

## **[1. Einleitung und Problemstellung]**

Beziehungen sind in unserem Leben allgegenwärtig: Menschen verbinden sich mit Menschen, Organisationen mit Organisationen – und Menschen stehen mit Organisationen in Beziehung.

Stellen wir uns vor: Wir entwickeln eine Datenbank für eine Stiftung. Diese Stiftung betreut Pflegekinder und sie unterstützt Personen mit Beistandschaft\*. Sie arbeitet mit Pflegefamilien, Behörden und Sozialarbeitern zusammen.

Und hier haben wir unsere Herausforderung: Wie können wir all diese unterschiedlichen Beziehungen effektiv modellieren und verwalten?

Es wäre ziemlich hilfreich, wenn wir diese Beziehungen so lebensnah wie möglich in unserer Anwendung abbilden könnten.

\* Beistandschaft: Das ist eine gesetzliche Hilfestellung für Menschen, die in einem Aspekt ihres Lebens Unterstützung benötigen, ohne dabei ihre volle Selbstständigkeit aufgeben zu müssen.

## [2. Einführung des Party Role Models und seine Vorteile]

Und hier kommt das Party Role Model ins Spiel.

Es hilft uns, diese komplexen Beziehungen effizient abzubilden!

Es bringt unseren Anwendern eine ganze Reihe von Vorteilen:

- Wir können Daten flexibler verwalten
- Wir vermeiden Datendopplungen
- Wir behalten alle Beziehungen besser im Blick
- Wir können die Daten einfacher pflegen
- Wir erhöhen die Datenqualität und -konsistenz

Nachher werden wir diese Vorteile genauer anschauen und verstehen, wie uns das Party Role Model hilft, das zu erreichen.

Mit diesem Modell passt sich unsere App kontinuierlich an neue Anforderungen an. Das spart Zeit, reduziert Fehler und macht unsere Datenbank zukunftssicher.

### **[3. Detaillierte Erklärung des Party Role Models]**

Schauen wir uns genauer an, wie das Party Role Model funktioniert. Im Kern strukturiert es Daten über Personen und Organisationen auf eine clevere Art und Weise.

In unserem Stiftungs-Beispiel arbeiten wir mit verschiedenen Parteien:

- Die Stiftung selbst
- Behörden wie Jugendamt und Sozialamt
- Pflegekinder
- Pflegefamilien
- Personen mit Beistandschaft
- Betreuer und Sozialarbeiter
- Andere Dienstleister wie Therapeuten und Ärzte

Hier wird's spannend: Jede dieser Parteien kann verschiedene Rollen einnehmen.

#### [4. Praktisches Beispiel]

Nehmen wir **Maria** als Beispiel.

In unserer Datenbank taucht sie auf als:

- Pflegekind
- Schülerin
- Klientin der Stiftung

Gleichzeitig finden wir **Ralf** in unserer Datenbank. Er ist:

- Pflegevater in einer Pflegefamilie
- Ehrenamtlicher Helfer der Stiftung
- Angestellter bei einem lokalen Unternehmen

Auch die **Stiftung** selbst spielt mehrere Rollen:

- Sie dient als Dienstleister für das Jugendamt
- Sie fungiert als Arbeitgeber für Sozialarbeiter
- Sie betreut Pflegekinder und Personen mit Beistandschaft

## **[5. Vergleich mit traditionellem Modell]**

In den meisten unserer FM-Datenbanken würden wir mehrere Einträge für Maria, Ralf oder die Stiftung anlegen müssen, um jede der Rollen abbilden zu können.

Das Party Role Model geht anders vor: Wir erfassen jede Entität nur einmal in der Haupttabelle. Ihre verschiedenen Rollen verknüpfen wir dann als separate Einträge in der PartyRole-Tabelle.

## **[6. Demonstration der Flexibilität]**

Gehen wir nochmal zu unserem Beispiel Maria: Die ist inzwischen volljährig und hat eine Ausbildung zur Sozialarbeiterin gemacht. Und jetzt will sie als Sozialarbeiterin für die Stiftung arbeiten. In einem starren System stünden wir vor einem Problem:

Wir müssten Maria ein zweites Mal in der Datenbank erfassen, diesmal als Mitarbeiterin. Das würde zu Redundanz führen und möglicherweise Inkonsistenzen verursachen.

Mit dem Party Role Model können wir das Problem einfach lösen: Wir fügen Maria lediglich eine neue Rolle hinzu. Wir müssen ihren Haupteintrag nicht duplizieren. Maria bleibt eine einzige Entität in unserer Haupttabelle. Wir ergänzen nur einen neuen Eintrag in der PartyRole-Tabelle für ihre neue Rolle als Mitarbeiterin.

## [7. Implementierung in FileMaker]

Wie setzen wir dieses Party Role Model in FileMaker um?

Wir müssen uns um fünf Dinge kümmern:

**Tabelle Party:** Hier erfassen wir alle Akteure – Pflegekinder, Pflegeeltern, Sozialarbeiter, Behörden, die Stiftung selbst.

(**PartyType:** Hier definieren wir, ob es sich um eine Person oder Organisation handelt.)

**Tabelle Role:** Hier speichern wir alle möglichen Rollen wie Pflegekind, Betreuer, Auftraggeber, Dienstleister.

**Tabelle PartyRole:** Diese Verbindungstabelle verknüpft Partys mit ihren Rollen.

**Tabelle Relationship:** Hier verbinden wir Parteien miteinander, z.B. Pflegekind mit Pflegefamilie, Stiftung mit Jugendamt.

## **[8. Vorteile für Anwender]**

Für die Anwender bedeutet das:

Wir erfassen jede Person oder Organisation nur einmal, unabhängig von der Anzahl ihrer Rollen.

Wir sehen alle Rollen und Beziehungen einer Partei an einem Ort.

Wir fügen jederzeit neue Rollen hinzu, ohne bestehende Daten zu überschreiben oder zu duplizieren.

Wir vermeiden mögliche widersprüchliche Informationen durch mehrfache Datensätze derselben Partei.

Wir suchen und aktualisieren Informationen schneller, da jede Partei nur einmal existiert.

## [9. Erweitertes Konzept: Supertypen und Subtypen]

Jetzt erweitern wir die Geschichte um die Hierarchie von Rollen. Hier kommen zwei wichtige Konzepte ins Spiel: Supertypen und Subtypen.

Stellen wir uns das wie einen Baum vor:

Der Supertyp bildet den Stamm – die übergeordnete, allgemeinere Kategorie.

Die Subtypen verzweigen sich als Äste – die sind spezieller und leiten sich vom Stamm ab.

Ein Beispiel:

Supertyp: 'Betreuer'

Subtypen: 'Sozialarbeiter', 'Pflegeeltern', 'Ehrenamtlicher Helfer'

Oder:

Supertyp: 'Klient'

Subtypen: 'Pflegekind', 'Person mit Beistandschaft'



In FileMaker setzen wir das so um:

Wir fügen in der Role-Tabelle ein Feld 'ParentRoleID' hinzu.

Bei Supertypen lassen wir dieses Feld leer.

Bei Subtypen verweisen wir auf die ID des Supertyps.

So bauen wir eine ganze Hierarchie von Rollen auf, ohne die Grundstruktur zu ändern. Das ermöglicht uns, die komplexen Strukturen und Beziehungen in der Arbeit unserer Stiftung effizient abzubilden und zu verwalten.

## [10. Zusammenfassung]

Zum Abschluss unserer Reise durch das Party Role Model möchte ich die Kernpunkte noch einmal hervorheben:

**Flexibilität:** Wir können komplexe, sich ändernde Beziehungen abbilden, ohne unsere Datenbankstruktur ständig umzubauen.

**Effizienz:** Jede Partei existiert nur einmal in der Datenbank, unabhängig von der Anzahl ihrer Rollen.

**Übersichtlichkeit:** Alle Rollen und Beziehungen einer Partei sind an einem Ort sichtbar.

**Zukunftssicherheit:** Wir können neue Rollen und Beziehungstypen hinzufügen, ohne an bestehenden Daten etwas ändern zu müssen.

**Hierarchische Strukturen:** Mit Super- und Subtypen bilden wir komplexe Organisationsstrukturen ab.

Ja, die Implementierung des Party Role Models erfordert anfangs mehr Planung und Aufwand. Aber für Projekte mit vielfältigen, sich entwickelnden Beziehungen zahlt sich dieser Aufwand aus. Wir gewinnen an Flexibilität und Anpassungsfähigkeit.

Ich möchte euch ermutigen, das Party Role Model für eure nächsten komplexen Projekte in Betracht zu ziehen. Experimentiert damit, passt es an eure Bedürfnisse an. Bestimmt entdeckt ihr dabei neue Möglichkeiten, die wir heute noch gar nicht besprochen haben.

























































































