

# Datenmodellierung - ein Praxisbeispiel

Karel Štípek

## Am Anfang war ein Wartungsauftrag

Kurz vor der letzten Nationalratswahl habe ich den Auftrag bekommen, das Programm für die Erfassung und Auswertung der Wahlergebnisse für die neue Wahl anzupassen. Das Werk war bei der vorigen Wahl erfolgreich im Einsatz und hat alles Notwendige enthalten - von den Masken für die Eingabe bis zu Auswertungen in Tabellen und Diagrammen nach verschiedensten Kriterien inklusive der Möglichkeit, die Ergebnisse von unterschiedlichen Wahlen miteinander zu vergleichen.

Der Autor der Applikation war kein Anfänger. Er hat keine Makros verwendet, hat die VBA-Programmierung gut verstanden, konnte z. B. mit den Ereignissen und Recordsets gut umgehen. Nur eines hat er bei der Arbeit nicht beachtet - die Grundsätze der Datenmodellierung. Es hat sich herausgestellt, dass aufgrund dieses Mangels das Anpassen des Programms sehr aufwändig wäre. Dann habe ich lieber aus der alten Lösung nur das Layout der Formulare und Berichte übernommen und sonst alles neu gemacht.

## Datenmodellierung macht die Datenbank

Die meisten PC-Anwender bekommen in der heutigen Zeit den ersten Kontakt mit der Datenverarbeitung über MS-Excel. Die Bedienung des Programms ist mehr oder weniger intuitiv, die Ergebnisse sind einfach formatierbar und die Begrenzung der maximalen Anzahl der Zeilen stört in den meisten Fällen auch nicht. Diese Tatsachen machen das Excel auch sehr beliebt.

Bei dem Aufstieg zum MS-Access sucht man dann Analogien zum Excel und der wichtigste Unterschied bleibt dabei unbemerkt - Access ist im Gegenteil zu Excel eine Datenbank. In einem Anfängerkurs, wie z.B. ECDL (Europäischer Computer-Führerschein) ist es zeitlich nicht möglich, die wichtigsten Prinzipien der Datenmodellierung den Teilnehmern ausreichend beizubringen. Dabei sollte man auch bei einfachen Applikationen, wie einer Adressen- oder Bücherverwaltung gerade damit anfangen. Schon nur durch die Beseitigung der redundanten Datenspeicherung spart man sich eine Menge Tipparbeit.

Der Vergleich zwischen Excel und Access soll nicht missverstanden werden. Auch Excel ist sehr mächtig und lässt sich mit VBA gut programmieren. Für die Entwicklung einer Datenbankapplikation ist jedoch Access viel besser geeignet.

## Theorie und Praxis

Im vorliegenden Artikel möchte ich Ihnen die Prinzipien der Datenmodellierung anhand eines praktischen Beispiels - der Wahldatenbank - erklären. Die Theorie ist dabei wichtig, scheint aber manchmal etwas kompliziert und unverständlich zu sein. Es gibt genug gute Beiträge zu dem Thema.

Damit das Verständnis der Problematik nicht so schwer fällt, lasse ich die Normalformen und andere Begriffe vorerst "schlafen" und werde bei der Analyse eher intuitiv vorgehen. Mit relativ einfachen Fragen und Überlegungen kommen wir bei diesem nicht sehr komplexen Beispiel auch ohne die exakte Theorie zum richtigen Ergebnis.

Die Datenmodellierung wird effizient durch spezielle Programme unterstützt. Ich werde in diesem Artikel die notwendigen Diagramme mit Microsoft Visio erstellen, weil sie besser lesbar sind, als z.B. die Ausdrucke des Access-Beziehungsfensters. Alle diskutierten Datenbankobjekte sind in der beigelegten Datenbank `DatModel1.mdb` enthalten.

## Die Lösung mit einer Tabelle

### Theoretisch kein Problem

Für die Eingabe und Auswertung der Wahlergebnisse einer Wahl selbst reicht theoretisch eine Tabelle vollkommen aus. In jeder Zeile werden folgende Daten für einen Sprengel gespeichert:

- Nummer und Name des Sprengels

- Anzahl der Wahlberechtigten, der abgegebenen und ungültigen Stimmen pro Sprengel

- Anzahl der abgegebenen Stimmen für jede Partei

Die Struktur einer solchen Tabelle (sie heißt **WahlErgebnis**) können Sie dem folgenden Bild entnehmen:

WahlErgebnis : Tabelle		
	Feldname	Felddatentyp
	SprengelNr	Zahl
	Sprengelbezeichnung	Text
	Wahlberechtigte	Zahl
	AbgegebeneStimmen	Zahl
	UngültigeStimmen	Zahl
	Partei1	Zahl
	Partei2	Zahl
	Partei3	Zahl
	Partei4	Zahl
	Partei5	Zahl

Abbildung 1: Die Wahlergebnisse in einer einzigen Tabelle

Für die Auswertung der Ergebnisse einer bestimmten Wahl in einer bestimmten Gemeinde ist gegen diese Tabelle nichts einzuwenden. Sie können mit Abfragen die Ergebnisse pro Partei und Gesamt aufsummieren und sie in verschiedenen Diagrammen und Berichten darstellen. Es fällt Ihnen allerdings jetzt schon auf, dass in dieser Tabelle nicht die Namen der Parteien gespeichert sind. Die werden erst als Bezeichnungsfelder in alle Präsentationsobjekte hinzugefügt.

### Die nächste Wahl kommt

Die entworfene Tabelle ist genau nur für eine Wahl gültig. Die Anzahl, Namen und Reihenfolge der Parteien können sich bei der nächsten Wahl ändern. Es wird notwendig sein, nicht nur eine neue Tabelle zu erstellen, sondern auch alle anderen Objekte der geänderten Struktur anzupassen. Der Wartungsaufwand ist groß und das Schlimmste daran ist, dass die notwendigen Arbeiten von einem Entwickler durchgeführt werden müssen. Mit anderen Worten - Sie werden ein solches Programm nie als eine allgemein brauchbare Lösung verwenden können.

### Wahlspezifische Eigenschaften

Die wichtigste Aufgabe ist, die Datenbankobjekte so zu entwerfen, dass ihre Struktur von einer Wahl zur anderen nicht geändert werden muss. Folgende Angaben sind vorgegeben und gelten meistens nur für eine Wahl:

- Datum der Wahl
- Gesamtanzahl der vergebenen Mandate
- Anzahl, Nummern und Bezeichnungen der Sprengel
- Anzahl, Namen, Logos, Diagrammfarben und Reihenfolge der kandidierenden Parteien

### Vergleiche der Wahlergebnisse

Eine besondere Herausforderung stellen die Vergleiche der Wahlergebnisse dar. In einem Formular, bzw. Bericht werden für zwei unterschiedliche Wahlen die Anzahl der Stimmen aller Parteien pro Sprengel oder gesamt einander gegenübergestellt. Aufgrund der wahlspezifischen Parteienstrukturen sind folgende Situationen zu beachten:

- Die gleiche Partei kann in zwei Wahlen unter einem anderen Namen auftreten.
- Eine Partei kann nur auf einer Seite des Vergleichs erscheinen, weil sie bei der anderen Wahl nicht kandidiert hat.

Bei der oben vorgestellten Lösung mit einer einzigen Tabelle wäre es also notwendig, für jede mögliche Wahlkombination ein eigenes Formular, bzw. Bericht zur Verfügung zu haben. Das bedeutet auch, dass im Rahmen der Anpassung des Programms für die neue Wahl ein neuer Satz an Präsentationsobjekten erstellt werden müsste.

- Die Anzahl der Stimmen pro Partei im Sprengel ist NICHT ein Attribut der Entitäten **Sprengel** oder **Partei**

**Schritt 3 - Beziehungen**

Um die letzten drei Aussagen richtig zu verstehen, müssen wir uns erinnern, was das Hauptproblem der Lösung mit einer einzigen Tabelle war - die Abhängigkeit verschiedener Eigenschaften von der konkreten Wahl.

Wenn wir also in einer Applikation, die mehrere Wahlen verwaltet, die Anzahl der Wahlberechtigten speichern wollen, können wir sie nicht einem Sprengel, sondern nur einer Kombination **Wahl-Sprengel** zuordnen. Diese Kombination ist die Ausprägung einer Beziehung zwischen der Entität **Wahl** und der Entität **Sprengel**.

Die zweite wichtige Kombination ist **Wahl-Partei**. Dieser Kombination werden die Attribute zugeordnet, die eigentlich eine Partei beschreiben, für jede Wahl aber anders sein können. Darunter gehört beispielsweise der wahl-spezifische Name oder das Logo. Unter dem in der letzten Zeile der **Tabelle 1** angeführten Name ist ein wahlenunabhängiger Name zu verstehen, der eigentlich in den Auswertungen nirgends angezeigt wird. Er dient nur dazu, beim Vergleich die gleichen Parteien einander gegenüberstellen zu können, auch wenn ihre Namen bei den unterschiedlichen Wahlen eventuell unterschiedlich sind.

Die letzte Kombination, die wir berücksichtigen müssen, ist **Wahl-Partei-Sprengel**. Nur dieser Kombination von allen drei Entitäten kann die Anzahl der Stimmen pro Wahl, Sprengel und Partei zugeordnet werden.

Eine Beziehung ist eigentlich auch eine Entität, die Attribute enthalten kann und wird als Beziehungsentität genannt. In folgender Tabelle werden die Attribute der drei Beziehungsentitäten zusammengefasst:

Beziehungsentität	Attribut
Wahl-Sprengel	Wahlberechtigte
	Abgegebene Stimmen
	Ungültige Stimmen
Wahl-Partei	Name (wahl-spezifisch)
	Logo
Wahl-Sprengel-Partei	Anzahl der Stimmen

Die Anzahl der abgegebenen Stimmen pro Wahl und Sprengel kann rein theoretisch aus der Summe der abgegebenen Stimmen für alle Parteien plus die Anzahl der ungültigen Stimmen berechnet werden. Sie muss nicht in der Beziehungsentität Wahl-Sprengel gespeichert werden. In der Praxis sollen Sie aber auch die realen Arbeitsabläufe und Notwendigkeit der Eingabeüberprüfung berücksichtigen. An die zentrale Datenerfassungsstelle werden Formulare angeliefert, auf denen die Zusammenfassung der Stimmen schon manuell berechnet wurde. Diese Zahl wird in die Beziehungsentität Wahl-Sprengel deswegen eingegeben, weil sie nach der Eingabe der Stimmen aller Parteien pro Sprengel als eine wichtige Kontrollsumme verwendet werden kann.

**Implementierung des Datenmodells in Access**

**Feldnamen-Konventionen**

Das Ergebnis der Implementierung eines Datenmodells sind schließlich die Tabellen, ihre Strukturen und Beziehungen. In einer komplexen Applikation ist auch die Übersichtlichkeit sehr wichtig. Bei der Feldnamenvergabe verwende ich deswegen konsequent folgende Regeln:

- Als Trennzeichen wird ausschließlich Underscore ("\_") verwendet. Wenn Sie Leerzeichen oder Bindestriche einsetzen, müssen Sie in Abfragen oder Programmcode die Namen der betroffenen Tabellenfeldern in eckige Klammern einschließen. Der Abfragenassistent macht das zwar für Sie automatisch, bei dynamischen Abfragen, wenn Sie den SQL-Ausdruck im Programmcode zusammenbauen, sparen Sie sich schon etwas Tipparbeit.

**Der intuitive Weg zu einem richtigen Datenmodell**

**Schritt 1 - Entitäten**

Auf der Suche nach einer allgemeinen Lösung müssen wir zuerst die für unsere Applikation relevanten so genannte Entitäten auffinden. Die Entitäten sind laut Definition die Objekte der realen Welt, die aber nicht nur physikalische Gegenstände oder Lebewesen sein müssen, sondern auch z.B. Ereignisse, Vorgänge u.v.a. sein können. Die Zusammenfassung aller Entitäten des gleichen Typs (ein so genannter Entitätstyp) wird in der Datenbank als eine Tabelle implementiert.

Bei der Erfassung und Auswertung der Wahlergebnisse sind offensichtlich drei Entitäten im Spiel:

- **Wahl**
- **Sprengel**
- **Partei**

Die Aufstellung der Entitäten ist von der konkreten Aufgabenstellung abhängig. Beispielsweise in einem Programm, dass zusätzlich auch die Vorzugsstimmen der Parteikandidaten auswerten soll, kommt die Entität Kandidat dazu. Sollten Sie die interne Struktur der Sprengel beobachten, werden Sie eine Entität für eine kleinere geografische Einheit (z.B. Straße) definieren müssen.

**Schritt 2 - Attribute**

Eine Entität hat Eigenschaften, die Attribute genannt werden. Ein Attribut einer Entität (exakt eines Entitätstyps) wird in der Datenbank als ein Tabellenfeld implementiert. Ihre Definition ist nicht immer ganz trivial, es ist etwas mehr abstraktes Denken angesagt. Die wichtigste Frage, die Sie sich stellen müssen bevor Sie ein Attribut der Entität zuordnen, lautet:

Ist die Eigenschaft nur für diese Entität spezifisch, bzw. nur von ihr abhängig?

Für die o.a. drei Entitäten kann die Frage für folgende Attribute mit JA beantwortet werden:

Entität	Attribut
Wahl	Typ
	Datum
	Anzahl der Mandate
	Anzahl der Sprengel
Sprengel	Nummer
	Bezeichnung
Partei	Name (wahlenunabhängig)

Es ist ersichtlich, dass mit den gerade angeführten Attributen (Tabellenfeldern) das Programm seine Aufgaben nicht erfüllen kann. Wo sind beispielsweise die Felder für die Eingabe der Anzahl der Wahlberechtigten pro Sprengel und der abgegebenen Stimmen pro Sprengel und pro Partei?

Folgende Aussagen sind der Grund dafür, dass die Felder für Stimmen nicht als Attribute der vorgestellten Entitäten auftreten. Sie scheinen zwar auf den ersten Blick unlogisch, sind aber doch wahr:

- Die Anzahl der Wahlberechtigten ist NICHT ein Attribut der Entität **Sprengel**.
- Das Partei-Logo ist NICHT ein Attribut der Entität **Partei**

- Die abgekürzten Tabellennamen werden groß geschrieben als Präfix zu jedem Namen eines Tabellenfelds (Spalte) hinzugefügt. Dadurch sind alle Feldnamen in der gesamten Datenbank eindeutig und klar der jeweiligen Tabelle zugeordnet.
- Die Namen der Schlüsselfelder werden groß geschrieben und fallen dadurch sofort auf.
- Die Namen der Primärschlüsselfelder, die als ein Autowert automatisch generiert werden, bestehen aus dem abgekürzten Tabellennamen und der Bezeichnung "\_ID".
- Die Namen von fremden Schlüsseln enthalten die Bezeichnung der eigenen Detailtabelle und den kompletten Namen des entsprechenden Primärschlüssels (enthält die Bezeichnung der Mastertabelle, wohin der fremde Schlüssel verweist).

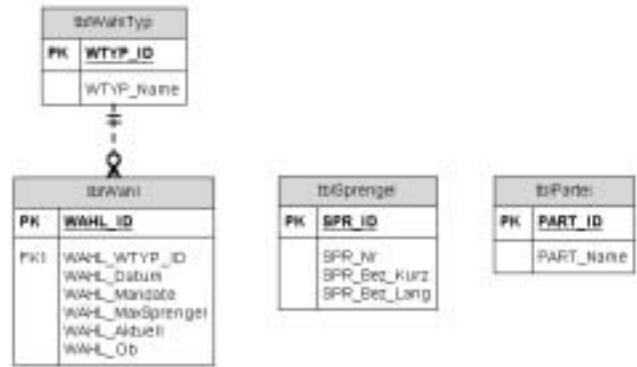


Abbildung 2: Die Entitätstabellen der Wahldatenbank

**Tabellen für Entitäten**

Die Strukturen der Tabellen **tb1Wahl**, **tb1Sprengel** und **tb1Partei**, die in der Datenbank den Entitäten **Wahl**, **Sprengel** und **Partei** entsprechen, werden bei der Umsetzung ins Access um einige technisch bedingte Felder erweitert.

- In allen drei Tabellen wird ein Autowert-Feld mit dem Namen **...\_ID** als primärer Schlüssel erstellt.
- Für die Eingabe des Wahltyps (Gemeindewahl, Nationalratswahl u.a.) wird eine Nachschlagetabelle **tb1WahlTyp** definiert und in der Tabelle **tb1Wahl** wird nicht der Text, sondern der primäre Schlüssel des Wahltyps als fremder Schlüssel gespeichert.
- Die Tabelle **tb1Wahl** wird noch um zwei Felder erweitert:
  1. **WAHL\_Aktuell** (Ja/Nein) kennzeichnet eine Wahl als die aktuelle. Ihre Daten werden dann standardmäßig in allen Masken und Berichten angezeigt, solange keine andere Wahl ausgewählt wird.
  2. **WAHL\_Ob** (Integer) ist ein Ordnungsbegriff, der die Reihenfolge der Wahlen bei der Auswahl (z.B. in Kombinationsfeldern) bestimmt.

In der Tabelle **tb1Sprengel** steht zusätzlich zu dem Textfeld **SPR\_Bez\_Kurz** ein Memo-Feld **SPR\_Bez\_Lang** für die Speicherung von eventuellen Zusatzinformationen zum Sprengel.

Die Darstellung aller Entitätstabellen der Applikation können Sie dem Datenbankmodell-Diagramm im folgenden Bild entnehmen:

Die Verbindungslinie zwischen den Tabellen **tb1WahlTyp** und **tb1Wahl** erfordert einen kurzen Exkurs in die Theorie.

**Die Tabellenbeziehungen und ihre graphische Darstellung**

Wenn wir eine Beziehung zwischen den Tabellen erstellen wollen, müssen wir zuerst festlegen, über welche Felder die Tabellen verknüpft werden. Auf dem vorigen Bild sind es das Feld **WTyp\_ID** in der Tabelle **tb1WahlTyp** und das Feld **WAHL\_WTyp\_ID** in der Tabelle **tb1Wahl**. Das erste wird als der primäre, das zweite als der fremde Schlüssel bezeichnet.

Auf jeder Seite der Beziehung können die Schlüsselwerte einmal oder mehrmals auftreten. Die Linie in der **Abbildung 2** präsentiert eine so genannte 1:N Beziehung. Sie kann wörtlich folgendermaßen formuliert werden:

- Ein Wahltyp kann in mehreren Wahlen auftreten.
- Eine Wahl darf dagegen nur von einem Wahltyp sein.
- In der graphischen Darstellung werden auf der 1-Seite der kurze Strich und auf der N-Seite das "Gänsefüßchen" verwendet.
- Der zweite kurze Strich auf der 1-Seite und der Kreis auf der N-Seite definieren die Abhängigkeiten zwischen den Entitäten. In unserem Beispiel kommen folgende Regeln zum Tragen:
  - Zu einem Wahltyp muss keine Wahl existieren.
  - Für jede Wahl muss dagegen eine Wahltyp definiert sein.

Die Beziehung 1:1 kommt in der Praxis relativ selten vor. Solche Tabellen können theoretisch in eine einzige zusammengefasst werden.

Die Beziehung M:N, wo auf beiden Seiten der primäre Schlüssel mehrmals vorkommen kann, ist dagegen sehr häufig. Die kann nur mit Hilfe einer dritten, einer Beziehungsentität dargestellt werden. Auf diesem Prinzip sind alle restlichen Beziehungen in unserem Beispiel aufgebaut.

**Beziehungsentitäten**

Die Beziehungsentitäten sind Tabellen, die eine Beziehung darstellen, die sich nicht nur durch die Speicherung des fremden Schlüssels abbilden lässt. Ein typisches Beispiel ist die oben erwähnte Beziehung M:N. Sie sind von den Entitäten abhängig. Das heißt, dass kein Datensatz in der Beziehungsentitätstabelle ohne die entsprechenden Datensätze in den Entitätstabellen existieren kann.

Aus diesem Grund müssen die Datensätze in den Beziehungsentitätstabellen nicht durch einen eigenen primären Schlüssel identifiziert werden. Die Rolle des eindeutigen Identifikators übernimmt die Kombination der primären Schlüssel aller an der Beziehung beteiligten Entitäten.

**Wahl-Sprengel**

Die Kombination einer Wahl mit einem Sprengel wird in der Tabelle **tb1WahlSprengel** implementiert. Beachten Sie, bitte, dass diese Tabelle nicht nur die Beziehung selbst (Kombination der primären Schlüssel der beteiligten Tabellen), sondern auch für diese Beziehung spezifischen Datenfelder speichert. Das sind in diesem Fall die Anzahl der Wahlberechtigten, die Anzahl der abgegebenen und ungültigen Stimmen in einem bestimmten Sprengel für eine bestimmte Wahl.

Die Tabellenstrukturen und Beziehungen sind in folgendem Diagramm dargestellt. Die Nachschlagetabelle **tb1WahlTyp** wurde aus dem Grund der besseren Übersichtlichkeit weggelassen.

Beachten Sie, dass die Felder **WSPR\_WAHL\_ID** und **WSPR\_SPR\_ID** gleichzeitig fremde Schlüssel sind und beide gemeinsam den primären Schlüssel der Tabelle **tb1WahlSprengel** bilden.



Abbildung 3: Die Beziehung zwischen den Entitäten Wahl und Sprengel.

**Wahl-Partei**

Analog wie eine Beziehung zwischen der **Wahl** und dem **Sprengel** wird auch die Beziehung zwischen der **Wahl** und der **Partei** realisiert.

Die Beziehungsentitätstabelle **tb1WahlPartei** wird außer dem im Entwurf des Datenmodells vorgeschlagenen wahl-spezifischen Namen und dem Logo-Bild noch um folgende Felder erweitert:

- Ordnungsbegriff **WPART\_Ob** für die Definition der Reihenfolge in der die Parteien in Formularen und Berichten angezeigt werden.
- Code der Farbe, mit der die Partei in Diagrammen dargestellt wird. Da die Access-internen Farbencodes nicht sprechend sind, werden sie in der Nachschlagetabelle **tb1Farbe** zusätzlich mit einer verständlichen Bezeichnung versehen. In der Tabelle **tb1WahlPartei** wird der primäre Schlüssel aus der Tabelle **tb1Farbe** als fremder Schlüssel **WPART\_FARBE\_ID** gespeichert.
- Im Feld **WPART\_Mandate** wird die Anzahl der gewonnenen Mandate nach der Berechnung gespeichert.

Die Beziehung Wahl-Partei schaut im Diagramm der Beziehung Wahl-Sprengel sehr ähnlich aus. Die Nachschlagetabelle **tblFarbe** und die Beziehung zu der Tabelle **tblWahlPartei** wurden wegen der besseren Übersichtlichkeit weggelassen. Das Prinzip der Nachschlagetabelle wurde bei den Entitätstabellen erklärt.

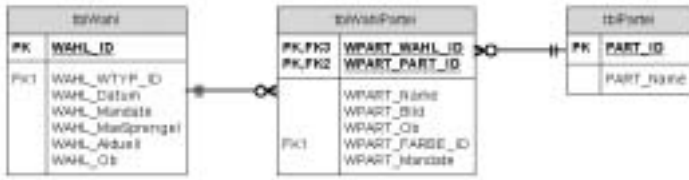


Abbildung 4: Die Beziehung zwischen den Entitäten Wahl und Partei

**Wahl-Sprengel-Partei**

Die letzte Beziehung, bzw. die letzte Tabelle die in der Wahl-datenbank zu erstellen ist, ist die Tabelle **tblErgebnis**. Darin werden die abgegebenen Stimmen bei der jeweiligen Wahl im Sprengel für eine Partei abgespeichert. Die Beziehungsentität wird also in diesem Fall die Beziehung zwischen aller drei Entitäten abbilden.

Der primäre Schlüssel der Tabelle **tblErgebnis** wird aus den primären Schlüssel aller beteiligten Entitäten zusammengesetzt. Das einzige Nicht-Schlüssel-Feld ist das Feld **ERG\_Stimmen** für die Eingabe der Stimmen.

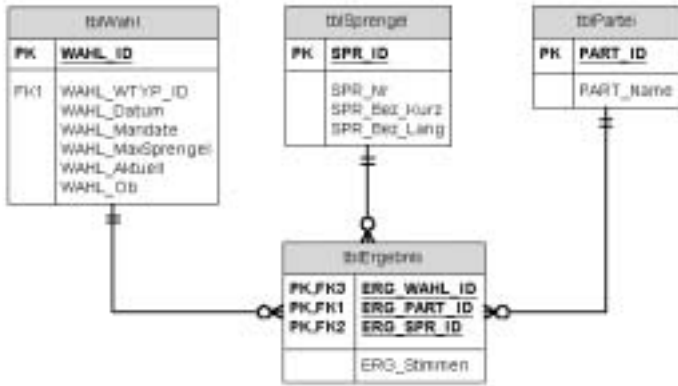


Abbildung 5: Tabelle tblErgebnis

**Beziehungen in Access**

Wenn das Datenmodell zuerst komplett mit Visio erstellt wurde, können alle Tabellen und ihre Beziehungen in der Access-Datenbank automatisch generiert werden. Sie können sie auch im Access selbst manuell erstellen. Die Beziehungen werden dann über den Menüpunkt **Extras->Beziehungen** festgelegt. Dabei ist es sinnvoll, auch die referenzielle Integrität zu definieren, damit die Datenkonsistenz sichergestellt wird.

Das Access-Beziehungsfenster (**Abbildung 6**) zeigt das komplette Datenmodell.

**Nachschlagefelder in Beziehungstabellen**

Nachdem Sie alle Tabellen und Beziehungen erstellt haben, wandeln Sie in allen Beziehungstabellen die Schlüsselfelder in Nachschlagefelder (mit Bezug auf die jeweilige Entität) um. Die Werte der primären Schlüssel der Entitäten bleiben dann für den Anwender des Programms verborgen. In den entsprechenden Spalten der Beziehungstabelle werden statt der Schlüssel die sprechenden Bezeichnungen der Entitäten angezeigt. Der Vorteil der Definition der Nachschlagefelder schon im Tabellenentwurf besteht darin, dass sie auch in Formulare automatisch übernommen werden.

Ein Beispiel der direkten Eingabe in die Tabelle **tblErgebnis** zeigt **Abbildung 7**.

Die Werte, die in der ersten Spalte angezeigt werden, werden nicht aus der Tabelle **tblWahl** direkt ausgewählt, sondern aus der Abfrage **qryWahl**. Die Abfrage verknüpft die Tabelle **tblWahl** mit der Nachschlagetabelle **tblWahlTyp** und berechnet den angezeigten Text aus den Feldern **WTYP\_Name** und **WAHL\_Datum**.

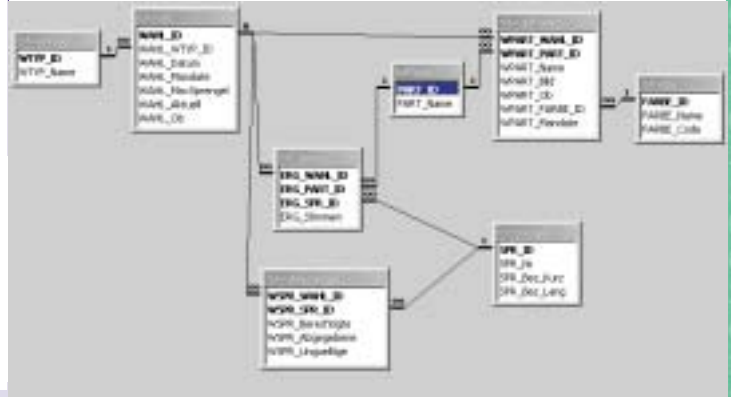


Abbildung -6: Das komplette Datenmodell der Wahl-datenbank.

In der Praxis werden die Daten nicht in die Tabellen direkt son-

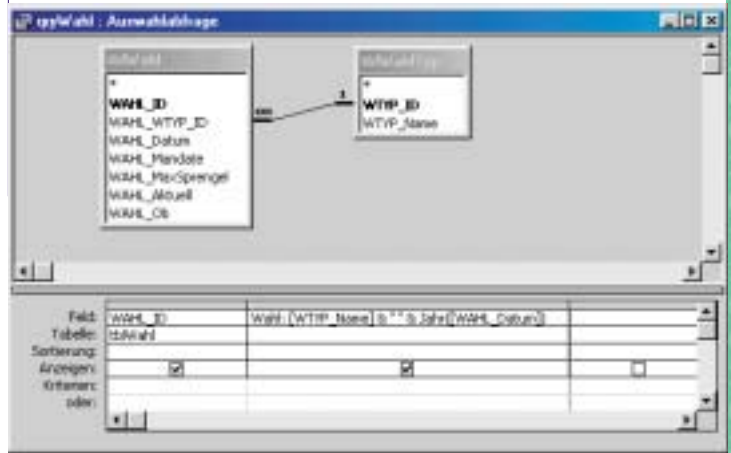


Abbildung -8: Entwurf der Abfrage qryWahl

dem in die Formulare eingegeben. Wenn die Beziehungen vorher definiert wurden, übernimmt sie auch der Formular-Assistent automatisch und generiert für die Nachschlagfelder die Kombinationsfelder statt der einfachen Textfelder.

**Fazit**

Der richtige Entwurf des Datenmodells garantiert die Zuverlässigkeit und Wartungsfreundlichkeit der Applikation. Das Programm selbst ist damit natürlich noch nicht fertig. Die nächste Aufgabe ist, die Präsentationsschicht der Applikation aufzubauen. Darunter gehören in erster Reihe die Abfragen, die die notwendigen Daten in gewünschter Form zur Verfügung stellen und dann die Objekte der Benutzeroberfläche, die Formulare, Diagramme und Berichte.

**Spendenaufwurf**



Christian Berger