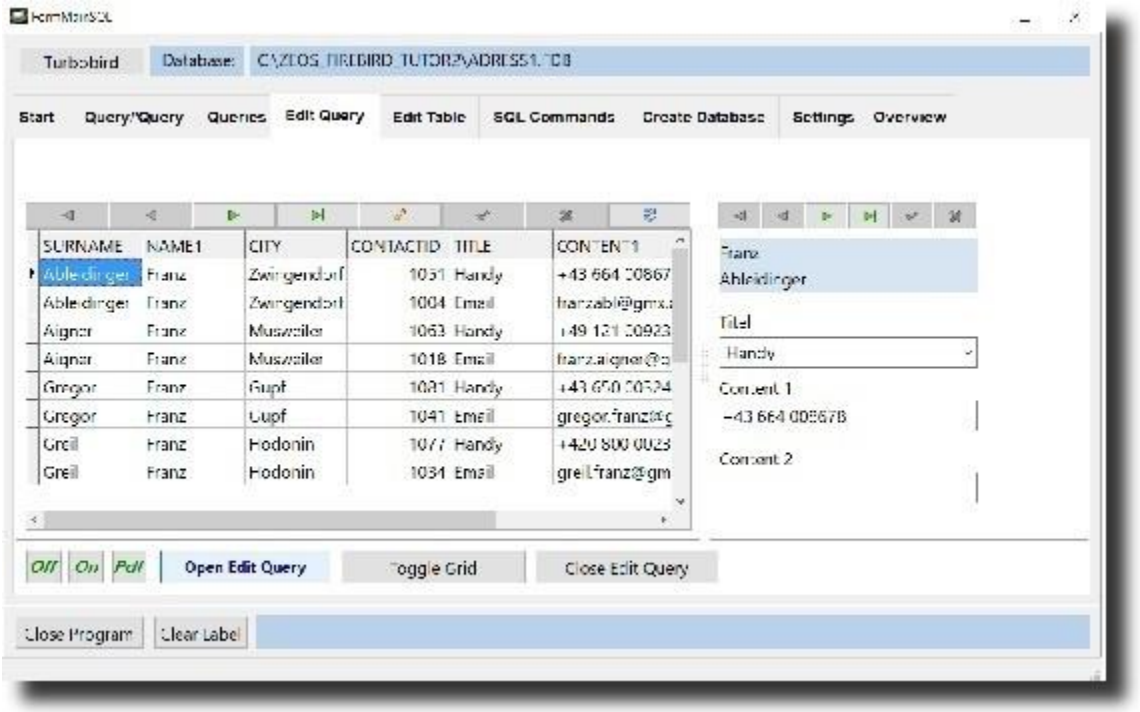
Edit Query



Bei "Edit Query" geht es darum, wie man mittels TZUpdateSQL eine nicht editierbare Datenbankabfrage ganz oder in Teilen editierbar machen kann. Das dahinterliegende Prinzip ist dasselbe wie bei QueryQuery, aber hier findet die Verlinkung zu der Datenbanktabelle, bei welcher die Daten verändert werden sollen, mithilfe der Komponente TZUpdateSQL statt. Die BDE von Delphi hat übrigens eine ähnliche Komponente.

Bei TZUpdateSQL ist es ebenso möglich wie bei TZQuery mehrere SQL Anweisungen abzuarbeiten. In unserer Abfrage, welche über ADRESS und CONTACT geht werden wir mit TZUpdateSQL die Änderung der Kontaktdaten CONTACT ermöglichen.

Die Daten von kombinierten Abfragen sind deshalb nicht editierbar, weil der Datenbankserver nicht weiß, welchen Tabellen er die geänderten Feldinhalte zuweisen soll. Dies wird durch TZUpdateSQL behoben, indem es der Datenbank mitteilt in welche Tabelle das entsprechende Feld zurückgeschrieben werden muss. Es kann durchaus möglich sein, dass es Situationen gibt, wo TZUpdateSQL von Vorteil ist. In der Regel findet man mit einer weiteren Query, die nur den zu ändernden Datensatz einliest und wieder zurückschreibt auch das Auslangen. Der Vorteil von TZUpdateSQL ist, dass ein kleiner SQL Generator mitgeliefert wird, mit welchem man die SQL Befehle für das Löschen, Einfügen und Ändern generieren kann. Aber jetzt zur Praxis.

Zuerst ziehen wir ein TZUpdateSQL Komponente auf unser Datenbankformular DataDMTutor2. Die Anleitung hier ist fiktiv, da alles schon im Tutorial vorhanden und erledigt worden ist. Aber sie ist gut zu gebrauchen, um sie analog für ein eigenes Programm zu verwenden. Als Nächstes müssen wir unserer ZQuery Komponente mitteilen, dass wir für sie eine ZUpdateSQL zur Verfügung stellen wollen. Jede ZQuery Komponente hat als vorletzte Property im Objektinspektor UpdateObject. Wenn wir unser ZUpdateSQL bereits erstellt haben, können wir es dort auswählen. In unserem Fall passiert das in ZQueryUpdate und der Name der ZUpdateSQL Komponente ist ZUpdateSQLUpdate. Jetzt sollten wir noch in ZConnection1.Connected auf True stellen und ZQueryUpdate.Active auf True, damit die Datenbankfelder zum Generieren der nötigen Argumente und SQL Befehle in ZUpdateSQLUpdate zur Verfügung stehen. Es muss in ZQueryUpdate.SQL ein entsprechender SQL Befehl stehen, sonst kommt eine Fehlermeldung. Im Tutorial ist das schon erledigt. Wenn alles geklappt hat sieht man jetzt im DBGrid die Daten der Abfrage.

Nun klicken wir auf ZUpdateSQLUpdate. Im Objektinspektor sind in den ersten drei Einträgen die SQL Befehle zum Löschen (DeleteSQL), Einfügen (InsertSQL) und Ändern (ModifySQL) die entsprechenden SQL Befehle für diese Aktionen beinhaltet. In Params sind die notwendigen Parameter zur Übergabe an die SQL Befehle enthalten.

In DeleteSQL, der SQL Befehl zum Löschen:

```
DELETE FROM CONTACT WHERE
  CONTACT.CONTACTID = :OLD_CONTACTID
```

***U**mgangssprachlich:* Lösche einen Datensatz aus der Tabelle CONTACT wobei der verwendete Schlüssel den Wert aus Params mit OLD_CONTACTID den Wert hat als die Daten am Anfang abgerufen wurden. Wenn CONTACTID geändert worden ist, aber noch nicht geschrieben und mit diesem WERT der Löschbefehl durchgeführt wird der falsche Datensatz gelöscht.

In InsertSQL, der SQL Befehl zum Einfügen:

```
INSERT INTO CONTACT
  (TITLE, CONTENT1, CONTENT2)
  VALUES   (:TITLE, :CONTENT1, :CONTENT2)
```

***U**mgangssprachlich:* Füge einen Datensatz in der Tabelle CONTACT ein und verwende dazu die Werte, die in den Parametern (aus Params) übergeben worden sind.

In ModifySQL, der SQL Befehl zum Ändern:

```
UPDATE CONTACT SET   TITLE = :TITLE,
   CONTENT1 = :CONTENT1,
   CONTENT2 = :CONTENT2
 WHERE
   CONTACT.CONTACTID = :OLD_CONTACTID
```

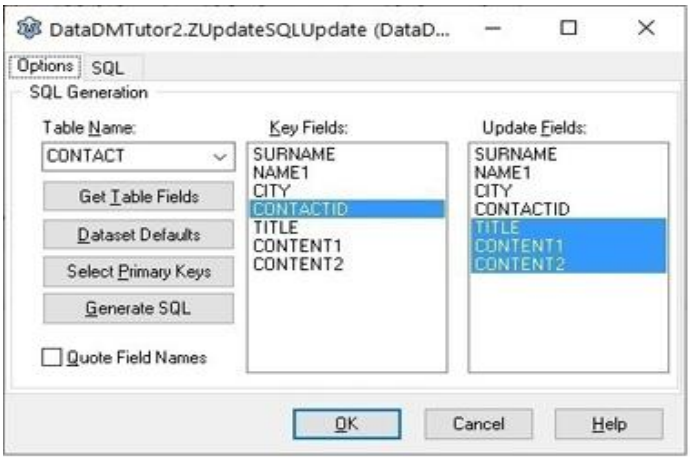
***U**mgangssprachlich:* Ändere die Daten im aktuellen Datensatz, wobei die aktuellen (geänderten) Daten aus Params übergeben werden sowie die ID mit OLD_CONTACTID damit durch Änderungen keine Fehler entstehen können. Mehr dazu siehe DeletSQL.

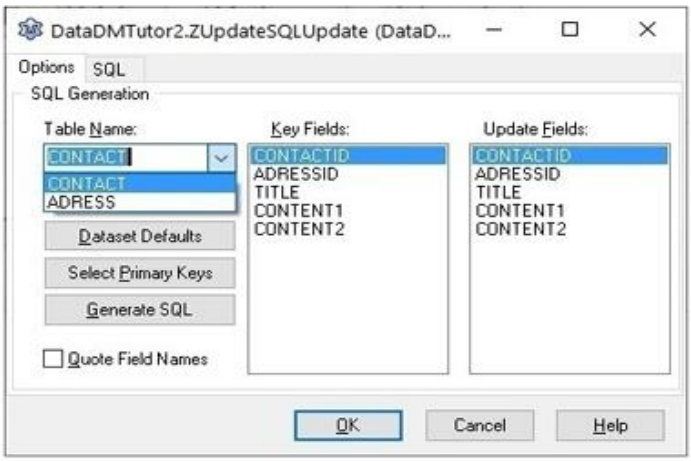
Und in Params die benötigten Parameter, welche an die SQL Befehle übergeben werden:

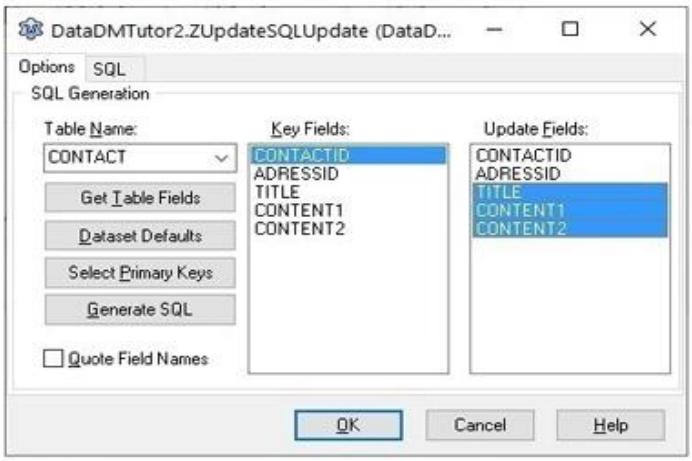
```
TITLE
CONTENT1
CONTENT2
OLD_CONTACTID
```

Jetzt einmal, was passiert da eigentlich? Wir haben oben gehört, dass in einer zusammengesetzten Datenmenge der Datenbankserver den Bezug zur Herkunft der Feldwerte verliert. Wir wissen aber diesen Bezug und können auf dem Weg von ZUpdateSQL dem Datenbankserver über die oben angeführten SQL Befehle zum Löschen, Einfügen und Ändern beauftragen. Das ist jetzt die Ähnlichkeit der Lösung mit verlinkten zusätzlichen ZQueries.

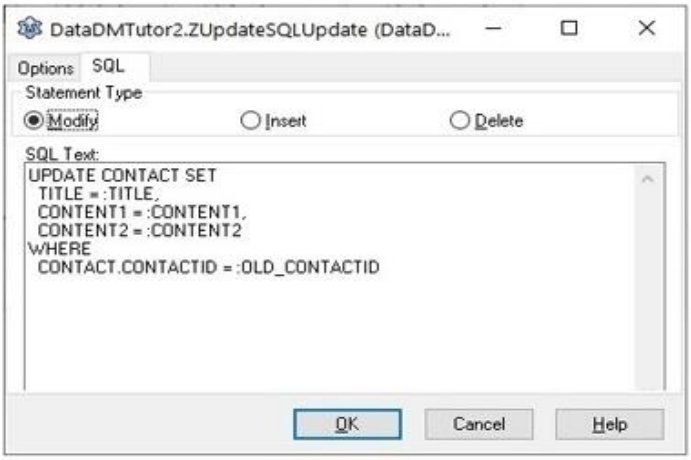
Die Parameter TITLE bis OLD_CONTACTID werden verwendet, um die aktuellen Daten an die SQL Befehle zu übergeben, beinhalten also die aktuellen Daten, mit Ausnahme von OLD_CONTACTID. OLD_CONTACTID beinhaltet den Wert von CONTACTID bevor irgend eine der drei Möglichkeiten mit den SQL Befehlen aufgerufen wurde. Das ist unser "Bindeglied" das uns ermöglicht, den aktuellen Datensatz zu identifizieren.

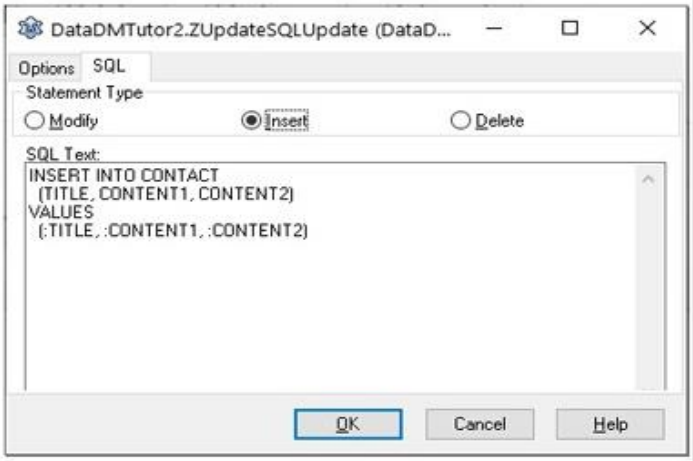

DataDMTutor2.ZUpdateSQLUpdate (DataD...
Options SQL
SQL Generation
Table Name:
CONTACT
Get Table Fields
Dataset Defaults
Select Primary Keys
Generate SQL
Quote Field Names
Key Fields:
SURNAME
NAME1
CITY
CONTACTID
TITLE
CONTENT1
CONTENT2
Update Fields:
SURNAME
NAME1
CITY
CONTACTID
TITLE
CONTENT1
CONTENT2
OK
Cancel
Help

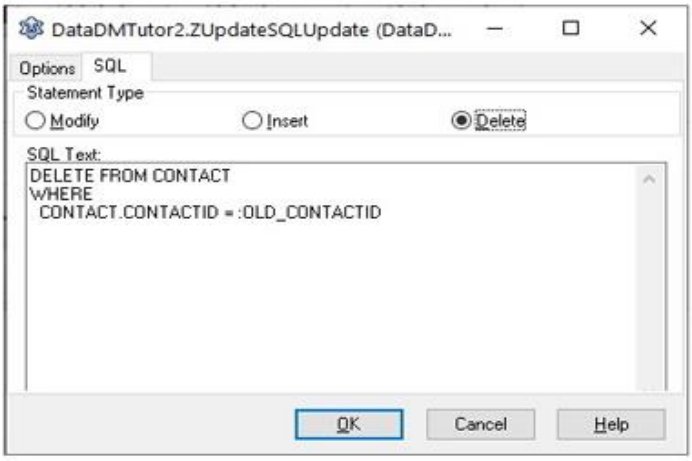

DataDMTutor2.ZUpdateSQLUpdate (DataD...
Options SQL
SQL Generation
Table Name:
CONTACT
CONTACT
ADRESS
Get Table Fields
Dataset Defaults
Select Primary Keys
Generate SQL
Quote Field Names
Key Fields:
CONTACTID
ADRESSID
TITLE
CONTENT1
CONTENT2
Update Fields:
CONTACTID
ADRESSID
TITLE
CONTENT1
CONTENT2
OK
Cancel
Help

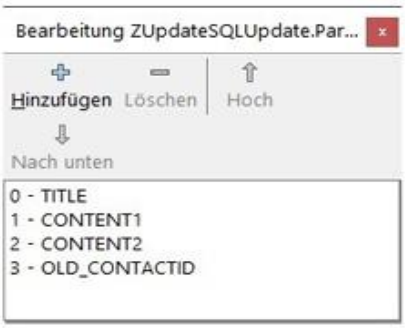

DataDMTutor2.ZUpdateSQLUpdate (DataD...
Options SQL
SQL Generation
Table Name:
CONTACT
Get Table Fields
Dataset Defaults
Select Primary Keys
Generate SQL
Quote Field Names
Key Fields:
CONTACTID
ADRESSID
TITLE
CONTENT1
CONTENT2
Update Fields:
CONTACTID
ADRESSID
TITLE
CONTENT1
CONTENT2
OK
Cancel
Help


DataDMTutor2.ZUpdateSQLUpdate (DataD...
Options SQL
SQL Generation
Table Name:
CONTACT
Get Table Fields
Key Fields:
CONTACTID
ADRESSID
TITLE
CONTENT1
CONTENT2
Update Fields:
CONTACTID
ADRESSID
TITLE
CONTENT1
CONTENT2
Warnung
The SQL property is not empty. Do you want to clear it before the generation?
Ja
Nein


DataDMTutor2.ZUpdateSQLUpdate (DataD...
Options SQL
Statement Type
Modify
Insert
Delete
SQL Text:
UPDATE CONTACT SET
TITLE = :TITLE,
CONTENT1 = :CONTENT1,
CONTENT2 = :CONTENT2
WHERE
CONTACT.CONTACTID = :OLD_CONTACTID
OK
Cancel
Help


DataDMTutor2.ZUpdateSQLUpdate (DataD...
Options SQL
Statement Type
Modify
Insert
Delete
SQL Text:
INSERT INTO CONTACT
(TITLE, CONTENT1, CONTENT2)
VALUES
(:TITLE, :CONTENT1, :CONTENT2)
OK
Cancel
Help


DataDMTutor2.ZUpdateSQLUpdate (DataD...
Options SQL
Statement Type
Modify
Insert
Delete
SQL Text:
DELETE FROM CONTACT
WHERE
CONTACT.CONTACTID = :OLD_CONTACTID
OK
Cancel
Help


Bearbeitung ZUpdateSQLUpdate.Par...
Hinzufügen
Löschen
Hoch
Nach unten
0 - TITLE
1 - CONTENT1
2 - CONTENT2
3 - OLD_CONTACTID

Als Nächstes werden wir uns mit dem SQL Generator von ZUpdateSQL beschäftigen. Wir rufen ihn mit Doppelklick auf ZUpdateSQLUpdate auf. Falls keine Datenfeldbezeichnungen angezeigt werden klicken wir auf "Get Table Fields". Als Nächstes wählen wir bei Table Name "CONTACT" aus. In der Box "Key Fields" wählen wir "CONTACTID" aus, da wir die Daten in CONTACT ändern wollen. In der Box "Update Fields" wählen wir TITLE, CONTENT1 und CONTENT2 aus. Mit der ID erfährt die Datenbank auf welchen Datensatz in der Tabelle CONTACTID sie zugreifen muss. Als nächstes klicken wir auf "Generate SQL". Wenn schon SQL Texte vorhanden sind, wird man gefragt, ob man die Inhalte leeren möchte. Wir bestätigen mit "Ja". Es werden automatisch die SQL Befehle erstellt und in Params die benötigten Parameter erstellt. Nun sollte Änderungen, Löschen und Einfügen nichts mehr im Wege stehen.

Zusätzlich zum DBGrid stehen rechts für die Kontaktdaten noch Eingabefelder zur Verfügung, um auch diese Variante vorzustellen.

Mit dem Button "Open Edit Query" wird die Datenbankabfrage geöffnet, mit "Close Edit Query" geschlossen. "Toggle Grid" dient dazu das Datenbankgrid unsichtbar zu machen und die Datenmenge so darzustellen, wie diese normalerweise einem Anwender dargeboten würde.

<>