orgAnice Software Solution GmbH

orgAnice SQL-Reportgenerator

Benutzerhandbuch

Stand: Februar 2022 | Version 2.0.0.4

Inhalt

1 Allgemeine Informationen

1.1 Einführung

Der orgAnice SQL-Reportgenerator stellt die folgenden Funktionalitäten zur Verfügung:

- Erstellung von Listen in Microsoft Excel™ anhand von SQL-Abfragen, die Daten aus der orgAnice-Datenbank auswerten
- Die SQL-Abfragen können mit Parametern versehen werden, sodass der Benutzer nur die Parameter ändern muss, ohne die zugrundeliegende SQL-Abfrage anpassen zu müssen
- Die Datensätze, die als Ergebnisse der Abfrage ausgegeben werden, können zusätzlich in orgAnice markiert werden.

1.2 Voraussetzungen

Folgende Voraussetzungen sind für die korrekte Funktion des orgAnice SQL-Reportgenerators nötig:

Systemvoraussetzungen:

- Windows 8 / 8.1/ 10/ 11
- .Net Framework 4
- Microsoft Visual C++ 2010 Runtimes

orgAnice-Voraussetzungen:

- orgAnice CRM 7

1.3 Zielgruppe

Das AddOn richtet sich an versierte orgAnice-Benutzer, die gute SQL-Kenntnisse besitzen und die Möglichkeiten, die SQL-Abfragen bieten, auf die orgAnice-Datenbank anwenden möchten. Gute Kenntnisse der relationalen Abhängigkeiten der orgAnice-Tabellen sind vom großen Vorteil, da die innerhalb orgAnice festdefinierten Relationen innerhalb von SQL-Abfragen mit dem JOIN-Schlüsselwort abgebildet werden müssen.

1.4 Warnung

Sie können mit Hilfe des SQL-Reportgenerators beliebige SQL-Abfragen an Ihre orgAnice SQL-Datenbank übermitteln, auch solche, die Datensätze oder gar Tabellen löschen. Lassen Sie die Standardeinstellung, dass die Konfiguration des AddOn ausschließlich Administratoren zugänglich ist, unbedingt bestehen. Denn damit dürfen unbedarfte Nutzer keine Änderungen an den Abfragen vornehmen.

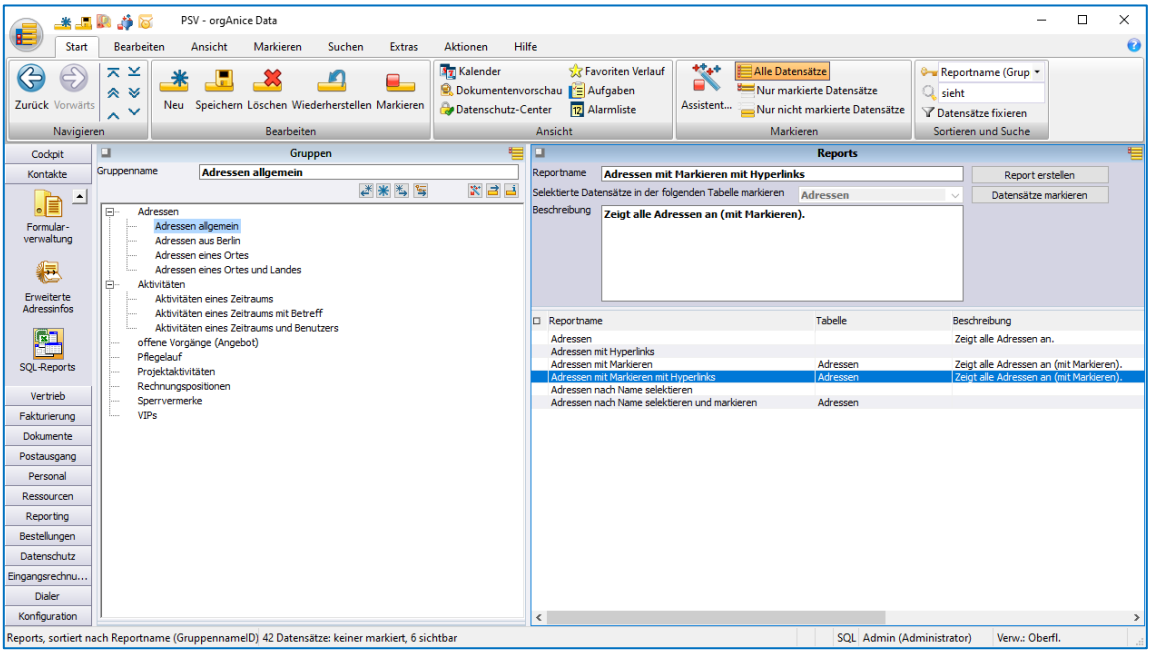
orgAnice Software Solution GmbH haftet für keinerlei Schäden, insbesondere nicht für Datenverlust, die durch fehlerhaft programmierte SQL-Abfragen entstehen.

2 Installation des orgAnice SQL-Reportgenerators

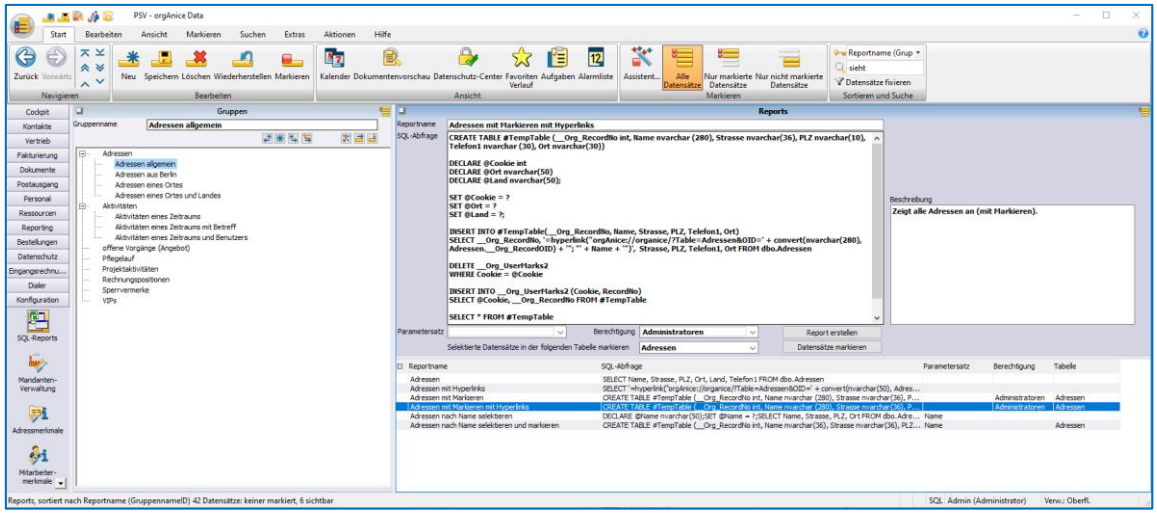
Das Datenbank-Update bereitet Ihre Datenbank auf den Einsatz des orgAnice SQL-Reportgenerators vor. Es werden eine Tabelle, zwei Tabellenlayouts sowie zwei Arbeitsbereiche angelegt. Stellen Sie vor dem Update sicher, dass zur Zeit des Datenbank-Updates kein Anwender mit Ihrer Datenbank arbeitet.

Starten Sie die Datei: „orgAnice-SQLReportgenerator-Setup*.exe*“ und folgen Sie den Schritten des orgAnice Installers.

Nach erfolgreicher Installation finden Sie in Ihrer Datenbank zwei neue Arbeitsbereiche. Der Arbeitsbereich „SQL-Reports“, den Sie innerhalb der Arbeitsbereichsgruppe „Kontakte“ sehen, ist für alle Benutzer zugänglich und dient der Ausführung der SQL-Reports.



Der Arbeitsbereich „SQL-Reports“, den Sie innerhalb der Arbeitsbereichsgruppe „Konfiguration“ sehen, ist nur für Administratoren zugänglich und dient der Konfiguration der SQL-Reports.



Lesen Sie im Folgenden, wie Sie Reports einrichten und verwenden können.

3 Einrichten von Reports

Damit die Datenbankbenutzer SQL-Reports ausführen können, müssen diese durch den Administrator konfiguriert werden. Ist die Baumansicht aktiviert, können Report in übergeordnete Gruppen sortiert werden. Es gibt vier Typen von Reports, die im Folgenden beschrieben werden:

- Parameterlose Reports
- Reports mit Parametern
- Reports mit Markierungsfunktionalität
- Reports mit Parametern und Markierungsfunktionalität (dieser Typ vereinheitlicht die zwei vorigen und wird deswegen nicht explizit beschrieben)

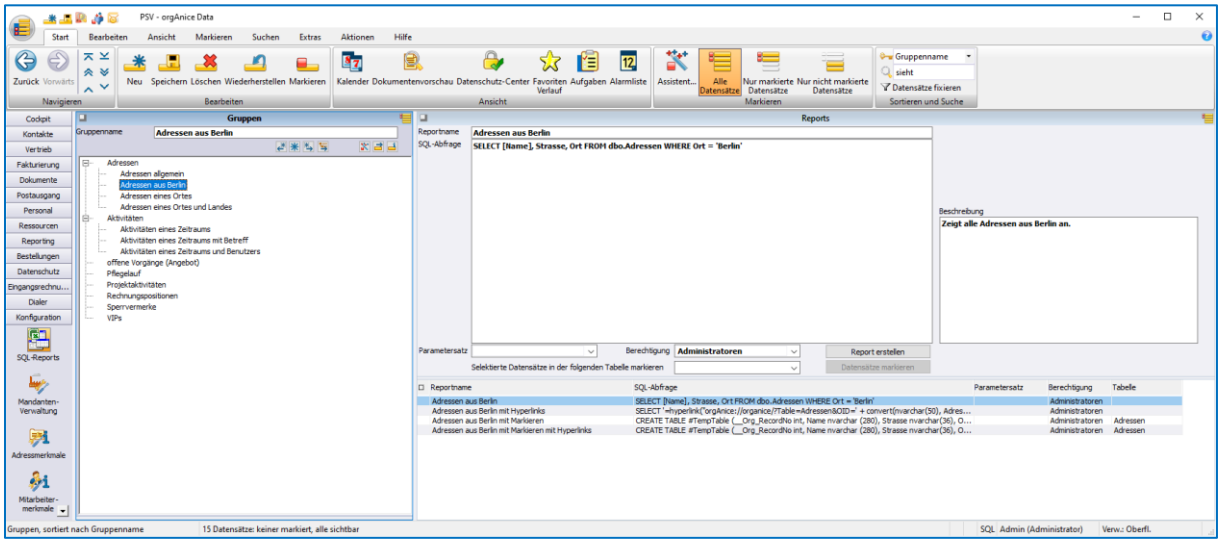
Als ein weiterer Spezialfall werden Reports mit Abfragen über mehrere Tabellen beschrieben.

3.1 Parameterlose Reports

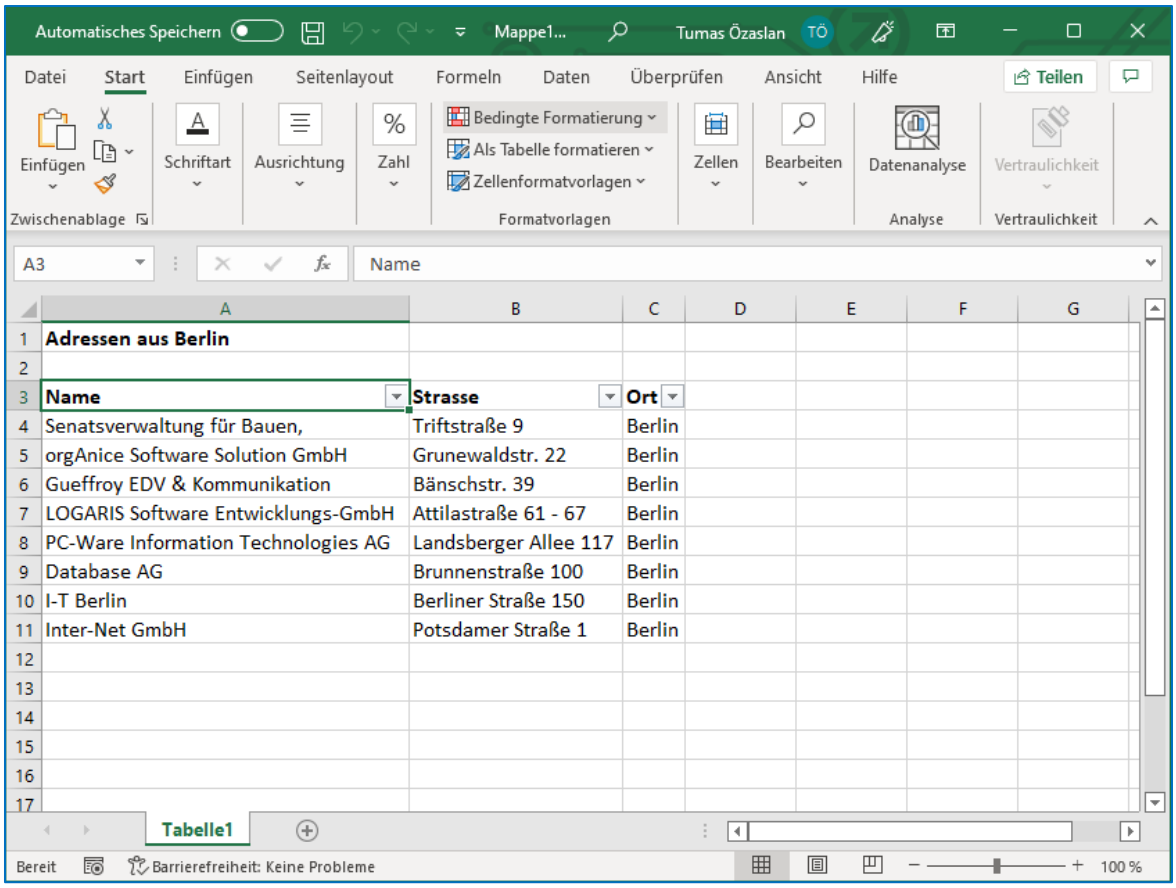
Die parameterlosen Reports bilden die einfachste Art der SQL-Reports, ihre Erstellung ist vergleichsweise einfach. Erstellen Sie in der Tabelle Reports einen neuen Datensatz, vergeben Sie einen sinnvollen Namen und tragen Sie im Feld SQL-Abfrage die gewünschte Abfrage ein.

Beispiel: Wir möchten in einem Report die Namen und Anschriften aller Adressen aus Berlin ausgeben. Die passende SQL-Abfrage lautet:

```
SELECT Name, Strasse, PLZ, Ort FROM dbo.Adressen WHERE Ort = 'Berlin'
```



Erstellen Sie den entsprechenden Datensatz und betätigen Sie die Schaltfläche „Report erstellen“. Das Ergebnis sieht dann folgendermaßen aus (alle Beispiele stammen aus der Demo-orgAnice CRM-Datenbank):



	A	B	C
1	**Adressen aus Berlin**		
2			
3	**Name**	**Strasse**	**Ort**
4	Senatsverwaltung für Bauen,	Triftstraße 9	Berlin
5	orgAnice Software Solution GmbH	Grunewaldstr. 22	Berlin
6	Gueffroy EDV & Kommunikation	Bänschstr. 39	Berlin
7	LOGARIS Software Entwicklungs-GmbH	Attilastraße 61 - 67	Berlin
8	PC-Ware Information Technologies AG	Landsberger Allee 117	Berlin
9	Database AG	Brunnenstraße 100	Berlin
10	I-T Berlin	Berliner Straße 150	Berlin
11	Inter-Net GmbH	Potsdamer Straße 1	Berlin

3.2 Reports mit Parametern

Reports können auch mit Parametern ausgestattet werden, die einen Report wiederverwendbar machen. Stellen wir uns vor, wir möchten nicht nur die Adressen aus Berlin, sondern auch aus Hamburg, München oder weiteren Städten auswerten. Die erste Möglichkeit, den oben erstellten Report mehrfach zu duplizieren, steht natürlich auch zur Verfügung, ist aber alles andere als pflegeleicht. Hier bietet es sich an, den Namen der Stadt als durch den Benutzer eingebbaren Parameter zu gestalten.

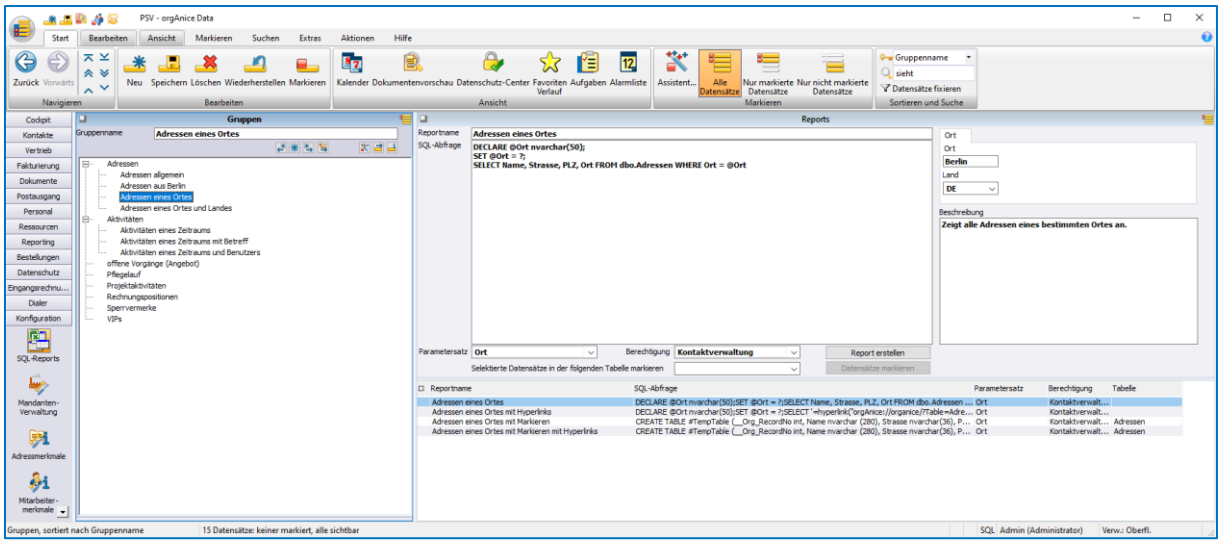
3.2.1 Nutzung vorbereiteter Parameter

Die Möglichkeit den Ort als Parameter einzustellen, ist bereits im Standard vorbereitet, später werden wir sehen, wie wir eigene Parameter erstellen können.

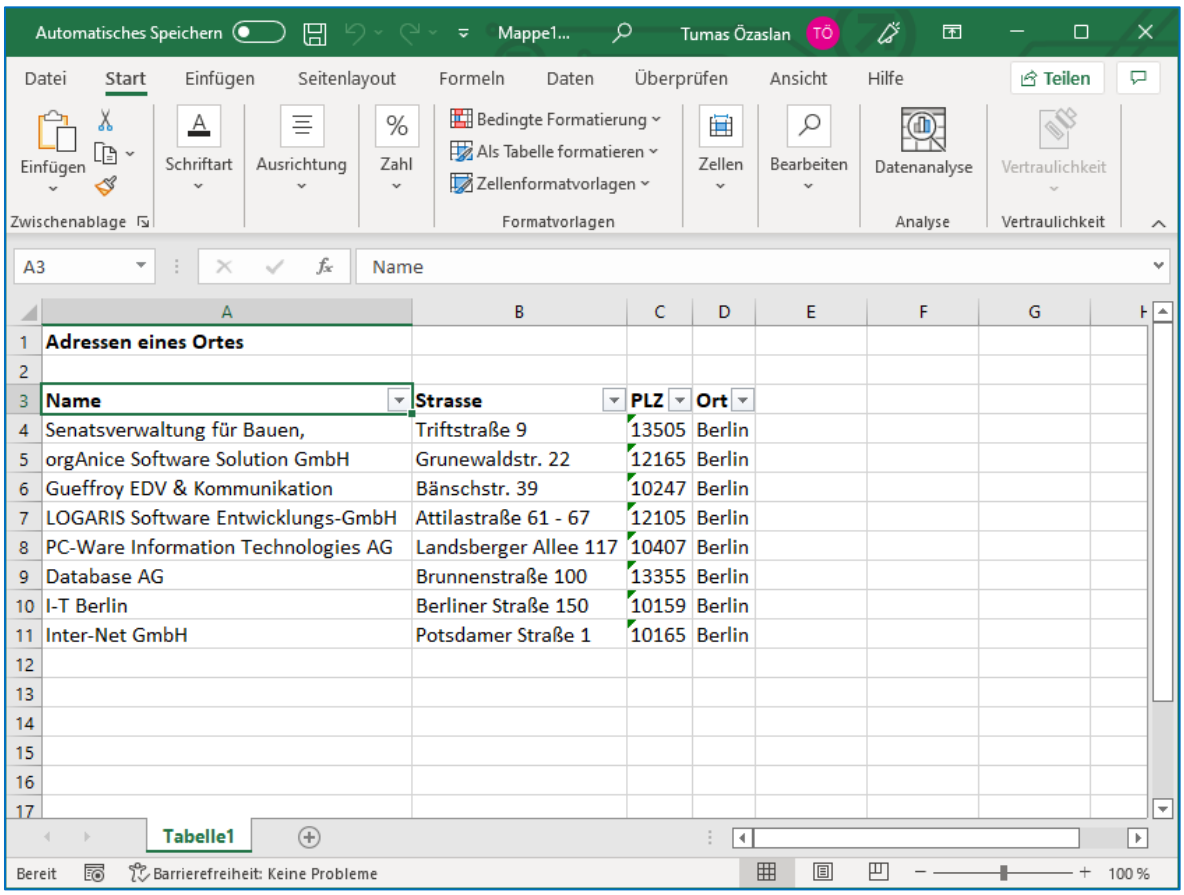
Duplizieren Sie den vorhandenen Datensatz, um den parameterlosen Report als Vergleich noch beizubehalten und geben Sie ihm einen anderen Namen. Wählen Sie dann als Parametersatz „Ort“ aus. Sofort wird die Eingabe des Parameters „Ort“ möglich. Wir müssen die SQL-Abfrage noch folgendermaßen anpassen:

```
DECLARE @Ort nvarchar(50);
SET @Ort = ?;
SELECT Name, Strasse, PLZ, Ort FROM dbo.Adressen WHERE Ort = @Ort
```

@Ort ist hier der Name des Parameters, der verwendet wird.



Geben Sie als Ort z.B. „Berlin“ ein, so betätigen Sie die Schaltfläche „Report erstellen“. Das Ergebnis sieht dann folgendermaßen aus:



	A	B	C	D	E	F	G
1	**Adressen eines Ortes**						
2							
3	**Name**	**Strasse**	**PLZ**	**Ort**			
4	Senatsverwaltung für Bauen,	Triftstraße 9	13505	Berlin			
5	orgAnice Software Solution GmbH	Grunewaldstr. 22	12165	Berlin			
6	Gueffroy EDV & Kommunikation	Bänschstr. 39	10247	Berlin			
7	LOGARIS Software Entwicklungs-GmbH	Attilastraße 61 - 67	12105	Berlin			
8	PC-Ware Information Technologies AG	Landsberger Allee 117	10407	Berlin			
9	Database AG	Brunnenstraße 100	13355	Berlin			
10	I-T Berlin	Berliner Straße 150	10159	Berlin			
11	Inter-Net GmbH	Potsdamer Straße 1	10165	Berlin			

Wollte man noch das Land auswerten, so müsste die Abfrage folgendermaßen erweitert werden:

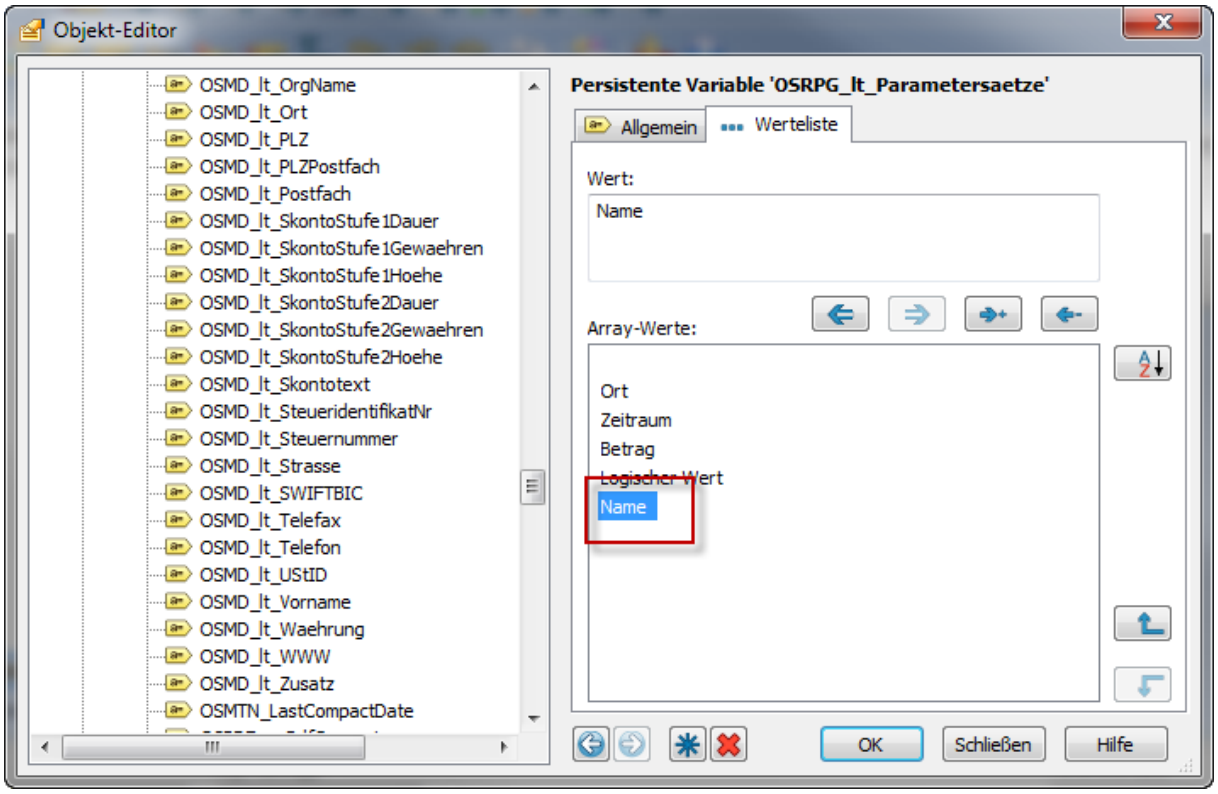
```
DECLARE @Ort nvarchar(50)
DECLARE @Land nvarchar(50);
SET @Ort = ?
SET @Land = ?;
SELECT Name, Strasse, PLZ, Ort FROM dbo.Adressen WHERE Ort = @Ort AND Land
= @Land
```

3.2.2 Hinzufügen von neuen Parametern

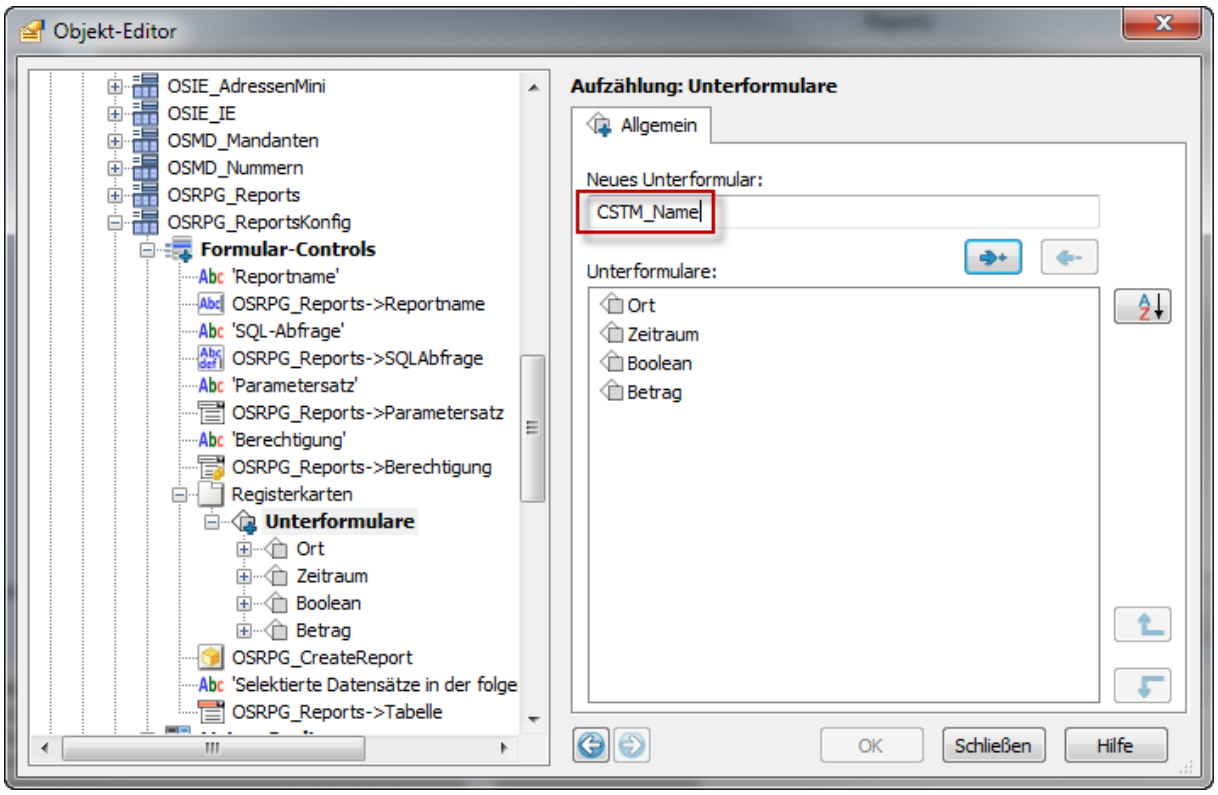
Sollen in einer Abfrage Parameter verwendet werden, nicht zu den ausgelieferten Parametern gehören, so müssen mehrere Schritte durchgeführt werden. Diese werden im Folgenden anhand eines Beispiels erklärt.

Beispiel: Wir möchten Adressen ausgeben, die im Namen eine bestimmte Zeichenkette beinhalten (z.B. „GmbH“). Die Zeichenkette soll dabei als Parameter eingebbar sein.

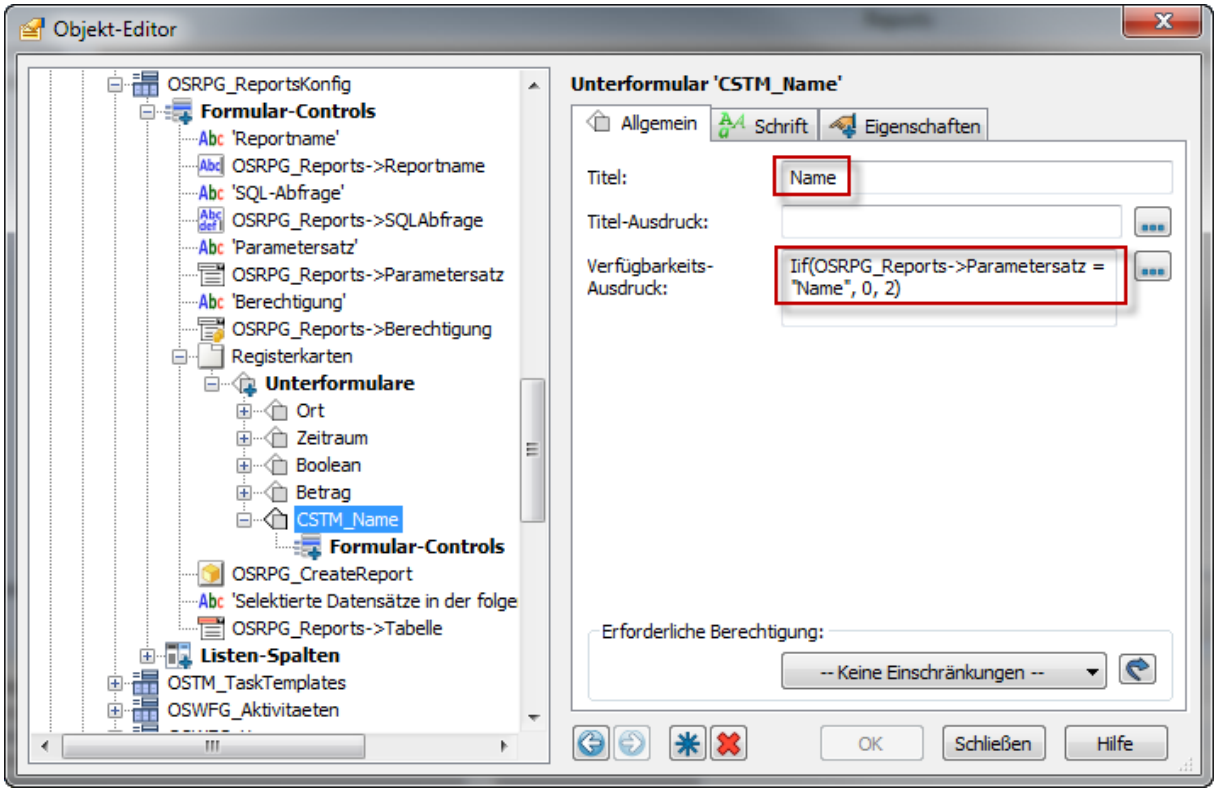
1. Erweitern Sie im Parser bei den persistenten Variablen „OSRPG_lt_Parametersaetze“ um den Wert „Name“:



2. Fügen Sie im Formular OSRPG_ReportsKonfig dem Registerkarten-Control ein neues Unterformular „CSTM_Name“ hinzu:

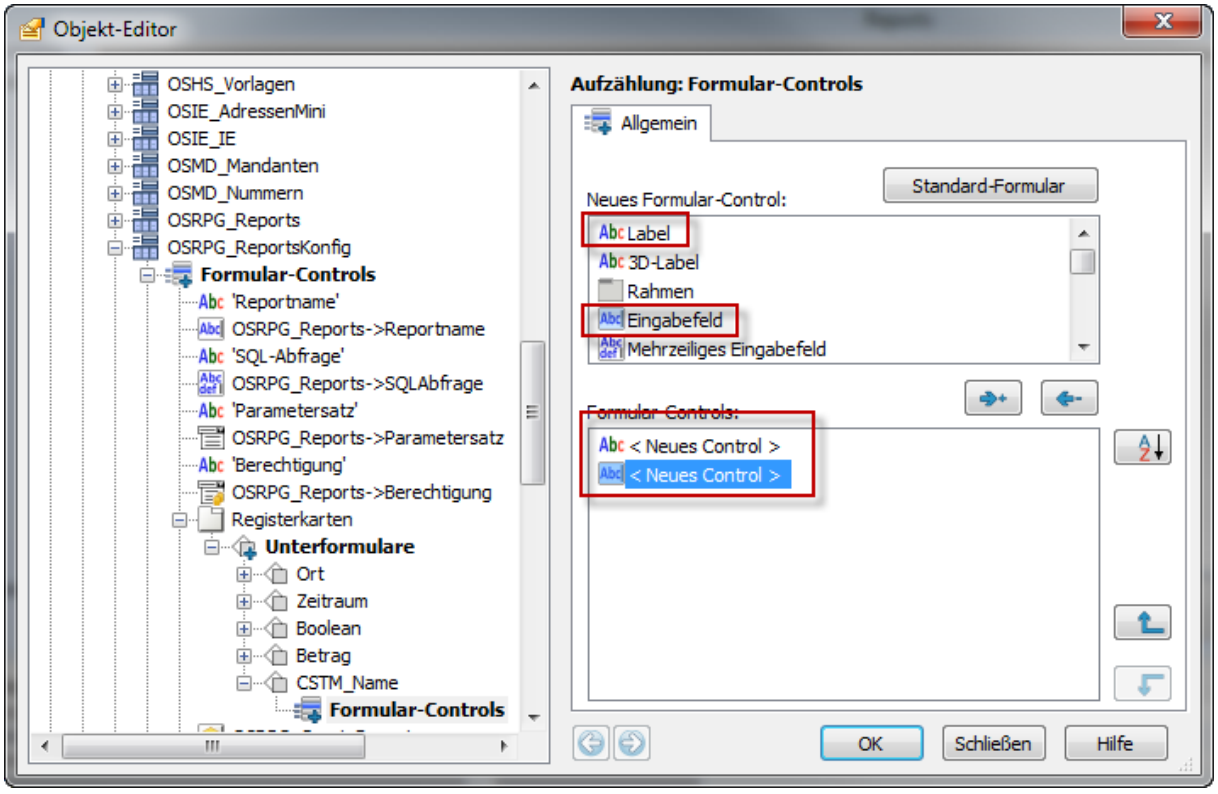


3. Passen Sie den Titel des Unterformulars und den Verfügbarkeits-Ausdruck an:

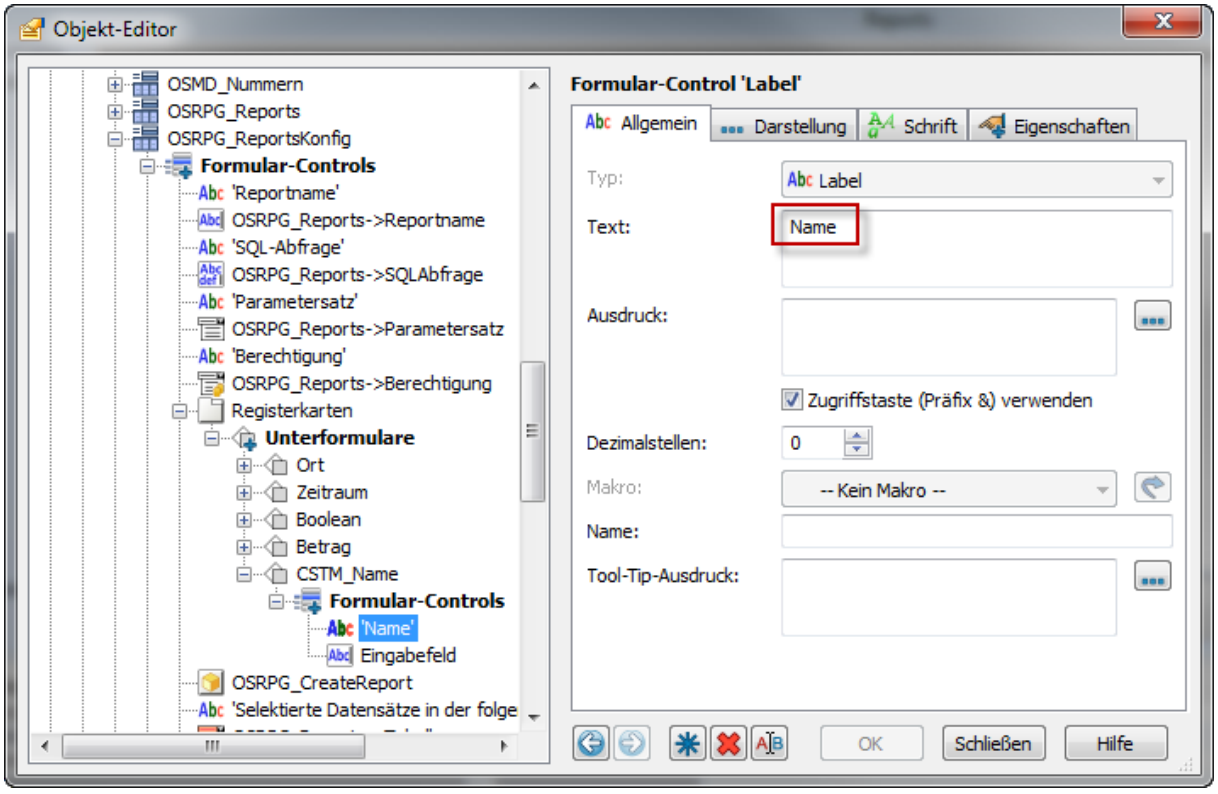


Der Verfügbarkeits-Ausdruck
Iif(OSRPG_Reports->Parametersatz = "Name", 0, 2)
gibt an, dass das Unterformular nur angezeigt wird, wenn als Parametersatz „Name“ ausgewählt ist.

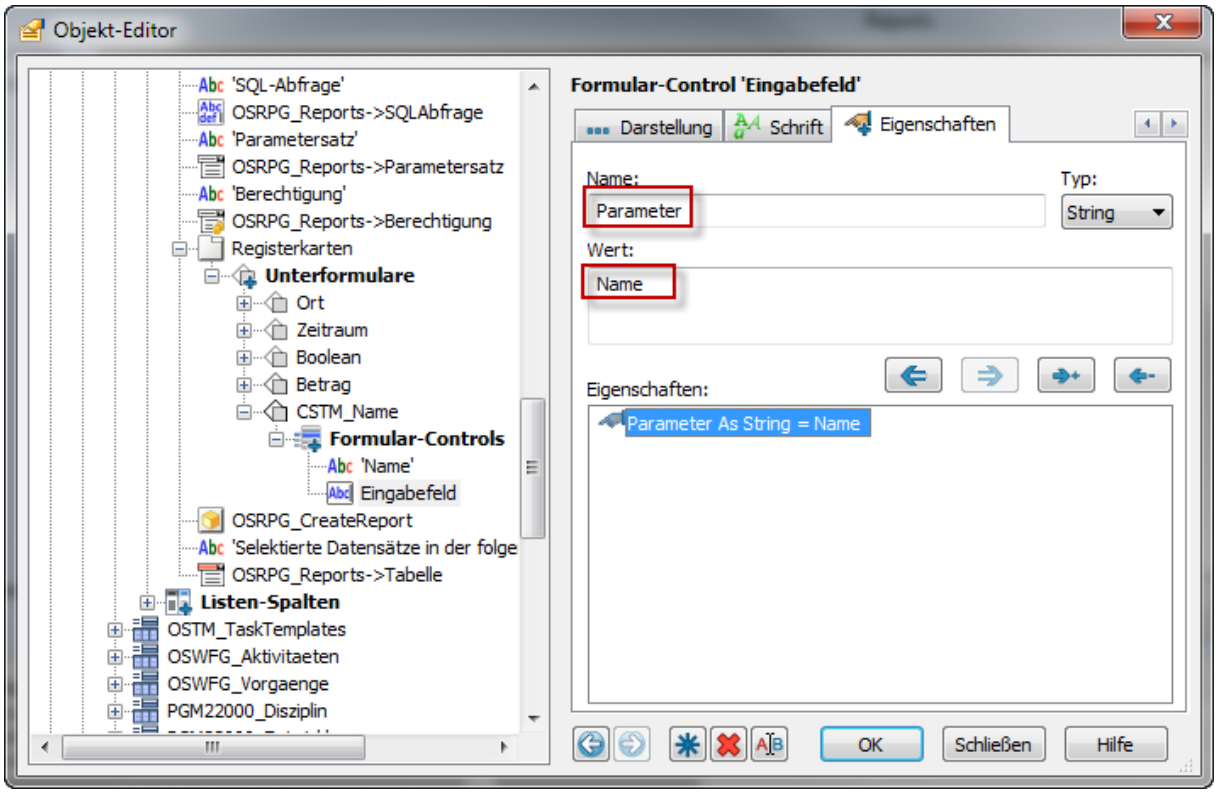
4. Fügen Sie dem Unterformular ein Label- und ein Eingabefeld-Control hinzu



5. Vergeben Sie dem Label den Text „Name“

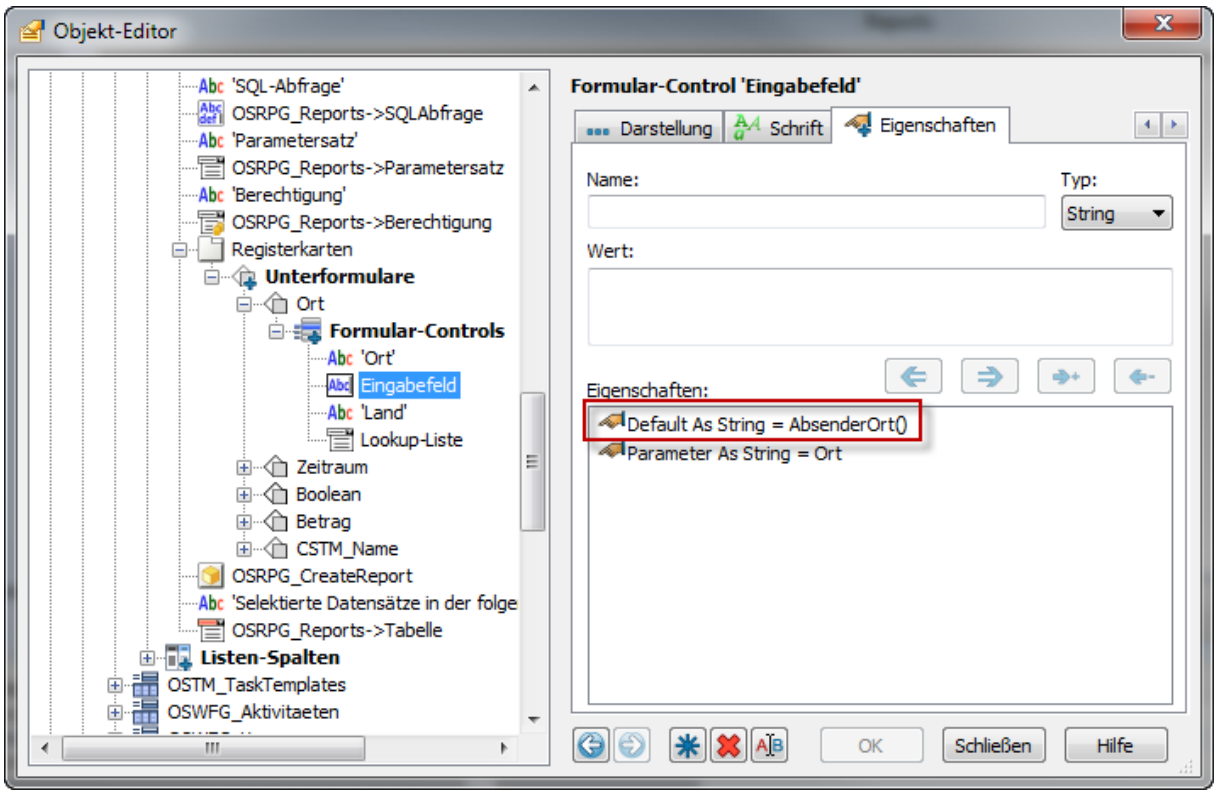


6. Positionieren Sie auf das Eingabefeld und wechseln zum Register „Eigenschaften“. Hier fügen Sie die Eigenschaft „Parameter“ mit dem Wert „Name“ hinzu:



Das ist der entscheidende Schritt, an dem die Verbindung zwischen dem Eingabefeld und der SQL-Abfrage geschaffen wird. In der SQL-Abfrage werden wir den Parameter @Name verwenden, um auf die Benutzereingabe aus dem Control zuzugreifen.

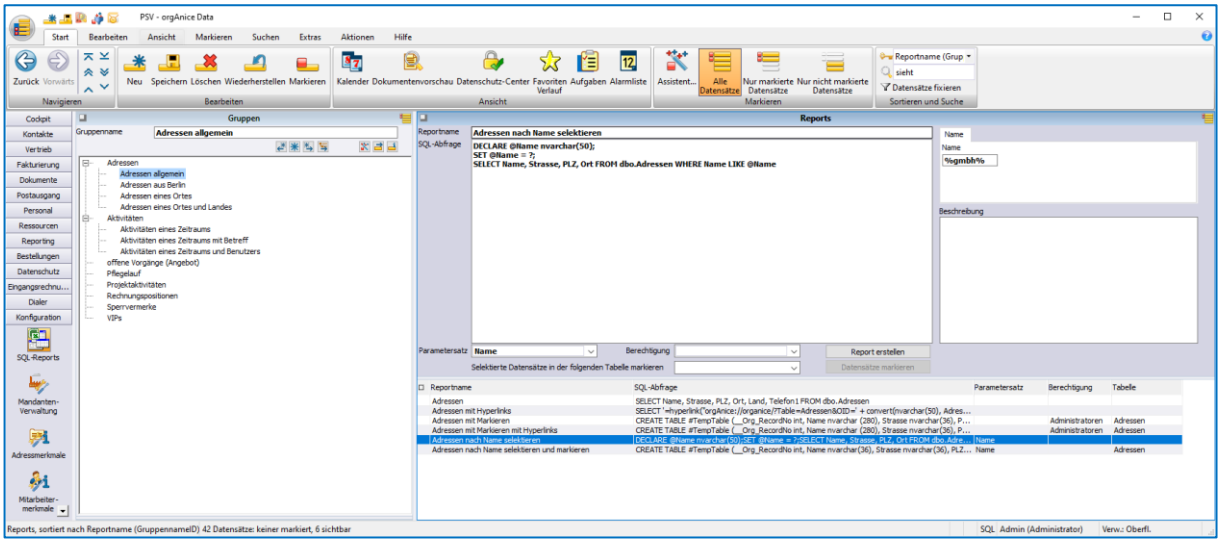
Hinweis: Als weitere mögliche Eigenschaft steht noch die Eigenschaft „Default“ zur Verfügung. Dadurch können wir eine Vorbelegung für den Parameter angeben, wie das beispielsweise beim Ort der Fall ist:



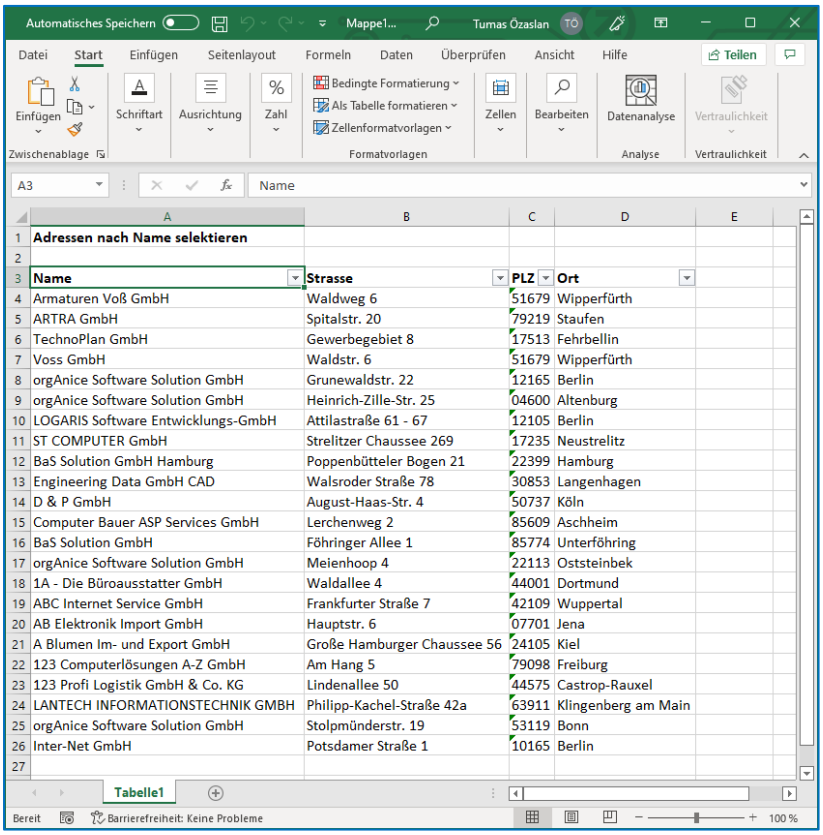
7. Führen Sie die Schritte 2-6 auch für das Tabellenlayout „OSRPG_Reports“ durch. Sie können auch stattdessen die Controls mit Hilfe der Copy&Paste-Funktionalität vom Tabellenlayout „OSRPG_ReportsKonfig“ in das Tabellenlayout „OSRPG_Reports“ kopieren.
8. Schließen Sie den Objekt-Editor und wechseln in den Benutzermodus.
9. Fügen Sie in der Tabelle „Reports“ einen neuen Datensatz hinzu und stellen Sie als Parametersatz „Name“ ein. Sofort steht die Parameterauswahl „Name“ zur Verfügung.

10. Die SQL-Abfrage, die jetzt einzugeben ist, ist der bereits bekannten Abfrage nach dem Ort ähnlich:

```
DECLARE @Name nvarchar(50);
SET @Name = ?;
   SELECT Name, Strasse, PLZ, Ort FROM dbo.Adressen WHERE Name LIKE
                                    @Name
```



Wenn Sie jetzt als Parameter “%gmbh%” eingeben (beachten Sie, dass in der Abfrage das Schlüsselwort LIKE verwendet wird), so erhalten Sie das folgende Ergebnis:



	A	B	C	D	E
1	**Adressen nach Name selektieren**				
2					
3	**Name**	**Strasse**	**PLZ**	**Ort**	
4	Armaturen Voß GmbH	Waldweg 6	51679	Wipperfürth	
5	ARTRA GmbH	Spitalstr. 20	79219	Staufen	
6	TechnoPlan GmbH	Gewerbegebiet 8	17513	Fehrbellin	
7	Voss GmbH	Waldstr. 6	51679	Wipperfürth	
8	orgAnice Software Solution GmbH	Grunewaldstr. 22	12165	Berlin	
9	orgAnice Software Solution GmbH	Heinrich-Zille-Str. 25	04600	Altenburg	
10	LOGARIS Software Entwicklungs-GmbH	Attilastraße 61 - 67	12105	Berlin	
11	ST COMPUTER GmbH	Strelitzer Chaussee 269	17235	Neustrelitz	
12	BaS Solution GmbH Hamburg	Poppenbütteler Bogen 21	22399	Hamburg	
13	Engineering Data GmbH CAD	Walsroder Straße 78	30853	Langenhagen	
14	D & P GmbH	August-Haas-Str. 4	50737	Köln	
15	Computer Bauer ASP Services GmbH	Lerchenweg 2	85609	Aschheim	
16	BaS Solution GmbH	Föhringer Allee 1	85774	Unterföhring	
17	orgAnice Software Solution GmbH	Meienhoop 4	22113	Oststeinbek	
18	1A - Die Büroausstatter GmbH	Waldallee 4	44001	Dortmund	
19	ABC Internet Service GmbH	Frankfurter Straße 7	42109	Wuppertal	
20	AB Elektronik Import GmbH	Hauptstr. 6	07701	Jena	
21	A Blumen Im- und Export GmbH	Große Hamburger Chaussee 56	24105	Kiel	
22	123 Computerlösungen A-Z GmbH	Am Hang 5	79098	Freiburg	
23	123 Profi Logistik GmbH & Co. KG	Lindenallee 50	44575	Castrop-Rauxel	
24	LANTECH INFORMATIONSTECHNIK GMBH	Philipp-Kachel-Straße 42a	63911	Klingenberg am Main	
25	orgAnice Software Solution GmbH	Stolpmünderstr. 19	53119	Bonn	
26	Inter-Net GmbH	Potsdamer Straße 1	10165	Berlin	
27					

3.3 Reports mit Markierungsfunktionalität

Wie bereits oben erwähnt, können Datensätze, die als Ergebnisse der Abfrage ausgegeben werden, zusätzlich in der orgAnice-Tabelle markiert werden. Dafür ist eine Erweiterung der SQL-Abfrage notwendig.

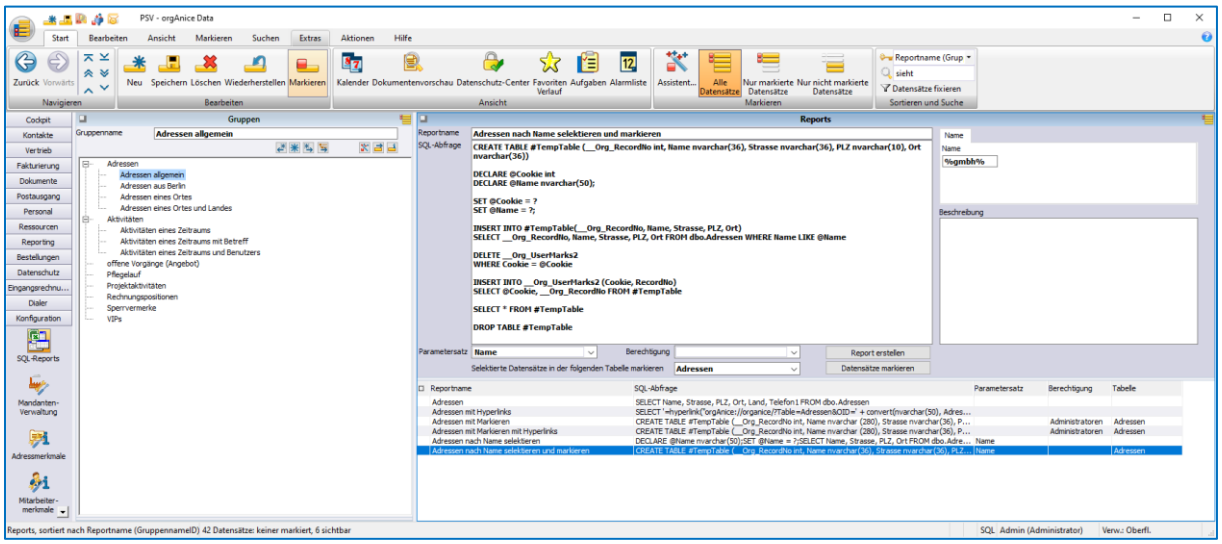
Beispiel: Wir möchten das vorige Beispiel (Adressen nach Namen selektieren) so erweitern, dass die selektierten Adressen in der orgAnice-Tabelle markiert werden.

1. Duplizieren Sie den Datensatz in der Report-Tabelle
2. Vergeben Sie einen neuen Namen
3. Stellen Sie in der Auswahlliste „Selektierte Datensätze in der folgenden Tabelle markieren“ die Tabelle „Adressen“ ein. Beachten Sie hierzu weitere Hinweise im Kapitel Auswahl der Tabelle für die Markierung, S.15.
4. Die SQL-Abfrage muss erweitert werden. In der untenstehenden Abfrage wurden dabei die Stellen gelb markiert, die bei einer SQL-Abfrage mit Markierungsfunktionalität variabel sind und von Ihnen angepasst werden können. Die nicht markierten Stellen sind für die Markierung notwendig und müssen in jeder SQL-Abfrage mit Markierungsfunktionalität vorhanden sein.

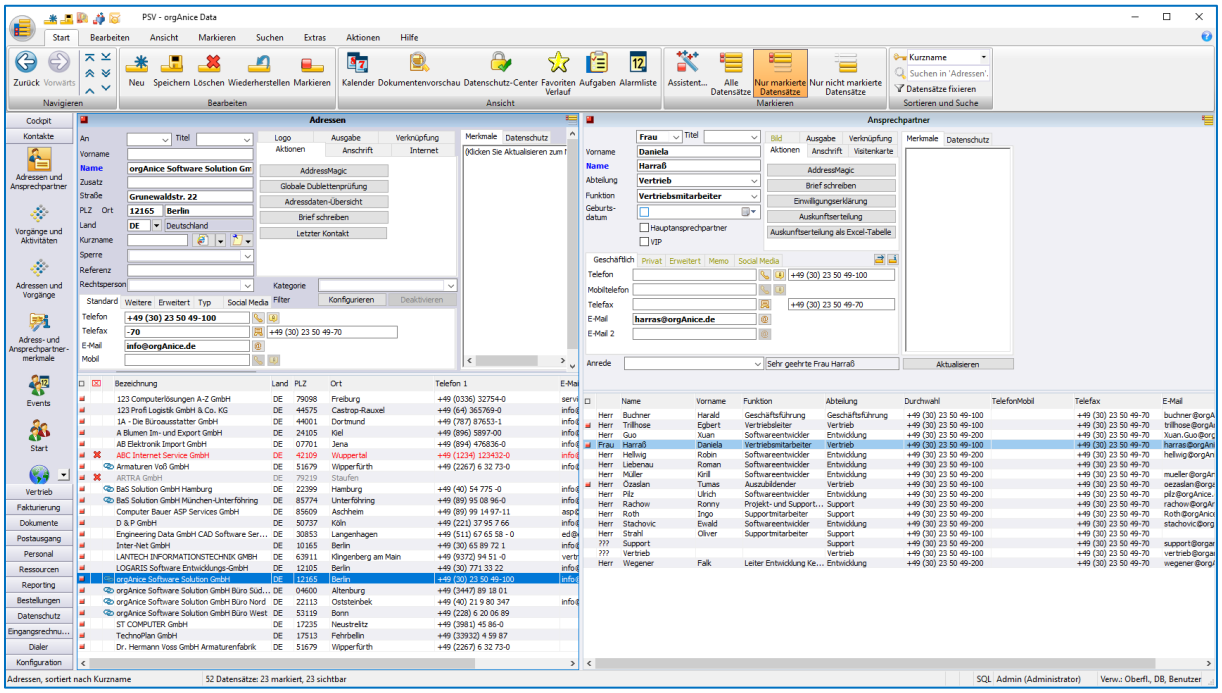
```
CREATE TABLE #TempTable (__Org_RecordNo int, Name nvarchar(36),
Strasse nvarchar(36), PLZ nvarchar(10), Ort nvarchar(36))

DECLARE @Cookie int
DECLARE @Name nvarchar(50);

SET @Cookie = ?
SET @Name = ?;

INSERT INTO #TempTable(__Org_RecordNo, Name, Strasse, PLZ, Ort)
SELECT __Org_RecordNo, Name, Strasse, PLZ, Ort FROM dbo.Adressen
WHERE Name LIKE @Name

DELETE __Org_UserMarks2
WHERE Cookie = @Cookie

INSERT INTO __Org_UserMarks2 (Cookie, RecordNo)
SELECT @Cookie, __Org_RecordNo FROM #TempTable

SELECT * FROM #TempTable

DROP TABLE #TempTable
```

Hier wird mit einer temporären Tabelle gearbeitet, die die gefundenen Datensätze zunächst zwischenspeichert. Die temporäre Tabelle muss die Spalte __Org_RecordNo beinhalten, da sie für das Setzen der Markierung benötigt wird. Die Selektion fügt zunächst die gewünschten Datensätze in die temporäre Tabelle ein.
Dann wird die bestehende Markierung entfernt und eine neue Markierung mit Hilfe der temporären Tabelle gesetzt. Die Markierung wird in der Tabelle __Org_UserMarks2 gespeichert.
Der Parameter @Cookie spielt hierbei eine besondere Rolle: er stellt einen Wert dar, der die Markierungen des aktuellen Benutzers in einer bestimmten Tabellen identifiziert. Dieser

Parameter wird programmatisch hinzugefügt, es muss kein Control dafür bereitgestellt werden.

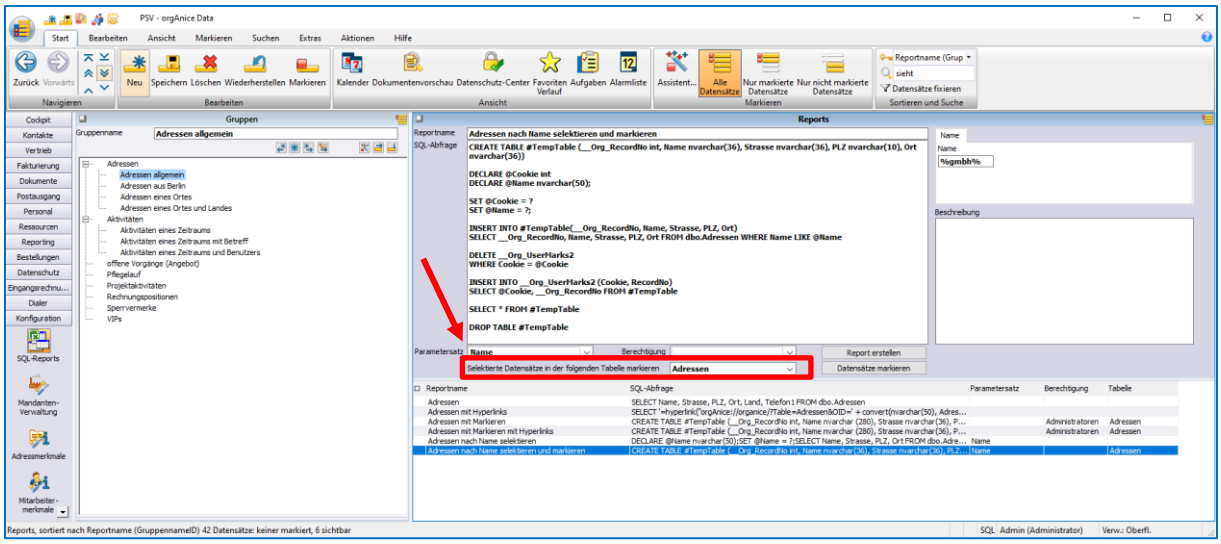


Das Ergebnis der Markierung sieht dann folgendermaßen aus:



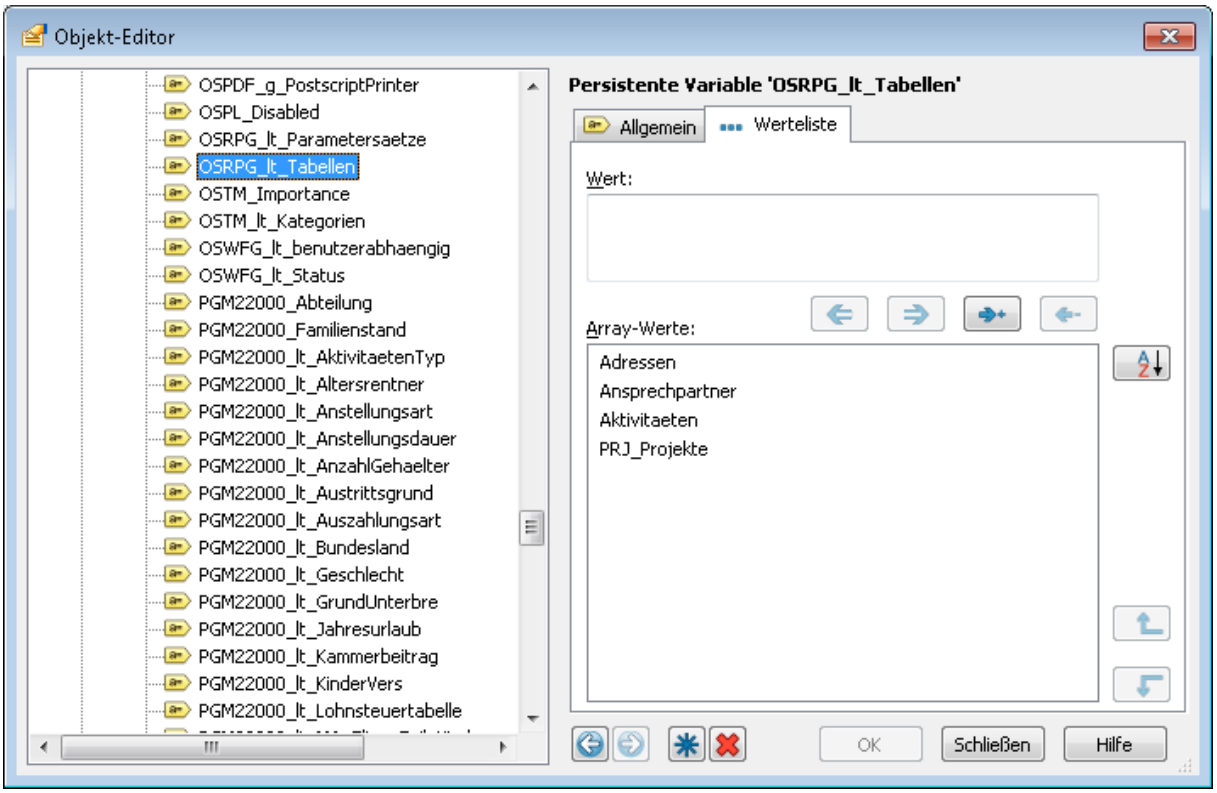
3.3.1 Auswahl der Tabelle für die Markierung

Mit Hilfe der Auswahlliste „Selektierte Datensätze in der folgenden Tabelle markieren“ wird die Tabelle ausgewählt, in der die Markierung vorgenommen wird. Nur wenn an dieser Stelle die richtige Tabelle eingestellt wird, wird die Markierung tatsächlich ausgeführt. Diese Einstellung ist aus Gründen der Programmlogik zusätzlich zur richtigen Angabe des Tabellennamens in der SQL-Abfrage notwendig.



In dem Auswahlfeld stehen einige orgAnice-Tabellen zur Verfügung, falls Ihre gewünschte Tabelle nicht dabei ist, können Sie entweder

a. Den Tabellennamen manuell eintragen, da es sich um eine editierbare Lookup-Liste handelt oder
b. Die persistente Variable „OSRPG_lt_Tabellen“ um den gewünschten Wert erweitern

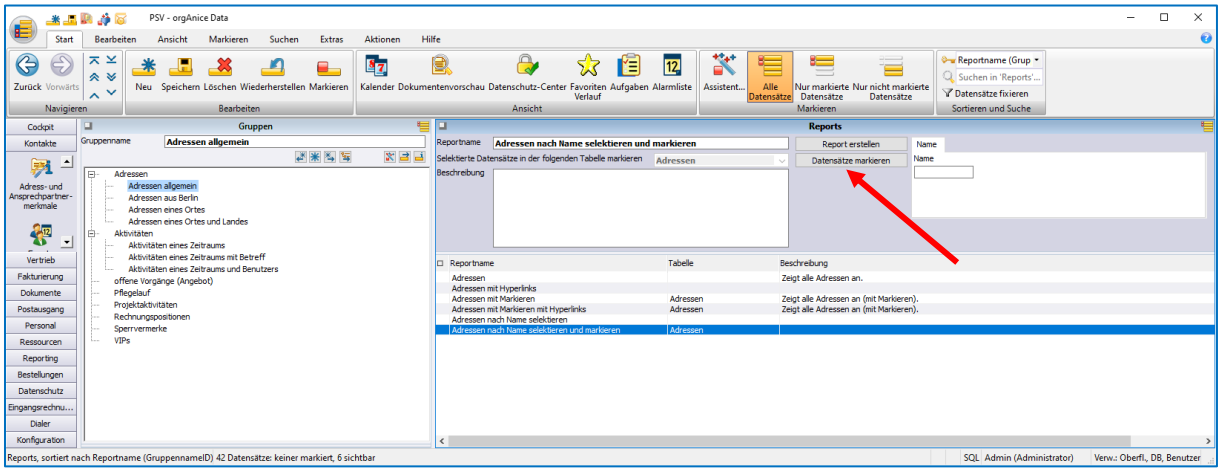


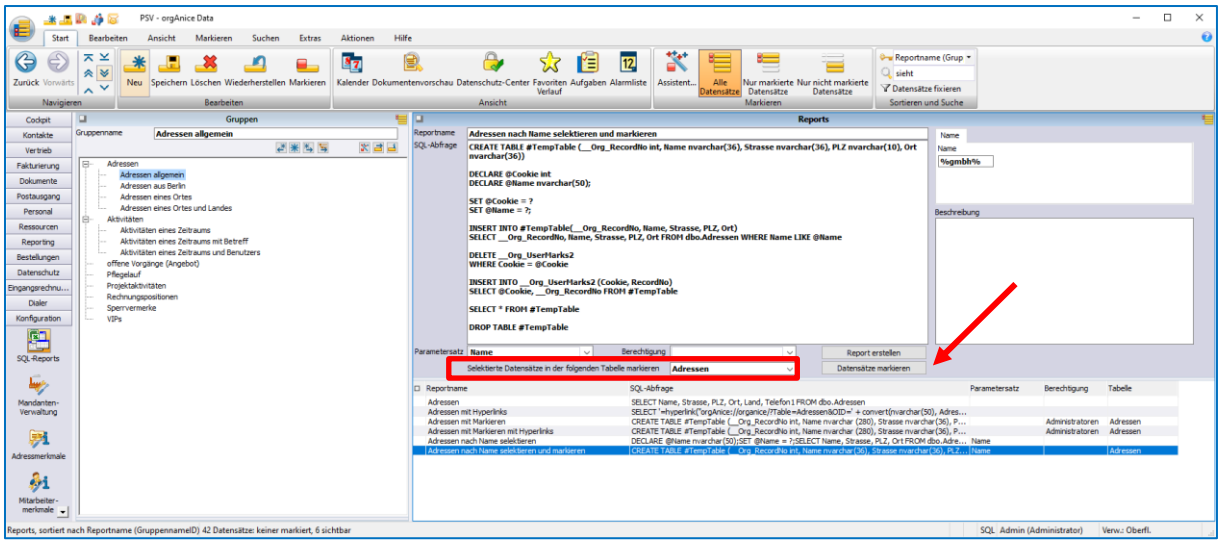
Zum Vergleich hier noch eine Erweiterung des obigen Beispiels, welches die Adressen eines Ortes selektiert. Beachten Sie, dass sich nur die gelb markierten Stellen von der vorigen SQL-Abfrage unterschieden.

```
CREATE TABLE #TempTable (__Org_RecordNo int, Name nvarchar(36), Strasse
nvarchar(36), PLZ nvarchar(10), Ort nvarchar(36))

DECLARE @Cookie int
DECLARE @Ort nvarchar(50)
DECLARE @Land nvarchar(2);

SET @Cookie = ?
SET @Ort = ?
SET @Land = ?;

INSERT INTO #TempTable(__Org_RecordNo, Name, Strasse, PLZ, Ort)
SELECT __Org_RecordNo, Name, Strasse, PLZ, Ort FROM dbo.Adressen WHERE Ort
= @Ort AND Land = @Land

DELETE __Org_UserMarks2
WHERE Cookie = @Cookie

INSERT INTO __Org_UserMarks2 (Cookie, RecordNo)
SELECT @Cookie, __Org_RecordNo FROM #TempTable

SELECT * FROM #TempTable

DROP TABLE #TempTable
```

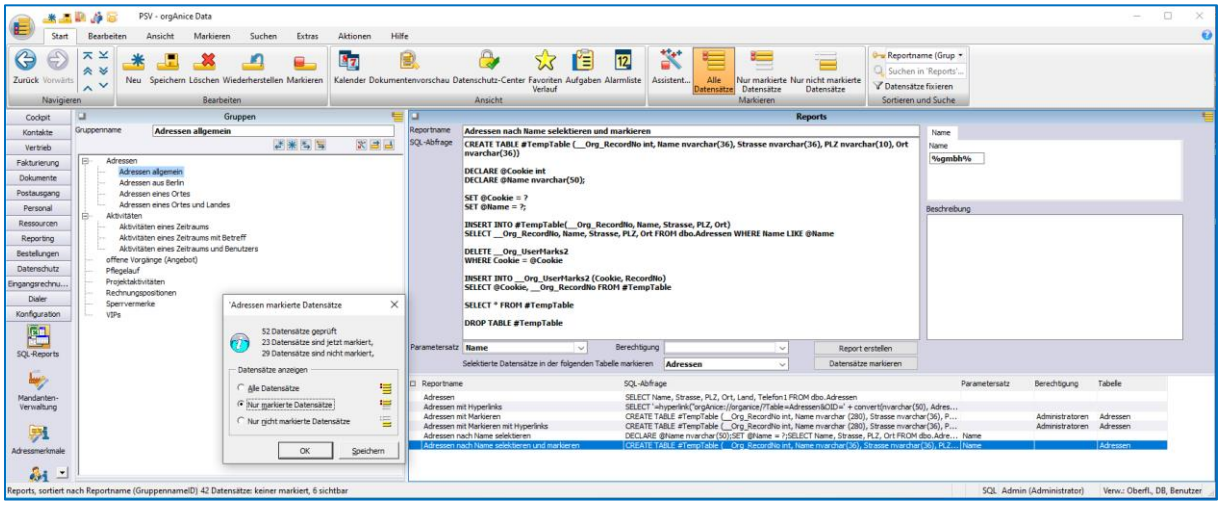
3.3.2 Datensätze markieren

Um vorab die Korrektheit Ihrer Abfrage schnell sicherzustellen, bietet der SQL Reportgenerator die Möglichkeit, Datensätze entsprechend Ihrer Abfrage und der eingestellten Tabelle in der orgAnice Datenbank zu markieren. Dies erleichtert die Kontrolle ohne Neuerstellung eines Reports. Beachten Sie, dass diese Funktionalität nur zur Verfügung steht, wenn Sie eine Tabelle zur Markierung ausgewählt haben.

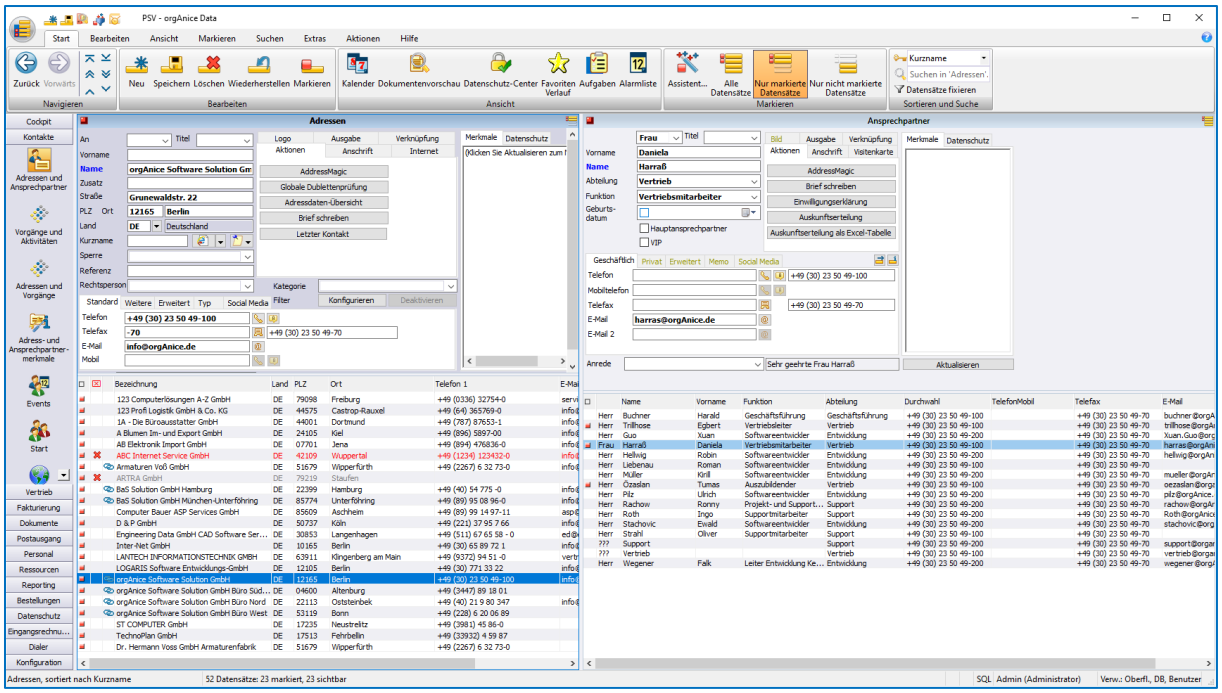




Um die Datensätze zu markieren, klicken Sie auf die Schaltfläche: „Datensätze markieren“. Es öffnet sich daraufhin die bekannte Dialogbox: „Markierte Datensätze“.



Nun können Sie sich die Markierungen anzeigen lassen, wie Sie es bisher aus orgAnice gewohnt sind.



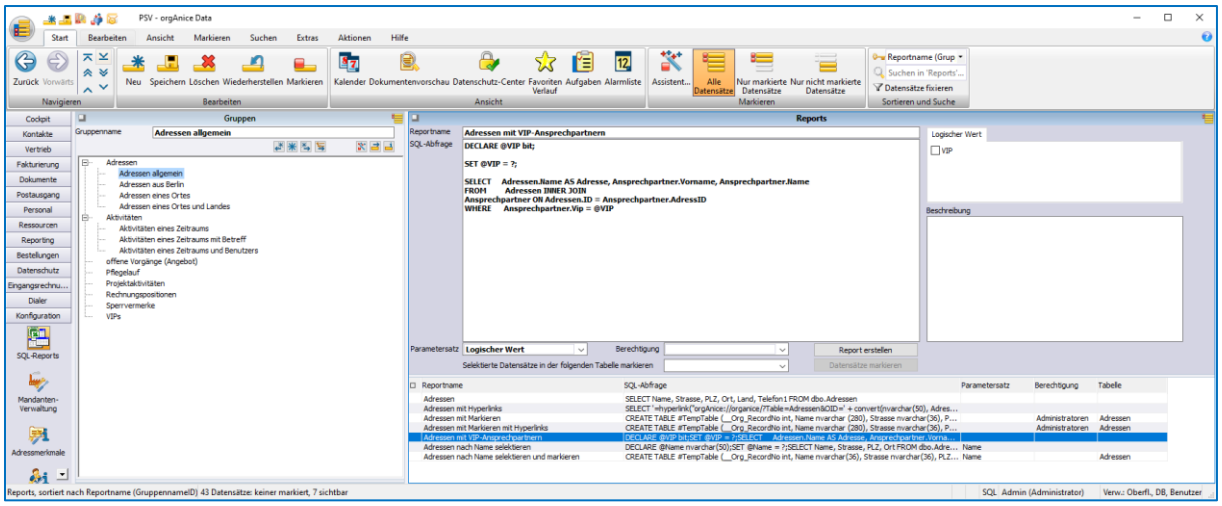
3.4 Reports mit Abfragen über mehrere Tabellen

Möchten Sie in Ihrer Abfrage Daten aus zwei (oder mehr) Tabellen verwenden, zwischen denen innerhalb orgAnice eine Relation definiert ist, so müssen Sie die Relation innerhalb der Abfrage mit dem Schlüsselwort JOIN nachbilden. Denn die orgAnice-Relationen sind innerhalb des SQL Servers nicht bekannt und müssen bei jeder Abfrage explizit angegeben werden.

Beispiel: Wir möchten alle Adressen ausgeben, die mindestens einen VIP-Ansprechpartner besitzen. Dabei greifen wir sowohl auf die Tabelle „Adressen“ als auch auf die Tabelle „Ansprechpartner“ zu.

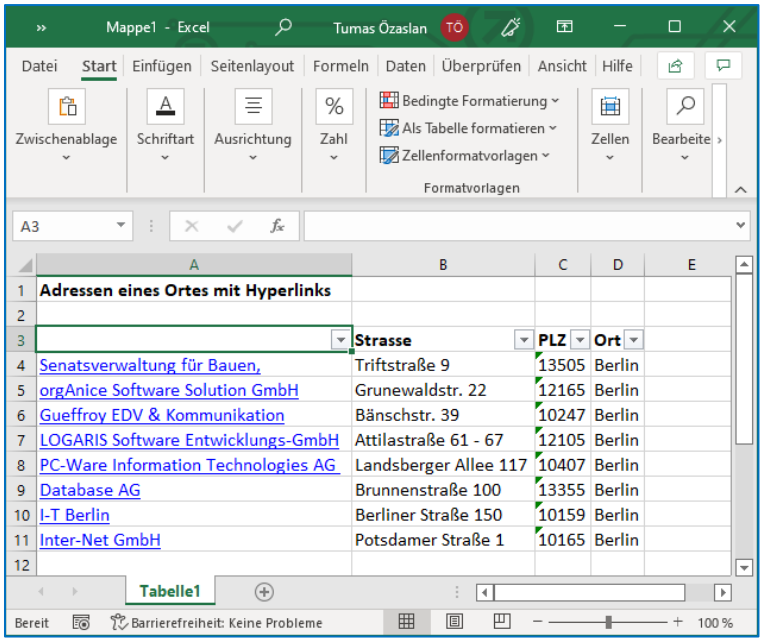
Die entsprechende SQL-Abfrage lautet:

```
DECLARE @VIP bit;

SET @VIP = ?;

SELECT      Adressen.Name AS Adresse, Ansprechpartner.Vorname,
Ansprechpartner.Name
FROM          Adressen INNER JOIN
Ansprechpartner ON Adressen.ID = Ansprechpartner.AdressID
WHERE       Ansprechpartner.Vip = @VIP
```



3.5 Reports mit Hyperlinks

Ab der Version 1.5 des SQL-Reportgenerators und orgAnice 7.1 ist es möglich, so genannte Hyperlinks zu integrieren. Dadurch sind einzelne Spalten „klickbar“:

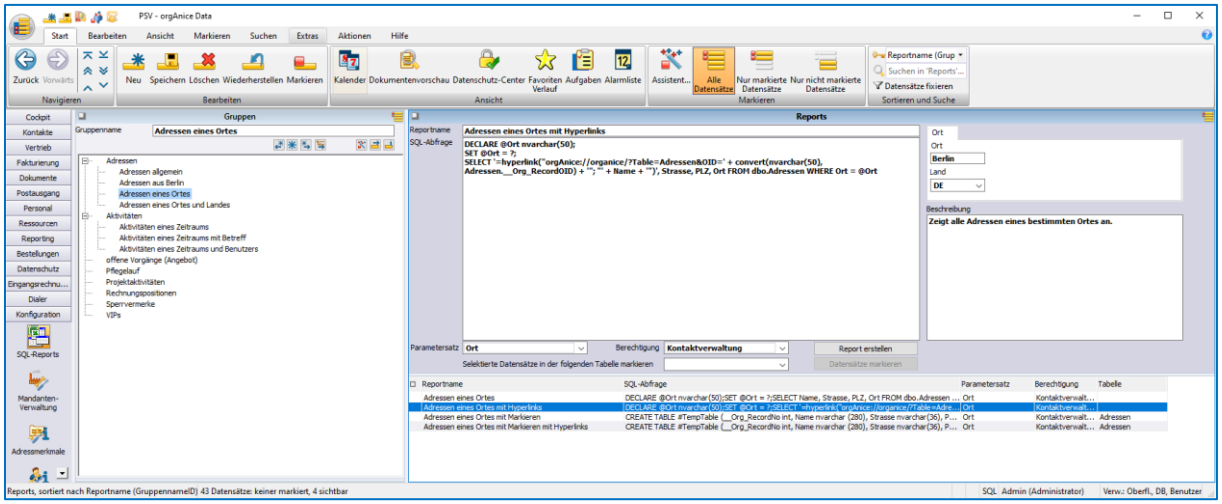


Umschließen Sie dazu die gewünschte Spalte mit dem orgAnice-Hyperlink. Ein Beispiel:

```
SELECT    '=hyperlink("orgAnice://organice/?Table=Adressen&OID=' + convert(nvarchar(50), Adressen.__Org_RecordOID) + '"; "' + Name + '")', Strasse, PLZ, Ort, Land, Telefon1
FROM    Adressen
```

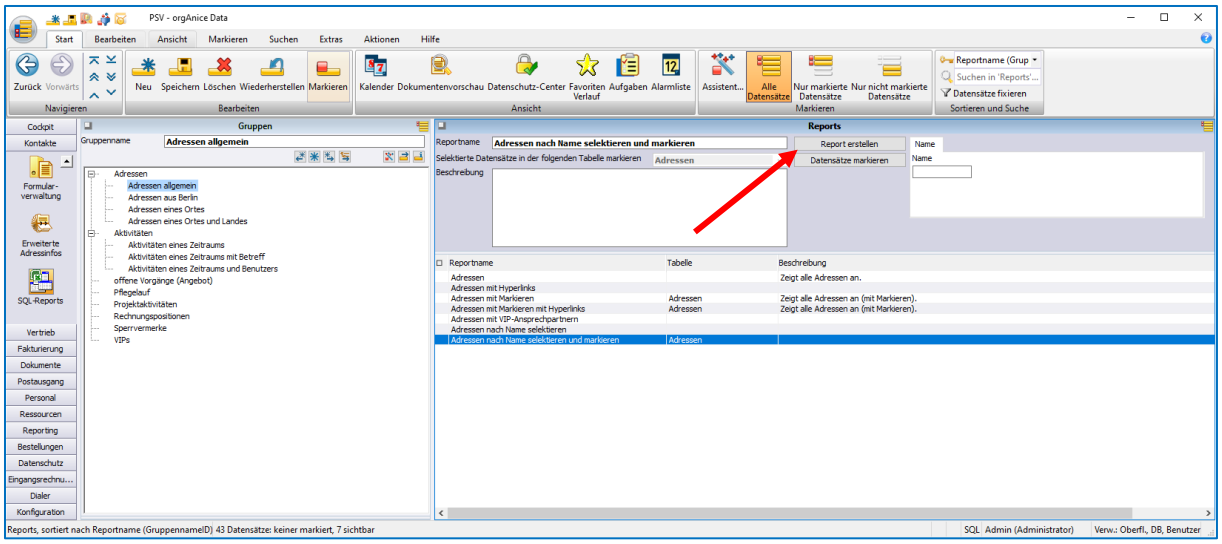
	Erklärung
'=hyperlink("orgAnice://organice/?.....	Dies ist der orgAnice Hyperlink eingebunden in die Excel-Formel „hyperlink()“.
Adressen	Gewünschte Tabelle
Name	Gewünschtes Feld für den anzuzeigenden Text des Hyperlinks
Strasse, PLZ, Ort, Land, Telefon1	„Restliche“ Felder

Beispiele dazu finden Sie auch in den mitgelieferten Demo-Datensätzen.



4 Verwendung der Reports

Die in der Arbeitsbereichsgruppe vordefinierten Reports können durch Nicht-Administratoren-Benutzer in der Arbeitsbereichsgruppe aufgerufen werden. Hier stehen keine Konfigurationsmöglichkeiten zur Verfügung, die Benutzer können nur die Parameter eingeben:



Der Report wird mit Hilfe der Schaltfläche „Report erstellen“ aufgerufen.