

## Inkonsistente Duplikate finden per Abfrage

Wenn Sie ein Datenmodell antreffen, das nicht vollständig normalisiert ist, kann es sein, dass dieses aufgrund seiner Beschaffenheit das Speichern von Duplikaten erlaubt. Das ist insbesondere interessant, wenn diese Daten in Tabellen vorliegen, die nicht der dritten Normalform entsprechen. Das bedeutet beispielsweise, dass Sie zu einer Bestellung in einer Bestellungen-Tabelle auch den Kunden inklusive Kundennummer und weiteren Kundendaten angeben. Bevor Sie eine solche Bestellungen-Tabelle in eine Bestellungen- und eine Kundentabelle aufteilen, sollten Sie sicherstellen, dass es keine Inkonsistenzen in den vermutlich bereits vorhandenen Duplikaten gibt und diese gegebenenfalls korrigieren. Um inkonsistente Daten in dieser Form zu finden, verwenden wir am einfachsten eine Abfrage. Wie Sie diese formulieren, zeigt Ihnen dieser Artikel.

### Beispieldatenbank zum Finden von inkonsistenten Daten

Die Beispiele dieses Artikels finden Sie in der Datenbank **2101\_InkonsistenteDaten.accdb**.

### Beispiel für inkonsistente Daten

Ein gutes Beispiel ist eine Tabelle mit Projekten oder Bestellungen, die direkt die jeweiligen Kundendaten speichern. Die Kundendaten landen dabei jeweils mit der Kundennummer und den übrigen Kundendaten in der Bestellungen-Tabelle.

Ein Beispiel für inkonsistente Daten finden Sie in der Tabelle aus Bild 1. Hier finden wir zwei Mal den Kunden mit dem Wert **1** im Feld **KundeID**.

Dabei taucht auch gleich eine Inkonsistenz auf: Der Firmenname des Kunden erscheint einmal mit **Minhorst** und einmal mit **Mienhorst**. Solche Inkonsistenzen können aus verschiedenen Gründen entstehen, zum Beispiel

- weil die Bestellannahme den Kundennamen einmal falsch und dann bei der nächsten Bestellung richtig eingetragen hat,
- weil der Kunde selbst sich bei einer Onlinebestellung vertippt hat oder

BestellungID	Bestelldatum	KundeID	Firma	Ansprechpartner
1	19.02.2021	1	André Minhorst Verlag	André Minhorst
2	20.02.2021	2	Klaus Müller GmbH	Klaus Müller
3	21.02.2021	3	Herbert Schmitt AG	Herbert Schmitt
4	22.02.2021	1	André Mienhorst Verlag	André Minhorst
5	24.02.2021	2	Klaus Müller GmbH	Klaus Müller
6		0	(Neu)	

Bild 1: Inkonsistente Daten

- weil sich die Adresse von einer Bestellung zur nächsten geändert hat – dann prüft und ändert man bei der Aufnahme einer neuen Bestellung eventuell nicht die zuvor verwendeten Daten.

Sie könnten einwenden, dass es ja sinnvoll ist, die Adresse des Kunden mit jeder Bestellung zu speichern, um später nachvollziehen zu können, wohin Sie die einzelnen Lieferungen geschickt haben. Das ist korrekt, aber hier wollen wir vereinfachend und zu Beispielszwecken davon ausgehen, dass die Adressen gleich bleiben und Unterschiede durch Tippfehler entstanden sind.

### Inkonsistente Daten per Assistent finden

Neben dem Assistenten für die Duplikatsuche, den wir im Artikel **Duplikate finden per Abfrage** ([www.access-basics.de/516](http://www.access-basics.de/516)) vorgestellt haben, gibt es auch noch einen Assistenten für die Inkonsistenzsuche.

Diesen finden Sie, wenn Sie im Ribbon auf den Befehl **Erstellen|Abfragen|Abfrage-Assistent** klicken und in dem dann erscheinenden Dialog **Neue Abfrage** den Eintrag **Abfrage-Assistent zur Inkonsistenzsuche** auswählen (siehe Bild 2).

Bereits der Beschreibungstext des Assistenten in diesem Dialog gibt uns jedoch einen Hinweis, dass der Assistent nicht das tut, was wir gern hätten: Er dient nämlich dazu, Inkonsistenzen bei verknüpften Datensätzen aufzuspüren. Wir wollen aber Inkonsistenzen innerhalb einer einzigen Tabelle identifizieren. Daher verschieben wir die Beschreibung dieses sehr nützlichen Assistenten in einen anderen Artikel namens **Inkonsistente Verknüpfungen finden** ([www.access-basics.de/518](http://www.access-basics.de/518)).

### Abfrage zum Suchen von Inkonsistenzen innerhalb einer Tabelle auffinden

Da wir die Abfrage nun selbst entwerfen müssen, wollen wir zunächst einmal definieren, welche Datensätze wir finden wollen. Wir gehen also davon aus, dass es in unserer Beispieltabelle **tblBestellungen** ein sogenanntes Nichtschlüssel­feld gibt, hier **KundeID**, sowie einige weitere Felder, die von diesem Nichtschlüssel­feld abhängig sind – in unserem Fall **Firma** und **Ansprechpartner**. In einem ausführlicheren Beispiel könnten auch noch die Adresse und weitere Daten dazugehören.

Grundsätzlich möchten wir alle Datensätze herausfinden, die zwar den gleichen Wert im Feld **KundeID** aufweisen, aber deren davon abhängige Felder **Firma** oder **Ansprechpartner** nicht übereinstimmen. »Oder« deshalb, weil es laut unseren Anforderungen für eine Inkonsistenz reicht, dass nur eines der von **KundeID**

abhängigen Felder nicht in allen Datensätzen mit dieser Kundennummer gleich ist.

Wir brauchen also erst einmal nur solche Datensätze zu untersuchen, von denen es mindestens zwei mit dem gleichen Wert im Feld **KundeID** gibt.

An dieser Stelle können wir den **Abfrage-Assistent zur Duplikatsuche** nutzen und diesem den Auftrag geben, eine Abfrage zu erstellen, die alle Werte im Feld **KundeID** ausgibt, die mehrmals auftreten. Dieser

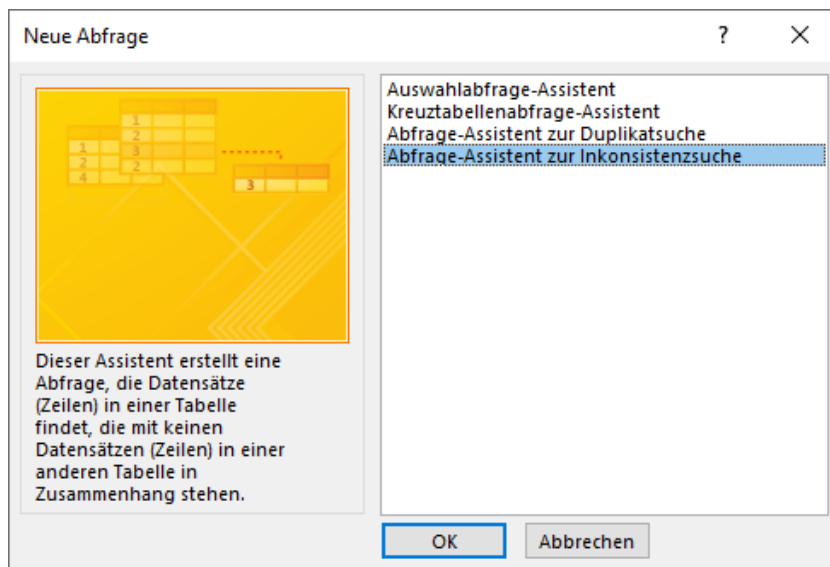


Bild 2: Abfrage-Assistent zur Inkonsistenzsuche

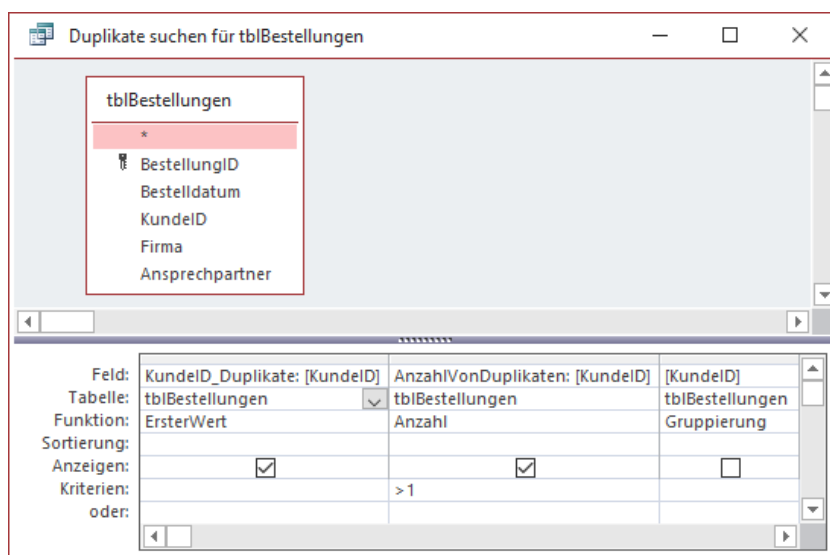


Bild 3: Abfrage zum Finden von doppelten Werten im Feld KundeID

liefert uns eine Abfrage, die im Entwurf wie in Bild 3 aussieht.

Diese ermittelt Datensätze, die erstens den Wert des Feldes **KundeID** enthält und zweitens die Anzahl der Datensätze, die den gleichen Wert im Feld **KundeID** enthalten.

KundeID_Duplikate	AnzahlVonDuplikaten
1	2
2	2

Bild 4: Doppelte Werte im Feld **KundeID**

Hier haben wir zwei Kunden beziehungsweise Werte für das Feld **KundeID** gefunden, die mehr als einmal vorkommen. Die Abfrage speichern wir unter dem Namen **qryKundenMitMehrAlsEinerBestellung** (siehe Bild 4). Schauen wir noch einmal in die Tabelle **tblBestellungen**, sehen wir, dass der Kunde mit dem Wert 1 im Feld **KundeID** doppelt vorkommt und eine Inkonsistenz im Feld **Firma** enthält. Der Kunde mit dem Wert 2 im Feld **KundeID** kommt auch zwei Mal vor, aber ohne Inkonsistenzen. Das heißt, die den Kunden betreffenden Daten sind in den beiden Datensätzen identisch (siehe Bild 5).

BestellungID	Bestelldatum	KundeID	Firma	Ansprechpartner
1	19.02.2021	1	André Minhorst Verlag	André Minhorst
2	20.02.2021	2	Klaus Müller GmbH	Klaus Müller
3	21.02.2021	3	Herbert Schmitt AG	Herbert Schmitt
4	22.02.2021	1	André Mienhorst Verlag	André Minhorst
5	24.02.2021	2	Klaus Müller GmbH	Klaus Müller
(Neu)		0		

Bild 5: Zu untersuchende Datensätze

Damit haben wir aber nur den ersten Teil des gesuchten Ergebnisses. Für den zweiten Teil gilt es herauszufinden, für welche der mit der ersten Abfrage gefundenen Datensätze wir mehr als einen unterschiedlichen Datensatz finden.

Das gehen wir in Zwischenschritten an:

- Im ersten Zwischenschritt erstellen wir eine Abfrage, die untersucht, welche Varianten von jedem der gefundenen Datensätze es gibt.
- Im zweiten Schritt finden wir aus dieser Abfrage die Datensätze für die Werte von **KundeID**, die mehr als einmal vorkommen.

### Verschiedene Varianten für Duplikate suchen

Der erste Schritt ist die Abfrage **qryEindeutige-KundenAusDuplikaten**, deren Entwurf Sie in Bild 6 finden.

Dieser Abfrage fügen Sie als Erstes die Tabelle **tblBestellungen** hinzu. Danach ziehen Sie auch noch die soeben erstellte Abfrage **qryKundenMitMehrAlsEinerBestellung** hinzu. Die beiden Datenquellen verbinden Sie über das Feld **KundeID**. Dazu ziehen Sie beispielsweise das Feld **KundeID** aus der Tabelle **tblBestellungen** auf das Feld **KundeID\_Duplikate** der Abfrage **qryKundenMitMehrAlsEinerBestellung**.

Dadurch sorgen wir dafür, dass nur Datensätze mit solchen Werten im Feld **KundeID** aus der Tabelle **tblBestellungen** geliefert werden, die wir mit der Abfrage **qryKundenMitMehrAlsEinerBestellung** bereits als Duplikate identifiziert haben.

Wenn wir nun in die Datenblattansicht wechseln, erhalten wir erst einmal alle Datensätze, deren Wert im Feld **KundeID** mehr als einmal in der Tabelle **tblBestellungen** vorkommt (siehe Bild 7). Hier taucht korrekterweise auch der Datensatz mit dem Wert 2 im Feld **KundeID** zwei Mal auf.

Damit dieser nur einmal auftaucht, stellen wir anschließend die Eigenschaft **Keine Duplikate** auf **Ja**.

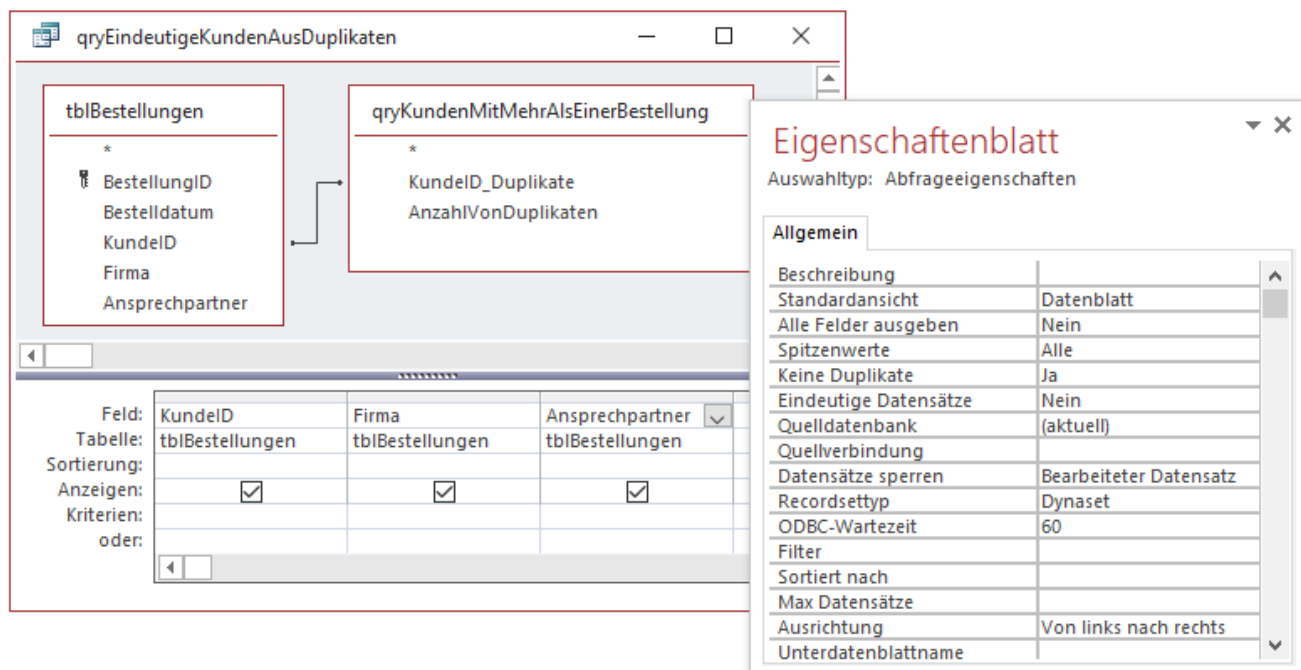


Bild 6: Eindeutige Kunden aus den Duplikaten finden

ein. Damit deaktivieren wir die mehrfache Anzeige von Datensätzen, die in allen im Entwurfsraster enthaltenen Feldern identisch sind.

Von den Datensätzen mit dem Wert 2 im Feld **KundeID** wird demzufolge nun nur noch ein Exemplar angezeigt (siehe Bild 8), während die beiden verschiedenen Version des Kunden mit dem Wert 1 im Feld **KundeID** noch erscheinen.

Nun folgt noch der dritte Schritt: Wir müssen diejenigen Datensätze aus dem Ergebnis entfernen, die nur eindeutige Daten enthalten. In diesem Fall handelt es sich um den Datensatz mit dem Wert 2 im Feld **KundeID**. Übrig bleiben sollen also nur die beiden Datensätze, in denen das Feld **KundeID** den Wert 1 aufweist.

### KundeID für mehrfach vorkommende, inkonsistente Datensätze ermitteln

Das erledigen wir in einer weiteren Abfrage, die diesmal nur auf der Abfrage **qryEindeutigeKundenAusDuplikaten** aufsetzt und aus dieser nur die Werte für das Feld **KundeID** ermitteln soll, die in der Abfra-

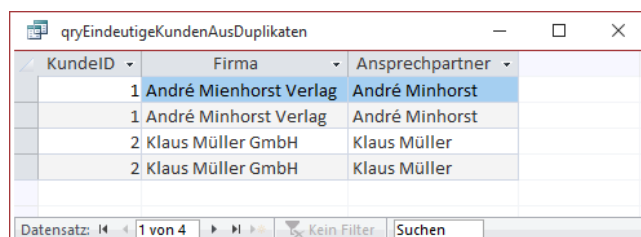


Bild 7: Erstes Ergebnis von qryEindeutigeKundenAusDuplikaten

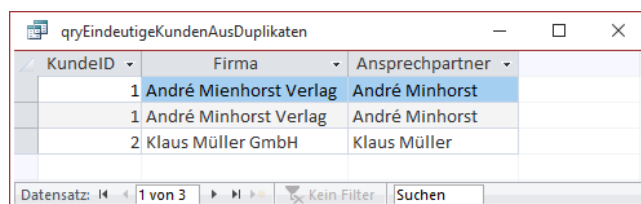


Bild 8: qryEindeutigeKundenAusDuplikaten nur mit eindeutigen Datensätzen

ge **qryEindeutigeKundenAusDuplikaten** mehrfach vorkommen.

Also erstellen wir eine Abfrage, der wir nur die Abfrage **qryEindeutigeKundenAusDuplikaten** hinzufügen. Aus dieser ziehen wir zwei Mal das

Feld **KundeID** in das Entwurfsgitter. Mit einem Klick auf den Ribbon-Befehl **Entwurf|Einblenden/Ausblenden|Summen** blenden wir die Zeile **Funktion** im Entwurfsgitter der Abfrage ein. Dann stellen wir für das erste Feld in der Zeile **Funktion** den Wert **Gruppierung** ein und für das zweite Feld den Wert **Anzahl**. Für das zweite Feld legen wir außerdem als Kriterium den Wert **>1** fest.

Damit erreichen wir, dass die Abfrage nach dem Wert **KundeID** gruppiert und in der zweiten Spalte die Anzahl der Datensätze je Gruppe ermittelt. Diese filtern wir so, dass nur diejenigen **KundeID**-Werte im Ergebnis erscheinen, die mehr als einmal in der Abfrage **qryEindeutigeKundenAusDuplikaten** vorkommen (siehe Bild 9).

Nachdem wir die neue Abfrage unter dem Namen **qryEindeutigeKundenAusDuplikatenMehrereJeKundeID** gespeichert haben, zeigen wir diese in der Datenblattansicht an. Als Ergebnis erhalten wir, dass der Kunde mit dem Wert **1** im Feld **KundeID** der einzige ist, für den mehrere Datensätze mit inkonsistenten Daten vorliegen (siehe Bild 10).

Wenn wir nun alle inkonsistenten Datensätze anzeigen wollen, brauchen wir nur noch eine letzte Abfrage, welche die Abfrage **qryEindeutigeKundenAusDuplikatenMehrereJeKundeID** nutzt.

### Ausgeben der Datensätze mit inkonsistenten Kundendaten

Im letzten Schritt erstellen wir also eine weitere Abfrage. Dieser fügen wir als Datenquelle die Tabelle **tblBestellungen** und die Abfrage **qryEindeutigeKundenAusDuplikatenMehrereJeKundeID** hinzu. Dann verknüpfen wir wieder das Feld **KundeID** der Tabelle **tblBestellungen** mit dem der Abfrage **qryEindeuti-**

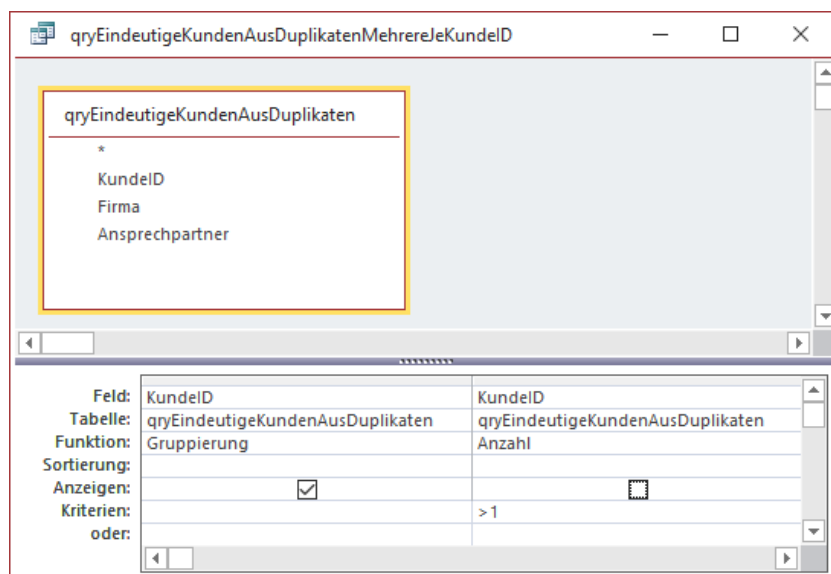


Bild 9: Alle Datensätze aus **qryEindeutigeKundenAusDuplikaten**, für die es mehrere Exemplare je **KundeID** gibt

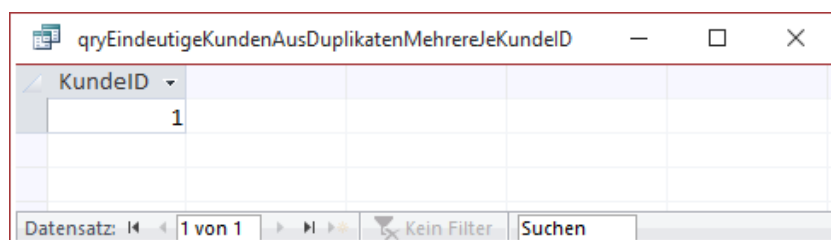


Bild 10: Ausgabe aller Werte des Feldes **KundeID**, für die es inkonsistente Datensätze gibt.

**geKundenAusDuplikatenMehrereJeKundeID**. Dazu ziehen Sie das Feld **KundeID** von der einen auf das gleichnamige Feld der anderen Datenquelle.

Anschließend ziehen Sie alle Felder der Tabelle **tblBestellungen** in das Entwurfsgitter der Abfrage. Damit ist die Abfrage bereits fertiggestellt – die Verknüpfung der beiden **KundeID**-Felder ist quasi das Kriterium, das dafür sorgt, dass nur die Datensätze der Tabelle **tblBestellungen** angezeigt werden, die inkonsistente Datensätze in den Feldern **KundeID**, **Firma** und **Ansprechpartner** enthalten (siehe Bild 11).

Diese Abfrage speichern wir unter dem Namen **qry-InkonsistenteKundenInBestellungen** und wechseln dann in die Datenblattansicht. Das Ergebnis zeigt genau die beiden Datensätze mit dem Wert **1** im Feld

KundeID an, die inkonsistente Daten enthalten (siehe Bild 12).

### Mehrere inkonsistente Datensätze

Wenn die Tabelle **tblBestellungen** mehrere Datensätze mit dem gleichen Wert im Feld **KundeID** enthält, von denen nur einer von den anderen abweicht oder von denen auch alle Unterschiede in den Feldern **Firma** oder **Ansprechpartner** aufweisen, werden immer alle betroffenen Datensätze angezeigt.

Nur so kann der Benutzer entscheiden, welche der Datensätze angepasst werden sollen und welcher der Datensatz ist, dessen Daten übernommen werden sollen.

### Zusammenfassung und Ausblick

Dieser Artikel zeigt, wie man Inkonsistenzen aufdeckt und diese mit verschiedenen, aufeinander aufbauenden Abfragen anzeigen kann.

Damit ist die Basis gelegt, mit welcher der Benutzer die inkonsistenten Daten ermitteln kann. Damit ist es wesentlich einfacher also zuvor, die doppelten und inkonsistenten Daten anzuzeigen und diese dann zu bearbeiten.

Dadurch, dass die Abfrage **qryInkonsistenteKundenInBestellungen** nur Felder aus der Tabelle **tblBestellungen** anzeigt, ist diese sogar aktualisierbar. Das heißt, der Benutzer kann die Inkonsistenzen direkt in dieser Abfrage korrigieren. Hat er die Kundendaten

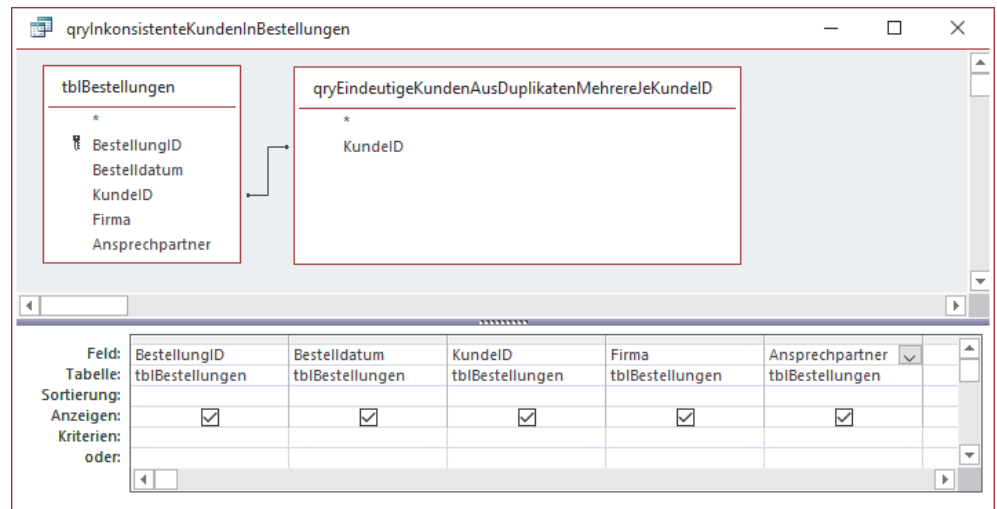


Bild 11: Abfrage zur Ermittlung aller Bestellungen, deren Kundendaten inkonsistent sind

BestellungID	Bestelldatum	KundeID	Firma	Ansprechpartner
4	22.02.2021	1	André Mienhorst Verlag	André Minhorst
1	19.02.2021	1	André Minhorst Verlag	André Minhorst

Bild 12: Ergebnis mit den inkonsistenten Daten

eines Datensatzes an einen anderen angeglichen und existieren somit keine Inkonsistenzen mehr für diesen Datensatz, kann er die noch angezeigten Daten mit diesem Wert im Feld **KundeID** durch Aktualisieren der Abfrage mit **F5** ausblenden.

Spannend wäre es noch, ein Formular zu entwickeln, mit dem der Benutzer nur noch den Datensatz für eine Menge von inkonsistenten Datensätzen auswählen müsste und dann per Mausklick die Kundendaten dieses Datensatzes auf die übrigen Datensätze mit dem gleichen Wert im Feld **KundeID** überträgt.

Das Beheben von Inkonsistenzen in einem Fall wie dem aus unserem Beispiel ist allerdings nur ein Zwischenschritt auf dem Weg zur Normalisierung des Datenmodells und hier speziell zum Überführen des Datenmodells in die dritte Normalform. Wie das gelingt, erfahren Sie im Artikel **Normalisierung, Teil 3: Die dritte Normalform** ([www.access-basics.de/512](http://www.access-basics.de/512)).