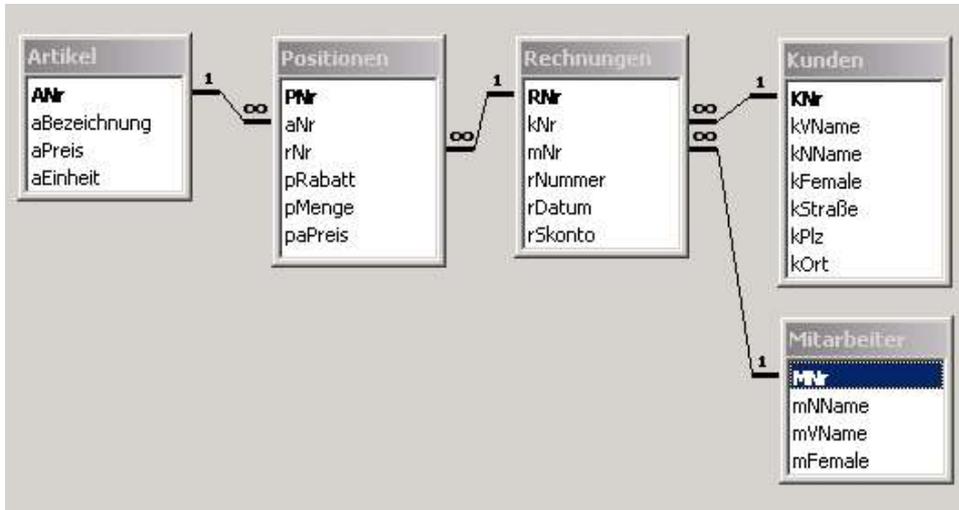


FRAGENKATALOG

Alle Fragen beziehen sich auf folgendes Datenmodell:



SQL-DML:

Zeige alle Artikel die in der Rechnungsnummer RNr = 4 verkauft worden sind.

```
SELECT ANr, a Bezeichnung
FROM Artikel JOIN (Rechnungen JOIN Positionen
ON Rechnungen.RNr= Positionen.rNr) ON Artikel.ANr = Positionen.aNr
WHERE Rnr = 4
```

```
SELECT Anr, aBezeichnung
FROM Artikel JOIN Positionen ON Artikel.ANr = Positionen.aNr
WHERE Positionen.rNr = 4
```

Zeige alle Daten (Kunde, Mitarbeiter, Artikel, Positionen) zur Rechnungsnummer RNr = 6

```
SELECT *
FROM (Artikel JOIN Positionen ON Artikel.ANr = Positionen.aNr)
JOIN ((Kunden JOIN Rechnungen ON Kunden.KNr = Rechnungen.kNr)
JOIN Mitarbeiter ON Mitarbeiter.MNr = Rechnungen.mNr)
ON Positionen.rNr = Rechnungen.RNr
WHERE Rnr = 6
```

Zeige alle Daten zum Kunden "Berger".

```
SELECT *
FROM (Artikel JOIN Positionen ON Artikel.ANr = Positionen.aNr)
JOIN ((Kunden JOIN Rechnungen ON Kunden.KNr = Rechnungen.kNr)
JOIN Mitarbeiter ON Mitarbeiter.MNr = Rechnungen.mNr)
ON Positionen.rNr = Rechnungen.RNr
WHERE kNName="Berger"
```

Welcher Mitarbeiter betreut welche Kunden?

```
SELECT Mnr, mNName
      FROM (Kunden JOIN Rechnungne ON kNR = KNr)
      JOIN Mitarbeiter ON mNr = MNr
```

Welcher Mitarbeiter hat den Kunden "Berger" zuletzt betreut?

```
SELECT mNName
      FROM (Kunden JOIN Rechnungen ON Kunden.KNr = Rechnungen.kNr)
      JOIN Mitarbeiter ON Rechnungen.mNR = Mitarbeiter.MNr)
      WHERE Kunden.kNName = „Berger“
      ORDER BY rDatum DESC LIMIT = 1
```

```
SELECT mNName MAX(rDatum) AS maxi
      FROM (Kunden JOIN Rechnungen ON Kunden.KNr = Rechnungen.kNr)
      JOIN Mitarbeiter ON Rechnungen.mNR = Mitarbeiter.MNr)
      WHERE Kunden.kNName = „Berge“ AND rDatum = maxi
```

Wieviele Hemdem wurden 2003 verkauft.

```
SELECT aBeziehung, SUM(pMenge) AS Menge
      FROM (Artikel JOIN Positionen ON Artikel.ANr = Positionen.aNr)
      JOIN Rechnungen ON Rechnungen.RNr = Positionen.rNr)
      GROUP BY aBezeichnung
      WHERE aBezeichnung LIKE „Hemd“
      AND rDatum >=01.01.2003 AND rDatum <=31.12.2003
```

Wie lautet der Gesamtumsatz für das Jahr 2003?

```
SELECT SUM(paPreis*pMenge*(1-pRabatt)) AS Umsatz
      FROM Rechnungen JOIN Positionen ON Rechnungen.RNr=Positionen.rNr
      WHERE Rechnungen.rDatum>=1.1.2003 AND Rechnungen.rDatum<=31.12.2003
```

Schramek Version:

```
select sum(pMenge*paPreis) as gessum
      from (Positionen Join Rechnungen on Positionen.rNr = Rechnungen.RNr)
      where rDatum <= #31.12.2003# and rDatum >= #01.01.2003#
```

Wie viel Umsatz mache ich mit welchem Artikel? (Gruppierung nach allen Artikeln)

```
SELECT ANr, aBezeichnung, SUM(paPreis x pMenge) AS summe
      FROM (Positionen JOIN Rechnungen ON rNr = RNr)
      LEFT JOIN Artikel ON aNr = ANr
      GROUP BY aBezeichnung
```

Welcher Kunde macht wie viel Umsatz?

```
SELECT kNName, summe(paPreis x pMenge)
      FROM (Kunden JOIN Rechnungen ON KNr = knr) JOIN Positionen ON RNr = rNr)
      GROUP BY kNName
```

Welcher Mitarbeiter macht wieviel Umsatz?

```
SELECT mNName, summe(paPreis*pMenge) as summe
      FROM (Mitarbeiter JOIN Rechnungen ON Mitarbeiter.MNr=Positionen.mNr)
      JOIN Positionen ON Positionen.RNr=Rechnungen.rNr)
      GROUP BY Mitarbeiter.mNName;
```

Schramek Version:

```
select sum(pMenge*paPreis) as gessumMitarbeiter, mNName
  from (Positionen Join Rechnungen on Positionen.rNr = Rechnungen.RNr)
  Join Mitarbeiter on Rechnungen.MNr = Mitarbeiter.MNr
 group by Mitarbeiter.mNName
```

Wieviel Umsatz wurde im Dezember 2003 gemacht?

```
SELEC SUM(pMenge x paPreis) AS summe, aBezeichnung, aNr
  FROM (Artikel Join Positionen On Anr = aNr)
  GROUP BY aBezeichnung
```

Wieviel Stück wurden 2003 von allen Artikeln verkauft? (bitte alle Artikel auflisten, auch die die nicht verkauft wurden, und absteigend nach Stückzahlen sortieren)

nicht lösbar momentan

Welche Artikel wurden 2003 nicht verkauft.

```
SELECT sum(pMenge) AS Stückzahl, aBezeichnung, Anr
  FROM Positionen JOIN Artikel ON Positionen.aNr = Artikel.ANr
  GROUP BY aBezeichnunge
```

Zeige alle Artikelumsätze absteigend sortiert.

```
SELECT ANr, aBezeichnung, SUM(pMenge*paPreis) AS Umsatz
  FROM(Artikel JOIN Positionen ON Artikel.ANr=Positionen.aNr)
  JOIN Rechnungen ON Positionen.rNr=Rechnungen.RNr
  GROUP BY Artikel.ANr
  ORDER BY SUM(pMenge*paPreis) DESC
```

Zeige alle Kundenumsätze für das Jahr 2003 absteigend sortiert.

```
Select kNNr, kNr, sum(paPreis*pMenge) as umsatz
  from (Kunden join Rechnungen on Kunden.kNr = Rechnungen.kNr)
  join Positionen on Positionen.rNr = Rechnungen.RNr
  order by sum (paPreis*pMenge) DESC
```

Welcher Kunde hat nichts gekauft?

```
select Kunden.*
  from (Rechnungen Right Join Kunden on Rechnungen.kNr = Kunden.kNr)
  where Kunden.kNr not in
  (select kNr from Rechnungen)
```

Welcher Kunde hat 2003 nichts gekauft.

```
select Kunden.*
  from (Rechnungen Right Join Kunden on Rechnungen.kNr = Kunden.kNr)
  where Kunden.kNr not in
  (select kNr from Rechnungen where rDatum <= #31.12.2003# and rDatum >=
  #1.1.2003# )
```

Lösche den Kunden "Huber"

```
DELETE * From Kunden
  WHERE KVName="Huber"
```

Lösche alle Kunden die noch nie etwas gekauft haben

```
DELETE * FROM Kunden
      WHERE kNR NOT IN
      (SELECT RNr, kNR FROM Rechnungen JOIN Kunden ON KNr = kNR)
```

Schramek Verison:

```
Delete Kunden.*
      From Kunden Left Join Rechnungen on Rechnungen.KNr = Kunden.KNr
      Where Is Null(Rechnungen.kNr)
```

Lösche die Position mit der Nummer 15.

```
DELETE * FROM Positionen
      WHERE PNr = 15
```

Erstelle aus allen Positionen vor dem Jahr 1.1.1998 eine neue Tabelle und nenne sie "PositionenArchiv"

```
SELECT Positionen.* INTO PositionenArchiv
      FROM Positionen JOIN Rechnungen ON Positionen.rNr=Rechnungen.RNr
      WHERE Rechnungen.rDatum <1.1.1998
```

Lösche alle Datensätze aus Positionen die vor dem 1.1.1998 erstellt worden sind.

```
DELETE PNr, aNr, rNr, pRabatt, pMenge, paPreis
      FROM Positionen JOIN Rechnungen ON rNr = RNr
      WHERE rDatum < 1.1.1998
```

Füge an die Tabelle PositionenArchiv alle Datensätze aus Positionen ein von 1.1.1998 bis 31.12.2000 erstellt worden sind

wie kann man erfahren wann ein datensatz erstellt wurde? (bitte um aufklärung!)

Lösche alle Datensätze aus Positionen die von 1.1.1998 bis 31.12.2000 erstellt worden sind.

```
DELETE PNr,aNr,rNr, pRabatt, pMenge,paPreis
      FROM Positionen JOIN Rechnungen ON rNr = RNr
      WHERE rDatum <=31.12.2000 and rDatum >= 1.1.1998
```

Union: Zeige alle Datensätze aus Positionen und PositionenArchiv an.

```
SELECT *
      FROM Positionen
      UNION SELECT * FROM PositionenArchiv
```

Erhöhe für alle Artikel aPreis um 3%.

```
UPDATE Artikel SET aPreis = aPreis * 1.03
```

Reduziere für den Artikel "Hemd" den Preis um 25%

```
APDATE Artikel SET aPreis = aPreis * 0.75
      WHERE aBezeichnung = „Hemd“
```

Ändere für das Feld paPreis alle leeren Datensätze auf den derzeit gültigen Artikelpreis aPreis.

```
UPDATE Positionen SET paPreis = Artikel.aPreis where paPreis = null
```

oder:

```
UPDATE Positionen SET paPreis = aPreis WHERE IS NULL(paPreis)
```

Füge einen neuen Kunden mit beliebigen Daten in die Tabelle Kunden ein.

```
INSERT INTO Kunden(KNr, kVName, kNName)
VALUES(3800,"Markus","Unger");
```

Füge eine neue Rechnungsnummer mit beliebigen Werten in die Tabelle Rechnungen ein

```
Insert INTO Rechnungen(RNr,rDatum)
VALUES(10,#17.06.2004#);
```

Füge eine neue Position mit beliebigen Werten in die Tabelle Positionen ein

```
INSERT INTO Positionen(PNr,pMenge,aNr)
VALUES(400,3,3);
```

Ändere die Menge pMenge auf 10 für die Position PNr = 234

```
UPDATE Positionen SET pMenge = 10 where PNr = 234
```

Zeige alle Artikel die in der Rechnungsnummer RNr = 4 verkauft worden sind.

```
select Artikel.*
from (Artikel Join Positionen on Artikel.ANr = Positionen.aNr)
Join Rechnungen on Positionen.rNr = Rechnungen.RNr
where Rechnungen.RNr = 4
```

einfacher:

```
select Artikel.*
from (Artikel Join Positionen on Artikel.ANr = Positionen.aNr)
where Positionen.RNr = 4
```

Gleiche Artikel mit unterschiedlichen Nummern sollen angezeigt werden (Redundanzüberprüfung)

```
Select count(Artikel.aBezeichnung) as redundant, Artikel.aBezeichnung
From Artikel
Group by Artikel.aBezeichnung
Having count(Artikel.aBezeichnung) > 1
```

Gleiche Mitarbeiter mit unterschiedlichen Nummern sollen angezeigt werden (Redundanzüberprüfung)

```
Select count(Mitarbeiter.mNName) as doppeld, Mitarbeiter.mNName
From Mitarbeiter
Group by Mitarbeiter.mNName
Having count(Mitarbeiter.mNName) >1
```

Alle Kunden, die ab 2003 nichts gekauft haben, in eine neue Tabelle Archiv

```
Select Kunden.* Into Archiv
From Kunden Join Rechnungen on Kunden.KNr = Rechnungen.kNr
Where Rechnungen.kNr Not in(Select Rechnungen.rDatum From Rechnungen
Where rDatum>=#1.1.2003#)
```

Lagerstand ermitteln

```
SELECT Artikel.*, SUM(Positionen.pMenge) AS Lagerstand
FROM Positionen JOIN Artikel ON Positionen.aNr = Artikel.ANr GROUP BY Artikel.ANr
```

SQL-DDL:

Erstelle eine neue Tabelle mit dem Namen "Lieferanten" und den Feldern LNr (Primärschlüssel,Autowert), IFirma, INName, IVName, IFemale, ITitel, IAdresse, IStraße, IPLz, IOrt.

```
CREATE TABLE Lieferanten(LNr autoincrement primary key not null, LFirma,...);
```

Schramek Version:

```
CREATE table Lieferanten(LNr autoincrement primary key not null,IFirma text,INName
text(30),IFemale boolean, ...)
```

Füge an die Tabelle Lieferanten weitere Felder mit den Namen ITel, IFax, und IKontoNr an.

```
ALTER TABLE Lieferanten ADD COLUMN ITel Text(255);
```

Lösche aus der Tabelle Lieferanten das Feld IAdresse.

```
Alter table Lieferanten drop IAdresse
```

Ändere die Tabelle Kunden so ab, dass sie die gleiche Struktur wie die Tabelle Lieferanten aufweist.

```
Select Kunden.* Into Lieferanten From Kunden
```

Union: Stelle Kunden und Lieferanten gemeinsam dar. (aufsteigend sortiert nach xNName). Welches Problem könnte sich daraus ergeben

in welcher tabelle stehen lieferanten?!

Theoriefragen:

Welchen Vorteil bietet eine GUID gegenüber einem Autowert.

Guid... Globally Unique Identifier

Der Vorteil einer GUID ist, dass jene fortlaufend gezählt wird (was bspw. für eine Rechnungsnummer ideal ist), was bei einem Autowert nicht zwangsweise der Fall sein muss

Was besagt die erste Normalform

Es gibt keine Attribute, deren Werte sich aus mehreren Elementen zusammensetzen (also wieder eine Relation sind). An allen Kreuzungspunkten zwischen Zeile und Spalte steht max. nur 1 Wert.

Was besagt die 2. Normalform

Eine Relation befindet sich in der 2NF, wenn jedes nicht dem Schlüssel angehörende Attribut vom Gesamtschlüssel funktional abhängig ist, nicht aber von Teilschlüsseln.

Was besagt die 3. Normalform

Eine Relation befindet sich in 3NF, wenn sie in 2NF ist und alle Nicht-Schlüsselspalten unabhängig voneinander sind.

Was versteht man unter Referenzieller Integrität

Die Regeln der referentiellen Integrität verbieten verwaiste Fremdschlüssel. Sie verbieten auch, eine Zeile mit einem Fremdschlüssel einzufügen, wenn es zu diesem Fremdschlüssel in der verbundenen Tabelle keinen Eintrag gibt.

Was versteht man unter Löschweitergabe

Die Löschweitergabe funktioniert mit der Referentiellen Integrität. Wurde diese ausgewählt, so werden automatisch auf dem gelöschten Primärschlüssel verweisende Spalten der Detailtabelle gelöscht.

Was ist eine Aktualisierungsweitergabe

Ist diese Einstellung aktiviert, so wird jede Änderung am Primärschlüssel der Mastertabelle automatisch an alle betroffenen Datensätze der Detailtabelle weitergegeben. Der Wert des Fremdschlüssels wird also entsprechend aktualisiert.

Was versteht man unter einer Aggregatsfunktion

Aggregatsfunktionen können in Abfragen und Formularen angewendet werden. Sie sind eine Anzahl von mathematischen Funktionen. Summe, Mittelwert, Min, Max, Gruppierung, Anzahl, Standardabweichung, Varianz, Erster Wert, Letzter Wert.

Wieviele Byte benötigt der Felddatentyp currency(Währung)

64 Bit

Können Daten nach einem Memo Felddatentyp sortiert werden?

Nein

Mit welcher Eigenschaft kann ich die Datenherkunft eines Formulars filtern

Leeb:

`DoCmd.ApplyFilter`

Praunias:

Die Eigenschaft `RecordSource` ist die Datenherkunft des Formulars, die eine Tabelle, eine gespeicherte Abfrage oder einen Abfragetext erhalten kann

Wann verwendet man Registerkarten

Wenn viele Details auf einem Formular dargestellt werden sollen. Dann besteht mit Registerkarten die Möglichkeit, die Details nach Themen sortiert auf verschiedenen Registerkarten abzulegen.

Wozu eignet sich eine Haupt/Unterformular-Konstruktion

Unterformulare sind eine elegante Möglichkeit, Daten aus Tabellen oder Abfragen darzustellen, die zueinander in 1:n Beziehungen stehen. Diese ist zwar auch mit Kombinations- oder Listenfelder möglich. Unterformulare können aber genauso wie die Mastertabelle in der Detailtabelle bearbeitet werden.

Wozu eignen sich Kombinationsfelder

Kombinationsfelder sind eine Mischung aus Eingabefeld und Listenfeld. Sie haben den Vorteil, dass sie Werte oder Texte zur Auswahl geben können. Datenherkunft kann eine Abfrage oder eine manuell eingetragene Liste sein.

Wie kann man vermeiden, dass eine Abfrage die Meldung "zu komplex" erzeugt.

Indem man inzwischen eine Tabelle erzeugt, in der die Zwischenergebnisse gespeichert werden und mit dieser weiterarbeitet.

Was bedeutet Datenredundanz bei einem Datenmodellentwurf.

Dass gewisse Daten einfach doppelt oder mehrfach in einer Datenbank gespeichert werden, zB die Adresse eines Kunden. Wird diese doppelt gespeichert, kann bei einer Aktualisierung der Adresse ein Fehler auftreten.

Was ist der Unterschied zwischen einem Textfeld und einem Bezeichnungsfeld

Das Textfeld ist das meistbenutzte Steuerelement in Access. Es wird für die Ein- und Ausgabe von Daten im Formular benutzt und hat eine Verbindung mit einer Tabelle. Das Bezeichnungsfeld hingegen hat keine Datenherkunft und wird meist für diverse Überschriften verwendet.

Was versteht man unter einer Ereignisprozedur

Windows-Programme (also auch Access) sind in fast allen Fällen ereignisgesteuert. Für jedes Ereignis (Klick, Fokusverlust, Aktualisierung usw.) in Formularen kann ein Makro oder ein Visual Basic Programm ablaufen. Diese Programmierung nennt man Ereignisprozedur.

Mit welcher Eigenschaft lässt sich ein Steuerelement ein bzw ausblenden.

Ausdruck.Visible

Mit welchen Eigenschaften kann in einem Bericht die Größe des Steuerelementes an den Steuerelementinhalt angepasst werden.

Eigenschaft: verkleinerbar = JA; ist für Textfelder möglich, die sich untereinander befinden; wenn sich jene nebeneinander befinden, muss sichergestellt sein, dass sie sich auf exakt selber Höhe befinden; (Ist ein Textfeld bspw. leer, nimmt es keinen Raum ein, werden Inhalte darin platziert, passt es sich in seiner Größe automatisch an!)

Mit welchem VBA Befehl kann ein Formular geöffnet werden?

DoCmd.OpenForm „Formularname“

Mit welchem VBA Befehl kann ein Formular geschlossen werden?

DoCmd.Close acForm, „Formularname“, acSave Yes

Mit welcher Eigenschaft kann ein Formular immer im Vordergrund positioniert werden?

Formular-Eigenschaften: POPUP = JA

Mit welcher Eigenschaft kann verhindert werden, dass ein anderes Fenster (als das momentan aktivierte) den Focus bekommen kann?

Formular-Eigenschaften: Gebunden = JA

Wie kann eine n:m Beziehung identifiziert werden?

Zwei Tabellen stehen in einer n:m Beziehung, wenn es zu einem Feld der ersten Tabelle eine, keine oder mehrere Zeilen der zweiten Tabelle gibt, aber umgekehrt auch zu einer Zeile der zweiten Tabelle keine, eine oder mehrere Zeilen der ersten Tabelle existieren.