# Maskierung von Daten

#### Motivation

- Inhalte bestimmter Spalten sollen nicht für jeden Benutzer sichtbar sein
  - > Z.B. Gehalt, Telefonnummer, Emailadresse
- Möglichkeit 1: steuern über Sichten + Rechtevergabe für Benutzer
- Möglichkeit 2: steuern über Tabellenwertfunktionen + Rechtevergabe für Benutzer
- Möglichkeit 3: maskieren bestimmter Spalten + Rechtevergabe für Benutzer
  - ➤ Vorteil: einfache und übersichtliche Vergabe der Rechte
  - Nachteil: all or nothing; entweder sieht der Benutzer alles maskiert oder alles demaskiert, nicht Spaltenweise definierbar

# Vorgehensweise

- Beim Erstellen der Tabelle Maskierungseigenschaft auf Spalte(n) legen
  - ➤ Kann auch nachträglich als Eigenschaft hinzugefügt werden
- Benutzern Rechte zum Selektieren von Daten auf die Tabelle geben
  - > Umfangreiches Thema, siehe "Anmeldungen, Rollen, Benutzer"
- Benutzern oder Rollen Rechte zum "unmaskieren" geben
  - > Ein Benutzer / eine Rolle mit diesem Recht kann die Daten sehen, welche für Andere maskiert sind

Thema Rechtevergabe, Anmeldungen, Rollen und Benutzer ist ein separates Thema ( >> Kursinhalte) <a href="https://msdn.microsoft.com/de-de/library/bb669084(v=vs.120).aspx">https://msdn.microsoft.com/de-de/library/bb669084(v=vs.120).aspx</a>

## Maskierungsarten

- Jede Maskierung wird mit MASKED WITH (FUNCTION = '<Funktionsname>')
  in die Wege geleitet
- Es stehen 4 verschiedene Funktionen für die Maskierung zur Verfügung

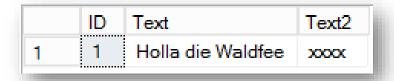
- Default
- > Email
- > Random
- Partial



# Default()

- Wählt eine zum Datentyp passende Maskierung
  - > Bei Zeichenfolgen wird der Inhalt mit XXXX (oder weniger X bei kleineren Längen) maskiert
  - Numerische Datentypen werden mit **0** maskiert
  - > Datentypen für Datum und Uhrzeit erhalten den Wert 01.01.1900 00:00:00.0000000
  - > Binäre Werte (binary, varbinary, image) bekommen ein Einzelbyte des ASCII-Werts 0
- Beispiel

```
ALTER TABLE TestTabelle
ALTER COLUMN [Text2]
ADD MASKED WITH (Function = 'default()');
```



## **Email**

- Für Zeichenfolgedatentypen
- Behält den ersten Buchstaben bei und ersetzt alles Weitere mit XXX@XXXX.com

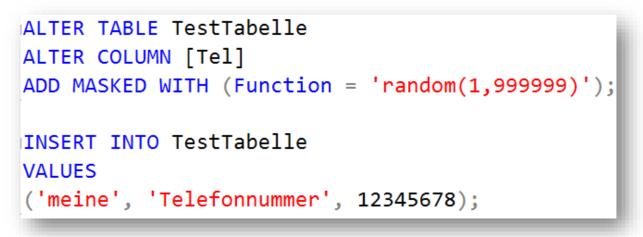
Beispiel

```
ALTER TABLE TestTabelle
ALTER COLUMN [Text2]
ADD MASKED WITH (Function = 'email()');
```

	ID	Text	Text2
1	1	Holla die Waldfee	HXXX@XXXX.com
2	2	Hui Buh	aXXX@XXXX.com

## Random

- Für numerische Datentypen
- Ein Zufallswert eines bestimmten Range maskiert die Ursprungsdaten
- Der Range sind zwei Übergabeparameter (start\_range, end\_range)
- Beispiel





```
ALTER TABLE TestTabelle
ALTER COLUMN [Tel]
ADD MASKED WITH (Function = 'random(1,5)');
```

	ID	Text	Text2	Tel
1	1	Holla die Waldfee	HXXX@XXXX.com	NULL
2	2	Hui Buh	aXXX@XXXX.com	NULL
3	4	meine	TXXX@XXXX.com	2
4	5	nochne	TXXX@XXXX.com	3

## **Partial**

- Erlaubt eine benutzerdefinierte Maskierung von Zeichenkettendatentypen
- Syntax: partial(prefix, padding, suffix)
  - > Prefix: die Anzahl der ersten sichtbaren Zeichen
  - > Suffix: die Anzahl der letzten sichtbaren Zeichen
  - Padding: die Zeichenkette, welche anstelle der zwischen Prefix und Suffix befindlichen Zeichen ausgegeben wird (bei Überlagerung hat *padding* Priorität)
- Beispiel:

```
ALTER TABLE TestTabelle
ALTER COLUMN [Text2]
ADD MASKED WITH (Function = 'partial(1, "Muuh", 2)');
```

	ID	Text	Text2	Tel
1	1	Holla die Waldfee	HMuuher	NULL
2	2	Hui Buh	Muuh	NULL
3	4	meine	TMuuher	1
4	5	nochne	TMuuher	5

# Weitere Eigenschaften

- Maskieren verhindert keine Einfügeoperationen (!)
  - Zugriffsteuerung sollte weiterhin verwendet werden
- Bei Kopiervorgängen werden die maskierten Einträge kopiert, sofern der Benutzer nicht über die Demaskierungsberechtigung verfügt
- Gleiches gilt für Ausführung von Import- bzw. Exportvorgängen
- Verschlüsselte Spalten, FILESTREAM-Spalten, Volltextindexspalten sind nicht maskierbar
- Berechnete Spalten können nicht maskiert werden; berechnete Spalten mit zugrundeliegender maskierter Spalte geben maskierte Werte aus
- Systemsicht aller maskierten Spalten: sys.masked\_columns