

Universität Fribourg

Fakultät für Wirtschafts- und Sozialwissenschaften

Betreuung: Andreea Jonas

Dozent: Prof. Dr. Andres Meier

Sommersemester 2004

# Informationsmanagement durch Kreierung einer Access-Datenbank

vorgelegt von: Stocker Fabian  
2. Semester Wirtschaftsinformatik  
Matrikelnummer: 03-214-095  
E-Mail: [fabian.stocker@unifr.ch](mailto:fabian.stocker@unifr.ch)  
Tel. 026-424 24 75 od: 078-819 68 71

## Kurzfassung

Diese Arbeit befasst sich hauptsächlich mit der Erstellung, aber auch mit der Benutzung einer selbst entworfenen Datenbank für einen Warenshop. Dieser Warenshop verfügt über unterschiedliche Mitarbeiter, welche Produkte bei Lieferanten kaufen und dann an Kunden weiterverkaufen. Das Ziel dieser Arbeit ist es, den Mitarbeitern eines solchen Shops das Leben durch effizientere Prozesse zu erleichtern. Ausgestattet mit sinnvollen SQL-Abfragen, soll die Datenbank zusätzlich als Unterstützung für das Management dienen.

Im ersten, von gesamthaft drei Überkapiteln, geht es um die Schaffung eines Datenbank-Modells, welches einem solchen Warenbetrieb gerecht wird. Es beinhaltet die Datenanalyse, die Schaffung eines Entitäten-Beziehungsmodells und die Überführung desselben in ein relationales Datenbankschema. Das zweite grosse Kapitel befasst sich mit der Implementierung dieses erarbeiteten Modells in dem Datenbank-Programm MS Access. Es wird durch Einblicke in die Funktionsweise von Access und weiteren Erklärungen ergänzt. Zum Schluss gibt es noch ein paar Hinweise zum Datenschutz und zur Datensicherheit.

Schlüsselwörter: ERM, Relationales Datenbankschema, MS Access, Datensicherheit, Datenschutz

# Inhaltsverzeichnis

1. Einleitung	5
1.1 Problemstellung	5
1.2 Zielsetzung	6
1.3 Vorgehensweise	6
2. Von der Idee zum Modell	7
2.1 Die Datenanalyse	7
2.2 Das Entitäten-Beziehungsmodell	11
2.3 Das Relationale Datenbankschema	12
3. Umsetzung in Access	14
3.1 Tabellen	15
3.2 Relationships	16
3.3 Werte eingeben	17
3.4 SQL-Abfragen	18
3.5 Management durch SQL	19
3.6 Entstehung von Formularen ( <i>Forms</i> )	21
3.7 Benutzeroberfläche - das <i>Mainswitchboard</i>	25
3.8 Datenschutz	26

4. Zur Datensicherheit	26
4.1 physikalische Datensicherheit	26
4.2 hardwaretechnische Datensicherheit	27
4.3 softwaretechnische Datensicherheit	27
5. Schlusswort	28
6. Literaturangaben	29
7. Abbildungsverzeichnis	29

# 1. Einleitung

In der heutigen Zeit der technischen Entwicklung ist es in nahezu jedem Betrieb möglich, die Effizienz durch den Einsatz von Computersystemen zu steigern. In den meisten Fällen wird dadurch der Mensch nicht ersetzt, sondern lediglich seine Arbeit um ein Vielfaches erleichtert. Genauso ist es beim Einsatz von Datenbanksystemen. Sie ermöglichen durch wenige Handgriffe, gezielt an wichtige Informationen heranzukommen, die sonst mühselig protokolliert und zusammengesucht werden müssten. Zusätzlich sind manche Zusammenhänge so komplexer Natur, dass sie ohne vernetzte Datenlagerung gar nicht einsehbar wären. Weitere Vorteile von DB sind, dass Daten geordnet gelagert und immer wieder verändert werden können. So können z.B. auch vorliegende Fehler jederzeit gezielt ausgemerzt werden. Und wenn mal neue Aspekte dazugefügt werden müssen, eventuell ein neuer Geschäftszweig, dann würde dies ganz sicher auch kein Problem darstellen.

Da sich diese Arbeit um die Kreation einer Access Datenbank für einen *Warenshop* handelt, ist folgendes zu beachten:

Das Beispiel eines Warenshops stellt nicht ein allgemeingültiges Modell zur Entwicklung dar. Die Struktur ist natürlich bei jedem Unternehmen verschieden und benötigt andere Gedankenschritte. Somit sollte man diese Arbeit eher als *Fallstudie* ansehen, die das Problem lediglich für bestimmte Problemstellungen löst, nämlich die eines Warenshops.

Wichtig ist es auch noch zu sagen, dass die Datenbank nicht realistisch viele Mitarbeiter und Produkte, wie auch Verkäufe und Bestellungen enthält. Dies einfach aus folgendem Grund: es wäre einfach zu mühsam und gleichsam nutzlos gewesen, viele weitere Tupel hinzuzufügen. Sicher ist aber, dass durch die ledigliche Erhöhung der Zahlen kein besseres Verständnis des Modells erreicht worden wäre.

## 1.1 Problemstellung:

Datenbanksysteme sind zwar sehr nützlich. Zuerst müssen sie aber entwickelt und an den Betrieb angepasst werden. Zur Problemstellung gehören folgende Fragen:

Wie kann jemand, der keine Erfahrung mit Datenbanken hat, eine solche entwerfen, editieren und für einen kleinen Warenshop verwenden?

Welche SQL-Abfragen benötigt das Management, um nützliche Aussagen über das Unternehmen machen zu können?

In welche Bereiche der Datenbank sollte der Kunde und in welche sollte der Mitarbeiter bzw. das Kader Einblick haben?

## 1.2 Zielsetzung

Diese Arbeit soll den ganzen Prozess der Datenbankerstellung beinhalten. Sie sollte quasi als Leitfaden dazu dienen. Dem Leser sollte es danach möglich sein, eine Datenbank zu entwerfen und auch wirkungsvoll in Access umzusetzen.

Dabei kann es nicht schaden, das Buch "Relationale Datenbanken" von Andres Meier zusätzlich zu konsultieren. Was die Zielsetzungen der Datenbank betrifft, ist folgendes zu beachten:

Die Datenbank soll den Mitarbeitern, wie auch den Kunden, Vorteile bringen. Den Verkäufern soll sie insofern eine Hilfe sein, als dass sie lästige Schreibarbeiten in Zeitersparnisse umwandelt. Das Management soll von einer ständig verfügbaren Informationsbasis profitieren können, auf der sie fundierte Entscheidungen treffen kann. Kunden soll es möglich sein, sich ein Bild über den Betrieb zu machen und Informationen zu sammeln welche ihre Verkaufsentscheidung positiv beeinflussen.

## 1.3 Vorgehensweise

Durch systematische Strukturierung, soll eine gewisse Übersicht und Logik entstehen, welche für den Leser nachvollziehbar ist. Da die Arbeit in einzelne Schritte der Datenbankgestaltung gegliedert ist, kann der Leser jeden Schritt für sich alleine betrachten. Dies ermöglicht einen grösseren Überblick. Viele Beispiele und Bilder sollen zusätzlich das Verständnis steigern und Interesse wecken. Im Kapitel über Access wird versucht auch die kleinen Schritte detailliert darzustellen, damit der Leser sie begreifen kann.

## 2. Von der Idee zum Modell

Jedes Projekt basiert auf einer Idee, wenn möglich auf einer guten Idee. Noch besser, einer gut ausformulierten Idee, die schon sehr viele Vorstellungen in sich birgt. Denn in der Anfangsidee steckt oftmals auch schon das Ziel, welches man zu erreichen hofft. Da diese Idee also für das Projekt entscheidend ist, sollte man sie sich gründlich durch den Kopf gehen lassen. Im Fallbeispiel des Warensshops benötigt man zuerst ein Konzept. Nun, wie beginnen?

Ein guter Ansatzpunkt ist es zunächst den Betrieb zu definieren. Es muss versucht werden ihn möglichst realistisch abzubilden. Dies setzt eine detaillierte Kenntnis seiner Strukturen voraus. Man muss wissen wie der Betrieb aufgebaut ist, wie er funktioniert und wie er sich im Laufe der Zeit verändert.

Die folgenden Abschnitte zeigen verschiedene Bereiche dieses Betriebs. In jedem Absatz wird geklärt, welche Elemente der jeweiligen Bereiche in die Datenbank aufgenommen werden sollten und vorallem wesshalb.

Bemerkung: Die, im folgenden Abschnitt, *kursiv* geschriebenen Wörter sind jeweils Merkmale von den späteren Tabellen.

### 2.1 Die Datenanalyse

Mitarbeiter:

Es wird angenommen, dass dieser Warensshop über mehrere Mitarbeiter verfügt, die in verschiedenen Funktionen tätig sind. Da gibt es zum einen den Chef, dem die Gesamtleitung und Koordination als Aufgabe zufällt. Daneben existiert eine Produktmanagerin, die für die Lieferantenbeziehungen (bestehende behalten und andere neugewinnen), sowie für die Auswahl des Produktsortimentes verantwortlich ist. Die Verkaufsleiterin hat schliesslich zur Aufgabe, kontinuierlich bei Bestandesänderungen die gegen Null gehen, Nachbestellungen zu machen, so dass der Kunde meistens ein vollständiges Produktesortiment vorfindet. Die Nachbestellungen sollten möglichst so organisiert sein, dass es zu keinem Warenüberschuss und dadurch zu verderbter Ware kommt. Verderbbare Ware wie etwa Brot oder Salat müssen also täglich und bloss in kleinen Mengen, die auf Verkaufserwartungswerten beruhen, eingekauft werden.

Aus dem oben gesagten wird klar, dass die *Funktionen* der Mitarbeiter in der Datenbank festzuhalten sind, um, aus der Sicht eines Managers Steuerungen vornehmen zu können. Daneben brauchen sie auch alle eine eindeutige *Mitarbeiternummer* die als Primärschlüssel und zu Identifizierungszwecken dient. Dass *Name*, *Vorname* und *Adresse* der Mitarbeiter in der Datenbank enthalten sein sollten, ist wohl offensichtlich, denn es muss immer möglich sein

Kontakt zu den Mitarbeitern herzustellen. Wichtig scheinen aber auch noch weitere Informationen, wie etwa *Einstellungsdatum* oder *Berufliche Vergangenheit* zu sein. So können obere Instanzen (Chef) jederzeit, ohne ständiges Nachfragen zu den gewünschten Informationen kommen.

Produkte:

Da es sich um einen Warenshop handelt, ist es offensichtlich, dass *Produkte* ein wichtiger Bestandteil der DB sein müssen. Versehen mit einer *Produktenummer* ist jedes Produkt eindeutig markiert und identifiziert. Den richtigen "Kontakt" zu Kunden stellt aber frühstens der *Produktname* her, der dadurch eine grosse Bedeutung erlangt. Er ist es, den die Kunden zuerst lesen, beim zweiten Blick kennen oder nicht kennen und dann kaufen oder nicht kaufen. Deshalb ist es auch wichtig, unter welchem Namen man das Produkt in die Datenbank setzt. Hat man zum Bsp. das Süssgetränk "Sprite" in seinem Sortiment, so schreibt man bestimmt nicht "Zitronenwasser" in die Datenbank. Des weiteren sollte man angesichts der Vielzahl an Produkten, diese in verschiedene *Kategorien* unterteilen, wie Fleischwaren, Früchte/Gemüse, Brot/Backwaren, Getränke, Molkerei und Lebensmittel allgemein. Grenzen sollten so gesetzt werden, dass Kategorien entstehen, die etwa eine gleich grosse Anzahl an Produkten enthalten. Ist es nötig eine neue Rubrik dazuzufügen, so sollte dies nur geschehen, wenn die Produkteanzahl den anderen Kategorien würdig ist. Um nicht bei jedem Produkt erneut den Kategorienamen notieren zu müssen, bildet man eine eigene Tabelle genannt *Kategorien*. Sie enthält lediglich eine *KategorienNr*, den *Kategorienamen* und eine kleine *Beschreibung*, damit man bei neuen Produkten direkt sieht, in welche Kategorie diese gehören.

Ein weiteres Merkmal in der Tabelle der Produkte, ist die *LieferantenNr*. Nur wenn man weiss, woher welche Produkte stammen, kann man die Bedeutung von Lieferanten richtig einschätzen (für Produktmanagerin wichtig).

Informationen für den Kunden, sind die Angaben *Menge pro Einheit* und *Preis*. Sie geben dem Kunden die nötige Transparenz und sind absolut unverzichtbar, denn nur so kann der Kunde direkte Vergleiche vornehmen. Dies führt zu einem grösseren Vertrauen.

Während der *Verkaufspreis* für den Kunden unabdingbar ist, ist die Differenz des *Einkaufspreises* und des *Verkaufspreises* für das Unternehmen enorm wichtig. Verbunden mit der Verkaufstabelle und somit der verkauften Mengenzahl pro Produkt, lässt sich sodann der Unternehmensgewinn berechnen. (Bemerkung: bei den Berechnungen wird davon ausgegangen, dass der Einkaufspreis konstant bleibt). Schliesslich sind auch noch die *Mengen im Lager* wichtig und zwar zur unumgänglichen Lagerhaltung und Erhaltung durch die Verkaufsleiterin. Immer wieder aktualisiert, z.B durch Inventuren, lassen sich so die Warenbewegungen problemlos kontrollieren.

## Lieferanten:

Die Lieferanten sind das Standbein eines jeden Warenunternehmens das keine eigenen Produkte verkauft. Deshalb sollte man immer versuchen die Kontrolle über die Lieferanten zu erhalten. Im Fallbeispiel hat man ganz verschiedene Arten von Lieferanten. Kleinlieferanten wie der lokale Metzger, Bäcker oder Konditor, bringen zwar häufig Waren aber in kleiner Menge pro Lieferung. Ihre Bedeutung für den Betrieb hält sich also im Grenzen. Dagegen sind Grosslieferanten wie Nestlé oder Coca-Cola AG für den Betrieb überlebenswichtig. Solche Grosslieferanten liefern nämlich sehr viele Produkte und ihre Mengen übersteigen die der Kleinlieferanten erheblich.

Der erste Schritt, um eine Übersicht über die Lieferanten zu gewährleisten, ist die Integrierung in diese Datenbank. Mit *Lieferantenummer, Lieferantename, Funktion, Adresse, Ort* und Kontakt (*Telefon, Internet*) ist das gleiche Verhältnis wie zu den Mitarbeitern geschaffen. Denn die Lieferanten sind auch eine Art Mitarbeiter, einfach in weiterem Sinne. Lieferanten müssen also wie die Mitarbeiter, immer erreichbar sein, um bei Problemen schnell eingreifen zu können. Das Merkmal *Art der Lieferung* dient zur Information der Betriebsleitung und somit der Organisation der Bestellungen. Ein weiterer guter Punkt, stellt die *Priorität* dar. Dadurch wird sofort signalisiert, dass Beziehungen zu diesem Lieferanten entweder sehr wichtig sind oder im Grenzfall gar auf sie verzichtet werden kann.

## Verkäufe:

Eine der wichtigsten Tabellen stellt aber definitiv jene für die Produktverkäufe dar. Wenn ein Unternehmen wenig verkauft, ist es ein schlechtes Unternehmen. Wenn es die Verkäufe aber nicht dauernt protokolliert und man somit die Übersicht über sie nicht behält, ist es ebenfalls ein schlechtes Unternehmen. Effizient ist es nur, wenn es die Verkäufe bis zu einem gewissen Grad kontrollieren kann. Z.B. kann schon die Angabe des Verkaufsdatums und Verkaufszeit, zusammengefasst in einer schönen Grafik, Rückschlüsse auf das Konsumverhalten der eigenen Kunden erlauben. Führt man empirische Untersuchungen über einen langen Zeitraum durch, so kann man dann mit einer gewissen Wahrscheinlichkeit voraussagen, wie viel man an einem bestimmten Tag, und zu einer bestimmten Uhrzeit verkaufen wird. Dieses ganze Verfahren setzt natürlich Kunden voraus, die regelmässig in diesem Geschäft einkaufen und somit annäherungsweise ein periodisches Einkaufsverhalten aufweisen. Um diese "Stammkunden", von den selten kaufenden zu unterscheiden, haben viele grosse Warenmärkte wie Migros oder Coop ein Kartensystem eingeführt. Dieses Kartensystem ist so ausgelegt, dass sich der Aufwand einer Kartenbestellung wahrscheinlich schon rein zeitlich nur für einen regelmässig einkaufenden Stammkunden lohnt. So abgegrenzt ist es nun möglich schon vorherzusehen, zu welchem Zeitpunkt

welche und wie viele Waren im Regal zu stehen zu haben, weil dann der Kundenansturm am grössten sein wird. Ein solches System ist sehr sinnvoll und kann ohne weiteres in einer Datenbank realisiert werden. Durch Zufügen des Merkmals *Karteninhaber* (also wahrscheinlich Stammkunde), mit den Merkmalsausprägungen Ja/Nein, wird bei jedem Verkauf festgehalten ob der jeweilige Kunde ein Stammkunde ist, somit regelmässig kommen wird und vielleicht ähnliche Produkte wieder einkaufen wird.

Was gehört nun an dieser Stelle sonst noch in die Datenbank? Zunächst die *VerkaufsNr* als Primärschlüssel. Dann die Tatsache welches Produkt verkauft wurde, also die *ProduktNr*. Auch festgehalten werden muss die verkaufte Menge, *Absatzmenge* genannt. Durch eine SQL-Abfrage verbunden mit dem jeweiligen Produktpreisgewinn lassen sich dann ganz einfach die Gewinne berechnen. Und wie schon oben erwähnt ermöglichen die Angaben *Verkaufsdatum*, *Verkaufszeit* und *Karteninhaber*, den Gang des Unternehmens teilweise zu kontrollieren.

Es gibt aber auch Möglichkeiten nicht nur den Kunden, sondern auch die Mitarbeiter zu kontrollieren. Wird bei jedem Verkauf zusätzlich die Verkaufende Person hinzugefügt, ist es möglich zu sehen, wer produktiv arbeitet, sprich, viel oder eben wenig verkauft und über welchen Zeitraum hinweg. Vielleicht kann man aus den Daten auch heraus lesen, dass jemand ein bestimmtes Talent zu haben scheint, eine gewisse Art von Waren an den Kunden zu bringen. Dann ist es natürlich sinnvoll Umstrukturierungen vorzunehmen. Eine Situation also in welcher systematische Datenerfassung einen entscheidenden Vorteil gebracht hätte. Also unbedingt hinzuzufügen ist die *VerkäuferNr*.

Einkäufe:

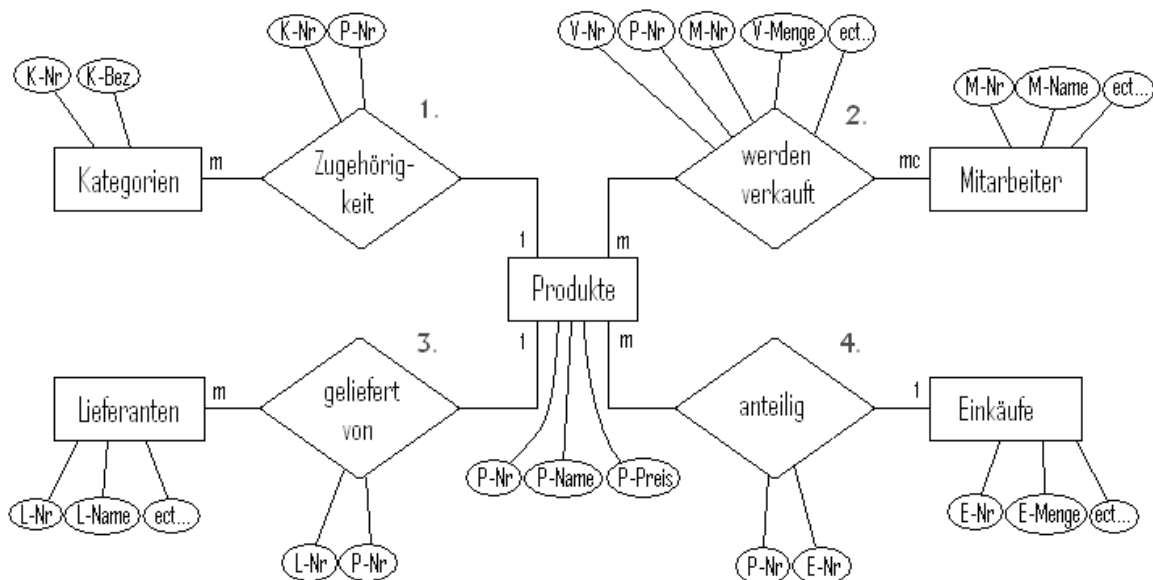
Parallel dazu kann man zur weiteren Verwaltung auch die Einkäufe bei den Lieferanten durch eine Tabelle festhalten. Diese Tabelle hat aber bei weitem nicht den Stellenwert der Verkaufstabelle, ist aber trotzdem nützlich.

Wie bei allen Tabellen kommt an erster Stelle der Primärschlüssel *EinkaufNr*. Sodann wird das eingekaufte Produkt durch die *ProduktNr* verknüpft. Das Ziel dieser Tabelle ist es die Einkäufe und somit auch die eingekauften Mengen im Überblick zu haben. Also ist ein weiteres Merkmal mit der Bezeichnung *Einkaufsmengen* nötig.

Zeitliche Orientierung wird durch das *Einkaufsdatum* geschaffen und sagt zusätzlich etwas über den Zustand der sich im Lager befindenden Waren aus. Am Schluss sollte man bei jedem Einkauf noch den jeweiligen Lieferanten hinzufügen. Wenn bekannt ist, von welchen Lieferanten man wie viel und zu welchem Zeitpunkt seiner Produkte eingekauft hat, dann weiss man auch wie nützlich dieser ist. Es lässt sich sogar berechnen wie viel Gewinn man an einem Lieferanten macht, so kann seine Bedeutung für den Betrieb bestimmen werden.

## 2.2 Das Entitäten-Beziehungsmodell

Hat man erst einmal die Informationen gesammelt, die man in die Datenbank einbringen möchte, ist schon ein grosser Teil der Arbeit getan. Aus der enorm wichtigen Datenanalyse folgt der nächste Schritt in der Datenmodellierung, die Kreierung eines Entitäten-Beziehungsmodells. Dadurch lässt sich testen, ob es überhaupt möglich ist die geplanten Informationen in einem Datenbankmodell umzusetzen. Ist dem nicht der Fall, müssen Änderungen möglichst noch in diesem Zwischenschritt vorgenommen werden. Wenn die Datenbank schon voller Informationen ist, wird es um ein vielfaches mühsamer, Fehler zu beheben. Das Entitäten-Beziehungsmodell eines Warensshops könnte folgendermassen aussehen:



**Abbildung 1: Das Entitäten-Beziehungsmodell**

Zur Erklärung der Beziehungen:

1. : Jedes Produkt gehört genau einer Kategorie an. Des weiteren werden aber nur Kategorien gebildet, die mindestens mehrere Produkte enthalten. Beziehung  $\rightarrow (1, m)$
2. : Produkte werden nicht immer vom gleichen Mitarbeiter verkauft. Meistens sind es die ausgebildeten Verkäufer, die die Verkäufe vornehmen. Es kann aber gut sein, dass auch mal die Verkaufsleiterin oder sogar der Chef persönlich an die Kasse geht. Es ist aber wiederum nicht zwingend, dass jeder Mitarbeiter etwas mit Verkäufen zu tun hat. Beziehung  $\rightarrow (m, mc)$
3. : Ein einzelnes Produkt, wird nur von einem Lieferanten geliefert. Hingegen liefert ein Lieferant meistens mehrere aber mindestens ein Produkt. Beziehung  $\rightarrow (1, m)$
4. : Pro Einkauf, wird ein einziges Produkt eingekauft. Produkte können mehrmals eingekauft werden  $\rightarrow (1, m)$

## 2.3 Das Relationale Datenbankschema

In diesem Schritt werden die Entitäten- und Beziehungsmengen in Tabellen abgebildet. Dabei ist zu beachten, dass manche Beziehungsmengen direkt mit den jeweiligen Enditätsmengen zusammengefasst werden können. Es entstehen somit ganz sicher Tabellen für Enditätsmengen, nur teilweise aber für Beziehungsmengen.

Wichtig: Die Frage, welche Merkmale die in die Tabellen aufgenommen werden und im besonderen weshalb, wurde schon im Kapitel der Datenanalyse geklärt. Alle Merkmale haben also ihren Ursprung in der Datenanalyse.

Produkte:

ProduktNr	Produktname	KategorieNr	LieferantNr	Menge pro Einheit	Einkaufspreis	Preis	Mengen in Lager

\* Die Mengen "Zugehörigkeit" und "geliefert von" werden hier durch die Merkmale *KategorieNr* und *LieferantNr* mit der Tabelle der Produkte zusammengefasst.

Kategorien:

KategorieNr	Kategoriename	Beschreibung

Jede Enditätsmenge muss in eine eigenständige Tabelle überführt werden. So z.B. die Tabelle der Produkte-Kategorien. Das gleiche gilt übrigens für die Tabelle "Mitarbeiter" und "Lieferanten", wird deshalb weiter vorne nicht mehr erwähnt.

### Mitarbeiter:

MitarbeiterNr	Name	Vorname	Strasse	Ort	Telefon	Geburtsdag	Einstellungsdatum	Funktion	Lebenslauf

### Lieferanten:

LieferantenNr	LieferantenName	Funktion	Adresse	Ort	Telefonnummer	Art der Lieferung	Priorität

### Verkäufe:

VerkaufNr	ProduktNr	Absatzmenge	Verkaufsdatum	Verkaufszeit	ZeitPeriode	VerkäuferNr	Karteninhaber

Die Verkäufe werden durch eine Beziehungsmenge verwaltet. Es muss aber unbedingt eine eigenständige Tabelle entstehen, da es sich um eine komplex-komplexe Beziehung handelt.

\*Siehe Meier`s "Relationale Datenbanken"

### Einkäufe:

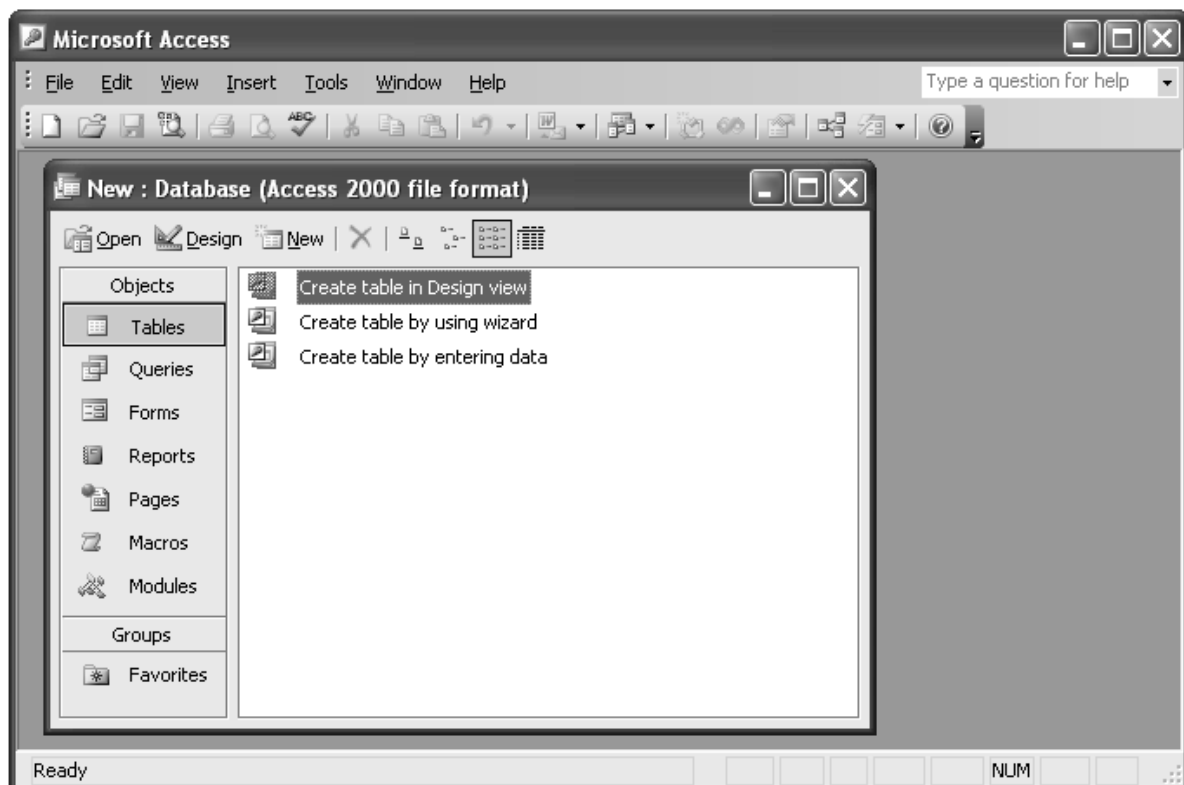
EinkaufNr	ProduktNr	Einkaufsmenge	Einkaufsdatum

Der gleiche Grund für die Entstehung einer alleinstehenden Tabelle tritt hier in Kraft: es handelt sich um eine komplex-komplexe Beziehungsmenge. Ein Produkt kann immer wieder eingekauft und verkauft werden. Es ist aber nicht nötig in dieser Tabelle, immer und immer wieder die Produkteigenschaften wie z.B. *Produktname* oder *Menge pro Einheit* zu notieren.

### 3. Umsetzung in Access

Dieses Kapitel befasst sich ausschließlich mit dem Programm MS Access 2004. Für jemanden, der das Programm noch nie benutzt hat, mag es schwierig erscheinen etwas Konstruktives damit anzufangen. Deshalb wird hier versucht durch viele Grafiken das Verständnis zu steigern.

Nach dem Start des Programms, geht man auf *File* → *New* → *Blank Database*, um eine neue Datenbank zu erstellen. Dann erscheint das folgende Dialogfenster:



**Abbildung 2: Startfenster in Access 2004**

Im kleinen Fenster, gibt es unter *Objects*, Unterteilungen in *Tables*, *Queries*, *Forms*, *Reports*, *ect...*

Klickt man das jeweilige Feld an, kann man entweder Tabellen, Queries, Forms oder Reports kreieren. Damit es kein durcheinander gibt, werden in diesem Kapitel die gleichen Unterteilungen wie im Programm vorgenommen.

Im ersten Unterkapitel geht es um die Erstellung von *Tabellen*. Dann werden die *SQL-Abfragen (Queries)* behandelt, und zu guter letzt noch die *Forms*. Hierbei handelt es sich um eine gestaltungsmöglichkeit für Tabellen.

*Reports* und *Pages* liefern nichts nenenswertes dazu, werden deshalb nicht weiter erwähnt.

Am Schluss gibt es noch ein Kapitel über *Mainswitchboards*. Sie sind quasi ein Steuerungsorgan für die Datenbank.

## 3.1 Tabellen

Ohne Tabellen hat es keinen Sinn Query-Abfragen zu formulieren oder irgendwelche Forms aufzusetzen. Folglich sollten zuerst Tabellen definiert werden. Im Abschnitt *Tables* hat man dafür drei Möglichkeiten: 1. *Design view*, 2. *using wizard*, 3. *entering data*. Bei der Möglichkeit 2 + 3, werden aber entweder bestimmte Werte vorgegeben oder man ist in der Handlung sehr eingeschränkt. Also klickt man am besten auf *Design view*:

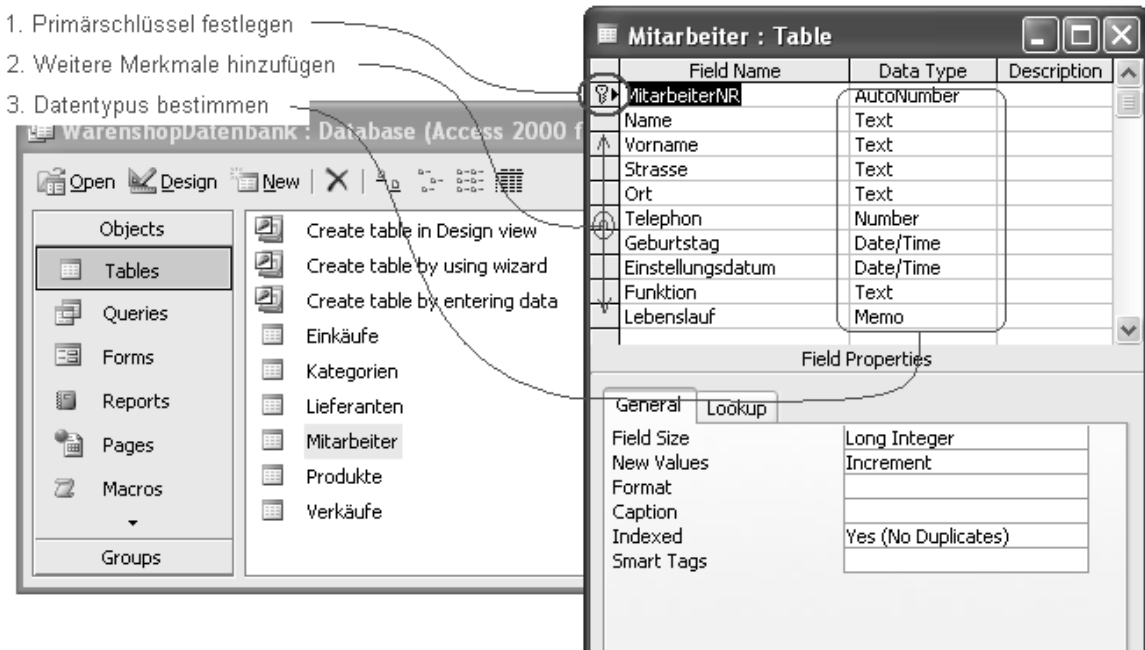


Abbildung 3: Design View von Tabellen

1. Unter *Field Name* gibt man die Merkmale der Tabelle ein. Als allererstes verwendet man den Primärschlüssel z.B. *MitarbeiterNr*. Durch einen Rechtsklick auf eines der kleinen Vierecke neben dem jeweiligen Merkmal, lässt sich der *Primary Key* definieren. Pro Tabelle gibt es nur einen, in diesem Fall ist es die *MitarbeiterNr*.
2. Danach kann man weitere Merkmale der Tabelle hinzufügen. Dabei ist zu beachten, dass Daten wie PLZ und Ort getrennt hineingeschrieben werden müssten, würden sie verwendet werden.
3. Unter *Data Type* gibt man den beim Merkmal vorliegenden Datentypus an. Da unter dem Merkmal Name, in der Tabelle lauter Namen folgen sollen, wählt man dort *Text*. Beim Primärschlüssel *AutoNumber*, was bedeutet, dass für jedes neue Tupel die nächstfolgende Zahl gewählt wird (gleiche Zahlen ausgeschlossen). Für normale Zahlen verwendet man *Number*. Handelt es sich aber um ein Datum oder eine Zeit, dann wählt man *Date/Time* und bei Geld/Währung nimmt man den Ausdruck *Currency*. Des weiteren wählt man für Sätze oder gar längere Beschreibungen den Typus *Memo*, hingegen verweist die Einstellung

*Hyperlink* auf Internetadressen oder Links allgemein. Dann gibt es noch die Funktion *OLE Object*, mit welcher man z.B. Bilder einfügen kann. Hat man dieses Fenster einmal geschlossen und abgespeichert, dann erscheint die Tabelle im kleinen Dialogfenster unter *Tables*. Wenn man nun einen Rechtsklick auf eine Tabelle ausführt, kann man unter *Design View* jederzeit die Tabelle wieder bearbeiten, neue Merkmale hinzufügen oder alte löschen und weitere Änderungen vornehmen.

### 3.2 Relationships

Hat man nun alle nötigen Tabellen in Access definiert, bleibt nur noch das Problem der Beziehungen zu lösen. Wie beim Entitäten-Beziehungsmodell, muss man in Access die Einzelnen Tabellen miteinander verbinden. Hierzu geht man einfach im Hauptfenster oben auf *Tools* → *Relationships*. Dann erscheint ein Fenster, indem alle Tabellen, mit den dazugehörigen Merkmalen abgebildet sind. Mit der Maus geht man dann auf ein Merkmal, dass man mit einem anderen verbinden möchte, klickt und lässt erst los wenn man sich über dem Zielmerkmal befindet. Im nun erscheinenden Fenster sollte man noch zusätzlich das Häkchen bei *Enforce Referential Integrity* anklicken. Somit wird referentielle Integrität erzwungen. Das fertige Relationship-Modell des Warenhops könnte dann folgendermassen aussehen:

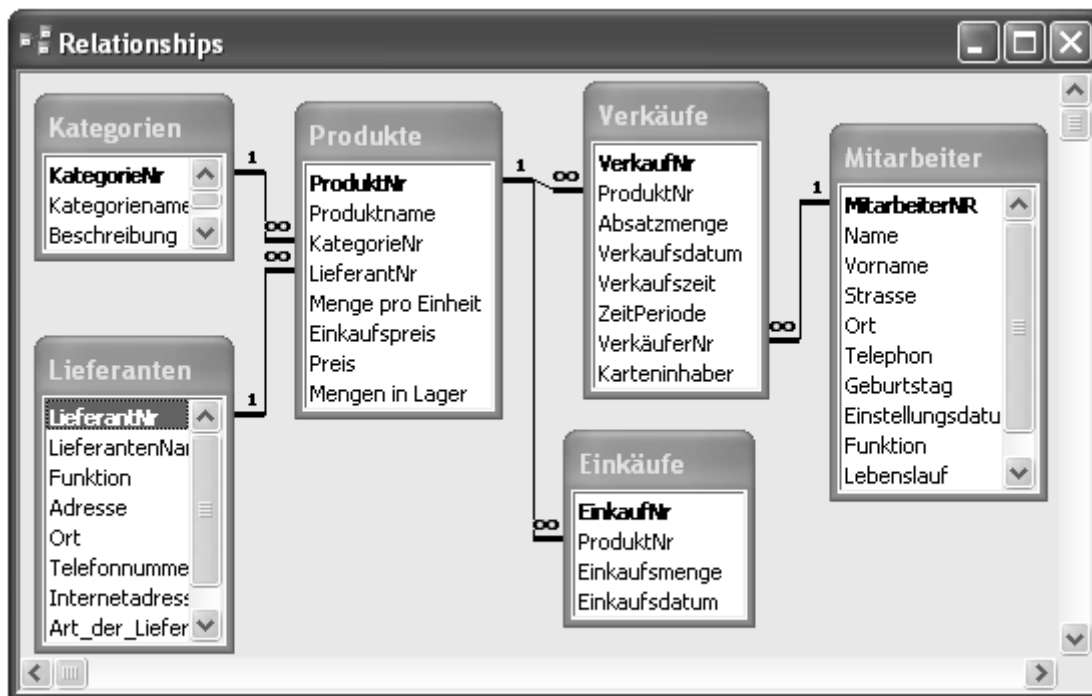


Abbildung 6: Die Tabellen und ihre Beziehungen

Macht man einen Rechtsklick auf eine Verbindung, kann man entweder die Verbindung löschen oder editieren. Will man in diese Grafik eine neue Tabelle hinzufügen, macht man einen Rechtsklick auf die graue Fläche und wählt *Show Table*, wählt die gewünschte Tabelle aus und geht dann auf *add*(hinzufügen). Man kann auch jederzeit das ganze Modell wieder verändern.

### 3.3 Werte eingeben:

Durch einen Doppelklick kann man die Tabellen nun verwenden und anfangen die Werte einzugeben.

Das sieht bei der Tabelle der Mitarbeiter etwa folgendermassen aus:

	MitarbeiterNR	Name	Vorname	Strasse	Ort	Telepho
+	1	Oberson	Denis	ch. de la madelaine 1	Granges-Pacco	305.
+	2	Zbinden	Barbara	rte de Villars 27	Fribourg	424.
+	3	Baumann	Eliane	ch. Monsejour 19	Fribourg	322.
+	4	Aerne	Christophe	rte Moncor 21	Villars-sur-Glân	408.
+	5	Sali	Hasan	av. Jean-Musy 2	Courtepin	684.
+	6	Schaerer	Héli			
*	(AutoNumber)					

**Abbildung 4: Die Tabelle der Mitarbeiter**

Bei der Tabelle der Verkäufe, kann es ein wenig anders aussehen. Hier besteht die Tabelle fast nur aus Zahlen, die sich teilweise auf andere Tabellen beziehen:

	VerkaufNr	ProduktNr	Absatzmenge	Verkaufsdatum	Verkaufszeit	ZeitPeriode	VerkäuferNr	Karteninhab
▶	1	3	3	01.01.2004	09:11:56	09-10	4	<input type="checkbox"/>
	2	11	1	01.01.2004	10:34:21	10-11	6	<input checked="" type="checkbox"/>
	3	34	2	01.01.2004	14:25:04	14-15	6	<input type="checkbox"/>
	4	2	4	01.01.2004	15:48:30	15-16	4	<input type="checkbox"/>
	5	7	3	02.01.2004	10:59:51	10-11	5	<input checked="" type="checkbox"/>
	6	23	7	02.01.2004	11:27:37	11-12	4	<input checked="" type="checkbox"/>
	7	29	2	02.01.2004	16:17:02	16-17	6	<input type="checkbox"/>
	8	31	20	03.01.2004	08:34:23	08-09	6	<input checked="" type="checkbox"/>
	9	2	4	03.01.2004	10:01:48	10-11	6	<input type="checkbox"/>
	10	10	2	04.01.2004	11:35:08	11-12	4	<input checked="" type="checkbox"/>

Record: 1 of 69

**Abbildung 5: Die Tabelle der Verkäufe**

Die folgende Tabelle zeigt in welcher Form man bei jedem Datentypus die Daten eingeben kann:

Text	Alle Arten von Zeichen sind möglich
Autonumber	Keine Eingabe möglich
Number	Nur Nummern, ohne Abstände
Date/Time	tt.mm.jjjj/hh:mm:ss
Currency	xx.xx → SFr. xx.xx
Memo	Längere Texte möglich
Hyperlink	www.keine-ahnung.com

### 3.4 SQL-Abfragen

Im Abschnitt Queries hat man zwei Möglichkeiten SQL-Abfragen zu formulieren. *Using wizard* ist zu oberflächlich und kann nicht zielgenau eingesetzt werden. Es schadet aber nicht es trotzdem mal zu versuchen. Wählt man *Design view*, komm zuerst ein kleiner Bildschirm den man direkt schliessen sollte. Mit einem Rechtsklick im folgenden Fenster, ruft man die bekannte *SQL-View* auf:

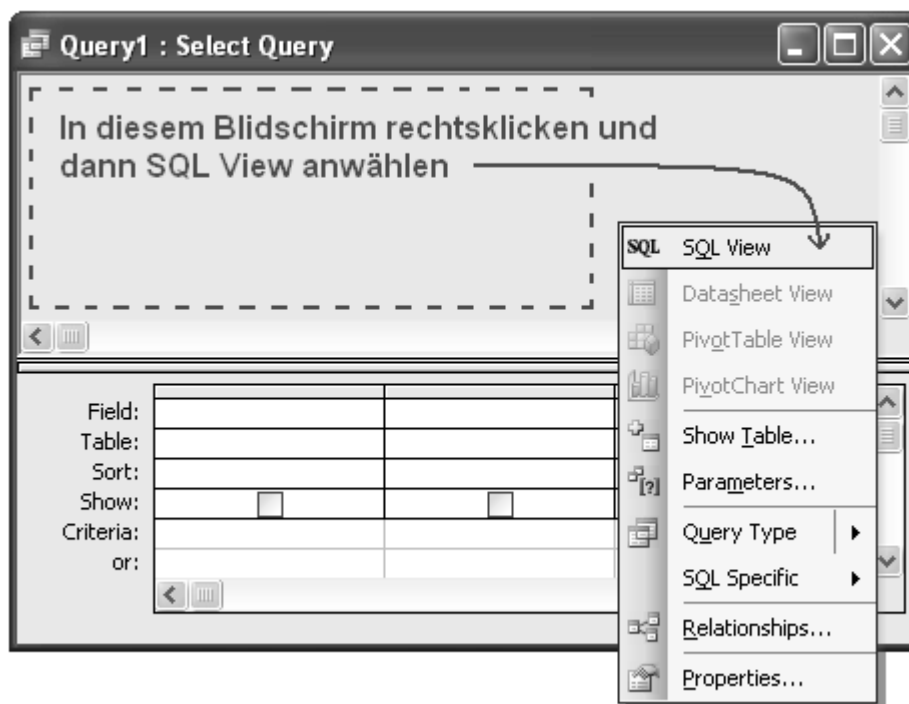


Abbildung 7: Beschreibung zur SQL-View

Im nächsten Bildschirm kann man die Abfragen mit dem standartisierten SELECT-FROM-WHERE-Schema ausführen. Bei Unklarheiten zur Funktionsweise dieses Abfragemechanismuses sollte an dieser Stelle das Buch "Relationale Datenbanken", welches im Literaturverzeichnis angegeben ist, zur Seite genommen werden.

### 3.5 Management durch SQL

SQL-Abfragen sind das nützlichste Werkzeug zur Datenbankbearbeitung. Sie erlauben beinahe alles möglich zu machen, was mit Dateninformationen zu tun hat. Um genauer darzustellen was damit gemeint ist, gibt es in diesem Kapitel Beispiele, die sich auf den Warehouse beziehen:

The screenshot shows a window titled 'Produktverkäufe diesen Monat : Select Query'. The SQL query is as follows:

```
SELECT Produktname, Sum(Absatzmenge) AS [Verkaufte Menge],
Sum(Absatzmenge*Stückgewinn) AS Produktgewinn
FROM Produkte, Verkäufe
WHERE Produkte.ProduktNr=Verkäufe.ProduktNr
GROUP BY Verkäufe.ProduktNr, Produktname, Stückgewinn
ORDER BY Produktname;
```

The result table displays the following data:

Produktname	Verkaufte Menge	Produktgewinn
Schwarzbrot	1	SFr. 0.35
Schwarzwäldertorte	2	SFr. 9.90
Sprite	9	SFr. 5.85
Suchard Express	3	SFr. 8.40

Diese Abfrage zeigt auf einen Blick den Produktgewinn jedes einzelnen Produktes seit anfang Monat. Vergleicht man die Tabellen mit anderen Monaten, lässt sich mit der Zeit sagen, welche Produkte sich rentieren. Hier sehr schön zu sehen ist, dass z.B. Schwarzbrot diesen Monat fast keinen Gewinn eingebracht hat. Würde sich dies mit der Zeit nicht ändern, müsste sich die Produktmanagerin ernsthafte Gedanken darüber machen ob man dieses Produkt weiter im Sortiment behalten möchte. Wäre die Verkaufszahl sehr hoch, der Gewinn aber niedrig, könnte man z.B. den Preis ein wenig zu erhöhen und so, ohne auf das Produkt zu verzichten, trotzdem einen rentablen Gewinn einfahren. Man sieht also, dass so ein komplexes System eine Menge Spielereien zulässt.

The screenshot shows a window titled 'Tageszahlen : Select Query'. The SQL query is as follows:

```
SELECT Verkäufe.Verkaufsdatum AS Datum, Sum(Absatzmenge*Preis) AS
Tagesertrag, Sum(Absatzmenge*Einkaufspreis) AS Tageskosten,
Sum(Absatzmenge*Stückgewinn) AS Tagesgewinn
FROM Produkte, Verkäufe
WHERE ((Produkte.ProduktNr)=Verkäufe.ProduktNr))
GROUP BY Verkäufe.Verkaufsdatum
ORDER BY Verkaufsdatum;
```

The result table displays the following data:

Datum	Tagesertrag	Tageskosten	Tagesgewinn
11.01.2004	SFr. 37.65	SFr. 12.90	SFr. 24.75
12.01.2004	SFr. 0.80	SFr. 0.40	SFr. 0.40
13.01.2004	SFr. 82.35	SFr. 53.45	SFr. 29.10
14.01.2004	SFr. 68.70	SFr. 42.85	SFr. 25.95

Durch diese Abfrage erhält man die wichtigen Zahlen eines jeden Tages. Noch aufgeteilt nach Wochentagen, wäre es möglich zu sagen, welche Tage besonders gewinnbringend sind und vielleicht noch gewinnbringender wären, wenn man den Schichtbetrieb an diesem Tag erhöhen würde.

The screenshot shows a window titled 'Lieferantenbeteiligung am Gewinn : ...'. The SQL query is as follows:

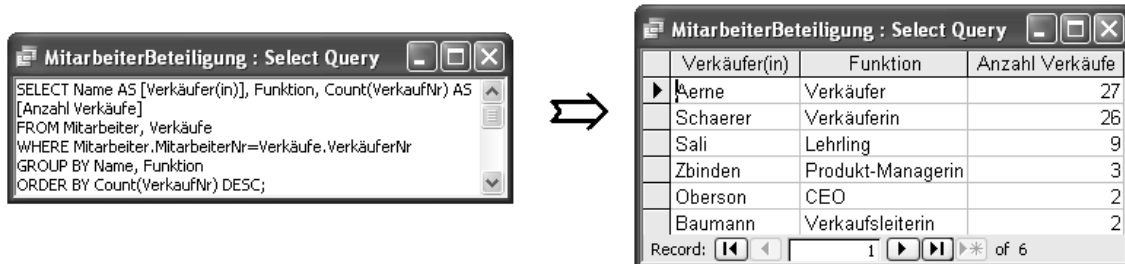
```
SELECT LieferantenName, Sum(Einkaufsmenge*Stückgewinn) AS
Gewinnbeteiligung
FROM Einkäufe, Produkte, Lieferanten
WHERE Produkte.ProduktNr=Einkäufe.ProduktNr and
Einkäufe.LieferantNr=Lieferanten.LieferantNr
GROUP BY Einkäufe.LieferantNr, LieferantenName
ORDER BY Sum(Einkaufsmenge*Stückgewinn) DESC;
```

The result table displays the following data:

LieferantenName	Gewinnbeteiligung
Nestlé SA	SFr. 164.00
Coca-Cola Beverages AG	SFr. 115.25
Kraft Foods Schweiz AG	SFr. 50.00
Fromage Gruyère S.A.	SFr. 49.00

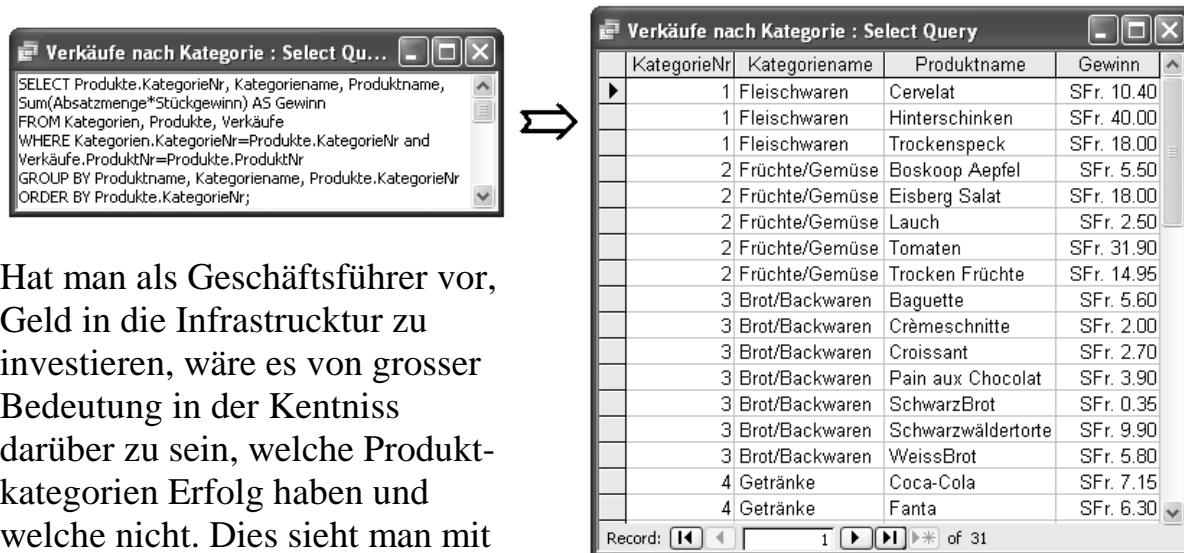
Wenn sich der Leser im Laufe der Arbeit mal gefragt hat wie man zu der Erkenntnis gekommen ist, wieso welcher Lieferant zu seiner Priorität gekommen

ist, dann hat er hier die Antwort. Mit dieser Abfrage hat man Einblick in die Gewinnbeteiligung eines jeden Lieferanten. Mit Gewinnbeteiligung ist die Einkaufsmenge mal den Stückgewinn eines jeden Produktes und zum Schluss noch zusammengefasst nach dem Lieferanten, gemeint. Dazu ist noch zu sagen, dass es eigentlich nicht darauf ankommt, ob man dafür die Einkäufe oder Verkäufe verwendet, da sowieso alle Waren immer wieder ersetzt werden.



Für Manger ist es doch schön, wenn man den Mitarbeiter nicht nach seiner Nase beurteilen muss, sondern ihn mit beinharten Fakten konfrontieren kann. Würde es sich beispielsweise um einen Autoverkaufs-Betrieb handeln und nicht um einen Warenshop, dann könnte man den Mitarbeiter mit seinen Verkaufszahlen konfrontieren.

Abbildungen 8-12: SQL-Abfragen



Hat man als Geschäftsführer vor, Geld in die Infrastrucktur zu investieren, wäre es von grosser Bedeutung in der Kentniss darüber zu sein, welche Produktkategorien Erfolg haben und welche nicht. Dies sieht man mit dieser Abfrage auf einen Blick.

Warum ist diese Abfrage so wichtig?: Macht man z.B. wenig Einnahmen mit Brot, lohnt es sich nicht ein neues Brotgestell zu kaufen, weil man bestimmt nicht dort expandieren wird. Hat man hingegen viel Erfolg mit Getränken, wäre es vielleicht nicht verkehrt ein neues Kühlfach für Getränke zu besorgen.

Man muss immer noch zu alledem sagen, dass solche Erkenntnisse, wie sie mit diesen SQL-Tabellen gemacht wurden, erst nach langen Empirischen Unter-

suchungen einen gewissen Sinn machen. Auch die Tatsache, dass in dieser Datenbank sehr wenige Informationen enthalten sind, ist ein Grund, warum anhand solcher Daten keine Entscheidungen getroffen werden sollten.

### 3.6 Entstehung von Formularen (*Forms*)

Dieses Tool erlaubt es mit wenigen Handgriffen, grafische Oberflächen zu kreieren. Das ganze Erscheinungsbild der langweiligen Datenbanktabellen kann komplett verändert werden. Für den Kontakt mit dem Kunden, ist das Erscheinungsbild natürlich entscheidend. Per Internet kann man so dem Kunden die Produkte und Produkteinformationen ohne Probleme vorstellen. Wie geht man nun aber vor?

Wählt man in Abschnitt Forms, *usign wizard*, erscheint folgendes Dialogfenster. Man kann unter *Tables/Queries*, Tabellen oder SQL-Tabellen auswählen, in denen man interessante Merkmale vermutet. Diese Merkmale erscheinen später auf der Benutzeroberfläche in genau der Reihenfolge wie im Feld *Selectet Fields*. Man kann auch von mehreren Tabellen Merkmale wählen. (SQL-Tabellen(Queries) und normale Tabellen dürfen nicht gemischt werden!)

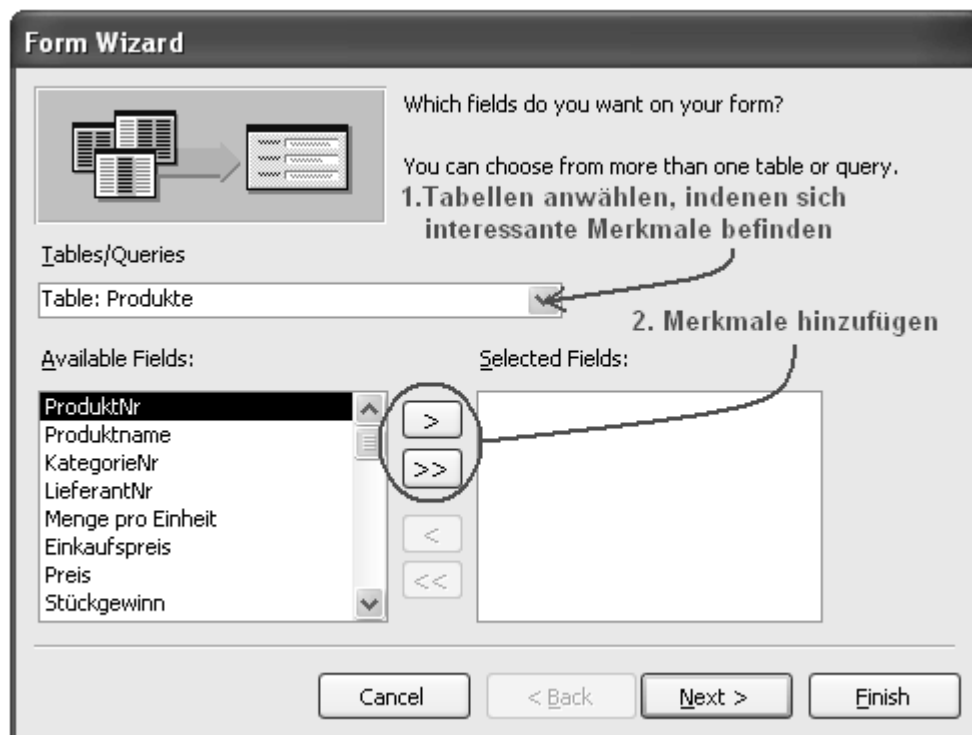


Abbildung 13: Hilfsassistent für Forms

Hat man die gewünschten Merkmale in der richtigen Reihenfolge, klickt man auf *Next*. Im nächsten Fenster kann man die Darstellungsarten wählen:

Columnar	Merkmale werden in einer Kolonne abgebildet
Tabular	Abbildung in einer Tabelle, aber mit Design Theme
Datasheet	gleich wie in DB-Ansicht (graue Tabelle)
Justified	spezielle Anordnung der Merkmale, wie bei einem Formular
PivotTable	Kreierung einer Tabelle
PivotChart	Kreierung eines Diagramms

Fenster Nummer drei, bietet eine grosse Auswahl an Darstellungsstilen (*Design Themes*). Die Namen haben etwas mit den Stilen zu tun. So ähnelt etwa *Rice-paper* dem Hintergrund einer chinsischen Restaurantkarte, hingegen erweckt *Sandstone* den Eindruck als wäre der Hintergrund aus Sandstein.

Hat man im letzten Fenster den Namen eingegeben und das Fenster geschlossen, erscheint die *Form* in der Unterteilung Forms. Durch einen Rechtsklick kann man *Design View* anwählen. In dieser Ansicht lassen sich, durch Anklicken, die Formen und Anordnungen der Merkmale verändern. Felder können länger und breiter gemacht werden und durch anklicken der Namen kann man diese nach belieben verändern.

Zum Schluss gibt es noch ein paar Beispiele für Forms:



Abbildung 14: Das Formular für Mitarbeiter

Man kann Fotos in die Form integrieren. Dazu bildet man in der Ursprungstabelle Mitarbeiter ein neues Merkmal genannt *Bild*, mit den Datentypus *OLE Object*. Bei der Tabelleneingabe, kann man durch einen Rechtsklick *Insert Object* anwählen. Klickt man im folgenden Dialogfenster auf *Create from File*, kann man den Pfad des Bildes angeben:



Abbildung 15: Erklärung zur Funktion *Insert Object*

Wenn man im Auswahlbildschirm (Abbildung 13), Merkmale von mehreren Tabellen anwählt, dann kann es zu verschachtelten Forms kommen, die sich auf andere Tabellen beziehen. So sieht man etwa hier auf einen Blick die zum Lieferanten gehörigen Produkte:

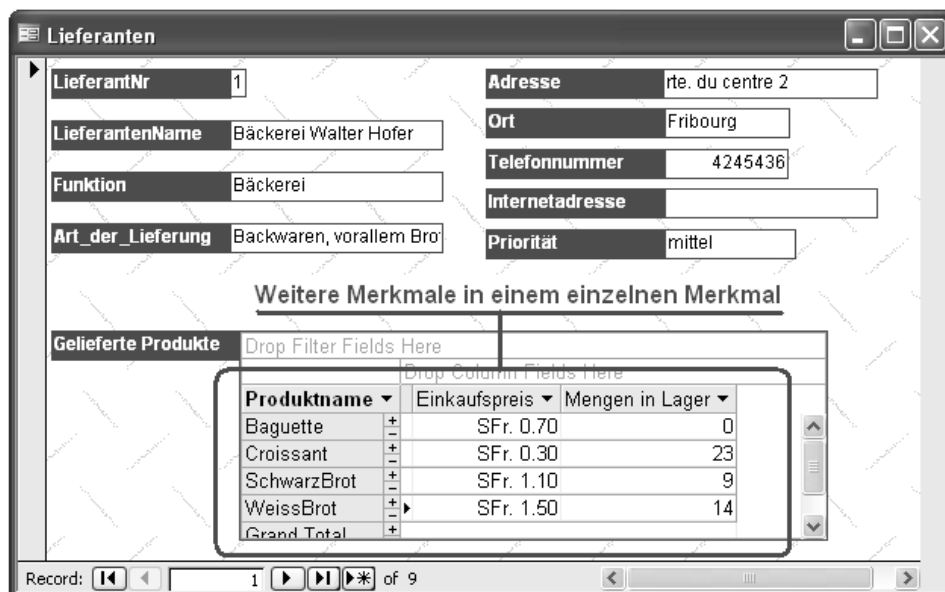
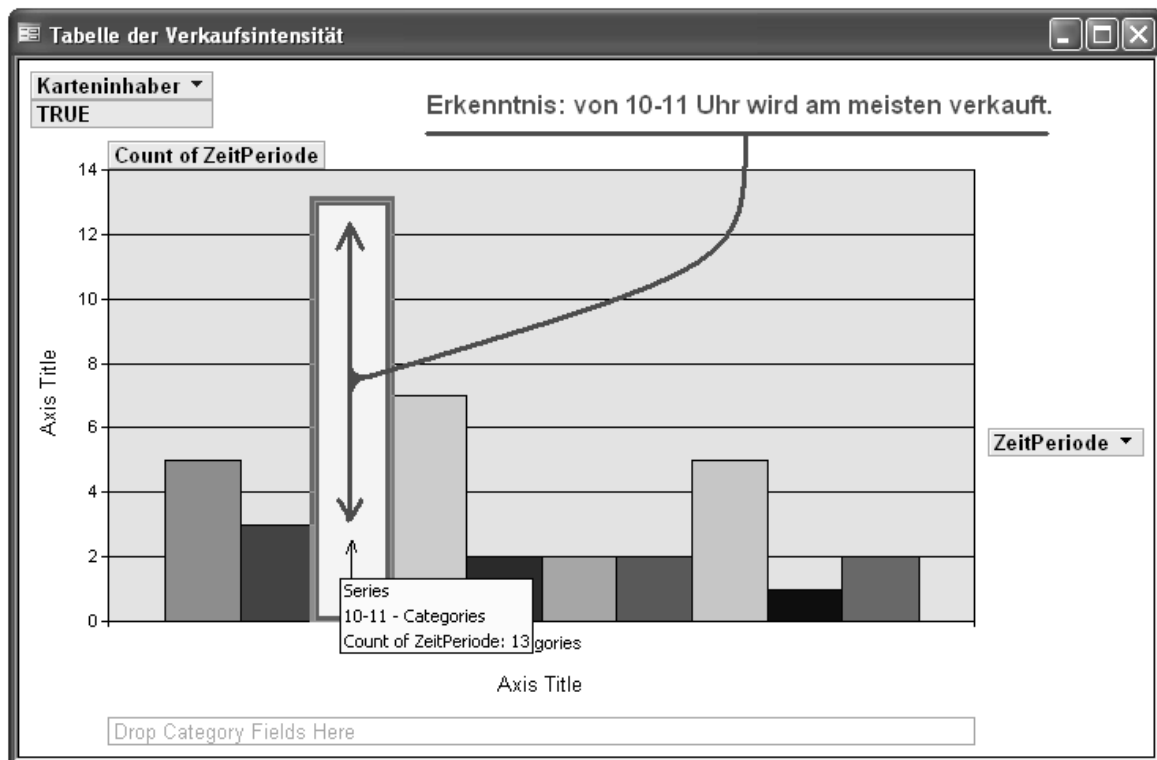


Abbildung 16: Beispiel für verschachtelte Forms

Will man in manchen Situationen mehr Übersicht haben, können *Diagramme* (Darstellungsart=Pivot Chart) die Lösung sein. In diesem Beispiel werden die Anzahl Verkäufe per Zeitperiode in einem Diagramm dargestellt. Zusätzlich ist es möglich Filterelemente in die Grafik miteinzubauen. In der linken oberen Ecke ist beim Merkmal Karteninhaber=TRUE, was bedeutet, dass nur Stammkunden abgebildet werden. Jetzt kann die Grafik sehr nützlich sein:



**Abbildung 17: Diagramm der Verkaufsintensität**

Mögliche Interpretation aus der Sicht des Managements:

Wie hier sehr schön zu sehen ist, kommen am Morgen sehr viele Kunden. Vielleicht besteht die Hauptzielgruppe aus Hausfrauen, die keinem Beruf nach gehen und so den Morgen frei haben. Somit könnte man die Produktauswahl an diese Zielgruppe anpassen. Nicht zu vernachlässigen scheint ebenfalls die Periode 16-17 Uhr zu sein, die genau mit dem beenden der Schule zusammen fällt. Es könnte sein, dass eine weitere wichtige Zielgruppe daher Jugendliche sind, welche ja auch einen beträchtlichen Teil des Geldes besitzen. Bei der letzten auffälligen Periode, zwischen 18 und 19 Uhr, wäre durchaus noch ein grösseres Verkaufspotenzial zu erwarten, wenn man die Ladenzeiten etwas verlängern könnte. Den dies ist genau der Zeitpunkt, an dem jeder die Arbeit beendet hat und auf dem Nachhauseweg noch feststellt, dass er unbedingt etwas einkaufen müsste.

### 3.7 Benutzeroberfläche - das *Mainswitchboard*

Unter Mainswitchboard, versteht man eine Art Haupt-Dialogfenster, dass erlaubt die ganze Datenbank zentral zu steuern. Unter *Tools*→ *Database Utilities*, findet man den *Switchboard Manager*. Dieser Manager erlaubt es, ganz einfach, eine Stuktur für eine Benutzeroberfläche festzulegen, von welcher aus man Forms, Tabellen und Reprints besichtigen und sogar bearbeiten kann.

Durch Kreierung zusätzlicher *Switchboards*, lässt sich die Datenbank in verschiedene Bereiche aufteilen. Als Beispiel wurde hier die Datenbank in einen Kundenbereich, einen Mitarbeiterbereich und Administratorenbereich aufgeteilt:



**Abbildung 18: Startfenster der Datenbank**

Im Bereich für Kunden, befinden sich Informationen über die Produkte (Preise, Mengen im Lager), Mitarbeiter und Lieferanten. Solche Informationen könnte man auch im Internet bereitstellen und so direkt von den Kunden Bestellungen entgegen nehmen.

Der Bereich für Mitarbeiter dient den Verkäufern. Dort können sie laufend, neue Verkäufe und Einkäufe eintragen.

Der DataBase-Administrator, ist der einzige, der die Tabellen und Forms verändern kann. Er hat die meisten Rechte. Dieser Titel fällt an den, der die Datenbank entworfen oder die Leitung darüber hat. In diesem Fall, sollte der CEO, die Produktmanagerin und auch die Verkaufsleiterin, zumindest beschränkt, Zugang zu diesem Bereich haben.

## 3.8 Datenschutz

Eine Datenbank verfügt meistens über wichtige Informationen des Betriebs, wie auch über persönliche Informationen von Mitarbeitern. Deshalb ist es wichtig, dass nur eine ausgewählte Gruppe an Personen Besichtigungs- und Bearbeitungsrechte (*Permission to Read/Edit Data*) haben.

So soll z.B. der Bereich des Administrators, durch ein Passwort geschützt und nur für das höhere Kader zugänglich sein.

Der Bereich der Mitarbeiter sollte auch durch ein Passwort geschützt werden. Da aber alle Mitarbeiter von Zeit zu Zeit mal für Verkäufe gebraucht werden, sollten alle Mitarbeiter Zugang zu diesem Bereich haben. Gefährlich ist dies aus diesem Grunde nicht, weil in diesem Bereich keine Daten gelöscht oder verändert werden können.

Der Bereich für Kunden ist für alle frei zugänglich. Es muss lediglich dafür gesorgt werden, dass Kunden keine Möglichkeit haben irgendetwas an der Datenbank zu verändern. Da für die Kunden nur Reports zu sehen bekommen, kann überhaupt nichts passieren. Denn diese kann man nicht durch anklicken verändern.

## 4. Zur Datensicherheit

Im vielen Firmen wird dieser Punkt langsam sehr ernst genommen. Da Betriebe immer abhängiger von Computersystemen werden, kann ein Ausfall einem Rückschreiten in die Steinzeit gleichkommen. Darum ist eine der höchsten Prioritäten der meisten Firmen in jedem Zeitpunkt die Datensicherheit gewährleisten zu können. Wie aber ist dies möglich? Nicht alle Leute sind mit Computern gross geworden und wissen was dabei der wesentliche Punkt ist. Folgende Punkte sollten berücksichtigt werden:

### 4.1 physikalische Datensicherheit

Eine Festplatte ist gar nicht so sicher wie man glauben könnte. Sie besteht aus kleinster Microtechnik und sehr fein justierter Elektronik. Die Funktionsweise ist komplizierter als man denkt. Wer schon mal eine Festplatte von innen gesehen hat, fragt sich oft, wieso wieso nicht noch mehr Pannen entstehen. Auf einer Scheibe die nicht mal einen Durchmesser von 10cm beträgt, können Unmengen an Daten gespeichert werden. Verständlicherweise können nun schon kleinste Veränderungen, grösste Schäden anrichten. Festplatten sind sehr empfindlich gegen Erschütterungen. Fällt eine zu Boden, kann es sehr gut sein, dass die Daten für immer verloren sind. Computer sollten daher an fixen Orten

stehen wo sie vor Erschütterungen absolut sicher sind und keinem Fussgänger im Weg stehen.

Da die Daten auf Festplatten magnetisch abgespeichert werden, können die Daten auch durch Magneten beeinflusst werden. Ist dies der Fall, sind die Daten in diesem Sektor auf jeden Fall zerstört. Deshalb sollte der Computer auch vor elektromagnetischer Strahlung gesichert werden. Dabei ist zu beachten, dass viele andere elektrische Geräte, wie etwa Fernseher oder Videogeräte, solche ebenfalls Strahlungen aussenden.

Des Weiteren, kann es immer wieder mal zu einem Stromausfall kommen. Und ohne Strom kann ein Betrieb, der auf Computersystemen aufbaut nicht mehr funktionieren. Ohne Strom schaltet sich aber nicht einfach nur der Computer aus, was schon schlimm genug wäre. Es können auch Daten, die gerade in Bearbeitung waren, ohne Vorwarnung, verloren gehen. Um solche Schäden vorzubeugen, sollte jede grössere Firma, die es sich leisten kann, ein Notstromagregat einbauen. Bei einem Ausfall, übernimmt es ohne Unterbruch die Stromversorgung.

## 4.2 hardwaretechnische Sicherheit

Aber auch von der rein technischen Seite her, gibt es sicherere und unsicherere Computer. Hochleistungsserver verfügen über Zwischenspeicherungs- und Backup-Funktionen.

Durch Zwischenspeicherung, sind auch Daten die gerade in Bearbeitung sind fortwährend gespeichert. Er zeichnet immer gerade die letzten Aktionen auf und speichert sie. Je grösser dieser Speicher ist, desto mehr Daten können so bei einem Stromausfall gerettet werden.

Die Backup-Funktion sorgt dafür, dass dauernd eine Kopie der vorhandenen Daten besteht. Diese Kopie (Backup) sollte auf einer anderen Festplatte gespeichert werden. Wenn möglich sogar in einem anderen Raum, übers Firmennetzwerk, oder sogar anderen Haus (Wegen Katastrophenfall: Brand, Überschwemmung, ect...). Funktioniert nämlich die Festplatte, aus welchem Grund auch immer, nicht mehr, dann hat man ohne Kopie ein ernsthaftes Problem.

Man unterscheidet zusätzlich zwischen kurzfristigem, mittelfristigem und langfristigem Backup. Sind die Daten mehr als 30 Tage alt, sollte man sie direkt zuhause auf Disketten oder CD's aufbewahren.

## 4.3 softwaretechnische Sicherheit

Wie wir wissen, handelt es sich bei Windows vielleicht um das weitverbreitetste Betriebsprogramm, aber längst nicht um das sicherste. Internet-Viren und

Hackerprogramme haben es auf gut funktionierende Computer abgesehen und wollen nur dessen Zerstörung. Solche Schädlinge werden, wie wir ja heute wissen, gemeinerweise auch unter dem Deckmantel schöner e-Mails versendet. Und da sich die Hacker dieser Welt immer neue Wege ausdenken werden, in Computersysteme einzudringen, ist äusserst davon abzuraten, den Datenbankcomputer mit dem Internet zu verbinden. Der Computer sollte von der Aussenwelt quasi Isoliert werden.

Es können aber auch schon installierte Computerprogramme, Probleme von riesigen Ausmassen auslösen. Auf dem Computer, sollte also neben Software- und Treiberupdates nur das nötigste installiert werden.

Sollte der Computer aber Erforderlicherweise trotzdem mit dem Internet verbunden sein, da es sich um eine Kundendatenbank handelt, können nur ausreichende Virens Scanner und Schutzschilder (Firewalls) die Sicherheit steigern. Da für Firmencomputer aber Lizenzen gekauft werden müssen, so übrigens auch für Access, kann dieser ganze Spass sehr teuer werden. Und hundertprozentiger Schutz ist auch dann nicht gewährleistet. So wie die Krankheiten der Medizin immer einen Schritt voraus sind, so sind auch die Hacker immer den Anti-Virenprogrammen einen Schritt voraus.

## 5. Schlusswort/Ergebnisse

Schon seitdem wir im ersten Semester in Wirtschaftsinformatik das Thema der Datenbanken durchgenommen haben, hat mich dieses Thema sehr interessiert. Und da ich mich in meiner Freizeit relativ oft mit Computern befasse, ist mir das Arbeiten mit Access gerade recht gekommen. Ich habe dieses Programm vorher noch nicht wirklich gekannt, bin aber relativ schnell damit klar gekommen. Mir hat es vorallem grossen Spass gemacht verschiedene Sachen auszuprobieren und dann wieder zu verändern. Es ist sehr befriedigend nun eine solche Datenbank ohne fremde Hilfe erstellen zu können. Vorallem erkenne ich jetzt die wirklichen Vorteile von solchen Computersystemen. Es ist mir vollkommen einleuchtend, dass jeder Betrieb durch solche Systeme viel effizienter und kostensparender Arbeiten kann. Und da diese Arbeit sehr Praxisbezogen ist, kann mir das neugewonnene Wissen auf diesem Gebiet bestimmt weiterhelfen.

Das Ergebnis dieser Arbeit, die WarenshopDatenbank, könnte als Basis für einen realen Warenshop verwendet werden. Sie müsste aber dauernd weiterentwickelt und verbessert werden. Man könnte in eine solche Datenbank über Monte hinweg Zeit investieren und sie würde dadurch noch effektiver und noch umfangreicher werden. Es könnte auch noch sehr viel mehr Automatisiert und verbessert werden, aber für solche Schritte habe ich einfach zu wenige Kenntnisse über das Programm MS Access.

Ich denke, dass ich aber die gesteckten Ziele erreicht habe und hoffe, dass die Arbeit so verstanden wird, wie ich sie auszudrücken versucht habe.

## 6. Literaturangaben

[Meier 2001] Meier, Andreas: Reationale Datenbanken - Leitfaden für die Praxis, 4.Auflage, Springer Verlag, Berlin, Heidelberg, NewYork, 2001

Als nützlichste Quelle, habe ich die *Northwind Sample Database* aus dem Programm *MS Access 2004*© zu Hilfe genommen. Sie befindet sich im Programm unter *Help*→ *Sample Databases*→ *Northwind Sample Database*

## 7. Abbildungsverzeichnis

Abbildung 1:	Das Entitäten-Beziehungsmodell
Abbildung 2:	Startfenster in Access 2004
Abbildung 3:	<i>Design View</i> von Tabellen
Abbildung 4:	Die Tabelle der Mitarbeiter
Abbildung 5:	Die Tabelle der Verkäufe
Abbildung 6:	Die Tabellen und ihre Beziehungen
Abbildung 7:	Beschreibung zur SQL-View
Abbildung 8-12:	SQL-Abfragen
Abbildung 13:	Hilfsassistent für Forms
Abbildung 14:	Das Formular für Mitarbeiter
Abbildung 15:	Erklärung zur Funktion <i>Insert Object</i>
Abbildung 16:	Beispiel für verschachtelte Forms
Abbildung 17:	Diagramm der Verkaufsintensität
Abbildung 18:	Startfenster der Datenbank