Gestaltung einer Datenbank für ein Hotel

Seminararbeit

Eingereicht bei

Prof. Dr. Andreas Meier

Betreuer: Daniel Wismer

Lehrstuhl für Wirtschaftsinformatik

Wirtschafts- und Sozialwissenschaftliche Fakultät

Universität Fribourg

Von

Martin Abderhalden

Via Leoncavallo 24

CH-6614 Brissago

Tel.: 091/793.12.41

Fax: 091/793.28.08

E-mail: martin.abderhalden@unifr.ch

Studentennr.: 02206290

Studium: Wirtschaft

2. Semester

Fribourg, September 2003

Inhaltsverzeichnis Seite

1 Einleitung

1.1 Problemstellung

Die Novo Gastro SA ist ein mittelgrosses Unternehmen in Brissago, im Kanton Tessin. Es ist ein Hotel (41 Zimmer und zirka 80 Betten) und gleichzeitig ein Restaurant (Zirka 70 Tische). Alle Kunden werden bei jeder Buchung wieder neu erfasst, auch wenn diese schon mal in diesem Hotel waren (Auch Stammkunden werden nicht ein für allemal registriert).
Seit langer Zeit werden Kundendaten mit Hilfe eines Karteisystems verwaltet und die Zimmerbuchungen und auch die Tischreservierungen werden separat in einem Kalender aufgezeichnet. Will man zum Beispiel eine Zimmerbuchung durchführen, werden zuerst die Kundendaten im Karteisystem erfasst und nachträglich die Buchung im Kalender aufgezeichnet (Doppelnennungen).Wenn dieser Kunde auch noch einen Tisch oder Saal reservieret, wird dies nochmals auf einem anderen Kalender markiert. Es ist offensichtlich, dass so eine separate Registrierung der diversen Daten ziemlich kompliziert werden kann. Speziell in der Hochsaison passiert es öfters dass die Verwaltung der diversen Daten ausser Kontrolle gerät. Auch die Sekretärin hat sich beschwert, dieses System wäre zu unübersichtlich und zeitraubend und möchte darum ein moderneres, mit Access definiertes Programm.
.

September

Zimmer/tag	1. 9	2. 9	3. 9	4. 9	5. 9	6. 9	7. 9	…….	…..
Nr 1	Baendle	Baendle	Rossi	Rossi	Rossi	Rossi		……	….
Nr 2	Giorgi			Peyer	Peyer	Peyer		……	…..
Nr 3	Baendle	Baendle				Brughelli	Brughelli	……..	…..
Nr 4	Maechler			Burkhardt	Burkhardt			……...	…..
Nr 5	Machler							……	……
Nr 6		Tobler	Tobler	Tobler	Tobler	Tobler	Tobler	……	…..
Nr 7				Schmidt	Schmidt			……	

Abb.1: Zimmerbuchungen mittels Kalender (Bsp. September).

1.2 Zielsetzung

Mit diesem neuem System sollte es möglich sein, Kundendaten, Zimmerbuchungen, Tischreservationen und Saalreservationen schneller und bequemer zu handhaben. Ein einziges Programm verwaltet also diese verschiedenen Komponenten und die diversen Integritätsbedingungen helfen dem Anwender, die Daten korrekt einzutragen. Daten von Stammkunden sollten ein für allemal registriert werden. Auch bei allfälligen Änderungen (Bsp: neue Zimmer, Zimmerrestaurierungen, Preisänderungen, usw.) sollte es möglich sein die Datenbank auszubauen oder abzuändern. Um eine Übersicht über den Geschäftsablauf zu kriegen, wäre es nützlich wenn man am Ende des Monats Rapporte erstellen könnte (zum Beispiel Anzahl Übernachtungen).

1.3 Vorgehensweise

Nach einem Treffen mit der Sekretärin werden zuerst die gewünschten Inhalte der Datenbank erläutert und daraus wird ein Entitäten-Beziehungsmodell entwickelt. Dieses

Modell wird in ein relationales Datenbankschema überführt und anhand der Normalformen kontrolliert. Dies wiederum ist der erste Baustein, die nächsten Schritte zu verfolgen, mit Access eine voll funktionierende Datenbank zu erstellen.

2 Datenanalyse

Nach einem langen Gespräch mit der Sekretärin wurden die vorhandenen Daten und deren Beziehungen analysiert und vorgehoben.
Mit der gewünschten Datenbank sollte man Zimmerbuchungen, Tischreservationen und Saalreservationen verwalten können.

Wichtig für die Buchung der Zimmer ist: der Kunde (Name, Vorname, Adresse, Land) bucht an einem bestimmten Tag, für eine gewisse Zeit ein oder mehrere Zimmer. Es muss also auch das An-und Abreisedatum erfasst werden. Der Gast kann zwischen diversen Zimmern auswählen (Diverse Preise, mit oder ohne Dusche, mit oder ohne WC ,Lage, Anzahl Betten).

Für die Reservation der Tische ist der Kunde, das Datum, die Uhrzeit und der gewünschte Tisch (Anzahl Plätze) von relevanter Bedeutung.

Für die Reservation der Säle ist der Kunde, der Saal (Anzahl Plätze), das Datum und die Uhrzeit von relevanter Bedeutung.

3 Datenmodellierung

3.1 Entitäten-Beziehungsmodell erstellen

Nach der Datenanalyse wird nun ein Entitäten-Beziehungsmodell (Entity-Relationship-modell, kurz: ERM) entwickelt um die Wirklichkeit zu vereinfachen, diskretisieren und graphisch darzustellen. In diesem Fall wird das ER-modell , welches auf dem relationalen Ansatz von Codd (1970) basiert und von Chen (1978) entwickelt wurde [vgl. Zehnder 98, S.61], benützt. Dieses ERM besteht aus Entitätsmengen, Beziehungsmengen und deren Assoziationen und Attributen.
Bevor nun damit angefangen wird, müssen zuerst einige Definitionen erklärt werden:

«Eine Entität ist eine Sache, welche eindeutig identifizierbar ist. Entitäten des gleichen Typs werden zu Entitätsmengen zusammengefasst und durch Merkmale weiter charakterisiert». [Zehnder 98, S.63] Im ERM werden sie als Rechtecke dargestellt.

«Eine Beziehung ist eine Zuordnung zwischen den Entitäten. Auch hier werden Beziehungen des gleichen Typs zu Beziehungsmengen zusammengefasst und durch Merkmale weiter definiert». [Zehnder 98, S.68] Im ERM werden sie als Rhomben dargestellt.

Um eine Beziehung genauer zu definieren existieren diverse Assoziationstypen. Eine Assoziation (EM1, EM2) legt fest, wie viele Entitäten aus der Entitätsmenge 2 (EM2) einer Entität aus der Entitätsmenge 1 (EM1) zugeordnet sein können. Insgesamt gibt es vier verschiedene Assoziationstypen:

Typ 1: Jeder Entität aus der Entitätsmenge EM1 wird „genau eine“ Entität aus der Entitätsmenge EM2 zugeordnet.
Typ c: Jeder Entität aus der Entitätsmenge EM1 wird „keine oder eine“ Entität aus der Entitätsmenge EM2 zugeordnet.
Typ m: Jeder Entität aus der Entitätsmenge EM1 werden „mehrere“ Entitäten aus der Entitätsmenge EM2 zugeordnet.
Typ mc: Jeder Entität aus der Entitätsmenge EM1 werden „keine, eine oder mehrere“ Entitäten aus der Entitätsmenge EM2 zugeordnet. [Vgl. Meier 01,S 18 f und Dizionario di informatica, S. 504]

Eine Beziehung umfasst immer zwei Assoziationstypen: die erste von EM1 nach EM2 und die zweite von EM2 nach EM1. Dies beschreibt die Mächtigkeit einer Beziehung und wird in der Überführung des ERM in ein Datenbankschema eine wichtige Rolle Spielen.

Ein Attribut ist die Beschreibung einer bestimmten Eigenschaft der Entitäten einer Entitätsmenge und definiert den Wertebereich der zugeordneten Datenwerte.
[vgl. Zehnder, S.70]

«Ein Identifikationsschlüssel ist ein Schlüssel, dessen Wert jede Entität einer Entitätsmenge eindeutig identifiziert». [Zehnder, S.42]

Somit sind die wichtigsten Elemente um ein Entitäten-Beziehungsmodell aufzubauen definiert worden.

Entitäten
Nun widmen wir uns den Entitäten und deren Attributen und Identifikationsschlüssel (fett gedruckt) zu. Man hat nun vier Entitätsmengen mit den zugehörigen Attributen:
KUNDE (**Knr**, Name, Vorname, Ort, Strasse, PLZ, Land)
ZIMMER (**Znr**, WC, Preis, Dusche, Betten, Lage)
TISCH (**Tnr,** Plätze)
SAAL (**Snr**, Plätze)

Beziehungen
Die Beziehungen zwischen diesen vier Entitäten lassen sich wie folgt definieren:
Jeder Kunde kann kein, ein, oder mehrere Zimmer reservieren (Typ mc).
Jedes Zimmer kann von Keinem, einem oder mehreren Kunden gebucht werden (Typ mc).
Jeder Kunde kann keinen, einen oder mehrere Tische reservieren (Typ mc).
Jeder Tisch kann von keinem, einem oder mehreren Kunden reserviert werden (Typ mc).
Jeder Kunde kann einen, keinen oder mehrere Säle reservieren (Typ mc).
Jeder Saal kann von keinem, einem oder mehreren Kunden reserviert werden (Typ mc).

Aus diesen Informationen erhält man die folgenden Beziehungen, dabei ist zu beachten dass diese alle komlex-komplex sind.
ZIMMERBUCHUNG (**Knr**, **Znr**, Anreisedatum, Abreisedatum) (mc ;mc)
TISCHRESERVIERUNG (**Tnr**, **Knr**, Datum, Zeit) (mc ;mc)
SAALRESERVIERUNG (**Snr**, **Knr**, Datum, Zeit) (mc ;mc)

Entitäten-Beziehungsmodell

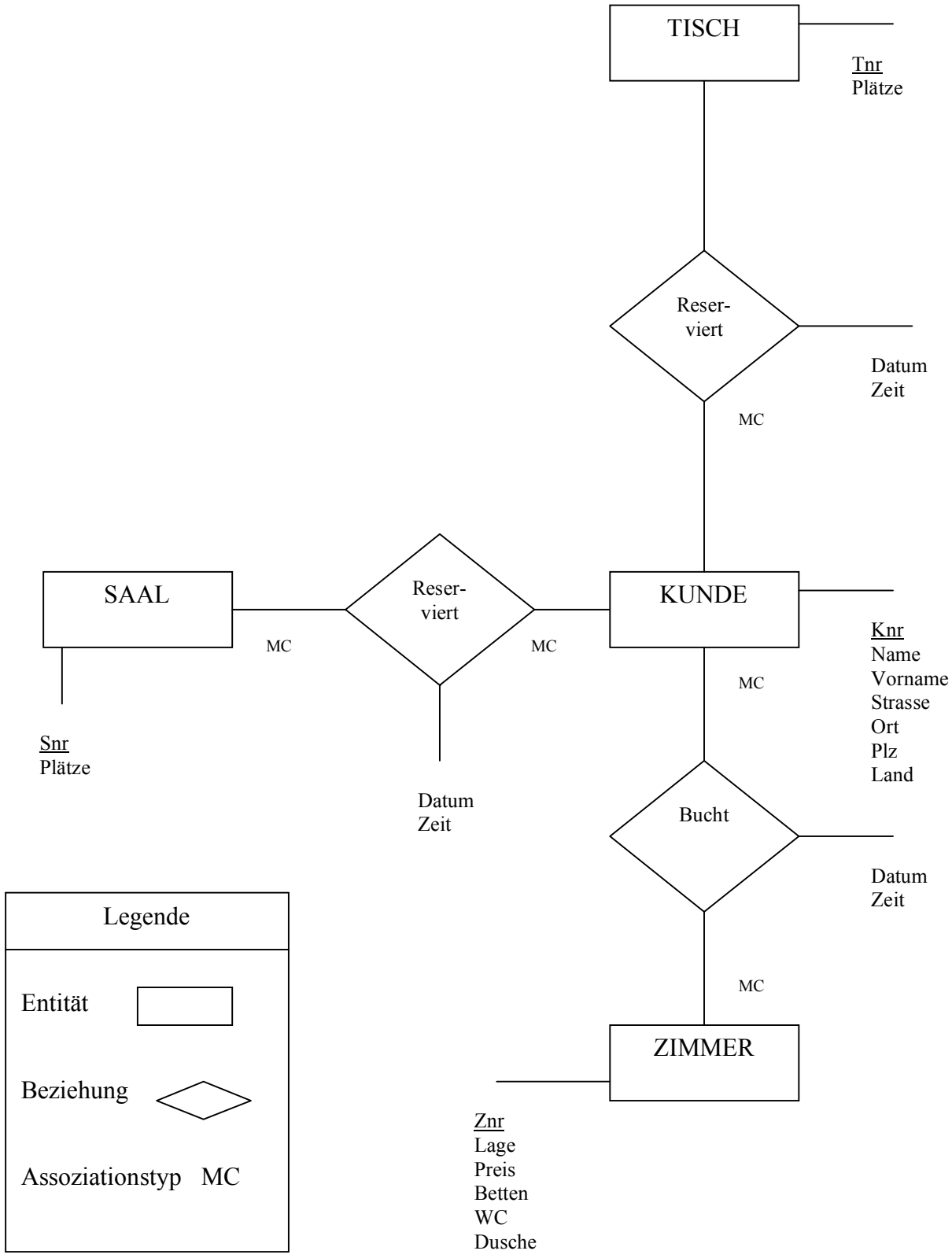

Abb.2: Entitäten-Beziehungsmodell.

3.2 Datenbankschema entwickeln

In diesem Abschnitt wird das in Punkt 3.1 definierte ERM in ein relationales Datenbankschema überführt. Unter einem Datenbankschema versteht man eine Datenbankbeschreibung, das heisst die Spezifikation von Datenstrukturen mitsamt ihren zugehörigen Integritätsbedingungen. Es enthält die Definition der Tabellen, der Merkmale und der Primärschlüssel. Integritätsbedingungen legen Einschränkungen für die Wertebereiche, für die Abhängigkeiten zwischen verschiedenen Tabellen sowie für die eigentlichen Datenvorkommen fest [vgl. Meier 01,S 23-24]. Um das ERM in ein Datenbankschema zu überführen sind Abbildungsregeln [Alle Abbildungsregeln stammen aus: Meier 01,S 23-24] von grosser Bedeutung, diese beschreiben auf genaue Weise die Überführung.

Regel 1 (Entitätsmengen)
«Jede Entität *muss* als eigenständige Tabelle mit einem eindeutigen Primärschlüssel definiert werden. Als Primärschlüssel der Tabelle dient entweder der entsprechende Schlüssel der Entitätsmenge oder ein Schlüsselkandidat. Die übrigen Merkmale der Entitätsmengen gehen in die korrespondierenden Attribute der Tabellen über»[Meier 01,S 25].

In diesem Fall werden alle Entitäten in eigenständige Tabellen verwandelt (Primärschlüssel sind unterstrichen) und erhalten somit die folgenden Relationen:

KUNDE

Knr	**Name**	**Vorname**	**Strasse**	**Ort**	**Plz**	**Land**

ZIMMER

Znr	**WC**	**Preis**	**Dusche**	**Betten**	**Lage**

TISCH

Tnr	**Plätze**

SAAL

Snr	**Plätze**

Abb.3: Entitäten in Tabellenform.

Regel 2 (Beziehungsmengen)
«Jede Beziehungsmenge *kann* als eigenständige Tabelle definiert werden, wobei die Identifikationsschlüssel der zugehörigen Entitätsmengen als Fremdschlüssel in dieser Tabelle auftreten müssen. Der Primärschlüssel der Beziehungsmengentabelle kann der aus den Fremdschlüsseln zusammengesetzte Identifikationsschlüssel sein oder ein anderer Schlüsselkandidat, zum Beispiel in Form eines künstlichen Schlüssels. Weitere Merkmale der Beziehungsmenge erscheinen als zusätzliche Attribute in der Tabelle».
[Meier 01,S 25]

Im Gegensatz zur ersten Regel ist diese eine Kann-Regel, sie muss also nicht unbedingt angewandt werden. Sie dient aber als Basis der nächsten Regeln, denn um Beziehungsmengen in Relationen zu überführen spielt die Spezifikation der Assoziationen eine wichtige Rolle. Es kommt also darauf an, was für eine Mächtigkeit die Beziehungsmenge hat. Die dritte Regel ist speziell für komplex-komplexe Beziehungen bestimmt. Für die Überführung von einfach-komplexen Beziehungen ist die Regel 4 notwendig und die Regel 5 spielt bei einfach-einfachen Beziehungen eine Rolle.

Regel 3 (komplex-komplexe Beziehungen)
«Jede komplex-komplexe Beziehungsmenge muss als eigenständige Tabelle definiert werden. Dabei treten mindestens die Identifikationsschlüssel der zugehörigen Entitätsmengen als Fremdschlüssel auf. Der Primärschlüssel der Beziehungsmengentabelle ist entweder der aus den Fremdschlüsseln zusammengesetzte Identifikationsschlüssel, oder ein anderer Schlüsselkandidat. Die weiteren Merkmale der Beziehungsmenge gehen in die Attribute der Tabelle über».[Meier 01,S 27]

Das bedeutet dass alle Beziehungsmenge die komplex-komplex sind, also vom Typ (m;m), (m;mc), (m;mc) oder (mc;mc), müssen als eigenständige Tabelle definiert werden. In diesem Projekt sind alle Beziehungsmengen komplex-komplex (mc;mc) und müssen darum als eigenständige Tabellen dargestellt werden. Als Primärschlüssel dient die Zusammensetzung der Fremdschlüssel, die aus den referenzierten Tabellen stammen, sie sind in der Tabelle unterstrichen.
Somit erhält man die folgenden Tabellen:

ZIMMERBUCHUNG

Knr	**Znr**	**Anreisedatum**	**Abreisedatum**

TISCHRESERVIERUNG

Tnr	**Knr**	**Datum**	**Zeit**

SAALRESERVIERUNG

Snr	**Knr**	**Datum**	**Zeit**

Abb.4: Beziehungen in Tabellenform.

Da in diesem Projekt keine einfach-komplexe oder einfach-einfache Beziehungen vorkommen, werden die Regeln 4 und 5 in diesem Kapitel nicht angewandt. Auch die Regeln 6 und 7 werden nicht erwähnt, da in diesem Fall keine Generalisationen und Aggregationen vorkommen.

3.3 Normalformen und Integritätsbedingungen

Kontrolle anhand Normalformen

Die Normalisation ist eine gleichwertige Entwurfsmethode wie die «Überführung anhand Abbildungsregeln».
Sie stammt von Codd und wurde entwickelt um sicher zu sein, dass eine Datenstruktur effizient ist. Da die Normalisation und die «Überführung anhand Abbildungsregeln» den gleichen Endzustand anstreben, kann die eine Methode benutzt werden, um die korrekte Anwendung der anderen Methode zu kontrollieren.
In anderen Worten wird in diesem Schritt nun also kontrolliert, ob die Abbildungsregeln korrekt angewandt worden sind.
Bei den verschiedenen Abhängigkeiten können eventuelle Unstimmigkeiten auftreten, diese werden hier entdeckt und behoben. Somit erhält man ein Datenbankschema welches frei von redundanten Informationen ist.
«Ein Merkmal einer Tabelle ist Redundant, wenn einzelne Werte dieses Merkmals innerhalb der Tabelle ohne Informationsverlust weggelassen werden können» [Meier 01, S.34]
Falls ein Datenbankschema doch Tabellen mit redundanten Informationen enthält, können sogenannte Mutationsanomalien auftreten (Einfügeanomalien, Löschanomalien oder Änderungsanomalien) [Vgl. Atzeni Ceri Paraboschi Torlone, S.48].
In diesem Projekt werden die ersten drei Normalformen genügen.

1. Normalform

«Eine Tabelle ist in erster Normalform, falls die Wertebereiche der Merkmale atomar sind. Die erste Normalform verlangt, dass jedes Merkmal Werte aus einem unstrukturierten Wertebereich bezieht. Somit dürfen keine Mengen, Aufzählungstypen, oder Wiederholungsgruppen in den einzelnen Merkmalen vorkommen».
[Meier 01, S.36].

Wie man erkennen kann, sind alle Tabellen in erster Normalform, da die Wertebereiche der Merkmale atomar sind.

2. Normalform

«Eine Tabelle ist in zweiter Normalform, wenn sie in erster Normalform ist und wenn jedes Nichtschlüsselmerkmal von jedem Schlüssel voll funktional abhängig bleibt»
Um zu kontrollieren ob die Tabellen in zweiter Normalform sind, muss zuerst erklärt werden was voll funktionale Abhängigkeit bedeutet.
Funktionale Abhängigkeit bedeutet, dass jeder Wert des Primärschlüssel eindeutig eine Wert der Merkmale bestimmt. Volle funktionale Abhängigkeit verlangt, dass nur der zusammengesetzte Schlüssel ein Nichtschlüsselmerkmal eindeutig bestimmt. Das heisst, dass ein Teil des Schlüssels ein Nichtschlüsselmerkmal nicht eindeutig bestimmen darf, sonst würde keine volle funktionale Abhängigkeit herrschen. [vgl. Meier 01, S.37]

Bei den drei Beziehungtsabellen ZIMMERBUCHUNG, TISCHRESERVIERUNG, SAALRESERVIERUNG, muss also kontrolliert werden, dass Teile des zusammengesetzten Schlüssels ein Nichtschlüsselmerkmal nicht eindeutig bestimmt.

Bei der Tabelle ZIMMERBUCHUNG ist der Schlüssel die Zusammensetzung der Kundennummer (Knr) und der Zimmernummer (Znr). Keiner dieser zwei Teile bestimmt alleine eines der anderen zwei Nichtschlüsselmerkmale (Anreisedatum, Abreisedatum). Nur der Zusammengesetzte Schlüssel bestimmt eindeutig das Anreisedatum und das Abreisedatum. Das bedeutet, dass diese Tabelle in zweiter Normalform ist.

Der Schlüssel der Tabelle TISCHRESERVIERUNG ist die Zusammensetzung der Kundennummer (Knr) und der Tischnummer (Tnr).
Da kein Teil des Schlüssels alleine das Merkmal Datum und das Merkmal Zeit eindeutig bestimmt, ist auch diese Tabelle in zweiter Normalform.

In der Tabelle SAALRESERVIERUNG ist die Situation analog. Nur der Zusammengesetzte Schlüssel (Snr und Knr) bestimmt das Merkmal Datum und das Merkmal Zeit. Darum ist auch diese Tabelle in Zweiter Normalform.

3. Normalform

«Eine Tabelle ist in dritter Normalform, wenn sie in zweiter Normalform ist und kein Nichtschlüsselmerkmal von irgendeinem Schlüssel transitiv abhängig ist».
Transitive Abhängigkeit bedeutet, über Umwege funktional abhängig zu sein. [Meier 01, S.37]. Hierzu eine kleine Grafik:

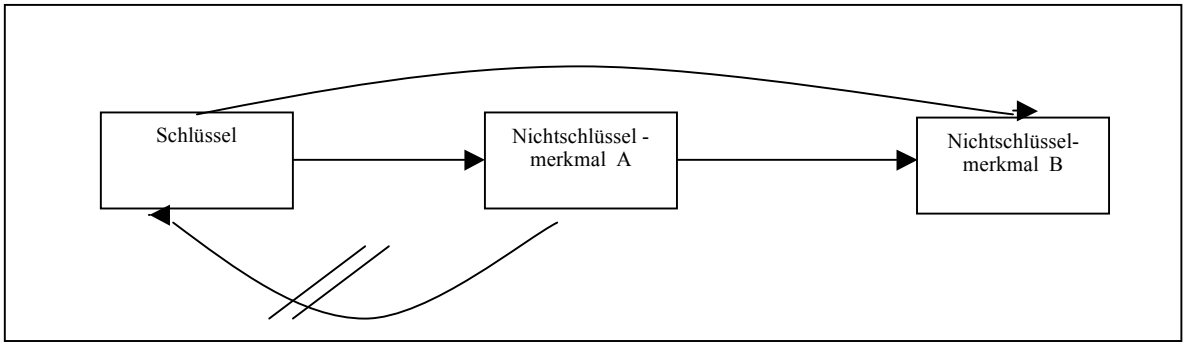

Abb.5: Transitive Abhängigkeit.

In diesem Projekt sind alle Tabellen in Dritter Normalform , ausser die Tabelle KUNDE welche noch einmal kontrolliert werden muss. Da das Merkmal Ort und das Merkmal Plz in Theorie das gleiche aussagen (Bsp. 6614=Brissag, 6612=Ronco, etc) existiert in diesem Fall eine transitive Abhängigkeit. Um diese Tabelle in die dritte Normalform zu bringen, teilt man die Tabelle in zwei Teile. Das redundante Merkmal Ort wird somit in eine eigenständige Tabelle verlegt und als Primärschlüssel dieser neuen Relation fungiert das Merkmal Plz:

KUNDE

Knr	Name	Vorname	Strasse	Plz	Land

ORT

Plz	Ort

Abb.6: Tabelle KUNDE in dritter Normalform.

Strukturelle Integritätsbedingungen

«Integrität oder Konsistenz bedeutet Widerspruchsfreiheit von Datenbeständen. Eine Datenbank ist integer oder konsistent, falls die gespeicherten Daten fehlerfrei erfasst sind und den gewünschten Informationsgehalt korrekt wiedergeben»[Meier 01, S.44]

Es existieren nun drei verschiedene Integritätsbedingungen die eine relationale Datenbank erfüllen muss um integer zu sein.
Die Eindeutigkeitsbedingung verlangt, dass jede Tabelle einen Identifikationsschlüssel besitzt, der jedes Tupel in der Tabelle auf eindeutige Art bestimmt.
In diesem Projekt wird diese Integritätsbedingung von allen Tabellen eingehalten.
Die Wertebereichsbedingung besagt, dass die Merkmale einer Tabelle nur Datenwerte aus einem vordefinierten Wertebereich annehmen dürfen, sie wird im Punkt 5.3 kurz erklärt. Die referentielle Integritätsbedingung will, dass jeder Wert eines Fremdschlüssels in einer Tabelle auch wirklich als Schlüsselwert in der referenzierten Tabelle existieren muss. Dass bedeutet zum Beispiel dass bei der Tabelle ZIMMERBUCHUNG der Fremdschlüsselwert Knr, auch wirklich in der Tabelle KUNDE als Primärschlüssel existieren muss. Bei Betrachtung der Tabellen sieht man, dass auch diese Bedingung erfüllt wird.

Die Datenmodellierung ist nun abgeschlossen und um eine klare Sicht über die Datenbank zu erhalten, wird das komplette Schema noch mal abgebildet:

ZIMMERBUCHUNG

Knr	Znr	Anreisedatum	Abreisedatum

TISCHRESERVIERUNG

Tnr	Knr	Datum	Zeit

SAALRESERVIERUNG

Snr	Knr	Datum	Zeit

ZIMMER

Znr	WC	Preis	Dusche	Betten	Lage

TISCH

Tnr	Plätze

SAAL

Snr	Plätze

KUNDE

Knr	Name	Vorname	Strasse	Plz	Land

ORT

Plz	Ort

Abb.7: Komplettes Datenbankschema

Dieses Datenbankschema ist in dritter Normalform und erfüllt die Integritätsbedingungen, im nächsten Kapitel wird es nun mit Access implementiert.

5 Implementierung

5.1 Access

Es existieren verschiedene Modelltypen von Datenbanken, wie zum Beispiel Netzwerkmodelle oder hierarchische Datenmodelle. Doch die bekannteste Datenbank ist die relationale Datenbank, die alle Daten in Tabellenform darstellt. Die Vorteile einer relationalen Datenbank ist, dass sie von der deskriptiven Abfragesprache SQL unterstützt wird, dass bedeutet, dass auch ein gelegentlicher Benutzer eine echte Chance hat, in kurzer zeit die Datenbank zu nützen und mit ihr zu arbeiten. Ausserdem ist die leichte Handhabung ein grosser Vorteil im Gegensatz zu anderen Datenbanktypen, denn diese erfordern meistens Programmierkenntnisse um die Datenbank auswerten zu können. Aus diesen Gründen ist die relationale Datenbank heute die meist angewandte Datenbank. [Vgl. Dizionario dei termini dell'informatica, S.343] Darum wird in diesem Projekt Access benutzt, um die Implementierung durchzuführen.

5.2 Datenbankschema in Access definieren

Nach dem Öffnen des Programms wird angegeben, dass man mit einer neuen Datenbank arbeiten möchte.
Als erstes wird die Tabelle KUNDE in drei Schritten wie folgt definiert:

Schritt 1

Im Register *Tables* wählt man zuerst die Schaltfläche *New* und dann die *Entwurfssicht*. Nun erscheint eine neue Tabelle, in der man die Tabelle Kunde definiert, in dem man die diversen Attribute hineinfügt. Für jedes Attribut müssen die folgenden Angaben festgelegt werden: Name des Attributs, Datentyp und eine Beschreibung des Attributs. In der zweiten Kolonne (*Data Type*) kann man den Datentyp eingeben, in dem man auf die Auswahlliste (kleiner Pfeil) Klickt und einen bestimmten Typ selektioniert. In der dritten Kolonne *Description* wird die Beschreibung hineingetippt. [vgl. Sherry Kinkoph,*1999. S. 228*]

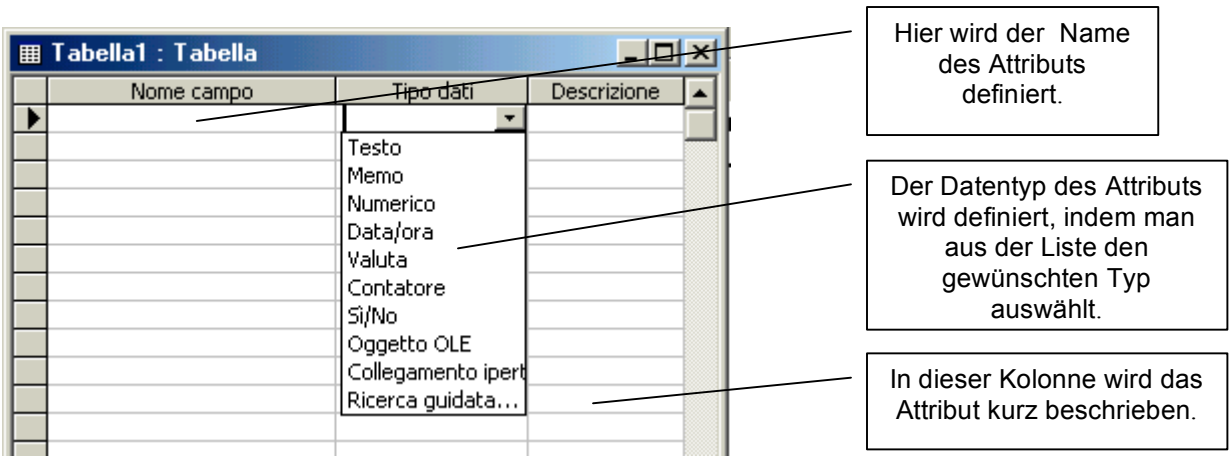

Abb.8: Definieren einer Tabelle.

Schritt 2

Hier wird nun angezeigt, welches Attribut der Tabelle KUNDE der Primärschlüssel ist. Man markiert das gewünscht Attribut das als Primärschlüssel dienen soll und verwendet dann den Befehl *Edit/primary Key*. (im Falle eines zusammengesetzten Schlüssels ,wird zuerst nur ein Schlüsselmerkmal markiert und dann die CTRL Taste gedrückt. Nun wird auch das zweite Schlüsselmerkmal markiert und somit sind beide Merkmale als Schlüssel definiert).Links von der Kolonne *Field Name* steht neben dem Attribut ein Symbol eines Schlüssels. [Vgl. Sherry Kinkoph,*1999. S. 229*]

Schritt 3

Nun schliesst man diesen Tabellenentwurf und es wird somit angefragt, wie diese Tabelle heissen soll. (in diesem Fall, KUNDE)

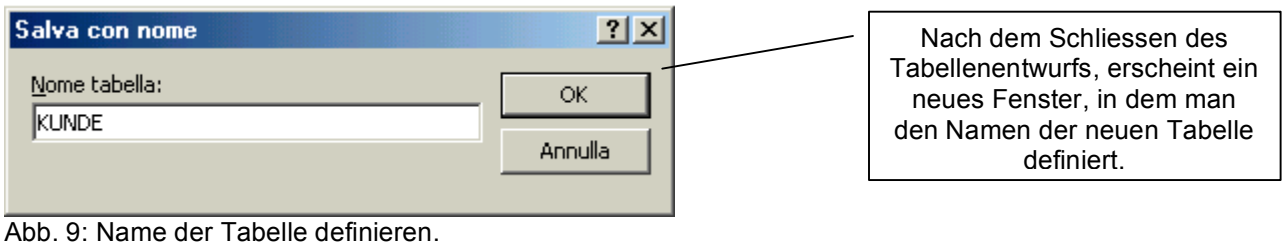

Abb. 9: Name der Tabelle definieren.

Nach diesen drei Schritten ist diese Tabelle vollständig definiert und die Entwurfssicht sieht folgendermassen aus:

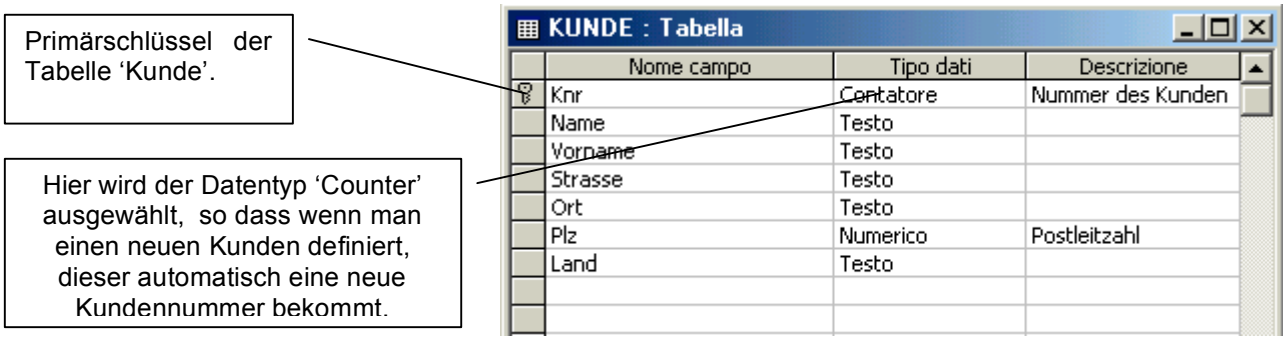

Abb.10: Entwurfssicht der Tabelle KUNDE.

Analog werden nun die anderen Tabellen definiert, dabei ist zu beachten, dass drei Tabellen einen Schlüssel besitzen der aus zwei Attributen besteht!

5.2 Wertebereichsbedingungen spezifizieren

In diesem Schritt werden Datenwerte für ein bestimmtes Attribut eingeschränkt .Wie kann man zum Beispiel eine Regel eingeben, die kontrolliert, dass der Preis eines Zimmers mehr als 55 SFr. beträgt? Oder, da das Hotel im Winter für zwei Wochen (vom 14. bis am 28. Dezember) geschlossen ist, dürfen für diese Zeit keine Buchungen angenommen werden. Wenn man doch in dieser Zeit eine Zimmerbuchung eingeben möchte, sollte das Programm mit einer Fehlermeldung Alarm schlagen.

Um die Wertebereichsbedingung für die Merkmale Anreise und Abreisedatum einzuschränken wird die betroffene Tabelle im Entwurfsmodus geöffnet und darin eines der beiden Merkmale markiert. Im unteren Bereich des Fensters erscheinen die detaillierten Informationen zum markierten Attribut. Nun wird im Feld *Validation Rule* die Prüfregel für das gewünschte Attribut eingegeben. [Vgl.Meier 01. S. 91 f]

In diesem Fall werden die Attribute Anreise und Abreisedatum eingeschränkt, in dem man die folgenden Eingaben eintippt:

«<14/12/2003 OR >28/12/2003».

Im Feld *Validation Text* kann nun auch noch angegeben werden, wieso diese Eingabe nicht akzeptiert wird. In diesem Fall wird ein Aussagekräftiger Text eingegeben: «Das Hotel ist in dieser Zeit geschlossen».

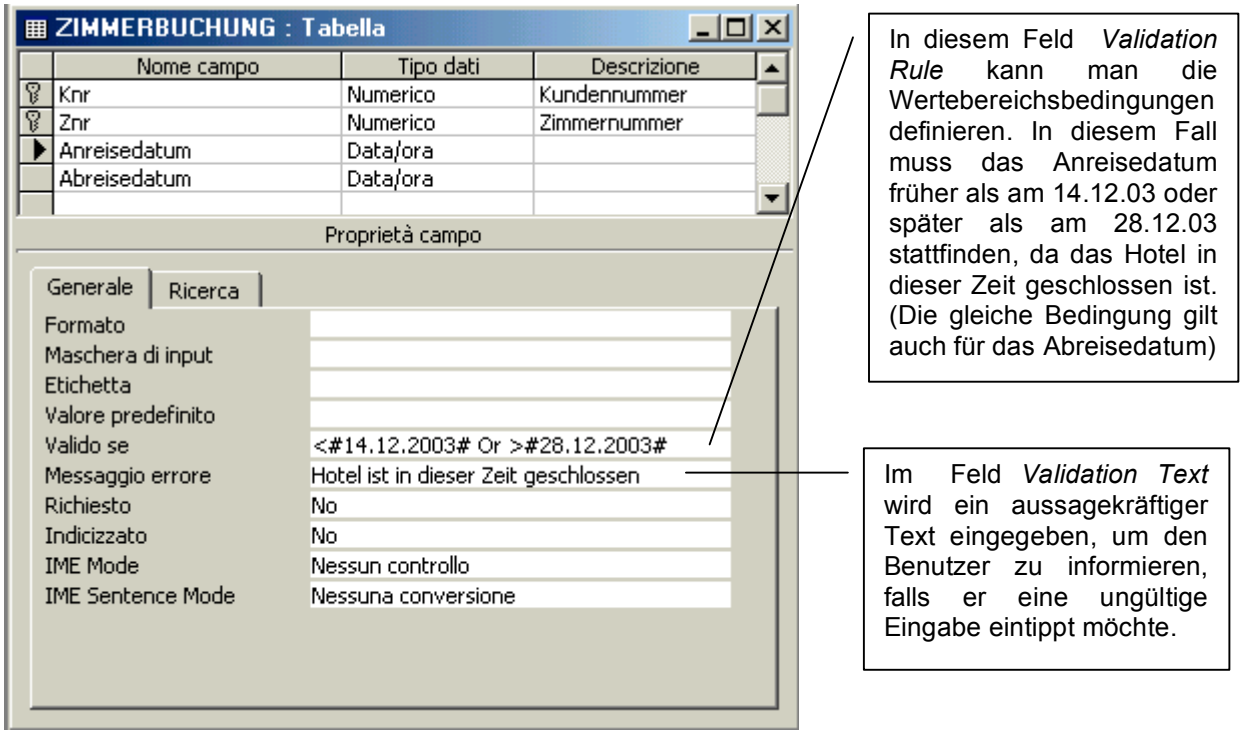

Abb.11: Wertebereichsbedingungen spezifizieren.

Das gleiche Vorgehen wird nun auch angewandt um zu kontrollieren das jedes Zimmer mindestens 55 SFr. Kostet. Im Feld *Validation Rule* wird die Regel«>55» eingetippt. Und Im Feld *ValidationText* wird die Aussage, «Jedes Zimmer kostet mindestens 55 SFr eingetragen».
Somit sind alle Wertebereichsbedingungen Dieses Projektes definiert worden.

5.3 Prüfregeln für Tupel festlegen

Eine zweite Integritätsbedingung ist, das Festlegen von Prüfregeln für Tupel. Sie bezieht sich auf mehrere Merkmale desselben Tupels und setzt deren Inhalte zueinander in Beziehung. [Vgl. Meier,01. S.92]
In diesem Projekt will man zum Beispiel, dass der Preis eines Zimmers mindestens 50 SFr. pro Bett kostet. Wenn ein Zimmer zum Beispiel 3 Betten besitzt, sollte der Preis mindestens 150 SFr. betragen. Diese Bedingung lässt sich durch die beiden Attributen Betten und Preis ausdrücken: Preis> Betten*50. Um diese Integritätsbedingung zu formulieren, wird die Tabelle ZIMMER im Entwurfsmodus geöffnet. Mit einem Klick auf die blaue Titelleiste erscheint eine Auswahlliste, in der man *Properties* auswählt [Vgl. Sherry Kinkoph, *S. 232*]. Es erscheint ein Fenster mit den Eigenschaften der Tabelle. In dem Feld *Validation Rule* wird nun die oben erwähnte Regel eingetippt . Ausserdem kann man auch hier einen Text (im Feld *Validation Text*) eingeben der als Fehlermeldung auftritt falls diese Inegritätsbedingung verletzt wird.

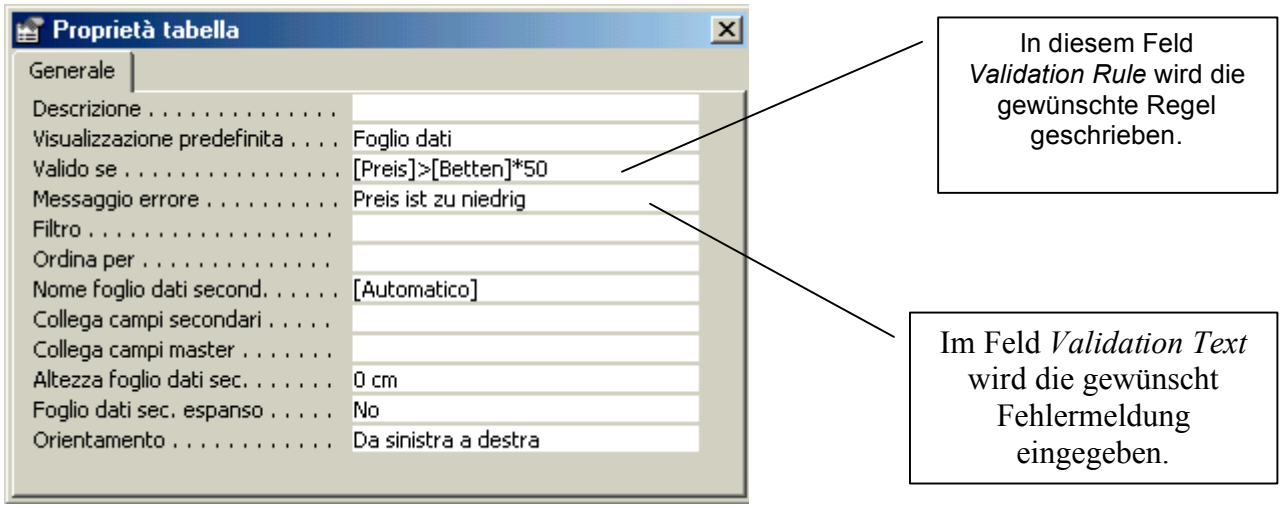

Abb.12: Prüfregel für Tupel festlegen.

5.4 Referentielle Integrität formulieren

Es ist notwendig dass zum Beispiel bei einer Buchung, Zimmer erwähnt werden, die auch wirklich existieren. Auch eine Kundennummer muss in der referenzierten Tabelle (in diesem Fall die Tabelle KUNDE) bei der Buchung übereinstimmen.
Die referentielle Integrität verhindert, dass in einem Fremdschlüsselmerkmal ein Wert eingetragen wird, zu dem in der referenzierten Tabelle kein Tupel existiert. Die referentielle Integrität sichert die Datenbank vor solchen ungültigen Dateneingaben ab[Vgl. Meier, 01. S.44]. Die Beziehungen zwischen den Fremdschlüsseln werden in der Entwurfssicht von Access als Linien dargestellt, die die Fremdschlüsselmerkmale mit den entsprechenden Primärschlüsseln der referenzierten Tabellen verbinden. [Vgl. Sherry Kinkoph, *S. 247*].
In den nächsten Schritten wird nun erklärt, wie man die referentielle Integrität formuliert:

Schritt 1

Man wählt den Befehl *Tools/Relationships* .Es wird ein Fenster angezeigt, aus dem man die gewünschten Tabellen, die miteinander verbunden werden sollen, auswählt (*Add*). In diesem Projekt müssen alle Tabellen selektioniert werden, da alle entweder ein Fremdschlüssel besitzen, oder einen Schlüssel besitzen, der als Fremdschlüssel in einer anderen Tabelle fungiert.

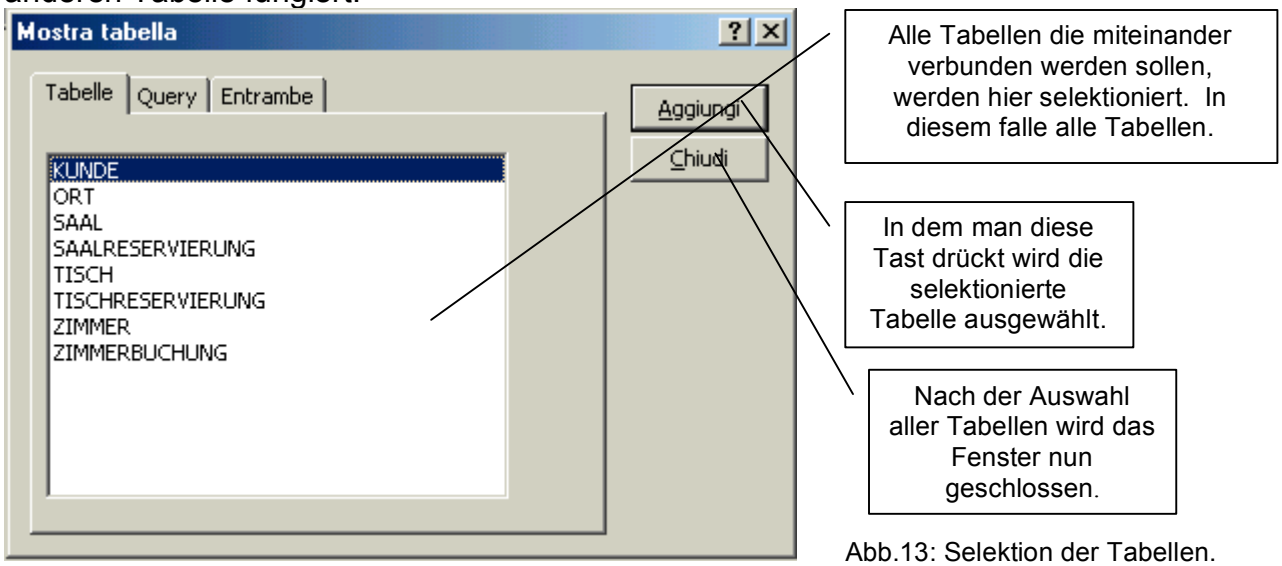

Abb.13: Selektion der Tabellen.

Schritt 2

Nach dem Schliessen des erwähnten Fensters, werden nun alle Tabellen schematisch in einem neuen Fenster (*Relationships*) angezeigt.

Nun werden die Fremdschlüssel und die Schlüssel in der referenzierten Tabelle miteinander verbunden. Als erstes wird in diesem Fall der Primärschlüssel der Tabelle Kunde (Knr) mit dem Schlüssel der Tabelle Zimmerbuchung verbunden. In dem man den Mauszeiger über dem Attribut Kunde.Knr gedrückt hält und ihn bis zum Attribut Zimmerbuchung.Knr zieht, wird diese Verbindung hergestellt. [Vgl. Sherry Kinkoph, *S. 250*] Nun erscheint ein neues Fenster, wo man die Definition der Fremdschlüsselbeziehung präziser definieren kann. Um die referentielle Integrität zu definieren, wird bei *Enforce Referential Integrity* ein Haken gesetzt. Nach dem Drücken des Befehls *Create,* ist die referentielle Integrität zwischen diesen zwei Tabellen abgeschlossen und sieht folgendermassen aus:

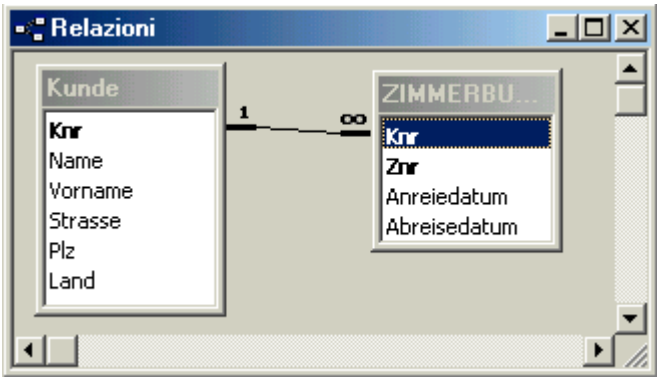

Abb.14: Referentielle Integrität zwischen zwei Tabellen.

Alle anderen Fremdschlüsselbeziehungen der restlichen Tabellen werden analog definiert und das *Relationship* Fenster sieht am Ende so aus:

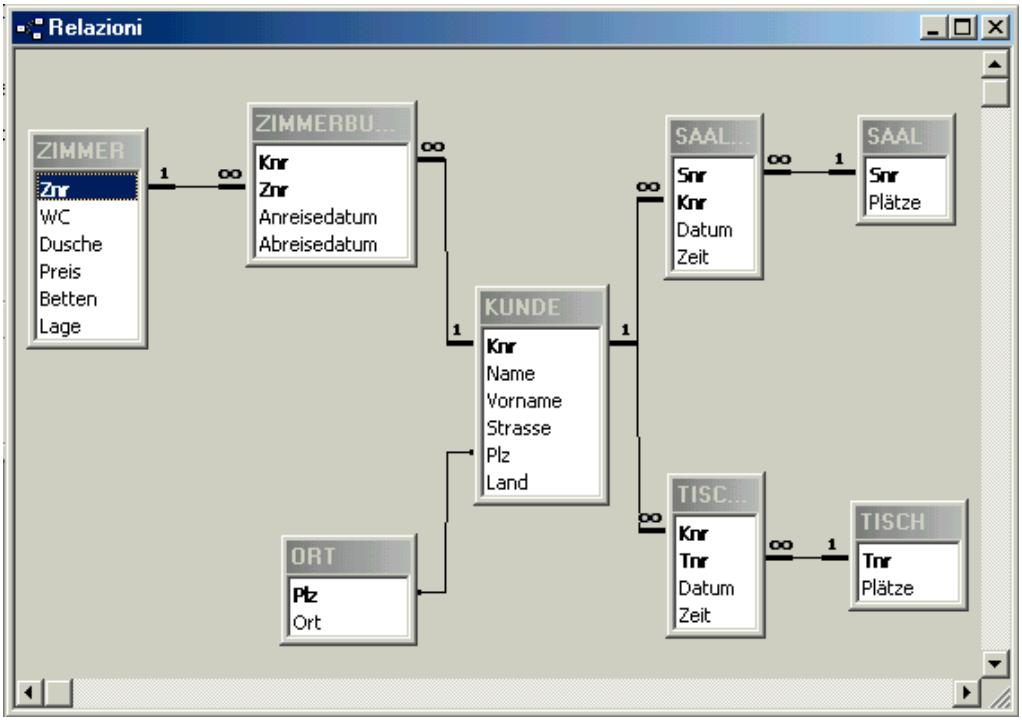

Abb.15: Referentielle Integrität zwischen allen Tabellen.

5.5 Die vorhandenen Daten eingeben

In diesem Punkt werden nun die Daten eingegeben. Wenn man sich im Datenbankfenster im Register Tables befindet, kann man die diversen Tabellen öffnen. Es erscheint eine leere Tabelle, in der man nun die vorhandenen Werte eingeben kann. Es ist zu beachten, dass Datenwerte die Integritätsbedingungen verletzen, nicht eingegeben werden können (das Programm verteidigt sich vor diesen falschen Angaben, in dem es Fehlermeldung angibt). Will man zum Beispiel eine Buchung am 26.12.03 eingeben, wird diese abgelehnt, da in dieser Zeit das Hotel geschlossen ist (Wertebereichsbedingung). Es ist auch zu beachten, dass Fremdschüssel auch wirklich in der referenzierten Tabelle als Schlüssel auftreten (referentielle integrität). Auch Prüfregel für Tupel müssen eingehalten werden.
Es ist zum Beispiel unmöglich den Preis eines Doppelzimmers auf 80 SFr. zu setzen, da eine Prüfregel verlangt, dass ein Bett mindestens 55 SFr. Kostet.

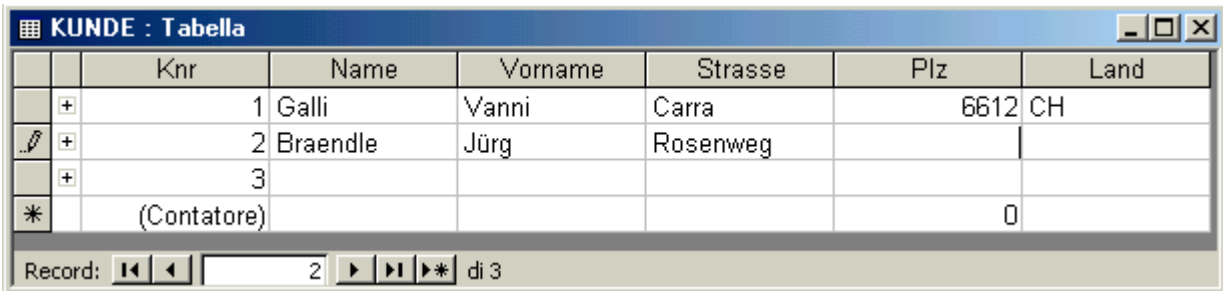

Abb.16: Eingabe der Daten. (Bsp.: Tabelle KUNDE.)

Somit ist die ganze Datenbank mit Access implementiert worden.

6. Einige SQL-Abfragen

Um die Datenbank auf ihre komplette Funktionalität zu überprüfen werden hier einige SQL-Abfragen ausprobiert:

Welche Zimmer befinden sich im Ruhigen Teil des Hotels, haben zwei Betten und kosten nicht mehr als 220 SFr.?
SELECT Znr
FROM Zimmer
WHERE Lage=' ruhig' **AND** Betten='2' AND Preis< 220

Znr
1
37
38
0

Welches Zimmer ist von Herr Braendle gebucht worden (mit an und Abreisedatum)?
SELECT Znr, Name, Anreisedatum, Abreisedatum
FROM ZIMMER,KUNDE, ZIMMERBUCHUNG
WHERE Name='Baendle' And Zimmer.Znr=Zimmerbuchung.Znr AND Kunde.Knr=Zimmerbuchung.Knr

	Znr	Name	Anreisedatum	Abreisedatum
▶	1	Braendle	29.12.2003	01.01.2004
	14	Braendle	01.02.2002	02.02.2004
	38	Braendle	08.01.2004	12.01.2004

Wie Teuer sind die Zimmer im Durchschnitt?
SELECT AVG (Preis)
FROM Zimmer

	Media Di Preis
▶	115

Welche Säle sind wann und von wem reserviert?
SELECT Snr, Name, Datum
FROM Saal, Kunde, Saalreservierung
WHERE Saal. Snr=Saalreservierun.Snr AND Kunde.Knr=Saalreservierung.Knr

	Snr	Name	Datum
▶	1	Galli	24.03.2003
	2	Abderhalden	25.01.2004

Welche Tische sind am 25.9.03 reserviert und von wem?
SELECT Tnr, Name
FROM Tisch, Tischreservierung
WHERE Tisch.Tnr=Tischreservierung.Tnr **AND** Datum=’25.9.03’

	Tnr	Name
	12	Hofer
	8	Galli

Ein Kunde Hätte gerne ein Ruhiges Zimmer mit WC und Dusche für zwei Personen und der Preis sollte nicht mehr als 120 Fr.- kosten. Welche Zimmer würden in Frage kommen?
SELECT Znr
FROM Zimmer
WHERE Preis< 220 **AND** Lage=’ruhig’ **AND** Dusche=’Ja’ **AND** WC=’Ja’

	Znr
▶	12
	17
	3

7. Schlusswort

Im ersten Semester hatte ich das erste mal etwas über Datenbanken Gehört, dieses Thema hat mich sehr interessiert . Da die Lektionen ziemlich theoretisch waren, habe ich mich entschieden, meine Seminararbeit auf diesem Gebiet zu schreiben, um das gelernte in die Praxis umzusetzen. Mit grossem Engagement habe ich mich dieser Aufgabe gestellt, doch schon am Anfang bei der Strukturierung der Arbeit wurde mir klar, dass es

nicht einfach sein würde, die Theorie in die Praxis umzusetzen. Vor allem die Datenmodellierung brachte mir einige Probleme, da ich mühe hatte die Realität in ein Entitäten-Beziehungsmodell zu überführen. Doch mit Geduld und gezielter Lektüre, habe ich auch dieses Problem lösen können. Die Implementierungssoftware Access war eine sehr gute Hilfe, um die gewünschte Datenbank zu implementieren. Dieses Projekt fand ich sehr interessant und lehrreich, da es mir einen kleinen Einblick in die Welt der Informatik erlaubt hat. Ich hoffe dass diese entwickelte Datenbank dem Unternehmen Novo Gastro SA die Arbeit erleichtert und dass somit eine bessere Übersicht der Zimmerbuchungen, Saal- und Tischreservationen möglich ist.

Literaturverzeichnis

- Atzeni Ceri Paraboschi Torlone,*Basi di dati.* McGraw-Hill.
- Daniela Dorbolo' e Andrea Guidi,*Guida a SQL*. McGraw-Hill.
- Meier, Andreas. *Relationale Datenbanken*. Vierte, überarbeitete und erweiterte Auflage. Springer-Verlag, 2001.
- Zhender, Carl-August*. Informationssysteme und Datenbanken*. Sechste Auflage. B.G. Teubner Stuttgart;1998.
- Sherry Kinkoph, *E' facile Office 2000.* Jackson Libri. Mailand,1999.
- *Dizionario dei termini dell'informatica*, Mondadori Informatica, 1992.
- *Microsoft. Dizionario di Informatica.* Fünfte Auflage. Mondadori Informatica. Milano.2002.