbarkeit der Anonymisierungsskripte von essentieller Bedeutung ist.
   Die Prozedur versucht die ID des Anonymisierungsskriptes zusammen mit Prozessschritt 1 in die Tabelle MGMT$MASK_CHECKPOINT einzutragen. Ist die ID des Anonymisierungsskriptes aber bereits in der Tabelle vorhanden, so bedeutet das, dass das Anonymisierungsskript bei der letztmaligen Ausführung mit Fehlern abgebrochen wurde. In diesem Fall liefert die Prozedur die Nummer des Prozessschrittes zurück, bei dem der Fehler aufgetreten ist und mit dem bei Wiederanlauf fortgefahren werden muss.
   Jede Prozedur des Anonymisierungsskriptes prüft als erstes nach ihrem Aufruf anhand der Tabelle MGMT$MASK_CHECKPOINT, ob sie überhaupt an der Reihe ist und trägt sich in diesem Fall in die Tabelle ein.
6. PROCEDURE mgmt$mask_deleteJobTableEntry:
   Die Prozedur löscht den Skripteintrag aus der Tabelle MGMT$MASK_CHECKPOINT, wenn das Skript erfolgreich gelaufen ist.

7. PROCEDURE mgmt$mask_setStep:
   Die Prozedur speichert die Nummer des aktuellen Prozessschrittes im Anonymisierungsskript in der Tabelle MGMT$MASK_CHECKPOINT. So kann von außen verfolgt werden, welcher Prozessschritt des Anonymisierungsskriptes gerade ausgeführt wird.
8. Package mgmt$mask_util:
   Package mit Hilfsprozeduren, z.B. Zufallszahlengenerierung, die für die verschiedenen Anonymisierungsstrategien gebraucht werden.
9. Tabelle mgmt$mask_checkpoint:
   Spalten:
   - SCRIPT_ID: Nummer des Anonymisierungsskriptes (wird bei der Generierung vergeben)
   - LAST_STEP: Nummer des Prozessschrittes, bzw. der Prozedur innerhalb des Anonymisierungsskriptes, die aktuell ausgeführt wird, bzw. in der ein Fehler aufgetreten ist und abgebrochen wurde.

Grundsätzlicher Ablauf der Datenmaskierung

Bei der Anonymisierung einer Tabelle werden grundsätzlich folgende Schritte durchlaufen:

1. Löschen aller Trigger, die auf dieser Tabelle definiert sind (pro Trigger 1 statische Prozedur).
2. Löschen aller Foreign-Key-Constraints, die sich auf diese Tabelle beziehen (generische Prozedur).
3. Entzug aller Zugriffsberechtigungen ("REVOKE <privilege> ON <table_name> FROM…"), die auf diese Tabelle vergeben sind (pro Privileg und Empfänger der Zugriffsberechtigung 1 statische Prozedur).
4. Löschen aller Constraints, die auf dieser Tabelle definiert sind, ohne die Foreign-Key-Constraints von dieser Tabelle auf andere (generische Prozedur).
5. Löschen aller Indizes, die auf dieser Tabelle definiert sind (pro Index 1 statische Prozedur).
6. Umbenennung der Tabelle in <table_name>$DMASK
7. Erzeugung der anonymisierten Daten aus der umbenannten Originaltabelle ("CREATE TABLE <table_name> … AS SELECT … FROM <table_name>$DMASK WHERE…"
8. Restaurierung aller Default-Werte der Tabellenspalten (1 statische Prozedur für alle Default-Werte und Tabelle).
9. Restaurierung des Parallelisierungsgrades der Tabelle.
10. Löschen der umbenannten Originaltabelle ("DROP TABLE <table_name>$DMASK").
11. Restaurierung aller Indizes (pro Index 1 statische Prozedur).
12. Restaurierung des Primary-Key-Constraints.
13. Restaurierung aller Unique-Key-Constraints (pro Constraint 1 statische Prozedur).
14. Restaurierung aller Check-Constraints (pro Constraint 1 statische Prozedur).
15. Restaurierung aller NOT NULL-Spalten (pro Spalte 1 statische Prozedur).
16. Restaurierung aller Foreign-Key-Constraints (pro Constraint 1 statische Prozedur).
17. Restaurierung aller Zugriffsberechtigungen ("GRANT <privilege> ON <table_name> TO…"), die auf diese Tabelle vergeben waren (pro Privileg und Empfänger der Zugriffsberechtigung 1 statische Prozedur).
18. Restaurierung aller Foreign-Key-Constraints, die sich auf den Primary-Key dieser Tabelle beziehen (pro Constraint 1 statische Prozedur).
19. Restaurierung aller Trigger (pro Trigger 1 statische Prozedur).
20. Neu-Compilierung aller Views, Packages, Package Bodies, etc., die sich auf diese Tabelle beziehen (pro Datenbankobjekt 1 statische Prozedur).
21. Restaurierung aller Spaltenkommentare (pro Kommentar 1 statische Prozedur).
22. Neuberechnung der Tabellenstatistiken.
23. Clean Up der erzeugten PL/SQLProzeduren.

Wartung/Neugenerierung der Anonymisierungsskripte

Da die generierten Anonymisierungsskripte zu 95% aus statischen Prozeduren bestehen, können sie nur als Snapshot der Tabellenkonstellation zum Zeitpunkt der Generierung angesehen werden.
Schon geringste Abweichungen in der Tabellenkonstellation zum Zeitpunkt der Generierung von der zum Zeitpunkt der Ausführung machen eine Neu-Generierung der Skripte unbedingt erforderlich.

1. Änderungen an der Tabellenstruktur
    - Neue Spalten:
      keine Fehler bei Ausführung des Skriptes, aber nach Erzeugung der anonymisierten Daten aus der umbenannten Originaltabelle (Schritt 7) existieren die neuen Spalten in der anonymisierten Tabelle nicht mehr.
    - Geänderte Spalten:
        - Datentyp: Fehler im Skript bei Erzeugung der anonymisierten Daten aus der umbenannten Originaltabelle (Schritt 7).
        - Datenlänge: nicht notwendigerweise Fehler bei Ausführung des Skriptes, aber nach Erzeugung der anonymisierten Daten aus der umbenannten Originaltabelle (Schritt 7) existieren die Änderungen in der anonymisierten Tabelle nicht mehr.
        - Nachkommastellen: analog Datenlänge.
        - Nullable: keine Fehler bei Ausführung des Skriptes, aber nach Erzeugung der anonymisierten Daten aus der umbenannten Originaltabelle (Schritt 7) existieren die Änderungen in der anonymisierten Tabelle nicht mehr.
        - Defaultwerte: analog Nullable
    - Gelöschte Spalten:
      Fehler im Skript bei Erzeugung der anonymisierten Daten aus der umbenannten Originaltabelle (Schritt 7)
2. Änderungen am Index-Design
    - Neue Indizes:
      keine Fehler bei Ausführung des Skriptes aber nach Restaurierung aller Indizes (Schritt 11) sind die neue Indizes nicht mehr vorhanden.
    - Geänderte Indizes:
      keine Fehler bei Ausführung des Skriptes aber nach Restaurierung aller Indizes (Schritt 11) sind die Indizes wieder so angelegt wie zum Zeitpunkt der Skriptgenerierung definiert.
    - Gelöschte Indizes:
      Fehler im Skript bei Löschen aller Indizes (Schritt 5)

3. Neue, geänderte oder gelöschte Grants: analog zu Indizes
4. Neue, geänderte oder gelöschte Constraints: analog zu Indizes
5. Neue, geänderte oder gelöschte Trigger: analog zu Indizes
6. Views:
    - Neue Views:
      invalid, da durch Anonymisierungsskript nicht neu-compiliert.
    - Gelöschte Views:
      Fehler im Skript bei Schritt 20
7. Packages: analog zu Views

Wiederholbarkeit der Anonymisierungsskripte
Bei Abbruch eines Anonymisierungsskriptes im Fehlerfall ist in der Tabelle MGMT$MASK_CHECKPOINT die Script-ID und der Prozessschritt (Spalte LAST_STEP) gespeichert, in dem der Fehler aufgetreten ist.
Bei Neustart des Skriptes werden alle Prozessschritte übersprungen, deren Prozessschrittnummer STEP_NUM <> LAST_STEP ist und der Prozess mit Schritt STEP_NUM = LAST_STEP fortgeführt.
Tritt bei der Ausführung eines Prozessschrittes ein Fehler auf, so wird die skriptglobale Variable STEP_NUM = -1 gesetzt, so dass alle nachfolgenden Prozessschritte nicht mehr ausgeführt werden.

Don'ts
1. Data Masking Pack => Maskierungsdefinitionen => erweiterte Optionen:
   Option "Während der Maskierung erstellte temporäre Tabellen löschen": niemals aktivieren:
   Wenn die temporär erstellten Tabellen gelöscht werden, ist die Wiederholbarkeit des Skriptes gefährdet. Die temporär erstellten Tabellen werden bei der Erzeugung der anonymisierten Tabelle ("CREATE TABLE <tablename>…") aus der umbenannten Originaltabelle ("… FROM <tablename>$DMASK …") benötigt. Wenn bei diesem Schritt ein Fehler auftritt, ist der CREATE-TABLE-Befehl nicht mehr ausführbar, wenn die temporär erstellten Tabellen am Ende des Anonymisierungsskriptes gelöscht worden sind, da bei Neuanlauf die Erzeugung der temporären Tabellen übersprungen wird.
2. Niemals mehr als eine Spalte pro Tabelle durch bedingtes Mischen anonymisieren:
   Beim bedingten Mischen werden temporär erstellte Tabellen verwendet, die bei der Erzeugung der anonymisierten Daten zu einem „Merge Join Cartesian" führen.
   Abgesehen von horrenden Laufzeiten sprengt ein kartesisches Produkt über 2 große Tabellen oftmals den zur Verfügung stehenden temporären Tablespace.
3. Niemals die Tabelle MGMT$MASK_CHECKPOINT inhaltlich verändern oder löschen:
   Alle Einträge in der Tabelle gehören zu Anonymisierungsskripten, die nach einem Fehler abgebrochen wurden. Diese Einträge dienen als Aufsetzpunkte für eine Wiederholung der Abarbeitung der Skripte.

Bekannte Fehler
1. Die Generierung der Skripte funktioniert nicht auf Datenbankschemen, die einen "." im Namen haben.
2. PROCEDURE mgmt$mask_checkDBAPrivs: Exception-Handling funktioniert nicht, weil die Exception "NO_DATA_FOUND" nicht explizit behandelt wird.
3. Die Reihenfolge bei der Neu-Compilierung in den Anonymisierungsskripten ist zufällig und berücksichtigt nicht die Abhängigkeiten der Datenbankobjekte untereinander. Es kommt oftmals vor, dass der Package Body vor der Package Specification neu compiliert wird. Dabei kann es zum Fehler "ORA-24344: success with compilation error" kommen. Das Problem ist in My Oracle Support bekannt (Bug 10243572) und der angegebene Workaround funktioniert.

Fazit:
Vorteile des Data Masking Packs:

1. Eingängige und komfortable graphische Oberfläche:
   Von der Installation bis zum 1. generierten Maskierungsskript dauert es ca. 1 Stunde.
2. Überdurchschnittlich gute Online-Hilfe
3. Umfangreiche Maskierungsbibliotheken
4. Zahlreiche mitgelieferte Maskierungsstrategien
5. Möglichkeit zur Definition virtueller Foreign-Keys
6. Berücksichtigung abhängiger Spalten.
7. Einfache Einbindung selbstgeschriebener Maskierungsstrategien möglich.
8. Einfache Einbindung von Pre- und Postprocessing-Skripten.

Nachteile des Data Masking Packs:

1. Generierte Skripten sind viel zu statisch. Generische Prozeduren zum Wiederaufbau der Indizes, Constraints, etc. sind unumgänglich.
   a. Die Notwendigkeit der Neugenerierung ist eigentlich vor jeder Ausführung des Skriptes erforderlich.
   b. Eine Übertragbarkeit der Anonymisierungsskripte auf andere DB-Umgebungen scheitert schon an Unterschieden bei den Grants.
2. Da absolut unformatiert sind die generierten Skripte ohne manuelle Überarbeitung sehr schwer lesbar.
3. Der Anonymisierungsprozess selbst ist durch die RENAME/CREATE-Strategie m.E. umständlich und alleine schon wegen der Restaurierung aller Indizes zu aufwendig (=> Performance).
4. Keine API zum Repository
5. Geringe Möglichkeiten der Parametrisierbarkeit von außen, z.B. Name und Pfad der Spool-Datei.
6. Keine Möglichkeit zusammengesetzte Felder als abhängige Felder zu definieren
7. Parallelentwicklung der Maskierungsdefinitionen nur unter einem zentralen OEM-Account möglich, da die Maskierungsdefinitionen eines Administrators nicht von einem anderen Administratoren-Account eingesehen werden können.

Kontaktadresse:

Bernhard Eichhorn
MIC Management Consulting GmbH
Kronacher Straße 4
D-81549 München

Telefon: +49 (89) 680 711 61
Fax: +49 (89) 680 711 62
E-Mail bernhard.eichhorn@mic-muenchen.de
Internet: www.mic-muenchen.de