Datenanalysen und Tests mit maskierten Daten

Oliver Gehlert
metafinanz Informationssysteme GmbH
München

Schlüsselworte
Data Masking, Compliance, Analyse, Fraud, Mining, Subsetting, Testdatenbank, Entwicklung

Einleitung
Die Anforderung maskierte Daten zum Testen und Entwickeln zu verwenden, stößt bei Entwicklern und Testern selten auf Begeisterung. Das Maskieren von Daten sei aufwändig, nicht flexibel genug und mit den Ergebnissen könnte man nicht weiterarbeiten.
In diesem Vortrag werde ich zeigen, dass maskierte Daten nicht nur für Entwicklung und Test verwendbar sind, sondern auch für – gesetzeskonforme – Auswertungen geeignet sind.

Aber seit der Einführung des Bundesdatenschutzgesetzes unterliegt die Nutzung von Produktivdaten strengen Regeln, die auch die Weitergabe von Daten außerhalb der EU betreffen. Nicht zu übersehen sind schließlich besondere Publikationspflichten bei Datenschutzverstößen.

Wachsenden Datenschutzherausforderungen begegnen Unternehmen der Finanz- und Versicherungswirtschaft auch noch auf einem anderen Gebiet: Für die Entwicklung neuer, innovativer Produkte sind sie zunehmend auf Data-Mining angewiesen, um aus den Kunden- und Geschäftsdaten der Data-Warehouses so auf viele wertvolle neue Zusammenhänge zu schließen. Dabei werden hier für viele Auswertungen gar nicht alle Daten benötigt, so dass insbesondere gerade personenbezogene Daten problemlos einzelner könnten anonymisiert werden können, ohne die Ergebnisse z.B. beispielsweise einer Warenkorbanalyse zu beeinträchtigen. Wie also lassen sich maximaler Schutz der Daten und optimale Nutzung der Datenschätze unter einem Hut vereinen? Die Antwort der Softwareindustrie lautet Data-Masking.
Data Masking nutzt Verfahren der realitätsnahen Datenanonymisierung, um geschäftliche Daten so zu verändern, dass sie sich weiterhin für aussagekräftige Tests und Auswertungen eignen. Dabei werden alle Bezüge zu realen Personen oder Geschäftsentitäten verschleiert, so dass keinerlei Datenschutzrechtliches Risiko mehr besteht. Richtig eingesetzt versetzt Data-Masking ein Unternehmen in die Lage, jederzeit Kunden- und Geschäftsdaten für Tests und Analysen weiterzugeben.
Zwei gängige Produkte kommen beim Data-Masking häufig zum Einsatz: Net 2000 und Oracle Data Masking. Das Grundprinzip dieser Werkzeuge ist recht einfach erklärt: Soll eine Datenbank für Testing oder Data-Mining zur Verfügung gestellt werden, anonymisiert ein solches Tool die Datenfelder nach zuvor festgelegten Methoden derart, dass die Inhalte die gleichen Eigenschaften wie die Produktivdaten aufweisen. Selbstverständlich ließen sich solche Aufgaben auch ohne externe Produkte erledigen, doch ersparen Data-Masking-Programme mühselige Eigenentwicklung und Handarbeit.
In diesem Vortrag werde ich die Funktionalitäten des Enterprise Managers 12c im Bereich Data Masking vorstellen. Neben den eigentlichen Maskierungsfunktionen stelle ich noch die Funktionen zur Erstellung von konsistenten Teildatenmengen für Testdatenbanken vor.

Anforderungen an die maskierten Daten

Zunächst ist bei einem Data-Masking-Projekt einmal zu klären, welche Felder maskiert werden müssen. Prinzipiell könnten alle Angaben maskiert werden, doch stünde der enorme Aufwand dafür in keinem Verhältnis zum Nutzen. Als grober Leitfaden gilt deshalb: So wenig wie möglich, aber so viel nötig. Die aus Data-Masking-Projekten gewonnene Erfahrung besagt, dass in enger Zusammenarbeit mit dem Kunden das richtige Maß gefunden werden muss.

Beim Maskieren von Feldern finden unterschiedliche Methoden Verwendung. Wenig hilfreich wäre ein einfaches Austauschen von die n Zeichenketten durch Zufallswerte, da solche Daten weder lesbar noch realistisch sind und Tests auf Plausibilität oder Format beeinträchtigen würden. Ein Beispiel dafür sind Kreditkartennummern: Checkt die Anwendung eine Kreditkartennummer auf Gültigkeit, so muss bei der Maskierung eine plausible Zahlenkombination eingesetzt werden. Manche Anbieter liefern hierzu eine eigene Maskierungsdatenbank, die plausible, aber künstliche Kreditkartennummern, Adressen oder Namen zum Austauschen bereitstellt.

Eine andere Möglichkeit der Verschleierung ist das Durchmischen von Feldinhalten, das sogenannte Shuffeln. Auch so können jegliche Bezüge auf reale Personen und ihre Daten verwischt werden, jedoch ist hierbei eine gewisse Mindestanzahl an Datensätzen erforderlich, um eine Rekonstruktion zu verhindern. Oft enthalten Datenbanken auch mehrere Felder, die zueinander in Bezug stehen. Die dabei zugrundeliegende Logik muss beim Erstellen der Maskierungsregeln berücksichtigt werden, damit beim Testen der Anwendung keine Fehler auftauchen.

Die wirksamste Verschleierungsmethode ist natürlich das Löschen von Feldinhalten. Doch auch ist ein planvolles Vorgehen erforderlich. Zunächst muss dabei definiert werden, welche Inhalte beim Testen oder Analysieren verzichtbar sind.

Wie wichtig eine gute Kenntnis bzw. Analyse über die zu maskierenden Daten ist, zeigt sich u.a. beispielsweise bei Freitextfeldern, die potenzielle Stolperfallen bergen. So könnten Bestellformulare oder Banküberweisungen in solchen Feldern vertrauliche Informationen enthalten, die möglicherweise bei der Maskierungskonfiguration übersehen werden.
Um von den Entwicklern und Anwendern akzeptiert zu werden, müssen bei der Definition der Maskierungsregeln und –umfänge die Ziele geklärt werden. Wofür sollen die maskierten Daten verwendet werden, wer kann darauf zugreifen? Haben diese Personen direkten Zugriff auf die echten Produktivdaten? Welche Relationen aus den vorhandenen Daten müssen erhalten bleiben?

Auswertungen

Datenanalysen

Möchte man Auswertungen über größere Datenmengen durchführen, bei denen nur gewisse Teilinformationen der einzelnen Datensätze benötigt werden, so kann man mit maskierten Daten gut durchführen. Für solche Analysen hat man dann mehrere Möglichkeiten, entweder man verwendet für die Analyse Daten, die weniger Spalten aufweisen oder man maskiert die entsprechenden Spalten.

In solchen Fällen kann eine Maskierung auch das Löschen von Feldinhalten sein.

Tabelle 1 Originaldaten

ID	Name	Vorname	Alter	VersNr	VersTyp	PLZ
4711	Mustermann	Hans	35	12325	HP	80804
4712	Musterfrau	Gabi	29	4566	KH	10303

Tabelle 2 Maskierung durch Ersetzung

ID	Name	Vorname	Alter	VersNr	VersTyp	PLZ
1235	Meier	Joseph	37	67534	HP	80803
67898	Huber	Johanna	25	333627	KH	10378

Oder

Tabelle 3 Maskierung durch Gruppierung

Alters-gruppe	VersTyp	PLZ-Bereich
30-39	HP	8
20-30	KH	1

Aus den Daten in Tabelle 3 lassen sich keine Rückschlüsse mehr auf einzelne Personen machen, aber für einzelne Analysen sind diese Daten ausreichend.

Die Daten in Tabelle 2 sind sinnvoll, wenn die Daten für sehr unterschiedliche Auswertungen, oder Auswertungen mit Einzelbezug verwendet werden sollen:

Beispiele:

- Durchschnittliche Anzahl Versicherungen je Kunde und PLZ Gebiet
- Korrelationen zwischen einzelnen Versicherungen
- Anzahl Versicherungen je Kunde pro Vertreter
- …

Für solche Auswertungen ist es wichtig, dass die Maskierung eines Kunden immer konsistent erfolgt, seine eigentliche Versicherungsnummer oder sein Name ist für die Auswertung nicht erforderlich.

Fraud Detection

Betrugserkennung ist im Versicherungsbereich aktuell ein großes Thema. Der Anteil an Betrugsfällen beträgt zwischen 5 und 15% aller Schadenfälle.

Bei den Betrugsfällen kann man zwei Kategorien unterscheiden, einmal den „spontanen“ Betrug und die organisierten Betrugsfälle.

Im ersten Fall handelt es sich meistens um kleinere Summen, bei denen der Versicherungsnehmer beschließt einen Schadenfall auszunutzen und den Umfang bzw. Aufwand erhöht. Hierunter fallen z. B. Reparaturen von Altschäden oder der Ersatz von bereits defekten Geräten.

Bei diesen Fällen ist eine automatisierte Erkennung wichtig, da eine manuelle Identifikation, in Relation zur Schadensumme, zu aufwendig ist. Andererseits darf der Algorithmus auch nicht zu „scharf“ eingestellt sein, so dass dieser zu viele echte Schadensfälle als Betrugsfall klassifiziert. Dies ist aus zwei Gründen ungünstig, einerseits einer negativen Außenwirkung und andererseits hohe Kosten für die Verfolgung.

Bei den organisierten Betrugsfällen handelt es sich in Summe, um weniger Einzelfälle, aber um hohe Summen je Fall. Diese Fälle sind schwieriger zu identifizieren, da hier ein großer Aufwand betrieben wird, den Betrug zu verdecken. In den meisten Fällen verfügen die Betrüger über ein umfangreiches Versicherungsknowhow und ein größeres Netzwerk an Mittätern.

Die Algorithmen zur Betrugserkennung sind recht komplex und werden von Spezialisten entwickelt und gepflegt. Dabei handelt es sich z. T. auch um externe Dienstleister der Versicherungen. Zum Entwickeln der Algorithmen dürfen diese natürlich keinen Zugriff auf die Echtdaten der Versicherung bekommen. Auf die eigentlichen Algorithmen zur Betrugserkennung gehen wir in diesem Vortrag nicht ein, sondern nur auf einige zentrale Anforderungen an die maskierten Daten:

- 2 Personen, die in derselben Straße wohnen, müssen nach der Maskierung wieder in derselben Straße wohnen
- Namen sind getrennt nach Vorname und Nachname zu maskieren, identische Vornamen müssen in identische Vornamen überführt werden, ebenso bei Nachnamen
- Personen aus dem selben Ort müssen nach der Maskierung wieder im selben Ort leben
- 2 Kontoinhaber bei einer Bank müssen nach der Maskierunbg wieder Konten bei einer Bank haben
- 2 Personen mit unterschiedlichen Vornamen haben nach der Maskierung auch wieder unterschiedliche Vornamen
- Eineindeutigkeit zwischen Original- und maskierten Personen

Ohne diese Anforderung entstehen bei den maskierten Daten ganz andere Netzwerke / Graphen zwischen den Personen.

Tabelle 4 Originaltabelle

Name	Vorname	Ort	PLZ	Straße	Konto	BLZ
Meier	Hans	Erding	85463	Am Markt 1	1234578	75050001
Huber	Sebastian	Erding	85463	Am Markt 2	68396709	70020000
Müller	Hans	Ebersberg	84342	Sternstraße	54784378	70020000

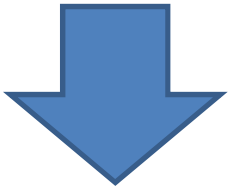

Tabelle 5 Maskierte Daten für Netzwerkanalysen

Name	Vorname	Ort	PLZ	Straße	Konto	BLZ
Graf	Boris	Übersee	88973	Uferstraße 3	765639	20041133
Becker	Herman	Übersee	88973	Uferstraße 3	7874267	74138900
Kahn	Boris	Grafing	84537	Almfeld 5	463114587	74138900

Während der Entwicklung neuer Algorithmen möchte man die Güte der neuen Algorithmen mit dem aktuellen Verfahren auf der Produktion vergleichen. Um dies zu ermöglichen muss man die Mappingtabellen, die während der Maskierung erzeugt werden, aufbewahren und nicht löschen. Diese Mappingtabellen dürfen aber nicht mit auf die Entwicklungsdatenbank mit transferiert werden.

Test und Entwicklung

Applikationsentwicklung

Im Bereich der Applikationsentwicklung sind die Anforderungen an die Daten üblicherweise deutlich geringer als für das Data Mining. Trotzdem muss mit den Entwicklern und Testern genau abgestimmt werden, welche Eigenschaften die maskierten Daten behalten sollen, bzw. welche Plausibilitätsprüfungen diese erfüllen müssen.

Für Testdatenbanken sind üblicherweise auch nur Teilmengen der Produktionsdaten notwendig bzw. erwünscht, da diese Systeme oft schwächer dimensioniert sind. Diese Teilmengen an Daten müssen aber in sich konsistent sein.

Die Erstellung von konsistenten Teilmengen wird nur von wenigen Tools unterstützt und erfordert daher umfangreichen manuellen Aufwand. Mit dem Release 12c von Grid Control ist es nun aber möglich solche konsistente Teilmengen an Testdaten zu erzeugen. Dort findet man die Funktionaliät unter dem Begriff Data Subsetting.

Für die Erstellung von Test- und Entwicklungsdatenbanken gibt es zwei Möglichkeiten:

- Datenbankcloning und Löschung der überflüssigen Daten
- Migration der Daten per Datapump zwischen Produktion und Entwicklung

Folgende Schritte sind in beiden Fällen notwendig:

- Application Data Modell anlegen
- Application Data Model anpassen
  - Beziehungen zwischen Tabellen werden durch FK-Beziehungen erkannt
  - Weitere Beziehungen können manuell hinzugefügt werden
- Subset definieren (über Quality Management)
- Prüfen des Platzbedarfes des Data Subsets
- Einmalig: Datenbankpackages per GridControl auf die Datenbank deployen

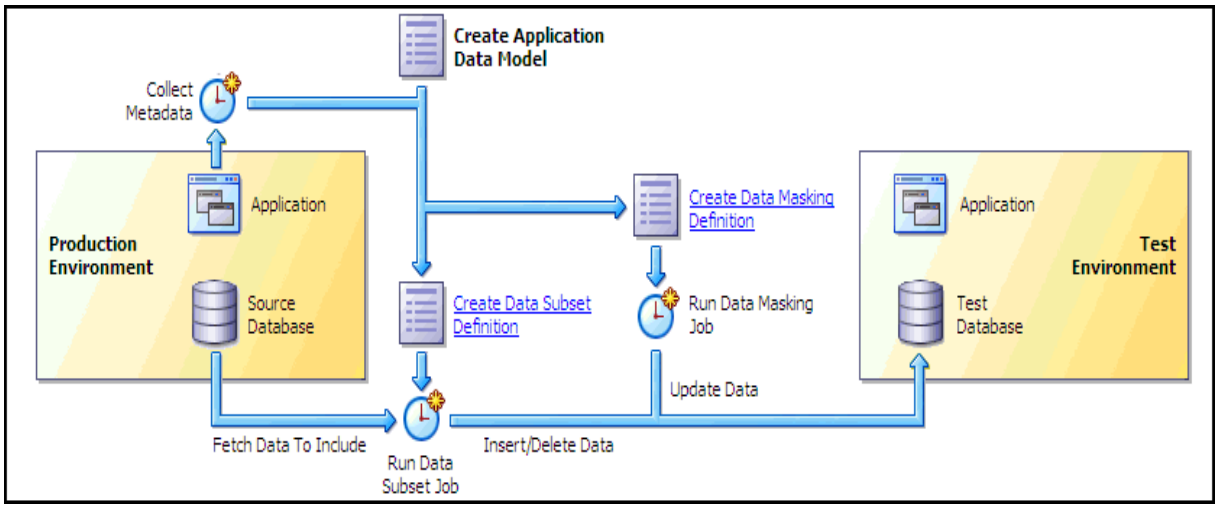

Tabelle 6 Vergleich der Verfahren zum Erzeugen von Data Subsets

Verfahren	Vorteile	Nachteile
Cloning	■ Test der Backups ■ Einfach zu skripten	■ Identischer Platzbedarf auf der Testdatenbank ■ Löschen inperformant ■ Nicht plattformübergreifend
Datapump	■ Geringer Platzbedarf ■ plattformübergreifend	■ Belastung der Produktivdatenbank ■ Export- und Importdauer ■ Höherer Skriptaufwand

Test

Im Testbereich muss man mehrere Arten von Tests unterscheiden:

- Fachliche Tests
- Technische Tests
  - Unittests
  - Integrationstests
  - Performancetests

Jede dieser Tests stellt unterschiedliche Anforderungen an die Tests.

Die umfangreichsten Anforderungen stellen die fachlichen Tests. Hierzu müssen die Daten bestimmte Ausprägungen aufweisen, damit die Tests durchgeführt werden können. Hier ist im Vorfeld der Maskierung eine Zusammenarbeit mit dem Fachbereich bzw. den Testern notwendig.

Performancetests stellen ebenfalls hohe Anforderungen an die maskierten Daten. Verändert man die Daten, so können sich Datenverteilungen ändern und somit auch die (optimalen) Zugriffspfade von Abfragen. Dies ist bei Performancetests natürlich nicht erwünscht und verhindert Aussagen über die eigentliche Performance auf Produktion.

Mit dem Release 12c von Grid Control haben wir die Möglichkeit Änderungen der Zugriffspfade auf Tuningset-Ebene zwischen den maskierten und den Originaldaten zu vergleichen. Um diesen Vergleich durchführen zu können, wird beim Maskieren der Datenbank auch das entsprechende Tuningset mit maskiert, sofern man diese Option auswählt.

Noch einen Schritt weiter geht die Integration von Real Application Testing und Data Masking. Hier werden die Protokolldateien, bei der Aufzeichnung der Produktivlast anfallen ebenfalls mit der

Datenbank zusammen maskiert. Dadurch lassen sich komplexe und umfangreiche Testläufe mit maskierten Testdaten durchführen. Da Real Application Testing Plattform und Versionsübergreifend anwendbar ist, eignet es sich gut um Auswirkungen von Plattformwechseln, neuer Hardware oder Storage bzw. Releasewechsel zu testen. Hierzu bietet Grid Control, auch zahlreiche Auswertungsmöglichkeiten.

Zusammenfassung

Richtiges Maskieren von Daten erlaubt die Verwendung dieser Daten für zahlreiche Anwendungsfälle. Von Tests über Entwicklung bis hin zum Data Mining reichen hier die Möglichkeiten. Mit dem richtigen Tool ist auch der Aufwand für die Maskierung gering. Mit dem aktuellen Release von Grid Control hat man nun zusätzliche Möglichkeiten für die Erstellung und Maskierung von Test- und Entwicklungsdatenbanken. Die Verknüpfung mit Real Application Testing erlaubt jetzt sichere Migrations- und Performancetests auf anderen Plattformen oder in den Labs der Hardwarehersteller.

Kontaktadresse:
Oliver Gehlert
Metafinanz Informationssysteme GmbH
Leopoldstraße 146
D-80804 München

Telefon:	+49 (0) 89-360531-0
Fax:	+49 (0) 89-360531-5015
E-Mail	oliver.gehlert@metafinanz.de
Internet:	www.metafinanz.de