1. DATENBANKDESIGN

Access ist die führende relationale Datenbank im Desktop-Bereich. Die Einfachheit in der Bedienung täuscht oft etwas darüber hinweg, dass eine Datenbank doch etwas ist, was nicht in fünf Minuten „quick and dirty“ erzeugt wird. Die Erstellung einer relationalen Datenbank erfordert eine genaue Planung. Jede Stunde, die vor der Erstellung der Datenbank in die Planung des Designs derselben investiert wird, kommt später 100-fach zurück.

Wir wollen damit beginnen, uns das relationale Datenbanksystem genauer anzusehen.

RELATIONALES DATENBANKMODELL

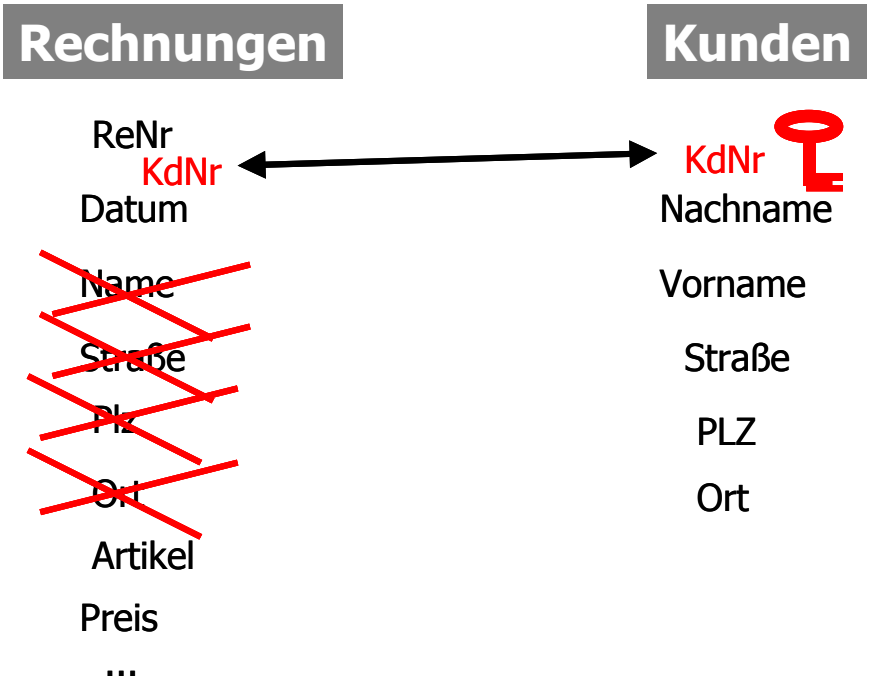
Stellen Sie sich vor, Sie müssen für einen Kunden eine Rechnung erstellen. Auf einer Rechnung werden folgende Informationen benötigt:

- Rechnungsnummer
- Rechnungsdatum
- Name des Kunden
- Strasse des Kunden
- Postleitzahl des Kunden
- Zahlungskonditionen
- Artikel
- Mengen
- Preise
- etc.

Stellen Sie sich folgende Situation vor:

Sie erfassen diese Daten in Word oder Excel und drucken die Rechnung aus. Der Kunde ist von Ihnen so begeistert, dass er zum Stammkunden wird und immer wieder kommt. Jedes Mal erfassen Sie die oben angeführten Daten um eine Rechnung für seinen Einkauf zu erstellen. Am Ende des Jahres haben Sie den Namen und die Adresse 20 Mal erfasst, dabei stoßen Sie nun zusätzlich auf fünf verschiedene Schreibweisen.

Abgesehen davon, dass Sie viel Zusatzarbeit beim wiederholten Erfassen der gleichen Adresse geleistet haben, wissen Sie nun nicht mehr, welche Schreibweise eigentlich die richtige ist.
Wäre es nicht einfacher, bei jeder Rechnung zu vermerken, dass es derselbe Kunde gewesen ist, der das letzte Mal schon hier gewesen ist? – Und genau hier hakt die relationale Datenbank ein.



Aus der Rechnungstabelle werden alle Adressbestandteile entfernt und anstelle dessen in eine eigene Kundentabelle geschrieben. Damit ein Kunde nun einer Rechnung zugeordnet werden kann, benötigen wir ein Feld, das in beiden Tabellen vorhanden und gleich ist, über das eine Verbindung hergestellt werden kann. Deshalb ergänzen wir in der Kundentabelle einen sogenannten Primärschlüssel in Form der Kundennummer. Diese Kundennummer wird ebenso anstelle der entfernten Kundendaten in der Rechungstabelle ergänzt. Wird jetzt eine Rechnung erstellt, wird nur mehr die Kundennummer vermerkt. Die übrigen Informationen werden aus der Kundentabelle "verlinkt".

Eine relationale Datenbank besteht aus mehreren Tabellen, die miteinander in Beziehung stehen, um Redundanzen (doppelte Dateneingaben) zu vermeiden. Ziel ist es, jede Information nur einmal eingeben und speichern zu müssen, aber beliebig oft nutzen zu können.

BEZIEHUNGSTYPEN

In einer relationalen Datenbank tauchen drei Beziehungstypen auf:

1:N – Beziehung

In einer 1:N-Beziehung gehören zu einem Datensatz in der so genannten Mastertabelle beliebig viele Datensätze in der so genannten Detailtabelle. Jeder Datensatz aus der Detailtabelle ist genau einem Datensatz in der Mastertabelle zugeordnet.
Beispiele:

- Jedem *Kunden* (Master) können mehrere *Rechungen* (Detail) zugewiesen werden, jeder *Rechnung* jedoch nur ein *Kunde*.
- Jedes *Elternpaar* (Master) kann mehrere *Kinder* (Detail) haben, jedes *Kind* aber nur ein *Elternpaar*.

1:1 – Beziehung

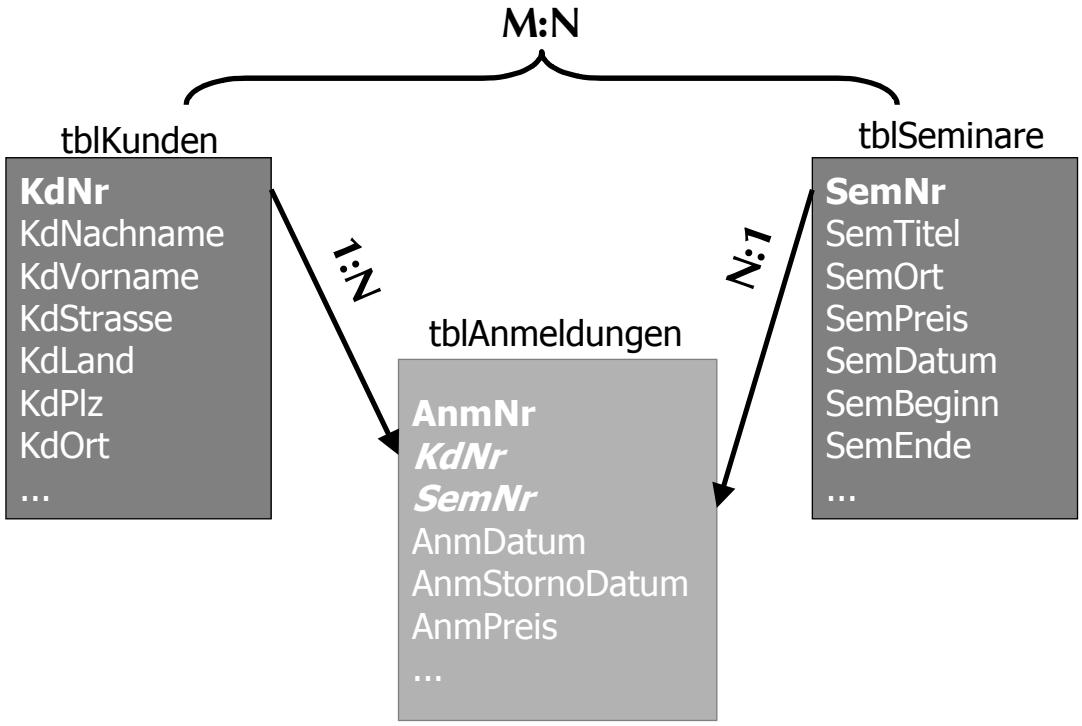
Bei einer 1:1-Beziehung gibt es zu jedem Datensatz in der einen Tabelle nur einen entsprechenden Datensatz in der anderen Tabelle und umgekehrt. Oft stellt sich die Frage, ob man nicht nur eine Tabelle daraus machen kann. Eine Trennung in zwei Tabellen ist oft sinnvoll, wenn diese Daten organisatorisch getrennt sind oder einen anderen zeitlichen Bestand haben.
Beispiele:

- Jeder *Posten* wird von einem *Mitarbeiter* ausgefüllt, jeder *Mitarbeiter* bekleidet einen *Posten*. (Zwischendurch kann ein Posten auch vakant sein.)
- Jeder *Außendienstmitarbeiter* fährt ein eigenes *Firmenfahrzeug*, jedes *Firmenfahrzeug* wird nur von einem *Außendienstmitarbeiter* gefahren.

M:N – Beziehung

Eine M:N-Beziehung existiert zwischen zwei Tabellen, wenn es für Datensätze in beiden Tabellen mehrere Entsprechungen in der jeweils anderen Tabelle geben kann.

Da eine M:N-Beziehung in der Datenbank nicht direkt umgesetzt werden kann, wird eine Zwischentabelle eingesetzt, welche die Beziehung in eine doppelte 1:N-Beziehung auflöst. Dabei ist zu beachten, dass die beiden ursprünglichen Tabellen immer die Mastertabellen bleiben und die neue Zwischentabelle die Detailtabelle für beide wird.



Beispiele:

- Jeder *Kunde* (Master) kann mehrere *Seminare* (Master) besuchen, jedes *Seminar* kann von mehreren (hoffentlich!) *Kunden* besucht werden. Zwischentabelle: z.B. *Anmeldung* (Detail für beide)
- Jeder *Artikel* (Master) kann auf mehreren *Rechnungen* (Master) aufscheinen, auf jeder *Rechnung* können mehrere *Artikel* enthalten sein. Zwischentabelle: *Rechnungspositionen* (Detail für beide)

SCHLÜSSEL

Um sinnvoll Beziehungen zwischen Tabellen aufbauen zu können, werden Schlüssel benötigt. Der Schlüssel wird immer für ein Feld einer Tabelle oder mehrere Felder gemeinsam vergeben.

Es werden drei Arten von Schlüssel unterschieden:

- *Primärschlüssel*: Ein Primärschlüssel sollte in jeder Tabelle vergeben werden. Die Felder eines Primärschlüssels dürfen *nicht leer (Null)* und müssen *eindeutig* sein. Ein Primärschlüssel ist für das sinnvolle Erstellen von Beziehungen notwendig. Ohne einen Primärschlüssel in der Mastertabelle kann keine Beziehung hergestellt werden, die den Regeln einer korrekten relationalen Datenbank entspricht.
- *Sekundärschlüssel*: Ein Sekundärschlüssel ist jeder weitere Schlüssel einer Tabelle. Jeder Schlüssel, der kein Primärschlüssel ist, wird in der Datenbanktheorie als ein Sekundärschlüssel bezeichnet. Dieser wird in Access nicht explizit mit dieser Bezeichnung verwendet.
- *Fremdschlüssel*: Als Fremdschlüssel wird jener Schlüssel bezeichnet, der auf den Primärschlüssel einer anderen Tabelle im Rahmen einer Beziehung zugreift.

Wird in Access eine Beziehung ohne einen Primärschlüssel oder wahlweise einem eindeutigen Index erstellt, kann für diese Beziehung keine referentielle Integrität (siehe dazu Seite 23) festgelegt werden. Dies wäre also eine „halbe Sache“ und daher absolut nicht zu empfehlen. Es stellt sich dann nämlich auch die Frage wozu überhaupt eine Beziehung? Eine saubere Beziehung wird in einer relationalen Datenbank immer mit einem Primärschlüssel in der Mastertabelle sowie mit referentieller Integrität gebildet. Eine Beziehung ohne diese Merkmale käme einem Auto gleich, das keine Bremsen besitzt – natürlich fährt es auch, aber sehr sicher ist dies wohl nicht.

INDEX

So wie ein Index in einem Buch die Suche nach einem Eintrag beschleunigt, beschleunigt ein Index für ein Feld die Suche in diesem.

Die Vorteile eines Indexes sind, dass

- die Suche nach Werten in dem indizierten Feld beschleunigt wird und
- Sortiervorgänge schneller vonstatten gehen.

Die Nachteile eine Indexes sind, dass

- der Speicherplatzbedarf der Datenbank steigt und
- Einfüge- und Änderungsvorgänge verlangsamt werden.

In einer Tabelle können mehrere Indizes vergeben werden. Sinn macht es jedoch nur für jene Felder, die in Beziehungen involviert sind und die häufig für Auswahlkriterien verwendet werden.

In Access gibt es zwei Arten von Indizes:

- *Index ohne Duplikate*
  In Feldern mich einem solchen Index werden keine mehrfach vorkommenden Werte erlaubt. Dieser Index wird automatisch bei der Vergabe eines Primärschlüssels in einer Tabelle für das Primärschlüsselfeld vergeben.
- *Index mit Duplikaten*
  Dieser Index wird für Felder vergeben, die zwar mehrfach gleiche Werte zulassen sollen (zum Beispiel Nachname, Postleitzahl, …), aber häufig für Suchkriterien verwendet werden.

NORMALISIERUNG

Die Normalisierung nach Dr. Codd, dem Erfinder des relationalen Datenbanksystems, ist ein Regelwerk, mit dem ein vorläufiger Datenbankentwurf auf seine Korrektheit geprüft wird. Sie können es auch als ein Checklistenverfahren bezeichnen. Es gibt einige Überarbeitungen dieser so genannten *Normalformen*, aber in der Grundform verwendet man drei Normalformen.

Anhand der Normalformen kann überprüft werden:

- Habe ich für meine Tabellen die richtigen Felder geplant?
- Benötige ich noch zusätzliche Tabellen in meinem Datenbankentwurf?

Erste Normalform

Jedes Feld einer Tabelle muss atomar und einwertig sein. (nicht mehr weiter unterteilbar)

ID	Name	Adresse	Email	Interessen
1	Frau Mag. Petra Deutschmann	Birkengasse 2, A-8020 Graz	petzi@aonline.at	Sport, Literatur, Musik
2	Herr Heimo Konrad	Weinsteig 3, A-8010 Graz	heimo@quality.at	Italien, Sport
3	Frau Dr. Ingeborg Preiss	Heldenstr. 17, A-8071 Hausmannstätten	inge@ahs.at	Sport, Familie, Fotografie
4	Frau Edith Neubauer	Rosenweg 22, A-8020 Graz	edita@rivercorner.com	Architektur, Garten

Die Spalten *Name*, *Adresse* und *Interessen* entsprechen nicht der ersten Normalform:

Name und *Adresse* müssen jeweils in mehrere Spalten unterteilt werden. (In der Praxis werden Straße und Hausnummer hingegen häufig in einer Spalte zusammengefasst, da eine separate Hausnummer nur dann sinnvoll ist, wenn man Sie für Sortier- und Selektionsvorgänge benötigt.)

ID	Name	Adresse	Email	Interessen
1	Frau Mag. Petra Deutschmann	Birkengasse 2, A-8020 Graz	petzi@aonline.at	Sport, Literatur, Musik
2	Herr Heimo Konrad	Weinsteig 3, A-8010 Graz	heimo@quality.at	Italien, Sport
3	Frau Dr. Ingeborg Preiss	Heldenstr. 17, A-8071 Hausmannstätten	inge@ahs.at	Sport, Familie, Fotografie
4	Frau Edith Neubauer	Rosenweg 22, A-8020 Graz	edita@rivercorner.com	Architektur, Garten

ID	Vorname	Nachname	Titel	Geschl	Strasse	Hausnr	Land	Plz	Ort	Email
1	Petra	Deutschmann	Mag.	1	Birkengasse	2	A	8020	Graz	petzi@aonline.at
2	Heimo	Konrad		2	Weinsteig	3	A	8010	Graz	heimo@quality.at
3	Ingeborg	Preiss	Dr.	1	Heldenstraße	17	A	8071	Hausmannstätten	inge@ahs.at
4	Edith	Neubauer		1	Rosenweg	22	A	8020	Graz	edita@rivercorner.com

Die Spalte *Interessen* ist mehrwertig, da für dieses Attribut für jeden Datensatz mehrere Ausprägungen möglich sind. Daher muss diese Spalte entfernt und dafür eine eigene Tabelle erstellt werden. In dieser werden die Interessen aufgeteilt und über die ID mit der ursprünglichen Tabelle verknüpft.

ID	Name	Adresse	Email	Interessen
1	Frau Mag. Petra Deutschmann	Birkengasse 2, A-8020 Graz	petzi@aonline.at	Sport, Literatur, Musik
2	Herr Heimo Konrad	Weinsteig 3, A-8010 Graz	heimo@quality.at	Italien, Sport
3	Frau Dr. Ingeborg Preiss	Heldenstr. 17, A-8071 Hausmannstätten	inge@ahs.at	Sport, Familie, Fotografie
4	Frau Edith Neubauer	Rosenweg 22, A-8020 Graz	edita@rivercorner.com	Architektur, Garten

ID	Interesse
1	Sport
1	Literatur
1	Musik
2	Italien
2	Sport
3	Sport
3	Familie
3	Fotografie
4	Architektur
4	Garten

Zweite Normalform

Jedes Feld einer Tabelle muss vom gesamten Primärschlüssel abhängig sein. (Dieser kann aus mehreren Spalten bestehen.)

ReNr	RePos	RePArtikel	RePMenge	RePPreis	ReDatum
200001	1	1005	1	86,99	16.01.2002
200001	2	1035	1	61,70	16.01.2002
200001	3	1082	3	16,35	16.01.2002
200002	1	1027	1	18,42	17.01.2002
200002	2	1065	2	21,58	17.01.2002
200002	3	1088	1	10,10	17.01.2002
200002	4	1013	5	7,12	17.01.2002
200003	1	1004	1	2,07	18.01.2002
200003	2	1079	1	31,25	18.01.2002
200003	3	1055	3	2,91	18.01.2002

Abhängig sein bedeutet, durch den Primärschlüssel eindeutig identifiziert zu werden. Dies gilt für Artikelnummer, Menge und Preis auf einer Rechnungsposition. Um diese Informationen genau zuordnen zu können, muss man sowohl die Rechnungsnummer als auch die Positionsnummer kennen.

Die Spalte *ReDatum*, die das Rechnungsdatum enthält, kann aber alleine durch eine Zuordnung zur Rechnungsnummer identifiziert werden. Somit ist sie nur von einem Teil des Primärschlüssels abhängig und gehört nicht in diese Tabelle, sondern in die Rechnungstabelle, in der die *ReNr* alleinige Primärschlüsselspalte ist.

ReNr	RePos	RePArtikel	RePMenge	RePPreis	ReDatum
200001	1	1005	1	86,99	16.01.2002
200001	2	1035	1	61,70	16.01.2002
200001	3	1082	3	16,35	16.01.2002
200002	1	1027	1	18,42	17.01.2002
200002	2	1065	2	21,58	17.01.2002
200002	3	1088	1	10,10	17.01.2002
200002	4	1013	5	7,12	17.01.2002
200003	1	1004	1	2,07	18.01.2002
200003	2	1079	1	31,25	18.01.2002
200003	3	1055	3	2,91	18.01.2002

nur von *ReNr* abhängig

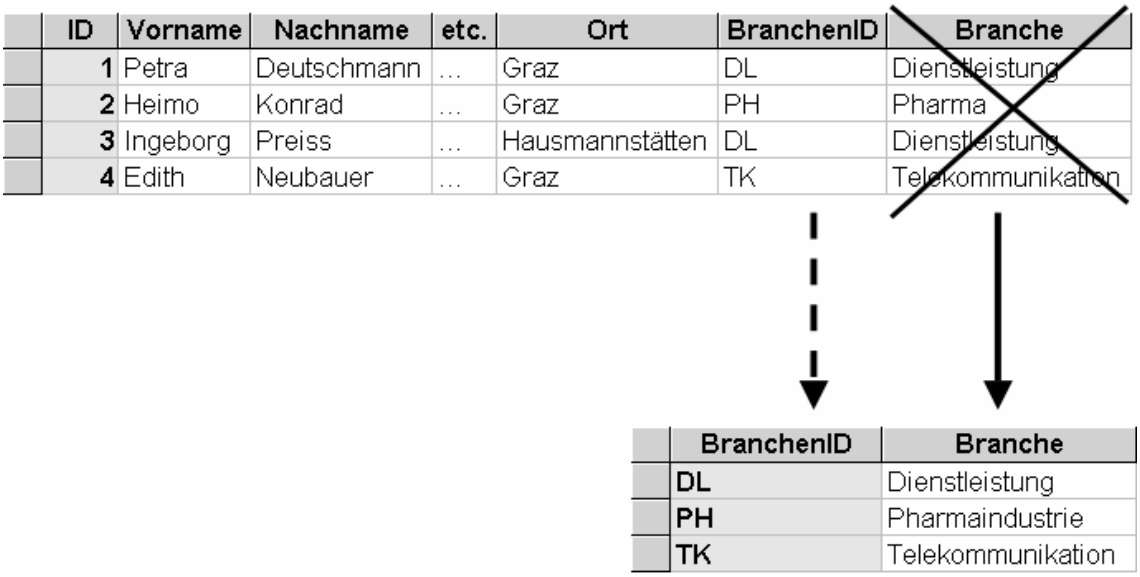
Dritte Normalform

Jedes Feld einer Tabelle muss vom Primärschlüssel und nicht von einer Nicht-Schlüsselspalte anhängig sein.

ID	Vorname	Nachname	etc.	Ort	BranchenID	Branche
1	Petra	Deutschmann	...	Graz	DL	Dienstleistung
2	Heimo	Konrad	...	Graz	PH	Pharma
3	Ingeborg	Preiss	...	Hausmannstätten	DL	Dienstleistung
4	Edith	Neubauer	...	Graz	TK	Telekommunikation

Vorname, *Nachname* und andere Daten wie der *Ort* und die *BranchenID* der Branche, der die Person angehört, sind vom Primärschlüssel *ID* abhängig.

Die *Branche* hingegen ist nicht von der *ID*, sondern von der *BranchenID* abhängig, welche kein Primärschlüssel und auch kein Teil eines Primärschlüssels ist. Für die Branchen muss eine eigene Tabelle angelegt werden, die mit der ursprünglichen Tabelle verknüpft wird. Der Nicht-Schlüssel wird dabei in der Ursprungstabelle zum Fremdschlüssel und in der neuen Tabelle zum Primärschlüssel.



ID	Vorname	Nachname	etc.	Ort	BranchenID	Branche
1	Petra	Deutschmann	...	Graz	DL	Dienstleistung
2	Heimo	Konrad	...	Graz	PH	Pharma
3	Ingeborg	Preiss	...	Hausmannstätten	DL	Dienstleistung
4	Edith	Neubauer	...	Graz	TK	Telekommunikation

BranchenID	Branche
DL	Dienstleistung
PH	Pharmaindustrie
TK	Telekommunikation

Hinweis:

Die drei Normalformen sind immer kumulativ zu sehen:

- *Der Entwurf ist in zweiter Normalform, wenn er in erster Normalform ist und ...*
- *Der Entwurf ist in dritter Normalform, wenn er in zweiter Normalform ist und ...*

Persönliche Notizen:

DATENBANKMODELL

In der Praxis ist es sinnvoll, das Datenbankmodell für die geplante Datenbank in schriftlicher Form festzuhalten. Dies hat den Vorteil, dass

- Sie selber einen besseren Überblick über die Datenbank bekommen,
- Designfehler leichter erkannt werden können und
- zugleich eine Dokumentation der Datenbank erstellt worden ist.

Vor allem, wenn mehrere Personen an einer Datenbank gemeinsam arbeiten, ist das Erstellen eines Datenbankdiagramms unumgänglich.

In einem Datenbankdiagramm zeichnen Sie für jede Tabelle ein Kästchen, in das der Name der Tabelle sowie der Primärschlüssel vermerkt wird. Die Beziehungen zu anderen Tabellen werden durch Pfeile dargestellt, wobei die Pfeile immer von der Master- zur Detailtabelle zeigen. 1:1-Beziehungen kennzeichnen Sie durch einen Doppelpfeil. Zusätzlich tragen Sie neben der Linie den Beziehungstyp ein, wobei Sie wiederum darauf achten müssen, dass der 1er auf jener Seite geschrieben wird, die näher zur Mastertabelle liegt.

CASE-Tools

In der Praxis können so genannte CASE-Tools das Erstellen eines Datenbankdiagramms vereinfachen. Dies sind grafische Tools, mit denen nicht nur ein Diagramm einfach „gezeichnet“ werden kann, sondern auch die Datenbank direkt aus dem Diagramm generiert werden kann. Dies erspart doppelte Arbeit.

Bekannte Programme sind MS Visio, ERWin DataModeler, Dezign for Databases oder Embarcadero.

Aufgabe:

Ergänzen Sie gemeinsam mit Ihrem Trainer das nachfolgende Datenbankmodell für eine Personalanwendung:

Der Personalschef benötigt eine Datenbank, mit der er den Ausbildungsstand seiner Mitarbeiter überwachen kann, und entscheidet, auf welche Seminare er diese schickt. Damit er wissen kann, welche Aus-

bildungen jeder Mitarbeiter benötigt, muss ihm die Datenbank eine Übersicht darüber liefern, welche Hard- und Software welcher Mitarbeiter im Einsatz hat. Dass die Dienstverträge und die Stellen im Unternehmen verwaltet werden müssen, versteht sich von selbst. Die Mitarbeiter haben weiters die Möglichkeit, für auswärtige Termine auf den Fuhrpark des Unternehmens zurückzugreifen. Auch diese Verwaltung soll über die Anwendung laufen.

Dazu werden folgende Tabellen verwendet:

- Mitarbeiter
- Hardware
- Software
- Lizenz
- Seminare
- Seminaranbieter
- Seminarteilnahme
- Fahrzeuge
- Fahrzeugbenutzung
- Stelle
- Dienstvertrag