

Studiengang: Informatik  
Prüfer/in: Prof. Mitschang  
Betreuer/in: Daniela Nicklas  
Sven Eckardt (IER)  
Beginn am: 02.01.2004  
Beendet am: 03.07.2004

CR-Nummer: H.2.2, J.1

# Universität Stuttgart

Entwicklung einer einheitlichen  
Datenstruktur und  
bedienungsfreundlichen  
Benutzerführung  
für ein EDV-Tool

Studienarbeit Nr.1932  
Gruppenarbeit  
Feng Ye&Yilian Shen

## Inhaltverzeichnis

1. Einleitung	4
1 Hinführung zum Thema	4
2 Grundlagen des Projekts und EDV-Tool	5
3 Aufgabestellung	6
4 Überblick über die Struktur der Arbeit	7
2. Problemstellung	9
2.1 Kernanforderungen	9
2 Grundlagen des EDV-Tools	9
3 Programmtechnische Anforderungen	10
4 Datenstruktur	13
3. Lösungsansatz	16
1 Grundlagen des Datenbankentwurfs	16
3.1.1 Datenbankentwurfsschritte	16
3.1.2 Entity-Relationship-Modell	19
Grundlage	19
Schlüssel	19
Funktionalitäten der Beziehungen	19
Ein Beispiel	20
2 Alternative von Datenbankmanagementsystemen	23
1 Access	23
2 MySQL	23
3 Webbasierte Datenbanken	26
PHP	27
MS SQL Server 2000	30
3 Lösung	31
1 Access wird hier benutzt.	31
2 Darstellung der Oberfläche	31
4. Entwurf	34
1 Kennzahl	34
4.1.1 Hinführung zum Thema	34
4.1.2 Darstellung der Arbeitsschritte und der angewandten Methoden	36
Konzeptionsphase	36
Entwicklungsphase	37
Datenstruktur	37
Implementierungsphase	41
4.1.3 Algorithmen zur Einheitsumrechnung	41
Standardeinheit	41
4 Kennzahlberechnung	42
Übernachtungsabhängige Kennzahlen	43
Selbstdefinierte Kennzahlen	44
5 Grafikerstellung	44
4.1.6 Microsoft Graph	46
Arbeitsschritt zur Diagrammerzeugung	
7 Andere Wege zur Kennzahlberechnung	49
Umweltmanagementsystem EMAS/ISO	49

Systematische Analyse	50
4.1.8 Zusammenfassung	
5 Evaluation	69
6 Ausblick	70
7 Webzugriff	71
8 Datenbankunabhängigkeit	74
Literaturverzeichnis	75
Anhang	77

Tabelle, Formulare und Abbildungen		
Abb. 2-4-1	Der Prozess im Mittelpunkt	15
Abb. 2-4-2	Das Entity-Relationship-Diagramm vom dem Prozess	15
Abb. 3-1	Die Phasen des Datenbankentwurfs	18
Abb. 3-1-2	ER-Diagramm für Mitarbeiter-Abteilung-Projekt-Beispiel	20
Abb. 3-2	Beispiel von einem PHP-Programm	27
Formular 3-3-1	Startfenster von E-KuH	31
Formular 3-3-2	Startfenster von Wirtschaftlichkeitsrechnung	32
Formular 3-3-3	Unterformular von Wirtschaftlichkeitsrechnung	33
Tabelle 4-1-1	Umweltkennzahlen einer Umwelterklärung	35
Formular4-1-1	Eingabeformular für Energieverbrauch	40
Tabelle 4-1-2	Tabelle für Einheitstyp Definition	41
Tabelle4-1-3	Tabelle für Einheitsdefinition	42
Formular 4-1-2	Formular für Diagrammtyp unter MS Graph	46
Formular 4-1-3	Diagramm-Assistent für MS Graph	48
Diagramm 4-1-1	Grafikdarstellung für Gesamtenergieverbrauch	49
Abb. 4-2-1	Rechengangmodell bei Wirtschaftlichkeitsrechnungen	52
Abb. 4-2-2	Rechenvorgang für Wassersparmaßnahmen	53
Abb. 4-2-3	Rechenvorgang für Heizungs- und Warmwasseranlagen	54
Abb.4-2-4	Struktur der Beziehungen von Wirtschaftlichkeitsrechnungen im Prototyp des EDV-Tool (MS Access)	60
Abb.4-2-5	Formular frm_Maßnahmen_Umweltbereich	64
Abb.4-2-6	Formular frm_Wirtschaftlichkeitsrechnung1	65
Abb.4-2-7	Formualr frm_Rechengang	66
Abb. 7-1	Startfenster der Intenetzugang	72
Abb.7-2	Vorstellungsformular von E-KuH	73

# 1. Einleitung

## 1 Hinführung zum Thema

Mehr und mehr Unternehmen in den Dienstleistungsbranchen, insbesondere im Tourismus (Reiseveranstalter, Hotellerie, Tourismuskommunen und Busreiseveranstalter), der nicht zuletzt durch seine rasante wirtschaftliche Entwicklung in den letzten zwei Dekaden für eine Vielzahl negativer Umweltauswirkungen verantwortlich ist, sehen Umweltschutz als Peripherie des unternehmerischen Handelns an. Diese Unternehmen möchten, dass die bisherigen Arbeitsergebnisse integriert werden können und der betrieblichen Umweltschutz durch eine geschlossene Systematik transparenter und effektiver gemacht werden kann. Ein geeignetes Instrument hierfür ist z.B. das EG-Umwelt-Audit-System.

Die deutschen Hotel- und Gaststättenbetriebe sind ein zentraler Leistungsträger des Tourismus. Allein in den rund 220.000 Unternehmen Deutschlands sind fast 1,2 Millionen Personen beschäftigt. Und das zunehmende Umweltbewusstsein der Gäste wird mittelfristig dazu führen, dass nicht mehr alleine die Gäste-, Betten- oder Umsatzzahlen entscheidend sind, sondern auch umweltrelevante Aspekte wie Abfallmanagement, Energieverbrauch oder Naturschutz. Die Handlungsleitlinien aus der Umwelterklärung der deutschen Tourismuswirtschaft sind:

- ◆ Umweltvorsorge und Stärkung der Eigenverantwortung
- ◆ Einrichtung von Umweltmanagementsystemen
- ◆ Umweltschonendes Beschaffungswesen
- ◆ Sparsamer Ressourcenverbrauch und Vermeidung von übermäßigem Flächenverbrauch
- ◆ Umweltgerechte Entsorgung
- ◆ Umweltbezogene Gästeinformation
- ◆ Aus- und Weiterbildung der Mitarbeiter in Umweltfragen
- ◆ Systematische Erfassung und Bewertung von Umweltauswirkungen
- ◆ Regelmäßige Berichterstattung

Investitionen in den Umweltschutz müssen als Investitionen in die Zukunft begriffen werden, denn eine intakte Natur und Umwelt ist für viele Hotels- und Gaststättenbetriebe der wohl bedeutsamste Standortfaktor. Eine umweltorientierte Wirtschaftsweise rechnet sich in der Regel auch ökonomisch. Sie weist die folgenden Vorteile auf:

- ◆ Einsparpotentiale

Einsparmöglichkeiten werden erkannt, z.B. im Bereich Energie. Dies kommt nicht nur der Umwelt zugute, sondern führt auch zur Entlastung des Betriebsbudgets.

- ◆ Öffentlichkeitswirkung

Durch aktive Information und Werbung mit dem EG-Umwelt-Audit läßt sich das Image des Betriebs langfristig steigern. Im Wettbewerb um Gäste stellen herausragende Leistungen im Umweltschutz einen werbewirksamen Faktor dar. Die Kunden festigen ihre Stellung gegenüber Verbänden, Behörden, Versicherungen und Banken.

- ◆ Informationsvorsprung

Der Betrieb ist künftig fortlaufend sowohl über aktuelle als auch langfristige Trends im Umweltbereich informiert.

- ◆ Kundenzufriedenheit

- ◆ Mitarbeitermotivation

## **1.2 Grundlagen des Projekts und EDV-Tools**

Im Rahmen des Projektes E-KUH (Entwicklung eines EDV-Tools für kostengünstige Umweltanalysen in Hotels und Gaststätten) soll ein Computerprogramm zur Unterstützung der umweltorientierten Betriebsführung im Gastgewerbe entwickelt werden. /2/

### **Umweltmanagement in Hotels**

An diesem Projekt nehmen vier „umweltorientierte“ Hotelbetriebe in Baden-Württemberg teil: Hotel Victoria in Freiburg, Seehotel Wiesler am Titisee, Erfurth's Vital-Hotel Bergfried in Hinterzarten und Hotel Rose auf der Schwäbischen Alb. Außerdem beteiligen sich auch ein Catering-Service (Herby's Party Service in Kraichtal) und die Mainau GmbH mit ihrem Restaurant ‚Schwedenschenke‘, die für die Umweltanalyse von Großküchen

eine nützliche Unterstützung sind. Bei diesen sechs Projektpartnern werden Begehungen in Zusammenarbeit mit DOS durchgeführt, um die relevanten Umweltaspekte und deren Auswirkungen zu erkennen. Ferner haben die Hotelbetriebe im Rahmen des Projektes die Möglichkeit, Informationen, Ideen und Erfahrungen auszutauschen. Sie können sich über den Stand der Technik in den verschiedenen Bereichen informieren sowie Adressen von Ansprechpartnern und Beratern austauschen. Mit den Hoteliers kann über die Vorbereitung des Einsatzes eines Umweltmanagementsystems im Hotel diskutiert werden. /3/

Wichtig für einen Vergleich zwischen Hotels ist die Bildung von Kennzahlen. Diese erfolgt mit der Sammlung von Informationen über Energieverbräuche für das Heizungssystem, Warmwasser- und Mahlzeitenvorbereitung und Abfallmengen (eingeteilt in Restmüll, Biomüll, Papier und Kartonagen, Sondermüll und Abwasser) in den Hotels. /3/

Am Ende des Projektes wird bei den Partnern die Installation und Erprobung des EDV-Tools getestet.

### **3 Aufgabenstellung**

Ziel des Projektes war die Entwicklung eines EDV-Tools für kostengünstige Umweltanalysen in Hotels und Gaststätten, da das Gastgewerbe einer der Dienstleistungssektoren mit den höchsten Energie- und Umweltbelastungen ist. Das EDV-Tool soll es Hotel- und Gaststättenbetrieben ermöglichen, Umweltbelastungen zu erkennen und zu beseitigen, Kosteneinsparpotenziale zu realisieren, und es soll den Betrieben umweltrelevante Informationen (z.B. Ansprechpartner, Adressen) bereitstellen.

Das EDV-Tool soll außerdem interessierten Hotels und Gaststätten den Einstieg in Umweltwettbewerbe ermöglichen bzw. ihnen aufzeigen, ob der Erwerb einer Lizenz von Viabono, der Dachmarke für umweltfreundlichen Tourismus in Deutschland, möglich ist. Nicht zuletzt soll das EDV-Tool auch zum Aufbau eines Umweltmanagement nach EMAS oder ISO 14001 dienen. Die Durchführung der jährlichen Umweltanalysen wird durch das EDV-gestützte Verfahren systematisiert und der erforderliche Aufwand dadurch minimiert.

#### 4 Überblick über die Struktur der Arbeit

Diese Studienarbeit befasst sich mit dem Projekt E-Kuh (Entwicklung eines EDV-Tools für kostengünstige Umweltanalysen in Hotels und Gaststätten), das von der *Deutscher Bundesstiftung Umwelt* unterstützt wird. Ziel des Projektes ist es, ein EDV-gestützte Verfahren zu entwickeln, welches einem ermöglicht, Umweltbelastungen zu erkennen und zu beseitigen, eine Umweltanalyse in Hotels jährlich zu wiederholen, aber mit geringem Aufwand. Alle Modelle werden in *Microsoft Access und VB (Visual Basic) Application* programmiert.

Zusammenfassend werden im ersten Kapitel das Projekt E-KuH vorgestellt. In Kapitel 2 wird näher auf das Projekt E-Kuh eingegangen, wobei die Schwerpunkte auf den Kernanforderungen, der Grundstruktur des EDV-Tools, den programmtechnischen Anforderungen bzw. der Datenstruktur von E-Kuh liegen. In Kapitel 3 werden die grundlegenden Techniken über Datenbanken vorgestellt, die in E-Kuh genutzt werden, nämlich die Datenbankentwurfsschritte, die Typen der *Datenbankmanagementsysteme* und die Konzepte für das *Entity-Relationship-Modell*. Im Kapitel 4, dem Hauptteil dieser Arbeit, werden separat die Kennzahlberechnung und die Wirtschaftlichkeitsrechnung behandelt. Das Kapitel 5 ist die Evaluierung des Projekts, meiste sind direkt von den Kunden, die dieses Tool getestet haben. Im Kapitel 6 werfen wir einen Blick im Webzugriff und die Datenunabhängigkeit.

Die ersten 3 Kapiteln, also der Theorieanteil haben wir zusammengearbeitet. Und im Kapitel 4 hat Feng Ye die Kennzahlberechnung und Yilian Shen die Wirtschaftlichkeitsrechnung behandelt. Und das Kapitel 5 und 6 haben wir wieder zusammen gearbeitet. Feng Ye's Thema — Kennzahlberechnung handelt es um die Kennzahlen. Umweltkennzahlen werden aus den In- und Outputkonten abgeleitet und liefern Belege für den kontinuierlichen Verbesserungsprozess im Umweltbereich eines Unternehmens. Deshalb sind die Kennzahlen sehr wichtig und sinnvoll für ein Unternehmen. Yilian Shen's Thema ist die Wirtschaftlichkeitsberechnung in E-Kuh. Mit Hilfe dieses Modells der Wirtschaftlichkeitsberechnung kann E-Kuh sowohl die technisch komplexen Anlagen berechnen als auch dem Benutzer die einfacheren Investitionsmaßnahmen empfehlen. Der Schwerpunkt in der Wirtschaftlichkeitsrechnung liegt in der Implementierung der

Berechnung, zum Beispiel von den Kosten. Dieser Algorithmus liegt in der Definition der Eingabe- und Ausgabendaten und ihren funktionalen Zusammenhängen. Ein ausführliches Beispiel wird im dritten Kapitel eingeführt. Außerdem werden alle Tabellen, Abfragen, Codes sowie Formulare in der Arbeit erläutert, um die Konzepte und das Verfahren der Wirtschaftlichkeitsberechnung deutlich zu verdeutlichen.

## **2. Problemstellung**

### **2.1 Kernanforderungen**

Ziel des Projektes ist es, ein praxisnahes Tool für die umweltorientierte Betriebsführung im Gastgewerbe zu entwickeln. Daraus ergeben sich folgende zentralen Anforderungen:

1. Benutzerfreundlichkeit,
2. Inhaltliche Konsistenz,
3. Flexibilität.

Diese werden im Folgenden konkretisiert. Im Anschluss daran wird das sich daraus ergebende EDV-technische Grundkonzept dargestellt. /2/

### **2.2 Grundlagen des EDV-Tools**

Das zu entwickelnde Tool umfasst mehrere Teile:

1. Kennzahlenermittlung und -vergleich
2. Betriebsmittelerfassung
3. Analyse von Prozessen und Dienstleistungen
4. Analyse von Standardmaßnahmen für den Umweltwettbewerb
5. Umweltberichterstattung
6. Maßnahmenvorschläge und -analyse

Mit Hilfe des Tools kann dem Kunden folgende Fragen beantwortet:

1. Wie kann ich in meinem Betrieb Umweltbelastungen reduzieren?
2. Wie ist die Umweltleistung meines Betriebes im Vergleich zu anderen?
3. Treffen meine Umweltaktivitäten die Erwartungshaltungen meiner Gäste und Kunden?
4. Rechnen sich Investitionen in neue Anlagen und Prozesse?
5. Wie kann ich systematisch und kontinuierlich meine Umweltleistung verbessern?
6. Wo finde ich weiterführende Informationen?
7. Zum kontinuierlichen Arbeiten mit dem EDV – Tool?

Mit Maßnahmenauswahl, Kennzahlenermittlung, viabono-Kriterienkatalog, Wirtschaftlichkeitsrechnung, Umweltmanagementsystem EMAS/ISO, Adressen und Ansprechpartner, systematische Analyse können die obigen Fragen beantwortet werden..

Die vorliegende Studienarbeit befasst sich zentral mit der Kennzahlenermittlung und der

Wirtschaftlichkeitsrechnung, die weiter zu den Maßnahmenvorschlägen und -analysen führt.

Zweck der Kennzahlenermittlung ist es, auf der Basis von Grund- und Verbrauchsdaten hotelspezifische Kennzahlen zu erstellen und damit Bereiche zu identifizieren, in denen vordringlicher Handlungsbedarf besteht. Dazu werden entsprechend einer Checkliste unternehmensspezifische Daten ermittelt, in das EDV-Programm eingegeben und anschließend standardisiert ausgewertet. Man versucht allgemeingültige Kennzahlen zu bilden, die jährlich oder monatlich erfasst werden und mit denen man im EDV-Tool in den Berechnungen arbeiten kann.

Basierend auf den ermittelten Grunddaten und Kennzahlen sowie der Analyse von Prozessen und Dienstleistungen werden in dieser Arbeit Maßnahmenvorschläge ermittelt. Die im Rahmen der Schwachstellenanalyse identifizierten Verbesserungsmaßnahmen werden hinsichtlich ihrer Umweltauswirkungen und Kosten quantifiziert und einem Maßnahmenvergleich unterzogen. Die Ergebnisse bieten die Möglichkeit, eine Auswahl der Maßnahmen z.B. nach dem Kosten-Nutzen-Verhältnis oder nach der Höhe der Investition zu treffen. Durch die Gegenüberstellung und den Vergleich der verschiedenen Maßnahmen soll ein Maßnahmenpaket identifiziert werden können, das die unternehmerische Entscheidung, unter Berücksichtigung möglicher Entwicklungen hinsichtlich der Umsetzung der Maßnahmen, erleichtert und einen wesentlichen Teil der Umweltberichterstattung darstellt. /2/

### **2.3 Programmtechnische Anforderungen**

Hier sind wesentliche Anforderungen für die programmtechnische Umsetzung aufgeführt:

1. *Datenbankprogramm*: Im Prinzip ist jede kommerziell verfügbare Software für relationale Datenbanken unter den Betriebssystemen Windows 95/98/ME, Windows NT, Windows 2000 und XP geeignet. Ein Einsatz unter LINUX ist wünschenswert, aber nicht Voraussetzung, da dieses Betriebssystem nach wie vor eine geringere Verbreitung aufweist. Wesentlich für die Auswahl des Datenbankprogramms ist, dass die Kopplung mit Benutzeroberflächen möglichst einfach ist und dass das Programm weit verbreitet ist. Daher wird die Nutzung von *Microsoft Access* empfohlen.

2. *Programmierungsumgebung für die Benutzeroberfläche:* Diese muss mit geringem Aufwand die Gestaltung ansprechender Benutzeroberflächen ermöglichen. Die Benutzeroberflächen sollten soweit als möglich das Look-and-Feel anderer Windows-Programme, insbesondere der Microsoft Office Produkte aufweisen. Dies legt nahe, die Benutzeroberflächen direkt unter *Microsoft Access* unter (Mit-)Verwendung von *Visual Basic for Applications* (VBA) zu implementieren.

3. *Programmierungsumgebung für Algorithmen:* Diese muss problemangepasst ausgewählt werden. Auswahloperationen sind, nach Möglichkeit, als Datenbankabfragen zu implementieren, Rechenoperationen können wahlweise unter *Microsoft Excel* oder unter *VBA* implementiert werden.

4. *Multi-User-Fähigkeit:* Diese ist zwar prinzipiell für größere Hotels wünschenswert, wird aber in der Grundversion nicht gefordert, da dies den Umsetzungsaufwand erheblich erhöht und die Zielgruppe zunächst eine große Anzahl mittelständischer Hotels ist.

5. *Internetanbindung:* Diese ist zumindest erforderlich für die Verlinkung von Informationsquellen. Daneben ist wünschenswert, den Kennzahlenvergleich mit über Internet upgedateten Vergleichswerten zu ermöglichen. /2/

6. *Log-Einträge:* Eine Protokollierung der wesentlichen vom Nutzer durchgeführten Operationen ist erforderlich, um dem Anwender bei einer neuen Sitzung die Möglichkeit zu geben, die Arbeiten dort fortzuführen, wo er sie beim letzten Mal beendet hat. Auch erleichtert ein solches Log-Buch insbesondere in der Anfangsphase der Tool-Nutzung die Identifikation möglicher Fehlerursachen. Außerdem ermöglicht eine solche Protokollierung den konsistenten Abgleich mit externen Datenquellen. /2/

## **VB for Application (VBA)**

### **Bitte nicht verwechseln: Basic/Visual Basic/VBA:**

**Basic** (<eng.>Abkürzung für the Beginner's All-purpose Symbolic Instruction Code = Allzweckprogrammiersprache für Anfänger) ist eine alte (1964 entwickelte), einfache Interpretersprache ohne Rekursionen etc. für den Heimcomputer unter MS-DOS. Basic ist »out«.

**Visual Basic** ist eine zwar auf Basic beruhende, objektorientierte Programmiersprache

von Microsoft, die aber inzwischen in keiner Weise mehr mit Basic vergleichbar ist. Sie bietet mächtige Werkzeuge, um mit hoher Geschwindigkeit professionelle Windowsprogramme zu erstellen.

**VBA** (Visual Basic for Application) ist eine eigenständige, objektorientierte Programmiersprache für die MS Office-Anwendungen. Sie dient der Erweiterung des Funktionsumfangs der MS-Office-Produkte und ist in diese integriert. Ursprünglich hatten die einzelnen Office-Anwendungen eigene, unterschiedliche Makro-Sprachen. Um eine einheitliche Programmierung im Office-Paket zu erreichen, wurde eine einheitliche Programmiersprache für alle Anwendungen entwickelt: VBA.

#### **Arbeitserleichterung durch Makros**

Das Office-Paket enthält Anwendungen, mit denen man Aufgaben der Textverarbeitung, Tabellenkalkulation, Datenverarbeitung, Präsentation und Planung erledigen kann. Alle Anwendungen können ohne Programmierkenntnisse eingesetzt werden

Im Praxiseinsatz der Office-Anwendungen treten häufig immer gleiche Handlungen auf, die zeitsparend automatisch ausgeführt werden könnten. Zu diesem Zweck wurden so genannte Makros eingeführt, die eine Reihe von Tastaturanschlägen oder Mausbedienungen zu einer Einheit zusammenfassen konnten. Makros waren jedoch keine Entwicklungswerkzeuge für Programmierer, sondern nur Arbeitserleichterungen für die Anwender. Im Laufe der Zeit wurde die Makro-Sprache der Office-Anwendungen erweitert und immer weiter zu einer strukturierten Programmiersprache entwickelt.

#### **Programmierung mit VBA**

Die umfangreichen Makro-Sprachen der einzelnen Office-Anwendungen (Excel-Makros, Word-Basic, Access-Basic) erlaubten ursprünglich keine einheitliche Programmierung im Office-Paket. Aus diesem Grund wurde eine einheitliche Programmiersprache entwickelt und in alle Anwendungen integriert: Visual Basic for Applications (VBA).

VBA ist eine objektorientierte, vollwertige Programmiersprache, die sich an Visual Basic anlehnt. Sie verfügt über umfassende Objektmodelle für alle Office-Anwendungen und

erlaubt eine strukturierte und modulatorientierte Programmierung. Auch wenn VBA speziell auf die Eigenschaften und Anforderungen für den Einsatz innerhalb einer Office-Anwendung ausgerichtet ist, verfügt es über viele Fähigkeiten und Merkmale einer allgemeinen Programmiersprache.

Um die Programmierung zu erleichtern, verfügt das Office-Paket über eine VBA-Entwicklungsumgebung, die als eigenständige Anwendung ausgeführt wird. Die Entwicklungsumgebung bietet eine übersichtliche Arbeitsumgebung zum Programmieren, die darüber hinaus noch einige Programmierhilfen und Arbeitserleichterungen zur Verfügung stellt.

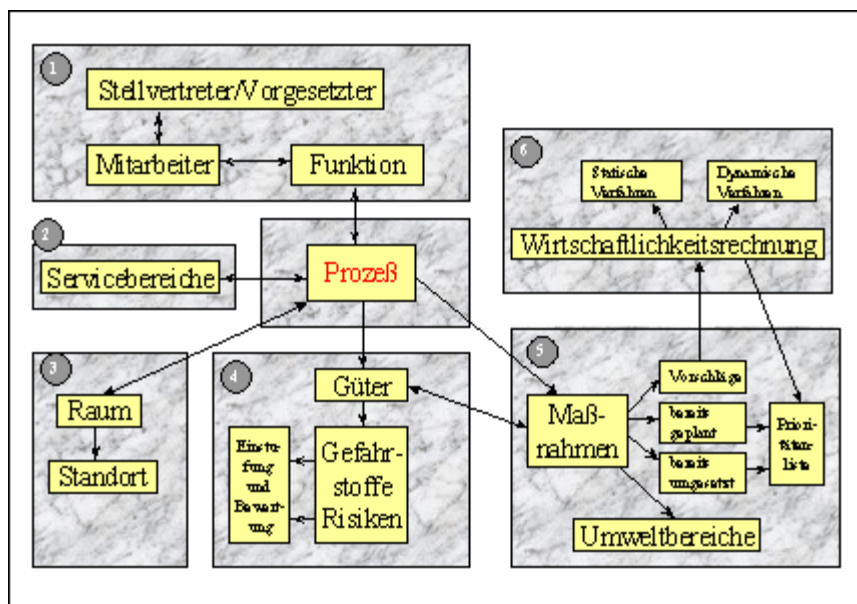
Makros, die möglicherweise für ältere Office-Pakete erstellt wurden, sind auch im aktuellen Office 2000 mit einigen Änderungen lauffähig. So verfügt beispielsweise Word über das Wordbasic-Objekt, das alle alten Makro-Befehle aus der Programmierung von Word 6.0 zur Verfügung stellt.

## **2.4 Datenstruktur**

Alle Prozesse stehen in Beziehung zu „Entitäten“, d.h. Objektkategorien, die für die Umweltanalyse und das Umweltmanagement bedeutsam sind. Solche Entitäten sind beispielsweise die Prozessverantwortlichen, hier in der Gruppe „Wirtschaftlichkeitsrechnung“ zusammengefasst, oder die Gruppe „Servicebereiche“. Ein Umweltmanagementsystem kann nur dann funktionieren und aufrechterhalten werden, wenn diese Bereiche mit den Prozessen direkt verknüpft sind. Prozesse sind z.B. „Zimmerwäsche waschen“ oder „heizen“.

Nach dem Prinzip des Qualitäts- und Umweltmanagements soll für jeden Prozess immer ein Verantwortlicher festgelegt werden. Dieser wird nicht namentlich, sondern über die Funktion, die er innerhalb seiner Arbeit hat, bezeichnet (Gruppe 1). Die Prozesse müssen bei der Analyse entweder einem Servicebereich oder einem bestimmten Raum zugeordnet werden (Gruppe 2 und 3). In einem Hotelbetrieb ist es auch wichtig, dass die Güter und Ressourcen, die in den verschiedenen Prozessen verbraucht werden, erfasst und kontrolliert werden, vor allem wenn es sich um Gefahrstoffe handelt (Gruppe 4.).

Maßnahmen (Gruppe 5) setzen ebenfalls an Prozessen bzw. deren Umgestaltung an. Die Maßnahmen, die berücksichtigt werden, sind in die vier Bereiche Energie, Wasser/Abwasser, Abfall und Sonstiges unterteilt. In der Struktur des EDV-Tools sind die Maßnahmen in „bereits umgesetzte“, „bereits geplante“ und als „Vorschlag“ aufgeteilt. Die Vorschläge sollen durch die Analyse des Betriebes herausgearbeitet werden, unter ökonomischen Gesichtspunkten (mit statischen oder dynamischen Verfahren) analysiert (vgl. Gruppe 6) und zuletzt im Umweltprogramm mit einer Prioritätenliste aufgelistet werden. /2/



**Abb. 2-4-1: Der Prozess im Mittelpunkt**

Die wesentlichen Verknüpfungen zwischen den Tabellen, die dem Prozess grunde liegen, werden nachfolgend durch ein Entity-Relationship-Diagramm dargestellt.



### 3. Lösungsansatz

#### 3.1 Grundlagen des Datenbankentwurfs

##### 3.1.1 Datenbankentwurfsschritte

Man nutzt die „top-down“-Entwurfsmethodik, die nun auf den Lebenszyklus eines Datenbankentwurfs zugeschnitten wird. Der Datenbankentwurf orientiert sich an den Abstraktionsebenen einer Datenbankanwendung.

Wie in jeder systematischen Entwurfsmethodik beginnt man auch im Datenbankentwurf mit der *Anforderungsanalyse*, die man auch *Anforderungsspezifikation* oder *Pflichtenheft* nennt. In der Anforderungsanalyse müssen zum einen die Informationsanforderungen der zu modellierenden Welt (bzw. des relevanten Ausschnitts der realen Welt - auch Miniwelt genannt) und zum anderen die Datenverarbeitungsvorgänge berücksichtigt werden. Eine sorgfältig ausgeführte Anforderungsanalyse, die in enger Zusammenarbeit mit dem projektierten Anwender des Systems ausgeführt wird, ist die Grundvoraussetzung für die spätere Akzeptanz der Datenbankanwendung.

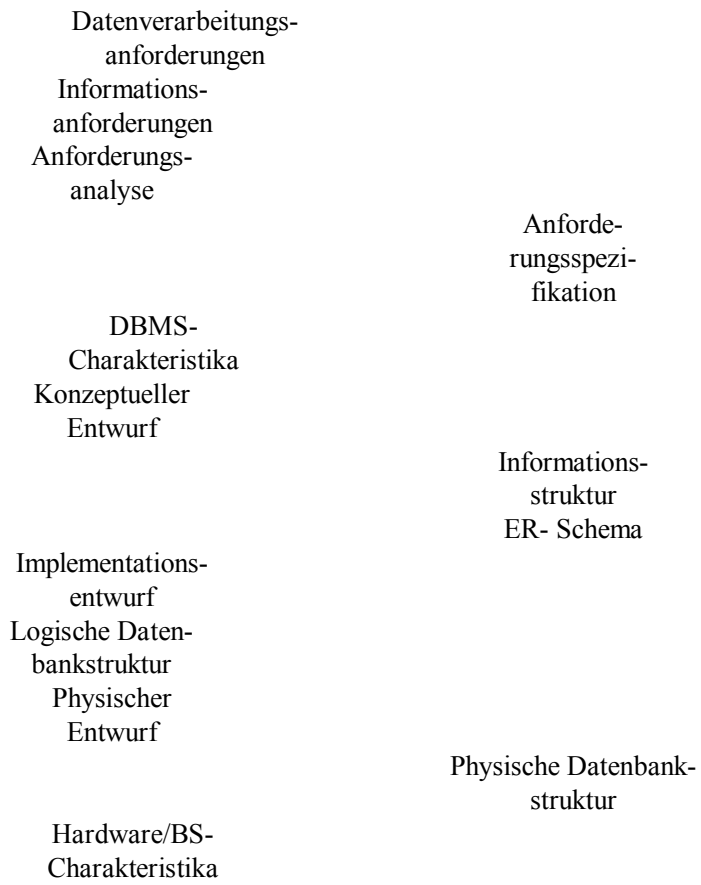
Anschließend – nach Fertigstellung der Anforderungsspezifikation – erfolgt der *konzeptuelle Entwurf*. In diesem Entwurfsschritt wird die Informationsstruktur auf einer konzeptuellen, d.h. anwenderorientierten Ebene festgelegt. Das am häufigsten für den konzeptuellen Entwurf verwendete Datenmodell ist das Entity-Relationship-Modell.

Als Ausgabe des konzeptuellen Entwurfs erhält man dann die Informationsstrukturbeschreibung in der Form eines Entity-Relationship-Schemas (kurz ER-Schema). Es ist wichtig zu betonen, dass dieser Entwurfsschritt noch gänzlich unabhängig vom eingesetzten Datenbanksystem durchgeführt wird. Das Datenmodell des eingesetzten Database Management Systems (DBMS) kommt erst im *Implementationsentwurf* zum Tragen. Dabei wird das ER-Schema in ein entsprechendes Implementationsschema - oft auch logische Datenbankstruktur genannt - überführt. Beim Implementationsentwurf müssen aber auch die Datenverarbeitungsanforderungen berücksichtigt werden, um ein geeignetes Datenbankschema erstellen zu können.

Der letzte Schritt des Datenbankentwurfs, der *physische Entwurf*, verfolgt das Ziel der Effizienzsteigerung, ohne dabei die logische Struktur der Daten zu verändern. Für den physischen Entwurf ist eine detaillierte Kenntnis des zugrunde liegenden Datenbanksystems, aber auch der Hard- und Software (z.B. des Betriebssystems), auf der das DBMS installiert ist, notwendig.

Die Abfolge und der Zusammenhang dieser Entwurfsschritte sind graphisch in Abbildung 3-1 gezeigt.

Schließlich gibt es noch den Einsatz des Systems, das durch die Überwachung (engl.: Monitoring) des laufenden Systems entsteht, um daraus Rückschlüsse auf eventuell notwendige Änderungen (Adaptionen) ziehen zu können. /1/



Quelle: /1/

**Abb. 3-1: Die Phasen des Datenbankentwurfs**

## 2 Entity-Relationship-Modell

### Grundlage

Die grundlegendsten Modellierungsstrukturen sind die Entities (Gegenstände) und die Relationships (Beziehungen) zwischen den Entities. Zusätzlich kennt das Entity-Relationship-Modell noch Attribute und Rollen.

Gegenstände (bzw. Entities) sind wohlunterscheidbare physisch oder gedanklich existierende Konzepte der zu modellierenden Welt. Man abstrahiert ähnliche Gegenstände zu Gegenstandstypen (Entitytypen oder Entitymenge), die man graphisch als Rechtecke darstellt, wobei der Name des Entitytyps innerhalb des Rechtecks angegeben wird.

Beziehungen werden auf analoge Weise zu Beziehungstypen zwischen den Gegenstandstypen abstrahiert. Die Beziehungstypen werden als Rauten mit entsprechender Beschriftung repräsentiert. Die Rauten werden mit den beteiligten Gegenstandstypen über ungerichtete Kanten verbunden.

### Schlüssel

Eine minimale Menge von Attributen, deren Werte die zugeordnete Entity eindeutig innerhalb aller Entities seines Typs identifiziert, nennt man Schlüssel. Sehr oft gibt es einzelne Attribute, die als Schlüssel „künstlich“ eingebaut werden, wie z.B. Personalnummer (PersNr) etc. Schlüsselattribute werden durch Unterstreichung gekennzeichnet.

Manchmal gibt es auch zwei unterschiedliche Schlüsselkandidaten, dann wählt man einen dieser Kandidaten-Schlüssel als Primärschlüssel aus.

### Funktionalitäten der Beziehungen

Man kann Beziehungstypen hinsichtlich ihrer Funktionalität charakterisieren. Ein binärer Beziehungstyp R zwischen den Entitytypen E1 und E2 heißt:

- ◆ 1 : 1 –Beziehung, falls jedem Entity e1 aus E1 höchstens ein Entity e2 aus E2 zugeordnet ist und umgekehrt jedem Entity e2 aus E2 maximal ein Entity e1 aus E1 zugeordnet ist.

- ◆ 1 : N –Beziehung, falls jedem Entity e1 aus E1 beliebig viele (mehrere oder gar keine) Entities aus E2 zugeordnet sein können, aber jedes Entity e2 aus E2 mit maximal einem Entity aus E1 in Beziehung stehen kann.
- ◆ N : 1 –Beziehung, falls Analoges zu Obigem gilt.
- ◆ N : M –Beziehung, wenn keinerlei Restriktionen gelten müssen.

### Ein Beispiel

Demnach könnte ein spezieller Mitarbeiter (ein Entity aus der Entitymenge MITARBEITER) durch die Attribute [MID: 3932, NAME: Peter Müller, GJAHR: 1947] vollständig beschrieben sein.

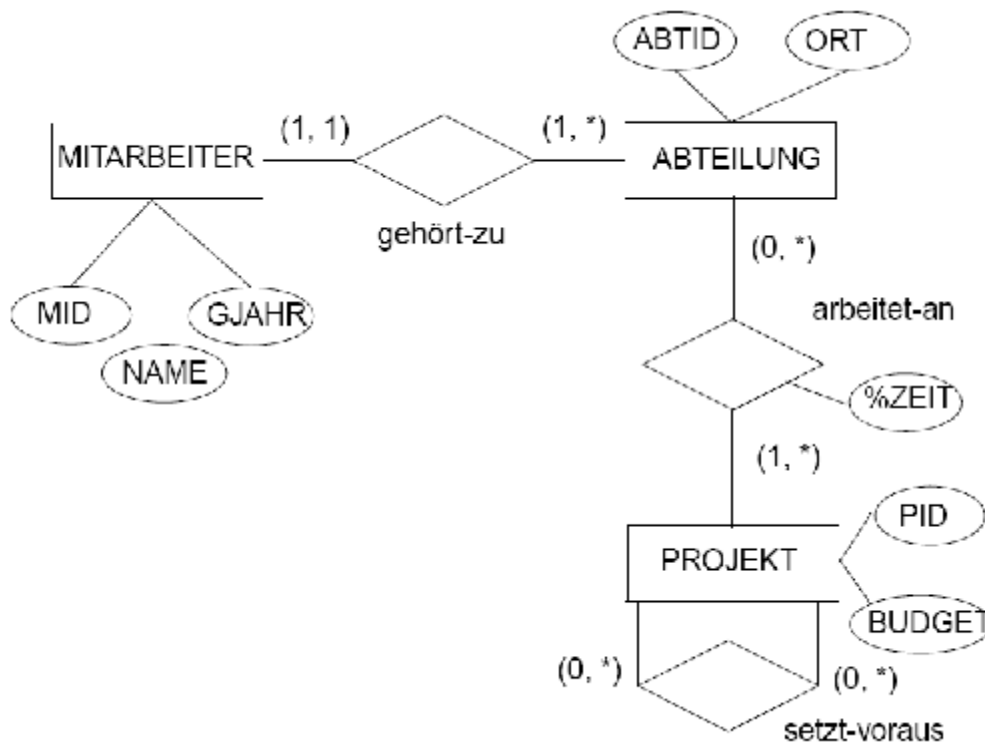


Abbildung 3.1.2: ER-Diagramm für Mitarbeiter-Abteilung-Projekt-Beispiel

Attribute werden durch Festlegung des *Wertebereichs* (engl. domain), also die Nennung des Attribut-Typs (hier z.B. numerisch), und der Attributbezeichner (hier MID, NAME, ... ) bestimmt. Existiert für ein Attribut gegebenenfalls kein Wert, kann stattdessen ein sog. Nullwert gespeichert werden (Details siehe unten).

Zwischen den Entitäten herrschen Beziehungen (engl. relationships), z.B. zwischen MITARBEITER und PROJEKT die Beziehung ARBEITET\_AN, bzw. die Umkehrung WIRD\_BEARBEITET\_VON. Dabei können zwischen zwei Entity- Typen auch mehrere Beziehungstypen existieren, z.B. zwischen Mitarbeitern und Projekten die Beziehungen ARBEITET\_AN und LEITET. Man spricht dann von mehreren *Rollen*, die ein Entity übernimmt.

Beziehungen können auch wieder Attribute zur Beschreibung allgemeiner Eigenschaften haben, z.B. könnte ARBEITET\_AN zwischen MITARBEITER und PROJEKT mit PROZENT\_ARBEITSZEIT, die Beziehung GEHÖRT\_ZU zwischen MITARBEITER und ABTEILUNG mit dem Attribut SEIT (vom Typ Datum) versehen sein.

Das ER-Modell hat eine anschauliche graphische Darstellung: Im einfachsten Fall werden Entity-Mengen durch Rechtecke, Attribute durch Ellipsen, Beziehungen durch Rauten dargestellt, die durch Linien verbunden sind. Sind, wie in Bild 3.1.2, zwei Entity-Mengen an einer Beziehung beteiligt, spricht man von binären Beziehungen (Grad 2, die weitaus gebräuchlichste Form). Daneben sind Beziehungen vom Grad 1, wie z.B. „setzt-voraus“ in Bild 3.1.2, und vom Grad 3 üblich.

An den Kanten können, wie in Bild 3.1.2, Teilnahmebedingungen und Kardinalitäten eines Beziehungstyps angegeben werden. Unter Teilnahmebedingungen (auch Mitgliedschaften genannt) versteht man die Forderung, dass jedes Entity an der Beziehung teilnehmen muss, d.h. dass die Beziehung total sein muss, bzw. die Erlaubnis, dass sie partiell sein darf (optionale Mitgliedschaft).

So besagt Bild 3-1-2 oben, dass nicht jede Abteilung ein Projekt, ggf. aber eine Abteilung

viele Projekte bearbeitet und dass jedes Projekt von mindestens einer Abteilung, ggf. vielen, bearbeitet wird. Die Mitgliedschaft von ABTEILUNG an der Beziehung ist optional, PROJEKT dagegen nimmt zwangsweise an der Beziehung teil.

Man nennt dies dann auch eine existentielle Abhängigkeit, da ein Projekt nicht ohne eine „bearbeitet-Beziehung“ existieren kann (entfernt man die Abteilung, sterben auch alle von ihr bearbeiteten Projekte).

Der eigentliche Zahlenwert, der angibt, an wievielen Beziehungen  $r \in R$  eine Entität  $e \in E$  teilnehmen kann, heißt Kardinalität und sollte mit Minimal- und Maximalwert angegeben werden, wobei \* für „beliebig viele“ steht (vgl. Bild 3.1.2).

1:1-, 1:n-, n:m-Beziehungen

Am gebräuchlichsten ist bei Zweierbeziehungen jedoch die recht ungenaue Angabe des Verhältnisses der Kardinalitäten, d.h. man klassifiziert sie als 1:1-, 1:n-, n:1-, n:m-Beziehungen, und zwar als

1:1 wenn die Kardinalitäten links und rechts (0, 1) oder (1, 1)

1:n ~ rechts (0, 1) oder (1, 1) und links (0, \*) oder (1, \*)

n:1 ~ rechts (0, \*) oder (1, \*) und links (0, 1) oder (1, 1)

n:m ~ links und rechts (0, \*) oder (1, \*) sind.

GEHÖRT\_ZU zwischen MITARBEITER und ABTEILUNG ist demnach eine n:1-Beziehung

(viele Mitarbeiter in einer Abteilung). In der englischsprachigen Literatur ist dann von one-to-one, one-to-many, many-to-many relationships die Rede.

Mitgliedsklassen und Kardinalitäten haben großen Einfluss auf den Systementwurf. Dazu gehört die Befragung der „Miniwelt-Experten“ zur Ermittlung der Entity-Typen, Beziehungen und Restriktionen (z.B. Kardinalitäten), die als Systemanalyse bezeichnet wird. Auf der Basis des (erweiterten) ER-Modells, das auf Chen zurückgeht, wird der Entwurf des konzeptuellen Schemas heute von vielen DB-Werkzeugen (z.B. ERwin, S-Designer, SELECT-SE) mit graphischen Editoren und automatisierten Übersetzungen in das relationale Modell unterstützt. Er bestimmt, wo Nullwerte erlaubt und wie Tabellen

aufzuteilen sind (die sog. Normalisierung).

## **3.2 Alternative von Datenbankmanagementsystemen**

### **3.2.1 Access**

Microsoft Access ist ein leistungsfähiges Programm zur Herstellung und Handhabung von Datenbanken. Es unterstützt die Benutzer, wenn es ihre Informationen konstruiert und ansieht.

Innerhalb von Access gibt es sieben Schwerpunkte: Tabellen, Abfragen, Formulare, Berichte, Seiten, Makros und Module, wodurch man eine neue Datenbank aufbauen kann.

Grundsätzlich hat Access folgende Vorteile:

1. einfache und benutzerfreundliche Möglichkeiten für Modellierung und Entwurf der Datenbank,
2. einfache und benutzerfreundliche Erstellung von Formularen und Reports. Und es benötigt keinen großen Verwaltungsaufwand.

Trotzdem ist Access nicht perfekt. Access ist normalerweise nur für Kleinbetrieb geeignet, denn es benötigt eine hohe Speicheranforderung. Weiterhin rät selbst Microsoft, bei größeren Benutzerzahlen (also auch bei Zugriffen aus dem Internet) nicht Access, sondern MS-SQL-Server einzusetzen. /4/ Denn die Geschwindigkeit ist bei vielen Datensätzen und vielen R/W-Zugriffen (simultan) sehr langsam, falls Access dort genutzt wird.

### **3.2.2 MySQL**

MySQL ist die populärste Open Source SQL-Datenbank. Open Source bedeutet, dass es für jeden möglich ist, solche Software zu benutzen und zu verändern. Jeder kann MySQL aus dem Internet herunterladen und benutzen, ohne irgend etwas zu bezahlen. Jeder, der daran interessiert ist, kann den Quelltext studieren und den eigenen Bedürfnissen entsprechend verändern. MySQL benutzt die GPL (GNU General Public License), um festzulegen, was man mit der Software tun darf und was man nicht tun darf, abhängig von unterschiedlichen Situationen. Wenn einen die GPL Probleme bereitet oder man Sie

MySQL in eine kommerzielle Anwendung einbetten muss, kann eine kommerziell lizenzierte Version von IER (Universität Stuttgart) erworben.

Sie wird von MySQL AB zur Verfügung gestellt. MySQL AB ist ein kommerzielles Unternehmen, dessen Geschäft darin besteht, Serviceleistungen rund um die MySQL-Datenbank zur Verfügung zu stellen.

MySQL ist ein relationales Datenbank-Managementsystem. Eine relationale Datenbank speichert Daten in separaten Tabellen, anstatt sie alle in einem einzigen großen Speicherraum unterzubringen. Hierdurch werden hohe Geschwindigkeit und Flexibilität erreicht. Die Tabellen werden durch definierte Beziehungen verbunden (Relationen), was es möglich macht, Daten aus verschiedenen Tabellen auf Nachfrage zu kombinieren. Der SQL-Teil von MySQL steht für "Structured Query Language" (strukturierte Abfragesprache) - die verbreitetste standardisierte Sprache für Datenbankzugriffe.

Die Entstehung des Namens MySQL ist nicht völlig geklärt. Das Basis-Verzeichnis und eine große Anzahl der Bibliotheken und Werkzeuge hatten immer schon das Präfix „my“ während mehr als 10 Jahren.

MySQL ist sehr schnell, zuverlässig und leicht zu benutzen. MySQL besitzt eine ganze Reihe praktischer Features, die in enger Kooperation mit unseren Benutzern entwickelt wurden.

MySQL wurde ursprünglich entwickelt, um sehr große Datenbanken handhaben zu können, und zwar sehr viel schneller als existierende Lösungen. Es wurde mehrere Jahre in höchst anspruchsvollen Produktionsumgebungen eingesetzt. Heutzutage bietet MySQL eine umfangreiche Reihe sehr nützlicher Funktionen. Vernetzung, Geschwindigkeit und Sicherheit machen MySQL äußerst geeignet, um auf Datenbanken über das Internet zuzugreifen.

MySQL ist ein Client-Server-System, das aus einem multi-thread SQL-Server besteht, der unterschiedliche Backends, verschiedene Client-Programme und -Bibliotheken,

Verwaltungswerkzeuge und etliche Programmschnittstellen unterstützt. Man stellt MySQL auch als multi-thread Bibliothek zur Verfügung, die man mit seinen Anwendungen verknüpfen kann, um ein kleineres, schnelleres und leichter zu bedienendes Produkt zu erhalten.

MySQL stellt beigesteuerte (contributed) Software in großer Menge zur Verfügung. Es ist sehr wahrscheinlich, dass die Lieblingsanwendung oder -sprache bereits MySQL unterstützt.

Offiziell wird MySQL 'Mai Ess Ku Ell' ausgesprochen (nicht 'Mai Siekwel').

### **Oracle**

Oracle ist eine Software-Firma. Diese in 1977 gegründete Firma in Kalifornien ist die erste in der Welt, die Relationale Datenbankmanagement System(RDBMS) herausgebracht hat. Jetzt wird ihr RDBMS zunehmend in anderen Betriebsumgebungen angewendet: Windows NT, kleine auf UNIX System basierende Maschinen, große Maschinen von IBM usw. Oracle ist jetzt der größte RDBMS- Lieferant in der Welt geworden. Jetzt ist Oracle die zweitgrößte Softwarefirma in der Welt direkt nach Microsoft.

Die Oracle-Technologie ist weltweit führend und in 98 der Fortune 100-Unternehmen vertreten. Oracle ist die erste Software-Firma, die zu 100 Prozent im Internet f r e i g e g e b e n e . Unternehmenssoftware entwickelte und quer durch ihre gesamte Produktlinie einsetzte: Datenbanken, kommerziellen Anwendungen, Anwendungsentwicklungen und entscheidungsunterstützende Tools. Oracle ist der weltweit führende Software-Lieferant für Informationsmanagement und das weltweit zweitgrößte unabhängige S o f t w a r e - U n t e r n e h m e n .

Oracle war schon immer eine innovative Firma. Sie war eine der ersten Firmen, die ihre kommerziellen Anwendungen im Internet verfügbar machten - ein Konzept, das heutzutage überall anzutreffen ist. Jetzt setzt sich Oracle dafür ein, dass ihre gesamte Software modular untereinander kompatibel konzipiert wird und andere Unternehmen,

Analysten und die Presse fangen an zu begreifen, dass Oracle den richtigen Weg eingeschlagen hat. Auch in Zukunft wird Oracle innovativ sein und die Industrie anführen, wobei stets die Lösung von Problemen ihrer Software-Anwender im Vordergrund stehen wird.

### 3.2.3 Webbasierte Datenbanken

Webbasierte Datenbanken sind der Trend zukünftiger Datenbanken. Deshalb sollte man über PHP, die besondere Fähigkeiten durch ihre flexible Datenbankanbindung (z.B. Access, MySQL u.a.) etwas informiert werden.

### PHP

PHP (Akronym für "PHP: Hypertext Preprocessor") ist eine weit verbreitete und für den allgemeinen Gebrauch bestimmte Open Source Skriptsprache, welche speziell für die Webprogrammierung geeignet ist, und in HTML eingebettet werden kann.

Ein Beispiel für ein PHP Programm:

```
<          h          t          m          l          >
                                     <head>
                                     <title>Beispiel</title>
                                     </head>
                                     <body>
                                     <?php
                                     echo  "Hallo,  ich  bin  ein  PHP-Skript!";
                                     ?>
```

</html>

</body>

### **Abb. 3-2 Beispiel von einem PHP-Programm**

Beachtet man den Unterschied zu einem Skript, welches in anderen Sprachen wie Perl oder C geschrieben wurde. Anstatt ein Programm mit vielen Anweisungen zur Ausgabe von HTML zu schreiben, schreibt man einen HTML-Code mit einigen eingebetteten Anweisungen, um etwas auszuführen (z.B. um - wie oben - Text auszugeben). Der PHP-Code steht zwischen speziellen Anfangs- und Schlusstags, mit denen man in den PHP-Modus und zurück wechseln kann.

Was PHP von clientseitigen Sprachen wie Javaskript unterscheidet, ist, dass der Code auf dem Server ausgeführt wird. Sollte man ein Skript wie das obige auf ihrem Server ausführen, würde der Besucher nur das Ergebnis empfangen, ohne die Möglichkeit herauszufinden, wie der zugrunde liegende Code aussieht. Man kann seinen Webserver auch anweisen, alle ihre HTML-Dateien mit PHP zu parsen, denn dann gibt es wirklich nichts, das dem Benutzer sagt, was man in petto hat.

Das Beste an der Verwendung von PHP ist, dass es für Neulinge extrem simpel ist, aber auch einen riesigen Funktionsumfang für den professionellen Programmierer bietet. Scheut man sich nicht, die lange Liste der PHP-Funktionen zu lesen. Man kann einsteigen und binnen weniger Stunden bereits mit dem Schreiben von einfachen Skripten beginnen.

Es gibt drei Hauptgebiete, in denen PHP Skripte genutzt werden.

- Serverseitige Skripte. Dies ist das traditionelle und auch Hauptfeld von PHP. Sie benötigen dazu drei Dinge: den PHP-Parser (CGI oder Server-Modul), einen Webserver und einen Webbrowser. Man muss den Webserver, verbunden mit einer PHP-Installation laufen lassen. Man kann sich die Ausgabe der PHP- Programme über den Server mit einem Webbrowser ansehen.
- Skripte auf der Kommandozeile. Man kann auch PHP-Skripte schreiben, um sie ohne einen Server oder einen Browser laufen zu lassen. Hierfür benötigt man nur den PHP-Parser. Diese Art der Verwendung ist für die regelmäßig auszuführenden

Skripte mittels cron (unter Unix oder Linux) oder dem Task Scheduler (unter Windows). Diese Skripte können auch für einfache Aufgaben zur Verarbeitung von Text verwendet werden.

- Schreiben clientseitiger GUI Applikationen. PHP ist wahrscheinlich nicht die allerbeste Sprache um GUI-Applikationen zu schreiben, aber wenn man PHP sehr gut kennen und einige weiterführende Funktionen in seinen clientseitigen Applikationen nutzen möchten, können Sie PHP-GTK nutzen, um derartige Programme zu schreiben. Auf diese Art haben Sie auch die Möglichkeit, plattformübergreifende Applikationen zu schreiben. PHP-GTK ist eine Erweiterung von PHP, welche in der Hauptdistribution nicht enthalten ist.

PHP kann auf allen gängigen Betriebssystemen verwendet werden, inkl. Linux, vielen Unix-Varianten (inkl. HP-UX, Solaris und OpenBSD), Microsoft Windows, Mac OS X, RISC OS und wahrscheinlich anderen. PHP unterstützt auch die meisten der heute gebräuchlichen Webserver. Dies umfasst Apache, Microsoft Internet Information Server, Personal Web Server, Netscape und iPlanet Server, O'Reilly Website Pro Server, Caudium, Xitami, OmniHTTPd und viele andere. Für den Großteil der Server bietet PHP ein eigenes Modul, für die anderen, welche den CGI Standard unterstützen, kann PHP als CGI-Prozessor arbeiten.

So hat man Freiheit, PHP auf dem Betriebssystem und dem Webserver Ihrer Wahl laufen zu lassen. Weiteres kann man je nach Vorliebe prozedural, objektorientiert oder gemischt programmieren. Auch wenn in der derzeitigen Version von PHP nicht jedes Standard OOP Feature realisiert ist, sind viele Bibliotheken und große Applikationen (inklusive der PEAR-Bibliothek) exklusiv unter Verwendung von OOP-Code geschrieben worden.

Mit PHP ist man nicht auf die Ausgabe von HTML beschränkt. Seine Fähigkeiten umfassen auch das dynamische Generieren von Bildern, PDF-Dateien und Flash Animationen (mittels libswf und Ming). Sie können auch leicht jede Art von Text, wie XHTML-oder irgendeine andere XML-Datei ausgeben. PHP kann diese Dateien automatisch generieren und im Dateisystem speichern, anstatt diese nur auszugeben bzw. auszudrucken, und baut für den serverseitigen Zwischenspeicher dynamische Inhalte.

Vielleicht die größte und bemerkenswerteste Stärke von PHP ist seine Unterstützung für eine breite Masse von Datenbanken. Eine datenbankgestützte Website zu erstellen ist unglaublich einfach.

Wenn man PHP im Bereich des E-Commerce nutzen, wird man Funktionen für Cybercash Payment, CyberMUT, VeriSign Payflow Pro und CCVS für seine Onlineprogramme zur Zahlungsabwicklung zu schätzen lernen.

### **Nachteil von PHP**

1. Erhöhter Netzwerktraffik:  
 Durch die Verwendung von PHP kann eine erhöhte Netzwerk- und Serverbelastung erfolgen, falls beispielsweise User-Eingaben in ein Formular immer nur auf dem Server geprüft werden. Mit dem Einsatz eines zusätzlichen Javascripts kann dieser Nachteil umgangen werden, wenn schon auf der Client-Seite eine gewisse Syntaxprüfung der Eingaben erfolgt. Dies ersetzt zwar nicht die Prüfung auf Korrektheit auf dem Server, aber die Prüfung muss dann vielleicht nur einmal durchgeführt werden.
2. Falls PHP nicht als Modul in den Web-Server eingebunden wird, so ergeben sich daraus die gleichen Probleme wie bei anderen CGI auch (Erzeugung von Prozessen mittels System-Calls, insbesondere fork()).
3. Sicherheitsrisiken: PHP wird auf dem Web-Server ausgeführt. Es ist ein mächtiges Werkzeug, das mit Dateien arbeiten, Netzwerkverbindungen öffnen sowie Systembefehle ausführen kann. Ein jegliches Programm, das diese Fähigkeiten mitbringt und auf einem Web-Server ausgeführt wird, stellt potentiell ein Sicherheitsrisiko dar.  
 Laut PHP-Manual ist PHP, da es speziell auf die Entwicklung von Web-Applikationen ausgerichtet ist, sicherer als andere Sprachen, mit denen Web-basierte Anwendungen programmiert werden (insbesondere auch C und Perl).  
 . Im WWW konnten jedoch auch ganz andere Stimmen dazu vernommen werden:  
 „...applications written in PHP [are] easy to attack and hard to defend“

### **MS SQL Server 2000**

MS SQL Server 2000 ist eins der populärsten Datenbankprodukte auf dem Markt. Die wichtigste Eigenschaft ist, dass es eine Datenbank ist, die Microsoft auf Grund der Windows-Serien-Plattform aufbaut. Deshalb ist es besonders beliebt bei den Windows-Benutzern.

Im Gegensatz zu solchen kleinen Datenbanken wie Access oder Foxpro, ist MS SQL Server 2000 ein funktionsfähiges Datenbankmanagementsystem.


## **3.3 Lösung**

### **3.3.1 Access wird hier benutzt.**

Der Markt des Tools orientiert sich an der Mittel – und kleinen Hotellerie. Meistens sind es die Familienhotels. Access ist einerseits sehr einfach für die Kunden.. Normalerweise nutzt der Chef, der kein Computerspezialist ist, selber die Software. Eine angenehmen und simple Oberfläche ist deshalb nötig. Andererseits ist Access auch schon technisch leistungsfähig ausreichend.

### **3.3.2 Darstellung der Oberfläche**

Die Hauptoberfläche (siehe Formular 3-3-1) wird den Kunden direkt angeboten, wo sie auf ihre Fragen entsprechende Antworten finden können, dadurch dass sie den entsprechenden Link anklicken.

**Willkommen bei E-KUH** 

---












**Welche Fragestellung möchten Sie bearbeiten:**

Wie kann ich in meinem Betrieb Umweltbelastungen reduzieren?	<a href="#">Maßnahmenauswahl</a>
Wie ist die Umweltleistung meines Betriebes im Vergleich zu anderen?	<a href="#">Kennzahlenmittlung</a>
Treffen meine Umwelaktivitäten die Erwartungen meiner Gäste und Kunden?	<a href="#">Verbesserungskatalog</a>
Rechnen sich Investitionen in neue Anlagen und Prozesse?	<a href="#">Wirtschaftlichkeitsrechnung</a>
Wie kann ich systematisch und kontinuierlich meine Umweltleistung verbessern?	<a href="#">Umsetzung des normensystem EMAS / ISO</a>
Wo finde ich weiterführende Informationen?	<a href="#">Adressen, Ansprechpartner</a>
Zum verbindlichen Arbeiten mit dem EDV Tool	<a href="#">Systematische Analyse</a>

Unsere Formulare sind öffentlich zugänglich und können verwendet werden.

### Formular 3-3-1 Startfenster von E-KuH

Beispiel: Hinter „Wirtschaftlichkeitsrechnung“ werden zuerst alle Umweltbereiche vorgestellt. Man wählt einen aus der Liste aus und dann klickt anschließend auf eine der angezeigten Maßnahmen (siehe Formular 3-3-2). Schließlich erhält man eine das Ergebnis dargestellt, nachdem Energie als Umweltbereich und „Gebrauch von energiesparenden Beleuchtungsarten im Außenbereich“ als Maßnahme ausgewählt worden sind. Abschätzung der Wirtschaftlichkeit dieser Maßnahme. In dem Formular 3-3-3 ist z.B.

<b>Wirtschaftlichkeitsrechnung</b>	
<p>In der folgenden Liste können Sie einen Umweltbereich auswählen.            Klicken Sie anschließend auf eine der angezeigten Massnahmen (Text oder Schaltfläche).            Sie erhalten dann eine Abschätzung der Wirtschaftlichkeit dieser Massnahme.</p> <p>Umweltbereich: <input type="text" value="Energie"/></p>	
<b>Gebrauch von energiesparenden Beleuchtungsarten im Außenbereich</b>	
Gebrauch von energiesparenden Beleuchtungsarten in der Garage/ im Bereich der Garageneinfahrt	
Gebrauch von energiesparenden Beleuchtungsarten im Eingangsbereich, in Hallen, Treppenhäusern, Fluren, Gängen	
Gebrauch von energiesparenden Beleuchtungsarten im Restaurant	
Gebrauch von energiesparenden Beleuchtungsarten in den Toiletten im Aufenthaltsbereich	
Gebrauch von energiesparenden Beleuchtungsarten in der Küche	
Gebrauch von energiesparenden Beleuchtungsarten in Nebenräumen	
Gebrauch von energiesparenden Beleuchtungsarten in den Gästezimmern	
Gebrauch von zeitgesteuerten Abschaltautomaten für Beleuchtungen	
Anschaffung einer neuen Waschmaschine	
Anschaffung eines Induktionsherdes	

F



## 4. Entwurf

In diesem Abschnitt werden die Kennzahlen und Wirtschaftlichrechnung vorgestellt. In dieser praktischen Phase fanden eine Aufgabenteilung statt. Der 1. Teil (Kennzahl) wurde von Feng Ye bearbeitet, und der 2. Teil (Wirtschaftlichrechnung) wurde von Yilian Shen bearbeitet.

### 4.1 Kennzahl

#### 4.1.1 Hinführung zum Thema

Umweltkennzahlen werden aus den In- und Outputkonten abgeleitet und liefern Belege für den kontinuierlichen Verbesserungsprozess im Umweltbereich eines Unternehmens. Sie eignen sich zur Information, Kontrolle, Planung und Steuerung der umweltrelevanten Tätigkeiten auf den verschiedenen Ebenen. Ebenso können sie zur Kommunikation mit den Anspruchsgruppen des Unternehmens herangezogen werden und für einen Kennzahlenvergleich zwischen Betrieben verwendet werden.

Tabelle 1 zeigt beispielhaft die Kennzahlen einer Umwelterklärung. Dabei werden sowohl Inputgrößen (z. B. Stromverbrauch) als auch Outputgrößen (z. B. Restmüllaufkommen) auf geeignete Bezugsgrößen bezogen. Im Beispielfall (Insel Mainau) werden im Wesentlichen besucherbezogene und flächenbezogene Größen gebildet. Für Betriebe im Gastgewerbe, die sowohl Übernachtungen als auch (warme) Mahlzeiten anbieten, erscheint es sinnvoll, sowohl die Anzahl der Übernachtungen als auch die Anzahl der warmen Mahlzeiten bei der Bildung von Kennzahlen zu berücksichtigen. Im Rahmen des EDV-Tools soll hier auch die Bildung eines gewichteten Mittels (vgl. Abschlussbericht Projekt Green Flag) implementiert werden.

**Tabelle 4-1-1:**Umweltkennzahlen einer Umwelterklärung

<b>besucherabhängige Kennzahlen (pro 1.000 Besucher)</b>		<b>1998</b>	<b>1999</b>	<b>2000</b>	<b>2001</b>
Mitarbeiter-Arbeitsstunden	h				
Stromverbrauch	kWh				
Verbrauch an Reinigungsmitteln	fest (kg)				
	fl ü s s i g (l)				
Papierverbrauch (Verwaltung, Geschäftsbriefe)	ohne Blatt				
Verbrauch an Werbematerialien	Stück				

Restmüllaufkommen	kg				
Biomüllaufkommen	kg				
Abfälle zur Verwertung (Papier, Glas, DSD)	kg				
Diesel	l				
Benzin	l				
Trinkwasser	m <sup>3</sup>				
CO <sub>2</sub> -Emissionen	kg				
Einweggeschirr					
Servietten	Stück				
Pappteller	Stück				
Pappbecher	Stück				
mit Bodensee-Erlebniskarte angereiste Besucher (Förderung des ÖPNV)	Gäste				
<b>Besucherunabhängige Kennzahlen (pro 1.000 m<sup>2</sup>)</b>					
Einsatz von Düngemitteln					
<i>mineralisch/organisch-mineralisch</i>	kg				
davon stickstoffhaltig	kg				
nicht stickstoffhaltig	kg				
<i>Biologisch</i>	kg				
Torf-Einsatz	m <sup>3</sup>				
Zukauf von Pflanzen	Stück				
<b>weitere Kennzahlen</b>					
Erdgasfahrzeugquote	%				

#### 4.1.2 Darstellung der Arbeitsschritte und der angewandten Methoden

Die Arbeit gliederte sich in drei große Phasen:

1. Konzeptionsphase,
2. Entwicklungsphase,
3. Implementierungsphase.

##### **Konzeptionsphase**

Um bestimmte Kennzahl zu bekommen, müssen die benötigten Daten gesammelt werden.

In dieser Phase entscheidet man sich, welche Daten man braucht und in welcher Form diese Daten dargestellt werden sollen. Diese kommen hauptsächlich von unseren Partnern. Diese stellte Erfahrungswerte des Hotelbetriebs dar. Parallel dazu erfolgten Begehungen in den Hotels, in denen die spezifische Situation vor Ort betrachtet wird.

In diesem Projekt sind folgende wesentliche umweltrelevante Jahresdaten berücksichtigt worden: Energieverbrauch, Abfallaufkommen, Wasserverbrauch und Abwasseraufkommen. Diese alle sind wichtige Faktoren für den Umweltbereich. Weitere wichtige Grunddaten eines Hotels, wie z.B. Betten, Übernachtung, Umsatz oder Gastronomie Gäste werden ebenfalls betrachtet. Diese dienen dazu, die Kennzahl zu definieren. Die Kennzahlen werden einerseits übernachtungsabhängig berechnet, um zu sehen, wie viel Energien oder Wasser werden in einem Jahr pro Übernachtung verbraucht wird. Andererseits kann man auch Kennzahlen selbst definieren bezüglich Bettenanzahl oder Umsatz usw., um einen selbst interessierende Umweltfaktoren zu bekommen, z. B. Energieverbrauch pro Bett, Wasserverbrauch pro Bett. Das alles kann man mit dem E-KUH sehr leicht realisieren.

Bei der Konzeption wurden weiterhin sowohl die Vorgaben von EMAS, ISO als auch vorhandene Maßnahmenkataloge von Öko-Labels analysiert.

## **Entwicklungsphase**

In der Entwicklungsphase wurden verschiedenen Komponenten des EDV-Tools implementiert. Für die Umsetzung waren Datenstruktur, Algorithmen und Benutzeroberfläche zu entwickeln. Für die Kennzahlermittlung konzentriert sich die Arbeit auf die Entwicklung der Grundstruktur dieses Tools (Datenstruktur und Verknüpfung von Benutzeroberflächen und Algorithmen zur Berechnung der Kennzahl mit der Datenstruktur).

## **Datenstruktur**

Der Datenstrukturentwurf ist der 1.Schritt in dieser Phase. Man muss genau überlegen, wie man die Daten organisieren soll. Unter Datenstruktur versteht man die umfangreichen

Tabellen, die definiert werden, um die Daten zu verwalten.

Um die Jahresdaten einzugeben und die Kennzahlen zu berechnen, benötigen wir folgende Tabellen :

### **Tabelle B\_Aspekte**

**B\_Aspekte** (ID\_Zeitreihe, Bezeichnung, ID\_Parameter, ID\_Einheit\_Zaehler, ID\_Einheit\_Nenner, ID\_Prozess1, ID\_Gut1, ID\_Prozess2, ID\_Gut2, flag\_berechnet)

In der Tabelle B\_Aspekte werden wichtige Informationen über die Güter dargestellt. Es gibt sehr viele Güter in einem Hotel, die alle aufgelistet werden müssen.

ID\_Parameter sind z.B. Anzahl, Preis, jährliche Menge oder Fläche.

Unter ID\_Einheit\_Zaehler und ID\_Einheit\_Nenner verstehen wir z.B. für Wasserverbrauch die Einheit m<sup>3</sup>/Jahr. Die ID\_Einheit\_Zaehler hier ist dann m<sup>3</sup>, und die ID\_Einheit\_Nenner ist Jahr. Für einfache Einheiten wie z.B. „Stück“ für Anzahl der Fenster ist „Stück“ wird die ID\_Einheit\_Nenner einfach freigelassen.

### **Tabelle Def\_einheit**

**Def\_einheit**( id\_einheit, einheit\_typ\_id, einheit, faktor)

In der Tabelle def\_einheit werden alle Einheiten mit IDs gekennzeichnet. Die Variable „faktor“ ist schließlich für die Umrechnung von verschiedenen Einheiten in eine Standardeinheit .

### **Tabelle Def\_einheit\_typ**

**Def\_einheit\_typ** ( id\_einheit\_typ, einheit\_typ, Standardeinheit)

In der Tabelle def\_einheit wird eine Variable “einheit\_typ\_id” definiert. Dieser entspricht genau in dieser Tabelle die Variable id\_einheit\_typ. Es gibt viele Einheitentypen, z.B. Energie, Geld, Volumen. Jedem Einheitentyp wird eine Standardeinheit zugeordnet.

### **Tabelle Def\_formular**

**Def\_formular**( id\_formular, formular, knopf, einheit\_typ\_id)

Die Formulare entsprechen der Jahresdateneingabe von verschiedenen Gütern. Man kann „Energie“, „Wasser“, „Abfall“ oder „Sonstiges“ auswählen.

### **Tabelle F\_Kennzahlen\_Gueter**

**F\_Kennzahlen\_Gueter** ( ID\_Zeitreihe, ID\_Zeitreihe\_Kosten, ID\_Formular, anzeigen)

In dieser Tabelle werden die Zeitreihe-IDs mit den Formular-IDs verbunden

### **Tabelle W\_Aspekte**

**W\_Aspekte** ( ID\_Zeitreihe, Datum, Hypothese, Wert)

Alle Daten, die als Jahresdaten eingegeben werden, werden in dieser Tabelle gespeichert, der wichtigsten Tabelle für die Bestimmung der Kennzahl.

### **Tabelle b\_kennzahl**

**b\_kennzahl** ( zaehler, nenner, einheit\_zaehler, einheit\_nenner, wert, loeschen, jahr)

Die selbst definierten Kennzahlen werden in dieser Tabelle angegeben. Die Kennzahlen werden in Bruchzahlen dargestellt.

### **Oberfläche**

In der Entwicklungsphase wurde überlegt, wie die Oberfläche aussehen sollte. Sie sollte knapp und klar sein, aber auch nicht so einfach. Die Dateneingabe und -darstellung sollte benutzerfreundlich sein. Das heißt, es sollte nicht passieren, dass nach der mühsamen Dateneingabe durch einen Klick alle Daten verlorengehen. Das ist ärgerlich. Zu viele

Knöpfe oder Bildern werden die Benutzer verwirren. Die Funktionen müssen reicht sein, aber nicht zu vielfältig, weil die Benutzer, wie zuvor geschrieben wurde, sind meistens Geschäftsführer selbst von kleinen und mittleren Betrieben der Hotel- und Gaststättenbranche. Das entwickelte Tool bietet solchen Betrieben eine praktikable Möglichkeit, selbstständig umweltrelevante Daten zu erfassen, zu quantifizieren, zu bewerten und als Grundlage für eine Umweltprüfung zu nutzen. Deshalb ist entschieden worden, die Oberfläche so übersichtlich wie möglich zu konstruieren. Nachfolgend ist ein Beispiel eines Formulars für den Energieverbrauch. Ähnliche Daten, wie z. B. Wasserverbrauch, Abwasseraufkommen oder Abfallaufkommen, werden auch in ähnlichen Formularen angegeben.

**Angaben zu Jahresdaten**

**Eingabe der Jahresdaten: Energie**

Energieträger auswählen

Zeitraum	Strom [kWh]	Heizöl [m³ Heizöl]	Erdgas [m³ Erdgas]
2002	1000	30	1000
2003	2000	25	2000

Daten übernehmen

Abbruch (Änderungen verwerfen)

▶\*

### **Formular4-1-1 Eingabeformular für Energieverbrauch**

Der Titel des Formulars besagt eindeutig, dass die Jahresdaten von Energien einzugeben. Es gibt viele verschiedene Energieträger, wie z. B. Benzin, Kohle, Erdgas, Heizöl usw. Die muss man auflisten und zur Auswahl stehen lassen. Durch den Knopf „Energieträger auswählen“ kann man alle Energieträger sehen und auch wählen. Dann wird eine neue Zeile darunter stehen. Den Zeitraum kann man durch den untersten Knopf einfügen. Jedes Mal wenn man diesen Knopf drückt, wird automatisch ein Jahr dazu addiert. Z. B. Steht für die obige Darstellung des Jahrs 2004 zur Auswahl. Man kann natürlich den Zeitraum ändern, indem man das gewünschte Jahr eintippt. Mit einem einzigen Knopf „Daten übernehmen“ werden die Änderungen, die der Benutzer gerade gemacht hat, vorgenommen und in einer Tabelle W\_Aspekte gespeichert, die zur Kennzahlberechnung zur Verfügung steht. Wenn man keine Änderungen vornehmen will, kann einfach auf „Abbruch“ geklickt werden. Es ist zu empfehlen, nach jedem Änderungsschritt die Daten sofort zu übernehmen, um ein Datenverlust zu vermeiden.

### **Implementierungsphase**

In der Implementierungsphase wurde das Tool von den Projektpartnern getestet. Die Funktionsfähigkeit und Bedienungsfreundlichkeit des Prototyps wurden überprüft und Änderungen angeregt. Außerdem wurde das Layout des Tools verbessert, so dass es auch für die zukünftigen Nutzer attraktiv ist.

#### **4.1.3 Algorithmen zur Einheitsumrechnung**

Die unterschiedlichen Einheiten stellen die größte Schwierigkeit bei der Kennzahlrechnung dar. Es gibt so viele unterschiedliche Einheiten bei den Grunddaten, so dass man nicht einfach die Datenmenge addieren kann. Manche Unterschiede zwischen zwei Einheiten sind so groß, dass sie zu großen Analysefehlern führen können. Deshalb müssen die Einheiten auf Standardeinheiten umgerechnet werden.

### **Standardeinheit**

Im Umweltbereich sind die Standardeinheiten schon definiert. Dies gilt insbesondere für die Energieträger. Für den Wasserverbrauch gibt es nur die Einheit  $m^3$ , ebenso für das Abfallaufkommen auch. Die Einheiten von verschiedenen Energieträgern sind nicht immer

gleich. Wir haben z.B. für Strom die Einheit „kWh“, für Erdgas die Einheit „m<sup>3</sup> Erdgas“ und für Heizöl die Einheit „m<sup>3</sup> Heizöl“. Die Standardeinheit für



id_einheit_typ	einheit_typ	Standardeinheit
1	Energie	kWh
2	Volumen	m <sup>3</sup>
3	Masse	kg
4	Anteile	%
5	Zeit	Stunde
6	Wasser	m <sup>3</sup>
7	Abfall	m <sup>3</sup>
16	Geld	Euro
17	Leistung	kW
18	sonstiges	

Tabelle 4-1-2 Tabelle für Einheitstyp Definition

Energie ist kWh. In der obenstehenden Tabelle Def\_einheit\_typ werden Standardeinheiten für verschiedene Einheitstypen definiert.

Die Umrechnung zu kWh erfolgt einfach durch einen Faktor. In der Tabelle def\_einheit werden die Faktoren für verschiedene Einheiten definiert.

def_einheit : Tabelle				
	id_einheit	einheit_typ_id	einheit	faktor
▶	1	1	kWh	1
	2	1	m <sup>3</sup> Heizöl	10000
	3	1	m <sup>3</sup> Erdgas	10
	4	1	l Heizöl	10
	5	2	m <sup>3</sup>	1
	6	2	l	0,001
	10	3	kg	1
	11	3	t	1000
	15	4	-	1
	16	4	%	0,01
	20	5	Jahr	
	21	5	Quartal	
	22	5	Monat	
	23	5	Tag	
	24	5	Min	
	30	16	€	
	35	18	Nacht	
	36	18	Gast	
	37	18	Übernachtung	
	45	19	l/Min	
	46	4	Stück	1
*				1

Datensatz: 1 von 21

### Tabelle4-1-3 Tabelle für Einheitsdefinition

#### 4.1.4 Kennzahlberechnung

Jetzt kommen wir zu einem der wichtigsten Punkte dieses Projekts, der Kennzahlberechnung. Wir haben bisher schon alle Jahresdaten und Grunddaten, die mit der Umwelt zu tun haben, gesammelt. Aber das reicht noch nicht. Durch die Kennzahlberechnung können wir die Gesamtsituation in den Griff bekommen, nicht nur von einer Seite aus.

Die Kennzahlberechnung wird in einem Modul „mod\_kennzahl“ definiert. Das ist ein VisualBasic for Applications Programm. Die Kennzahl wird einerseits übernachtungsabhängig definiert, andererseits kann man selbst eine Kennzahl definieren.

### **Übernachtungsabhängige Kennzahlen**

Drei SQL-Anfragen werden gemacht, um Energieverbrauch, Wasserverbrauch und Abfallaufkommen pro Übernachtung zu berechnen. Diese werden im Anhang dokumentiert.

Wir nehmen Energieverbrauch pro Übernachtung als Beispiel und erklären den Vorgang. Die andere zwei sind fast ähnlich.

### **Energieverbrauch pro Übernachtung**

```
SELECT  a.jahr, round(sum(a.menge*b.faktor/x.menge),2) AS kennzahl
FROM    W_Daten2 a, def_einheit b ,F_Kennzahlen_Gueter2 c,W_Daten2 x WHERE
  a.id_zeitreihe IN
    (SELECT id_zeitreihe
     FROM  F_Kennzahlen_Gueter2
     WHERE id_formular=1)
AND a.id_zeitreihe=c.id_zeitreihe
     AND c.id_einheit=b.id_einheit
     AND b.einheit_typ_id=1
     AND x.jahr=a.jahr
     AND x.id_zeitreihe=5
GROUP BY a.jahr ORDER BY a.jahr desc
```

In dieser Abfrage werden wiederum zwei Grundabfrage verwandt. Eine ist F\_Kennzahlen\_Gueter2, und die andere ist W\_Daten2. Die beiden Abfrage werden im Anhang erklärt.

Um der Energieverbrauch pro Übernachtung zu berechnen, brauchen wir folgende Faktoren: Anzahl der Übernachtungen, die Summe von Energieverbrauch. Die entscheidende Sätze in diesem Programmschnitt sind „ b.einheit\_typ\_id = 1“ und „x.id\_zeitreihe= 5“. Durch „ x.id\_zeitreihe = 5“ kann man der Anzahl von Übernachtungen bekommen. Und die Einheittyp\_id von Energie ist 1. Und der erste Satz berechnet genau die Funktion „ Energieverbrauch pro Übernachtung“. Daher können wir sehr leicht solche

Kennzahl bekommen und es ist sehr wichtig für die Gastronomiebranche, um statistisch zu analysieren, ob das Betriebsmodell passend zu der Umwelt ist.

### **Selbstdefinierte Kennzahlen**

Die Kennzahlen sind in Form von Bruchzahlen dargestellt, also mit Zähler und Nenner. Zähler und Nenner sind je eine Variable von den Jahresdaten oder Grunddaten. Bevor man eine neue Kennzahl definieren möchte, muss man sich für ein bestimmtes Jahr entscheiden. Nur wenn für dieses Jahr der Eintrag von Jahresdaten schon vorhanden ist, kann man die Zähler und Nenner für diese Kennzahl bestimmen. Sonst bekommt man eine Fehlermeldung. Die Variablen von dem Eintrag stehen dann zur Auswahl der Zähler und Nenner zur Verfügung. Die selbst definierte Kennzahl sollte sinnvoll sein. Das System kann natürlich auch eine nicht sinnvolle Kennzahl berechnen, aber das ist nutzlos. Man sollte vorher wissen, was man braucht oder einen interessiert. Man kann auch eine schon definierte Kennzahl löschen, indem man die zu löschende Kennzahl markiert. Das VB-Programm (im Anhang) erledigt das alles.

#### **4.1.5 Grafikerstellung**

Zur Vorbereitung für die Grafikerstellung haben wir drei Abfragen gemacht, um die benötigte Daten zu sammeln. Wie zuvor beschrieben, werden die eingegebenen Jahresdaten alle in der Tabelle W\_Aspekte gespeichert, und zwar ohne eine bestimmte Reihenfolge. Diese enthält alle Jahresdaten wie Energie, Wasser, Abwasser, Abfall usw. Deshalb sollen zuerst die Jahresdaten selektiert werden, um die einzelne umweltrelevanten Jahresdaten, wie Energieverbrauch, Wasserverbrauch usw. aufzulisten. Die drei Abfragen sind Abfrage\_Energie, Abfrage\_Wasser, Abfrage\_Abfall.

#### **SELECT**

```
B_Aspekte.Bezeichnung,  
W_Aspekte.Datum,
```

```

W_Aspekte.Wert,
def_einheit.einheit,
def_einheit.faktor,
[W_Aspekte.Wert]*[def_einheit.faktor] AS kWh_Umrechnung
FROM ((B_Aspekte INNER JOIN F_Kennzahlen_Gueter
ON B_Aspekte.ID_Zeitreihe = F_Kennzahlen_Gueter.ID_Zeitreihe)
INNER JOIN W_Aspekte
ON (B_Aspekte.ID_Zeitreihe=W_Aspekte.ID_Zeitreihe)
AND (F_Kennzahlen_Gueter.ID_Zeitreihe = W_Aspekte.ID_Zeitreihe))
INNER JOIN def_einheit
ON B_Aspekte.ID_Einheit_Zaehler = def_einheit.id_einheit
WHERE (((def_einheit.einheit_typ_id)=1));
ORDER BY W_Aspekte.Datum;

```

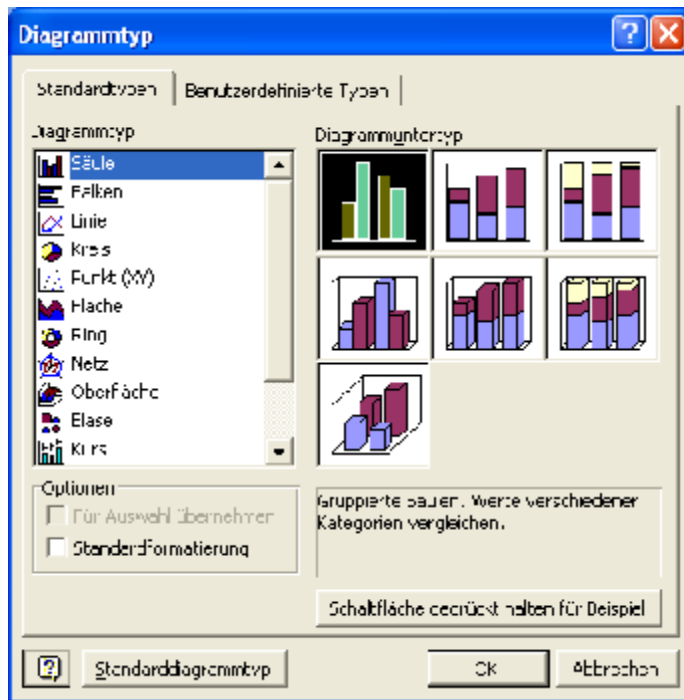
Nehmen wir hier auch nur eine Abfrage Abfrage\_Energie als Beispiel. Für dieser Abfrage braucht vier Tabellen, die miteinander verbunden sind. Die einheit\_typ\_id von Energie ist eins. Somit können die Energiejahresdaten von allen anderen Jahresdaten abgeleitet werden. Dazu braucht man noch folgende Attribute: Bezeichnung (Energieträger, Datum (welches Jahr?), Wert, Einheit, und Faktor (bezüglich Standardeinheit). Diese Abfrage dient zur Kennzahlberechnung und zur Erstellung von Grafiken über die Jahresdaten.

Für den Wasserverbrauch wird eine Tabelle weniger gebraucht. Der Grund hierfür ist, dass für Wasser kein Faktor zur Einheitenumrechnung benötigt wird.

#### 4.1.6 Microsoft Graph

Um das gewünschte Diagramm zu erstellen, benötigen wir ein besonderes Programm, -Microsoft Graph. Das ist nicht in MS Access integriert, sondern muss extra mit MS Office 2000 Premium installiert werden.

Das in die verschiedenen Office-Anwendungen (Word, Powerpoint) integrierte Programm **Microsoft Graph** bietet dem Anwender standardmäßig 73 unterschiedliche Diagrammtypen an, die auch in der Tabellenkalkulation Microsoft Excel verwendet werden.



#### Formular 4-1-2 Formular für Diagrammtyp unter MS Graph

Die 73 Diagrammtypen im Angebot von MS Graph sind in Diagrammtypen und Diagrammuntertypen eingeteilt. Die Diagrammauswahl erfolgt mit Hilfe des Dialogfeldes „Diagrammtypen“, sollte aber erst nach ein paar allgemeinen Vorüberlegungen zu Diagrammen erfolgen. Die Versuchung, sich einfach durch die Vielzahl der Diagrammvarianten zu "zappen", ist natürlich sehr groß.

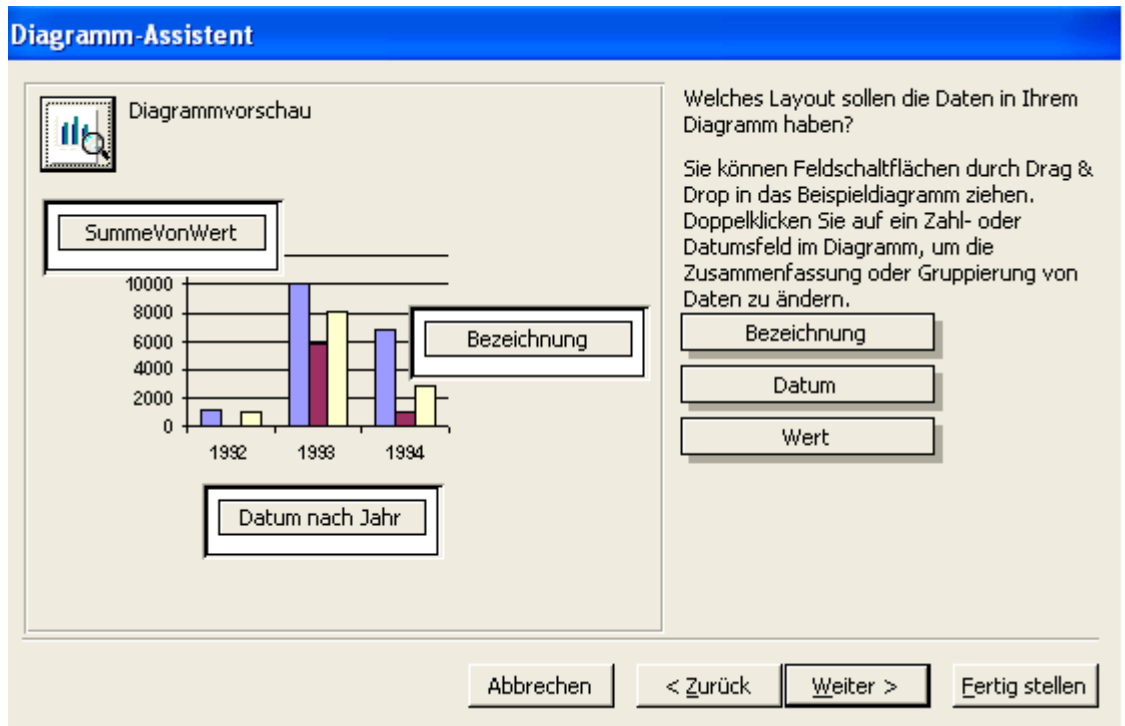
Die **14 Diagrammtypen** sind:

- Säule,
- Balken,
- Linie,
- Kreis,
- Punkt,
- Fläche,
- Ring,
- Netz,
- Oberfläche,
- Blase,
- Kurs,
- Zylinder,
- Kegel,
- Pyramide.

Die vielfältigen Diagrammtypen sind eine der wichtigsten Eigenschaften von Microsoft Graph. Ein anderer Vorteil liegt in der leichten Handhabung.

### **Arbeitsschritt zur Diagrammerzeugung**

1. Ein neues Formular wird erzeugt.
2. In diesem neuen Formular wird ein Diagramm eingefügt dadurch, dass man in dem Menü „Einfügen“ das untere Titel „Diagramm“ klickt.
3. Einen Platz in diesem Formular wird hier ausgewählt, wo das Diagramm stehen soll, und mit dem Linkentaste der Mause ziehen.
4. Jetzt kann man die richtige Abfrage oder Tabelle eingeben
5. Die benötigte Attribute können jetzt ausgewählt werden.
6. Die Diagrammtyp und Diagrammuntertyp werden ausgewählt, in unseren Fall haben wir dieses aufeinanderschichtenden Modell von Säuletyp gewählt, um die Jahresverbrauchvergleichen durchsichtig zu machen
7. Am Schluss darf man die Attribute für die X-Achse, Y-Achse bestimmen.

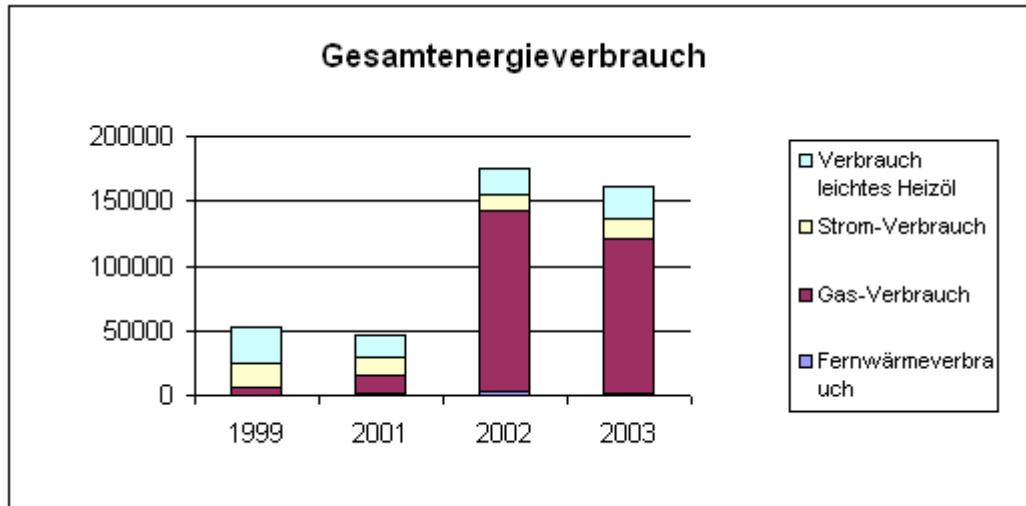


### Formular 4-1-3 Diagramm-Assistent für MS Graph

Man kann einfach Feldschaltflächen durch Drag & Drop in das Beispieldiagramm ziehen. Durch Doppelklicken auf ein Zahl- oder Datumsfeld im Diagramm kann man die Zusammenfassung oder Gruppierung von Daten zu ändern. Z. B. Kann das Datum nach Jahren gewählt werden, aber auch nach Monaten, nach Quartalen, nach Wochen, sogar nach Minuten. Als Wert kann man die Summe wählen, aber auch die Max- oder Minwerte oder die Mittelwerte. Die Legende steht neben dem Diagramm.

8. Es wird ein Titel für dieses Diagramm vergeben.

Schon bekommen wir das Diagramm. MS Graph ist für die Erstellung solcher kompliziertes Grafik besonders effizient. Als Beispiel schaut man das Diagramm des Gesamtenergieverbrauchs an.



#### **Diagramm 4-1-1 Grafikdarstellung für Gesamtenergieverbrauch**

In diesem Diagramm wird besonders deutlich, dass der Gesamtenergieverbrauch im Jahr 2002 viel höher als in den Jahren 1999 und 2001 ist. Die Bezeichnungen in der Legende sind abhängig von der in den Tabellen oder Abfragen verwendeten. Aber man kann die Beschreibungen nachher ändern.

#### **4.1.7 Andere Wege zur Kennzahlberechnung**

Im Startfenster gibt es einen Knopf , durch den man direkt auf die Kennzahlermittlung gehen kann. Aber das ist nicht der einzige Weg zur Kennzahlermittlung. Das Projekt E-Kuh hat für die Benutzern ausgedacht, wenn man zuerst nicht die Kennzahlermittlung schauen möchte, sondern etwas anderes, später kommt aber die Bedürfnis, doch die Kennzahlen zu studieren, dann braucht man nicht extra von dem gegenwärtigen Fenster auszuspringen, sondern man kann sehr bequem die benötigte Kennzahlen bekommen. Es gibt folgende zwei weitere Möglichkeiten, sich auf die Kennzahlen zuzugreifen.

#### **Umweltmanagementsystem EMAS/ISO**

Um ein Umweltmanagementsystem aufzubauen, wird empfohlen, es mit dem

Qualitätsmanagementsystem zu koppeln. Das Projekt E-kUH hat für die Benutzer sieben Schritte als wesentliche Grundlage für Umwelt- und Qualitätsmanagement definiert. Das Ziel ist das Erstellen einer Umweltbilanz, die Auskunft über den Zustand des Betriebs bezüglich der Umwelt gibt. Natürlich braucht man mehrere Faktoren, um eine Umweltbilanz zu erstellen. Einer der wichtigsten Schritte sind die Jahresdaten. Die Jahresdaten müssen genau untersucht und analysiert werden, bevor man eine Umweltbilanz machen kann. Durch den Klick auf „Jahresdaten“ wird man direkt mit dem Jahresdateneingabeformular verbunden.

### **Systematische Analyse**

Mit diesem Formular kann man alle Formulare direkt ansteuern und natürlich auch das Formular Jahresdaten/Kennzahl. Alle in diesem Formular gemachten Angaben können von anderen Formularen aus noch einmal überprüft oder auch eingegeben werden. Das heißt, die Jahresdaten können hier genauso eingegeben oder geändert werden wie direkt von dem Formular „Kennzahlermittlung“.

#### **4.1.8 Zusammenfassung**

In dieses Kapitel wird um die Kennzahlen in dem E-Kuh Tool behandelt. Es gibt zwei wichtige Punkte für ein Hotel: 1. die Einsparung von Rohstoffe; 2. die Umweltfreundlichkeit. Die Kennzahlen im Umweltbereich sind sehr wichtig für die Hotels, um die Umweltfreundlichkeit zu verbessern. Die Hotels können durch die Berechnung von verschiedenen Kennzahlen in einem Jahr erkennen, ob das Betriebsmodul für die Umwelt geeignet ist. Man kann Maßnahmen ergreifen, die gut für die Umwelt sind. Und durch das Kennzahlvergleichen mit vorherigen Jahren kann man feststellen, welche der bereits eingesetzten Maßnahmen, gut für die Erhöhung von Umweltfreundlichkeit und auch gut

für den Betrieb, (Einsparen von den Rohstoffen) sind.

Das Tool ist leichthandhabbar für die Hotels. Man braucht nur die Kennzahlen in den verschiedenen Formularen einzugeben die Rechnung in E-Kuh erfolgt automatisch. Schon nach wenigen Sekunden erhält man die gewünschten Ergebnisse. Später kann man sogar die Kennzahlen im Internet veröffentlichen, um am Wettbewerb zwischen den Hotels teilzunehmen. Die Hotels können sich somit jederzeit selbst kontrollieren.

Wichtig ist die Berechnungsmethode für die Kennzahlen. In diesem Artikel wurden der Energieverbrauch, der Wasserverbrauch und der Abfallaufkommen angegeben. Man muss darauf achten, dass nicht nur die Verbräuche berechnet werden sollen, sondern auch deren Auswirkungen, z. B Abwasser, Abgas oder Abfälle.

## 4.2 Umsetzung der Wirtschaftlichkeitsrechnung in EDV-Tools

### 4.2.1 Konzepte der Wirtschaftlichkeitsrechnung

Für die Wirtschaftlichkeitsrechnung ist ein allgemeines Modell für die Berechnung zu implementieren. Dieser Algorithmus muss sowohl die Eingabedaten als auch die zu berechnenden Daten definieren und eine Verbindung zwischen diesen herstellen. In Abb. 4-2-1-1 ist eine entsprechende Struktur dargestellt, die als Basis sowohl für die Berechnung von technisch komplexen Anlagen als auch für einfachere Investitionsmaßnahmen gültig ist. /2/

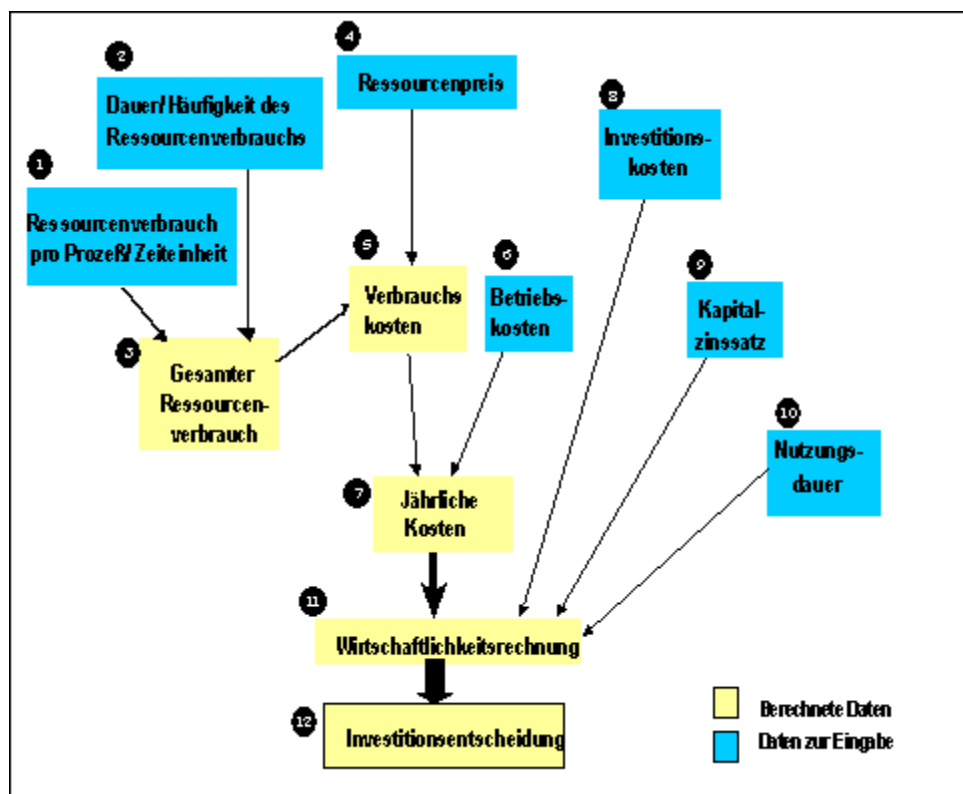
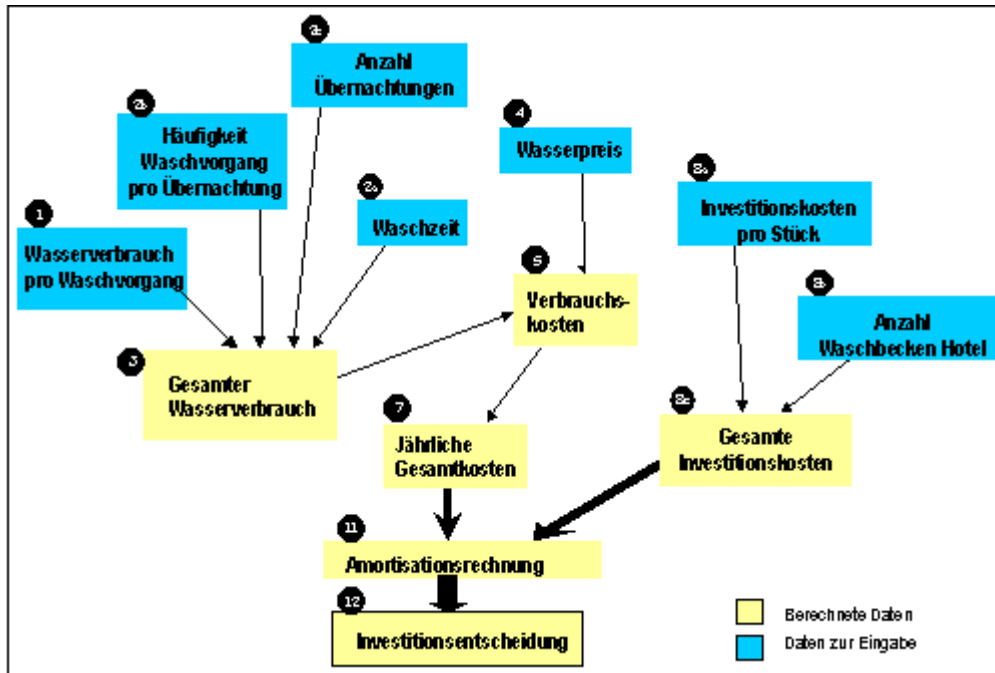


Abb. 4-2-1: Rechengangmodell bei Wirtschaftlichkeitsrechnungen

Bei den Kosten werden entsprechend VDI 2067 Betriebs-, Verbrauchs- und Kapitalkosten definiert. Zu den *Betriebskosten* zählen u.a. die Wartung, die Instandsetzung und Erneuerung der Anlage sowie die Dienstleistungen, die von der Art der Anlagen abhängen. Die *Verbrauchs-kosten* sind vom Preis der Ressourcen und von der verbrauchten Menge abhängig. Die *Kapitalkosten* werden durch die Höhe der Investitionskosten bedingt. Eine wichtige Rolle spielt dabei die Höhe des Kapitalzinssatzes des Darlehens, der von Banken und Kreditgebern festgelegt wird. Bei der Berechnung der Kapitalkosten muss die Nutzungsdauer der Maßnahme berücksichtigt werden. Auf der Basis dieser Informationen kann eine Wirtschaftlichkeitsrechnung durchgeführt werden .

Beispielhaft wird das Modell in Abb. 4-2-1-3 für die Maßnahme „Durchflussbegrenzer bei Waschbecken“ angewendet. Der jährliche Wasserverbrauch wird bei einer bestimmten Anzahl von Waschvorgängen pro Übernachtung und bei einem gewissen Wasserverbrauch pro Waschvorgang, der von der Art des Wasserhahns abhängt, berechnet. Mit dem Wasserpreis pro m<sup>3</sup> Wasser rechnet man die Verbrauchskosten aus. Diese Berechnung wird bei einem Waschbecken mit Durchflussbegrenzer und bei einem ohne Durchflussbegrenzer wiederholt. Aus der Differenz zwischen den Kosten mit oder ohne Wasserspareinrichtungen ergeben sich die ersparten Kosten des Wasserverbrauchs inklusiv des Erwärmens von Warmwasser und das Einsparungspotential in m<sup>3</sup> Wasser. Die vermiedenen Kosten hängen vom Einsatz von Durchflussbegrenzer an den Waschbecken ab. Die Anschaffungskosten dieser Maßnahme sind daher zu bestimmen. Die Amortisationszeit ergibt sich aus

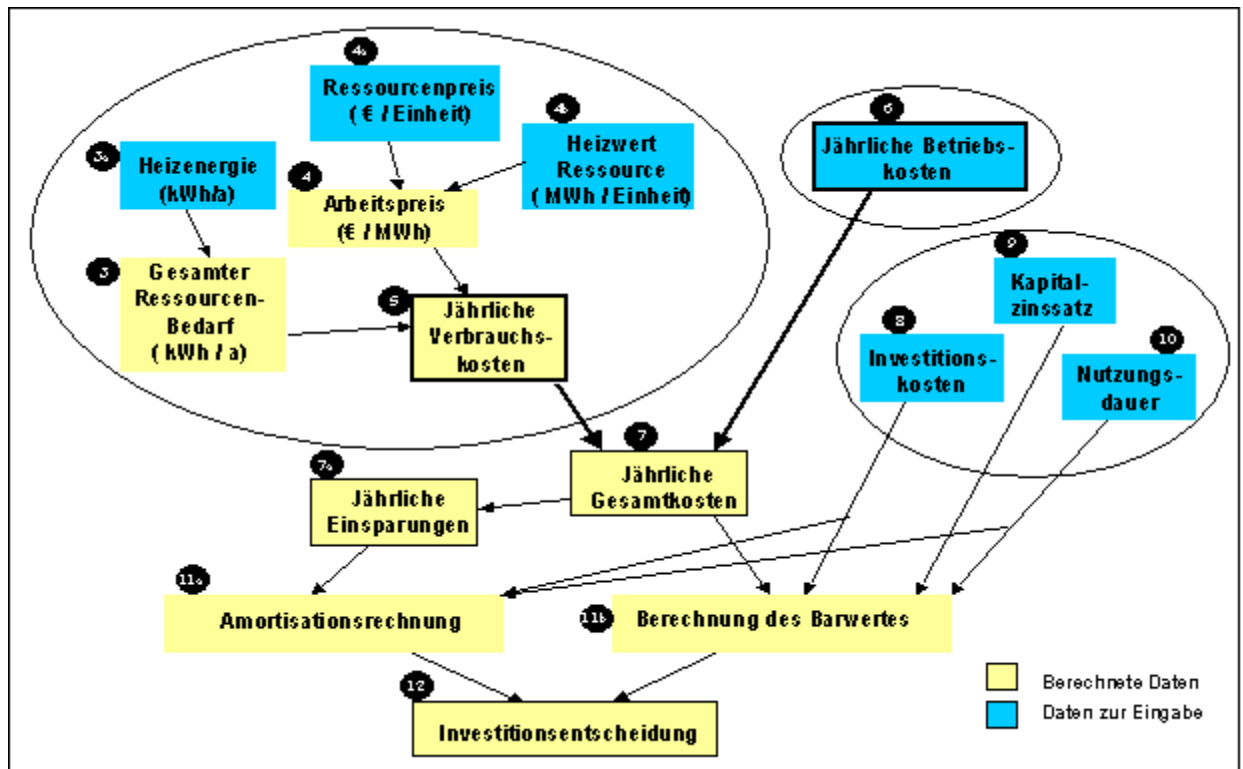


**Abb. 4-2-2:** Rechenvorgang für Wassersparmaßnahmen

dem Verhältnis zwischen Investitionskosten und ersparten Verbrauchskosten.

Die Baumstruktur des Rechenvorganges verändert sich wie in Abb 4-2-1 dargestellt.

Die Wirtschaftlichkeitsrechnung bei Warmwasser- und Heizungsanlagen erfordert eine deutlich größere Anzahl von Daten bei der Eingabe und Berechnungen. Zentral für die Berechnung ist der jährliche Wärmebedarf des Hotelgebäudes. Von da ausgehend wird der gesamte Ressourcenbedarf ausgerechnet und mit dem Arbeitspreis der Anlage ergeben sich die jährlichen Verbrauchskosten.



**Abb. 4-2-3:** Rechenvorgang für Heizungs- und Warmwasseranlagen

Die Investitionskosten für die Anlage werden über die Nutzungsdauer und in Abhängigkeit des Kapitalzinssatzes verzinst. Für die Kapitalkosten wird die Annuitätenmethode benutzt.

Mit den Ergebnissen dieser Berechnungen kann sowohl eine Amortisationsrechnung durchgeführt als auch der Barwert ausgerechnet werden.

Im EDV-Tool ist die Wirtschaftlichkeitsrechnung zumindest für die beiden dargestellten Beispiele sowie für die folgenden Anwendungsfälle zu implementieren:

- Wasser-Stop-Tasten in Toilettenspülungen
- Kleinere Spülkästen für Toilettenspülungen

- Regenwassernutzung
- Verzicht auf Portionsverpackungen

Die dargestellten Beispiele haben aufgezeigt, dass der Rechengang für verschiedene Maßnahmen jeweils spezifische Ausprägungen aufweist. Daher ist zu prüfen, ob ein allgemeiner Algorithmus oder mehrere maßnahmenspezifische Algorithmen für die Implementierung besser geeignet sind.

Die Implementierung der zur Wirtschaftlichkeitsrechnung notwendigen Rechenschritte kann auf zwei verschiedene Weisen erfolgen. Entweder werden die Berechnungen mit dem Softwareprogramm Microsoft Excel oder innerhalb von Microsoft Access ausgeführt.

## 4.2.2 Entwicklungsphase

### 4.2.2.1 Tabellen für Basisobjekte

Entsprechend den vorangegangenen Ausführungen sind die folgenden Basisobjekte zu unterscheiden:

- Aspekte → Tabelle B\_Aspekte
- Güter → Tabelle B\_Gueter
- Maßnahmen → Tabelle B\_Massnahmen
- Prozesse → Tabelle B\_Prozesse
- Parameter für Dateneingaben → Tabelle B\_Parameter
- Rechengang → Tabelle B\_Rechengang
- Rechengang Operanden → Tabelle B\_Rechengang Operanden
- Umweltauswirkungen → Tabelle B\_Umweltauswirkungen

Die entsprechenden Tabellen werden in den folgenden Abschnitten kurz dargestellt.

#### **Tabelle B\_Aspekte**

B\_Aspekte = {ID\_Zeitreihe Integer, Bezeichnung Varchar(255), ID\_Parameter Integer, ID\_Einheit\_Zaehler Integer, ID\_Einheit\_Nenner Integer, ID\_Prozess1 integer, ID\_Gut1 integer, ID\_Prozess2 integer, ID\_Gut2 integer, flag\_berechnet Ja/Nein}

Diese Tabelle definiert alle Bereiche, die ein Hotel zur Kontrolle braucht, zum Beispiel Anzahl der Betten, Öffnungstage, usw. Gleichzeitig wird die entsprechende Einheit durch

ID\_Einheit\_Zaehler oder ID\_Einheit\_Nenner mit eingegeben. Denn die Einheiten sind sehr wichtig während der Berechnung.

### **Tabelle B\_Gueter**

B\_Gueter = {ID\_Gut Integer, Gut Varchar(80), Gefahrstoff Ja/Nein, Beweglich Ja/Nein, Investitionsgut Ja/Nein, Input-Output-Bilanz Ja/Nein, sichtbar Ja/Nein, relevant Ja/Nein, ID\_Guetergruppe integer, Ebene Integer}

Hier werden alle Objekte des Hotelmanagements, z.B. Wasser, Getränke, usw. definiert und nach bestimmten Merkmalen klassifiziert.

Für die Güter kann eine interne Hierarchie in Form einer Baumstruktur definiert werden. Dadurch ist es möglich, ähnliche Güter, z. B. verschiedene Reinigungsmittel zusammenzufassen. Dies wird durch Eingabe eines entsprechenden übergeordneten Gutes (oder Gütergruppe) im Feld ID\_Gütergruppe erreicht. Das Feld ermöglicht eine interne 1:n Zuordnung, d. h. jedes Gut hat maximal einen Vorgänger, kann aber beliebig viele nachgeordnete Güter haben. Dies entspricht der mathematischen Definition einer Baumstruktur.

### **Tabelle B\_Massnahmen**

B\_Massnahmen = {ID\_Massnahme Integer, flag\_Wirtschaftlichkeitsrechnung Ja/Nein, Massnahme Memo, Kurzbezeichnung Varchar(100), Formularansicht Varchar(255), ID\_Umweltbereich Integer, ID\_Massnahmengruppe Integer, flag\_Wirtschaftlichkeitsrechnung\_alt Ja/Nein}

Diese Tabelle ist eine der wichtigsten Tabellen in der Wirtschaftlichkeitsberechnung. Alle Maßnahmen, die E-Kuh einem Hotel zur Verfügung stellt, sollten hier eingegeben werden. Damit wird mitgeteilt, welche Maßnahmen mit der Wirtschaftlichkeitsberechnung zu tun (durch flag\_Wirtschaftlichkeitsrechnung) hat, worauf sich die Maßnahme bezieht (durch Formularansicht, z.B. Einwegbesteck) und in welchem Umweltbereich sie liegt (durch ID\_Umweltbereich) usw.

Auch für die Maßnahmen kann eine interne Hierarchie in Form einer Baumstruktur

definiert werden. Dadurch ist es möglich, ähnliche Maßnahmen, z.B. Verwendung von Groß- und Mehrweggebinden beim Einkauf von Frischgemüse. Dies wird durch Eingabe einer entsprechenden übergeordneten Maßnahme im Feld ID\_Massnahme\_Gruppe erreicht. Die Zuordnung davon ist 1:n.

#### **Tabelle B\_Parameter**

B\_Parameter = {ID\_Parameter Integer, Parameter Varchar(50), Einheit Varchar(50), Kommentar Memo}

#### **Tabelle B\_Prozesse**

B\_Prozesse = {ID\_Prozess Integer, Prozess Varchar(255), Bezeichner Varchar(50), ID\_Betrieb Integer, Ebene Integer, sichtbar Ja/Nein, Kommentar Varchar, ID\_Funktion Integer, ID\_Prozessgruppe Integer}

Zu beachten ist, dass für die Prozesse eine interne Hierarchie in Form einer Baumstruktur definiert werden kann. Dadurch ist es möglich, ähnliche Prozesse, z. B. Reinigen des Bades und Reinigen der Toilette zusammenzufassen. Dies wird durch Eingabe eines entsprechenden übergeordneten Prozesses (oder Prozessgruppe) im Feld ID\_Prozessgruppe erreicht. Das Feld ermöglicht eine interne 1:n Zuordnung, d. h. jeder Prozess hat maximal einen Vorgänger, kann aber beliebig viele nachgeordnete Prozesse haben. Dies entspricht der mathematischen Definition einer Baumstruktur.

#### **Tabelle B\_Rechengang**

B\_Rechengang = {ID\_Berechnung Integer, Nr\_Rechenschritt Integer, ID\_Operator Integer, ID\_Ergebnis integer, Kommentar Memo}

#### **Tabelle B\_Rechengang Operanden**

B\_Rechengang Operanden = {ID\_Berechnung Integer, Nr\_Rechenschritt Integer, Nr\_Operand Integer, ID\_Operand integer, Flag\_Operd\_berechnet Ja/Nein, Kommentar Memo}

#### **Tabelle B\_Umweltauswirkungen**

B\_Umweltauswirkungen = {ID\_Umweltauswirkung Integer, Daten Varchar(50), Umweltbereich Varchar(50)}

### **Tabellen für Definitionen**

Bei einer Vielzahl von Eingabefeldern in den zuvor genannten Tabellen sind nur ausgewählte Einträge zulässig. Um diese Einträge sauber zu definieren und nachträgliche Veränderungen zu ermöglichen, sind die zulässigen Einträge jeweils in Nachschlage- oder Definitionstabellen zusammenzufassen.

Folgende Definitionstabellen sind erforderlich:

- Einheiten für B\_Gueter: def\_Einheit,
- Operatoren : def\_Operatoren.

#### **Tabelle def\_Einheit**

def\_Einheit = {id\_einheit Integer, einheit\_typ\_id Integer, einheit Varchar(50), faktor Integer}

#### **Tabelle def\_Operatoren**

def\_Operatoren = {ID\_Operator Integer, Symbol Varchar(1), Operator Varchar(50)}

#### **Tabelle W\_Aspekte**

Das ist eine Tabelle für Zeitreihen.

Insbesondere für Güter (z. B. Input- oder Output-Stoffströme) sind zeitliche Verläufe von Bedeutung. Diese sind in der Tabelle W\_Gueter abgelegt.

W\_Aspekte = {ID\_Zeitreihe Integer, Datum Datum/Uhrzeit, Hypothese Integer, Wert Integer}

#### **Tabelle Z\_Massnahmen\_Umweltauswirkungen**

Das ist eine Tabelle der Zuordnungen.

Sogenannte 1:n-Zuordnungen, wie zum Beispiel von Prozessverantwortlichen zu Prozessen, können direkt in den Tabellen der entsprechenden Basisobjekte implementiert werden. Z. B. gibt es für jeden Prozess nur einen Prozessverantwortlichen, und dieser

kann direkt in das entsprechende Feld der Tabelle B\_Prozesse eingetragen werden (vgl. Abschnitt 3.2.1.4). Eine Funktion im Unternehmen kann aber durchaus für mehrere Prozesse verantwortlich sein, daher handelt es sich bei der Zuordnung von Prozessverantwortlichen zu Prozessen um eine 1:n Zuordnung, d. h. 1 Prozessverantwortlicher zu n Prozessen.

Bei sogenannten n:m-Zuordnungen ist eine Abspeicherung der entsprechenden Informationen in keiner Tabelle der entsprechenden Basisobjekte möglich. Ein Beispiel für eine solche Zuordnung ist diejenige zwischen Maßnahmen und Prozessen. Eine Maßnahme kann mehr als einen Prozess betreffen und umgekehrt kann es bei einem Prozess mehrere mögliche Maßnahmen zur Verbesserung der Umweltrelevanz geben. D. h. es ist prinzipiell eine allgemeine Zuordnung von n Maßnahmen zu m Prozessen möglich. Die entsprechenden Verknüpfungen müssen in einer relationalen Datenbank in einer separaten Zuordnungstabelle abgelegt werden.

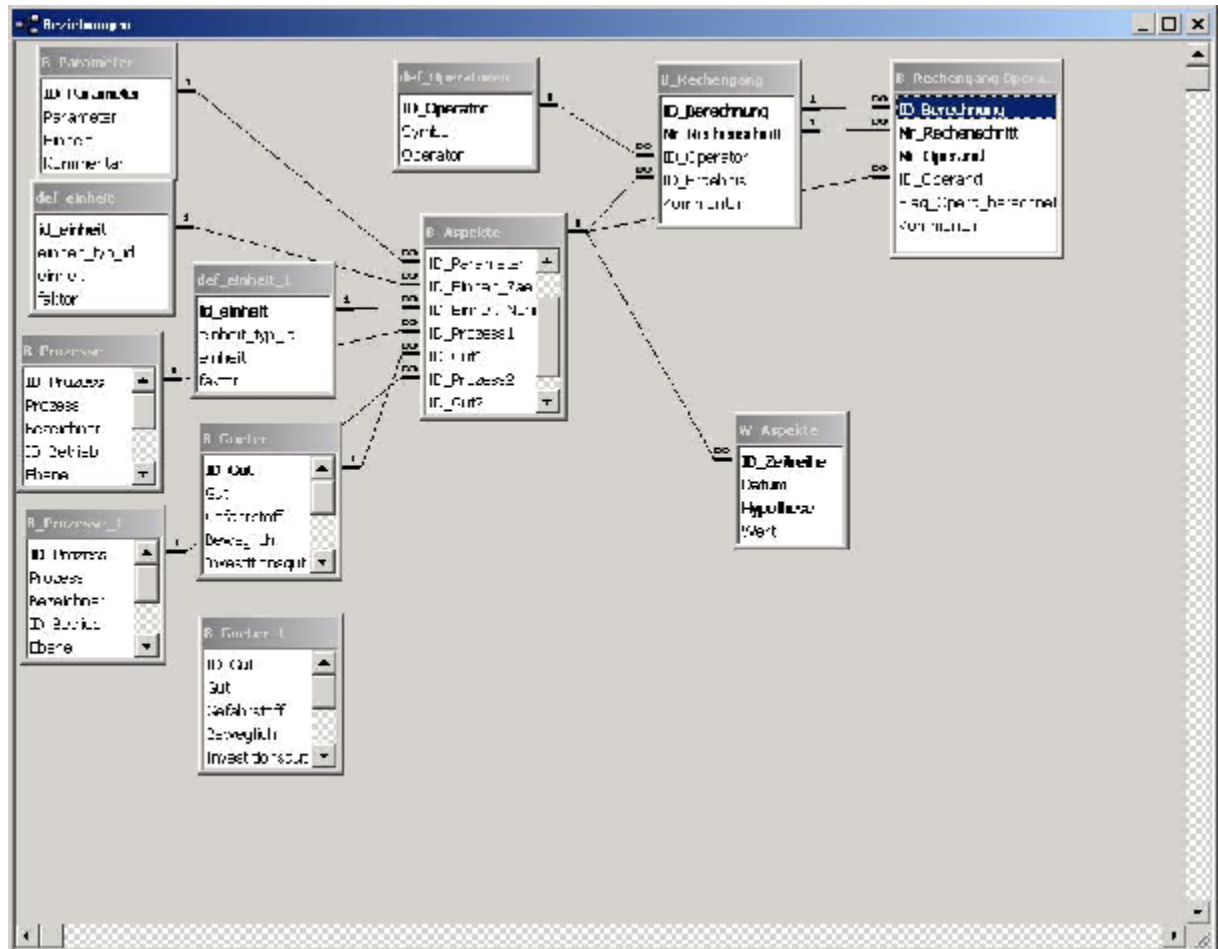
Im EDV-Tool E-Kuh sind im Bereich Wirtschaftlichkeitsrechnungen folgende m:n Zuordnungen erforderlich:

- Zuordnungen von Maßnahmen zu Umweltauswirkungen  
→ Tabelle Z\_Massnahme\_Umweltauswirkungen

VBAZ\_Massnahme\_Umweltauswirkungen = {ID\_Massnahme Integer, ID\_Umweltauswirkung Integer, flag\_HauptUmweltauswirkung Ja/Nein}

#### 4.2.2.2 Entity-Relationship-Diagramm

Die wesentlichen Verknüpfungen zwischen den zuvor dargestellten Tabellen sind in Abb. 4-2-2-2 dargestellt. Dabei wurde aber der Übersichtlichkeit halber darauf verzichtet, die Verknüpfungen zwischen den Nachschlagetabellen und den übrigen Tabellen darzustellen.



**Abb. 4-2-4:** Struktur der Beziehungen von Wirtschaftlichkeitsrechnungen im Prototyp des EDV-Tool (MS Access)

#### 4.2.2.3 Abfragen und Module für die Wirtschaftlichkeitsrechnung

**Abfragen ( Siehe bitte alle genauen Abfragen im Anhang!)**

Der Benutzung nach gibt es zwei Abfragenserien in der Wirtschaftlichkeitsrechnung. Eine unterstützt die Rechnung des Verbrauches,

nämlich Abfrage H\_Massnahmen\_Gueter\_Rechengang,

Abfrage H\_Rechengang\_Rechenschritte,

Abfrage H\_Rechengang\_Rechenschritte\_Operand,

Abfrage H\_Wirt\_Gueter\_Mengen0,

Abfrage H\_Wirt\_Gueter\_Mengen1,

und Abfrage H\_Wirt\_Gueter Mengen2.

Eine andere unterstützt die Rechnung der Investgüter,

nämlich Abfrage H\_Massnahmen\_InvestGueter\_Rechengang,

Abfrage H\_Rechengang\_Rechenschritte\_InvestGueter,

Abfrage H\_Rechengang\_Rechenschritte\_Operand\_InvestGueter,

Abfrage H\_Wirt\_InvestGueter0,

Abfrage H\_Wirt\_InvestGueter1,

Abfrage H\_Wirt\_InvestGueter2,

Abfrage H\_Wirt\_InvestGueter3,

Abfrage H\_Wirt\_InvestGueter4,

Abfrage H\_Wirt\_InvestGueter5,

und Abfrage H\_Wirt\_InvestGueter6.

Der Struktur nach werden alle Abfragen in Basis(normale Abfragen) und die zur Aktualisierung klassifiziert. Zur ersten Klasse gehören

Abfrage H\_Massnahmen\_Gueter\_Rechengang,

Abfrage H\_Rechengang\_Rechenschritte,

Abfrage H\_Rechengang\_Rechenschritte\_Operand,

Abfrage H\_Massnahmen\_InvestGueter\_Rechengang,

Abfrage H\_Rechengang\_Rechenschritte\_InvestGueter,

und Abfrage H\_Rechengang\_Rechenschritte\_Operand\_InvestGueter. Darin gibt es alle wichtigen Informationen, um alle Formulare aufzubauen und die Rechnungen in den Programmen richtig durchzuführen. Zur Aktualisierung allen Mengentabelle dienen

Abfragen H\_Wirt\_Gueter\_Mengen0,1,2

und Abfragen H\_Wirt\_InvestGueter0,1,2,3,4,5,6. Hier funktionieren die beiden Abfragen,

die mit 0 enden, als die Abfragen zu löschen. Das heißt, die alte Mengenangaben werden durch die beiden löschen. Die Abfragen, die mit 1 enden, arbeiten zur Einfügung neuen Datei. Recordset über aktuelle Mengen werden hier eingetragen. Und die Reste dienen zur Aktualisierung der jeweiligen Parameter. Zum Beispiel dienen Abfragen H\_Wirt\_InvestGueter 2,3,4,5,6 zur Änderung der Anzahl der Investgüter, Nutzungsdauer, spezifischem Kapitalzinssatz, jährliche Betriebskosten und Investitionskosten.

Außerdem gibt es noch sechs zusätzliche Abfragen.

Abfrage H\_Wirtschaftlichkeit\_Eingabedaten,

Abfrage H\_Wirtschaftlichkeit\_Eingabedaten0,

Abfrage H\_Kosten\_jährlich,

Abfrage H\_Zeitreihe\_Rechengang,

Abfrage H\_Rechengang\_Rechenschritte2,

und Abfrage H\_Rechengang\_Rechenschritte\_Operand2.

Analyse der Basisabfragen:

Abfrage H\_Maßnahmen\_Gueter\_Rechengang ist eine der wichtigsten Abfragen in der ersten Serien. Sie stellt vor, bei welchen Maßnahmen der Verbrauch von welchen Gütern durch welche Rechengänge berechnet werden. Und normalerweise dienen die Maßnahmen zu einem bestimmt Prozess. Deshalb bezieht sie sich auf die Informationen von der Maßnahmen, Prozess, Gütern und den bezogenen Rechengängen, welche durch die Tabelle von B\_Massnahmen, Z\_Massnahme\_Prozess, Z\_Prozess\_Gueter, B\_Gueter, B\_Aspekte und B\_Rechengang dargestellt werden.

Abfrage H\_Rechengang\_Rechenschritte spielt Hauptrolle bei der Rechnung. Sie zeigt genau, in einer Maßnahme es welchen Rechengang, insbesondere die konkreten Rechenschritten gibt. Dazu zeigen sich noch die bezogenen Operatoren. Offensichtlich werden hier die Tabelle H\_Massnahmen\_Gueter\_Rechengang und B\_Rechengang benutzt.

Bei der Abfrage H\_Rechengang\_Rechenschritte\_Operand sieht man alle Operanden in einer konkreten Rechenschritt genau an. Und bis hier wird eine Rechnung vollständig dargestellt.












Zusammenfassend werden diese drei Abfragen Schritte für Schritt definiert. Von einem großen Konzept „Maßnahme“ bis zur kleinsten Einheit „Operand“ wird die Rechnung dargestellt. Analog wie die erste Serie funktionieren alle drei Abfragen aus der zweiten Serie genauso. Der einzige Unterschied liegt in den Gütern zur Behandlung, welche die Investgüter sind. Dies ist eine andere Art von Rechnung. Das heißt, man berechnet, bei welchen Maßnahmen wie viele Güter investiert werden sollten.

### **3 Prozessklärung und Beispiele von Formularen**

#### **4.2.3.1 Hauptoberfläche der Wirtschaftlichkeitsrechnung**

Bei der Wirtschaftlichkeitsrechnung ist jede Maßnahme entscheidend. Nachdem der Umweltbereich bestimmt worden ist, muss die entsprechende Maßnahme ausgewählt werden, worauf die entsprechende Berechnung erfolgt.

Das folgende Formular zeigt insgesamt elf Maßnahmen im Umweltbereich Energie:

<b>Wirtschaftlichkeitsrechnung</b>	
<p>In der folgenden Liste können Sie einen Umweltbereich auswählen. Klicken Sie anschließend auf eine der angezeigten Massnahmen (Text oder Schaltfläche). Sie erhalten dann eine Abschätzung der Wirtschaftlichkeit dieser Massnahme.</p> <p>Umweltbereich: <b>Energie</b></p>	
<b>Gebrauch von energiesparenden Beleuchtungsarten im Außenbereich</b>	
Gebrauch von energiesparenden Beleuchtungsarten in der Garage/ im Bereich der Garageneinfahrt	
Gebrauch von energiesparenden Beleuchtungsarten im Eingangsbereich, in Hallen, Treppenhäusern, Fluren, Gängen	
Gebrauch von energiesparenden Beleuchtungsarten im Restaurant	
Gebrauch von energiesparenden Beleuchtungsarten in den Toiletten im Aufenthaltsbereich	
Gebrauch von energiesparenden Beleuchtungsarten in der Küche	
Gebrauch von energiesparenden Beleuchtungsarten in Nebenräumen	
Gebrauch von energiesparenden Beleuchtungsarten in den Gästezimmern	
Gebrauch von zeitgesteuerten Abschaltautomaten für Beleuchtungen	
Anschaffung einer neuen Waschmaschine	
Anschaffung eines Induktionsherdes	

#### 4-2-5 Formular frm\_Maßnahmen\_Umweltbereich

Mit der Auswahl der Maßnahmen wird die Wirtschaftlichkeitsrechnung gestartet.



Weiterhin werden die durchzuführenden Rechengänge, die einzelnen Rechenschritte und notwendige Operanden aufgelistet. Dann wird die Mengentabelle aktualisiert, indem alte Mengenangaben (Abfrage „H\_Wirt\_Gueter\_Mengen0“) gelöscht werden, Records in der Mengentabelle (Abfrage „H\_Wirt\_Gueter\_Mengen1“) angelegt werden und dann ein Recordset über aktuelle Mengen (Abfrage „H\_Wirt\_Gueter\_Mengen2“) eingetragen wird. Schließlich werden die Schleifen über die Rechengänge durchgeführt, wodurch sich das Ergebnis des Rechengangs ermitteln lässt.

Alle Rechengänge werden durch die Funktion *Rechengang* realisiert. Bei dieser Funktion zeigt sich, wie die Rechengänge im Allgemeinen durchgeführt werden. Es beginnt mit dem in der Liste aller Rechenschritte zu findenden ersten Rechenschritt für den Rechengang. Dann wird die Schleife über alle Rechenschritte im Rechengang durchgeführt, wodurch die richtigen Operanden und Operatoren gefunden und angewendet werden. Schließlich ergibt sich das Ergebnis, das zugewiesen wird.

Außerdem gibt es eine entsprechende Formel zur Berechnung für eine bestimmten Maßnahme.

Zum Beispiel das folgende Formular:

Rechengang	
<i>Brennstellen in Außenanlagen</i>	30
X	
<i>Brenndauer Beleuchtung in Außenanlagen</i>	13
X	
<i>Öffnungstage</i>	330
=	
<i>Betriebsstunden Beleuchtung Außenbereich pro Jahr</i>	128700
X	
<i>Leistung konventionelle Lampen im Schnitt</i>	60
X	
<i>Umrechnungsfaktor Wh in kWh</i>	0,001
=	

#### 4-2-7 Formualr frm\_Rechengang

Bei der anderen Berechnung *Rechnen\_InvestGueter* handelt es sich um die Berechnung von Investitionsgütern. Zuerst werden analog die entsprechenden Abfragen („H\_Massnahmen\_InvestGueter\_Rechengang“, „H\_Rechengang\_Rechenschritte\_InvestGueter“, „H\_Rechengang\_Rechenschritte\_Operand\_InvestGueter“) definiert. Zweitens wird eine betrachtete Maßnahme nach den Auswahl ermittelt. Weiterhin werden die durchzuführenden Rechengänge, die einzelnen Rechenschritte und notwendigen Operanden aufgelistet. Dann werden alte Mengenangaben (Abfrage „H\_Wirt\_InvestGueter0“) gelöscht. Es folgen weitere Schleifen über die Rechengänge, wodurch sich das Ergebnis des Rechengangs ermitteln lässt. Hier wird die Funktion *Rechengang* nach der entsprechenden Formel wieder aufgerufen. Abschließend werden die Mengentabellen aktualisiert, indem Recordset über aktuelle Mengen (Abfrage

„H\_Wirt\_InvestGueter1“) eingetragen wird, und Anzahl (Abfrage „H\_Wirt\_InvestGueter2“), Nutzungsdauer (Abfrage „H\_Wirt\_InvestGueter3“), spezifischer Kapitalzinssatz (Abfrage „H\_Wirt\_InvestGueter4“), jährliche Betriebskosten (Abfrage „H\_Wirt\_InvestGueter5“) und Investitionskosten (Abfrage „H\_Wirt\_InvestGueter6“) korrigiert werden.

#### **4 Zusammenfassung**

Was steht die Wirtschaftlichkeitsrechnung von E-Kuh zur Verfügung?

Dieses Modell der Wirtschaftlichkeitsberechnung hilft technisch komplexe Anlagen zu berechnen und bietet die entsprechende einfachere Investitionsmaßnahmenempfehlungen an. Die Berechnungsschwerpunkte liegen in der Ermittlung des Verbrauchs, sowie der Investitionsgüter. Der jeweilige Algorithmus liegt in der Definition der Eingabe- und Ausgabendaten und ihren funktionalen Zusammenhängen mit der Umwelt.

## 5. Evaluation

### Was sind alle Vorteile von E-Kuh?

Zunächst ist E-Kuh ein E-Produkt. Papier, das bisher immer als normales Medium bei der Informationsverwaltung eines Hotels genutzt wurde, wird gespart. So kann man E-Kuh „Leitfaden zum Umweltmanagement“ nennen, denn je weniger Papier gebraucht wird, desto weniger Holz wird gehackt und desto weniger Bäume werden gefällt. Somit trägt E-kuh zum Umweltschutz bei.

Im Wesentlichen kann E-Kuh alle Analyseschritte und Vorgehensanleitungen integrieren, Berechnungen und Routinevorgängen automatisieren sowie redundante Arbeiten vermeiden. Dies hilft, das Hotelmanagement einfacher und effizienter zu gestalten. Durch E-Kuh kann ein Manager die Analyseergebnisse und die Maßnahmenempfehlung viel schneller erhalten.

Dennoch ist E-Kuh nicht perfekt. Es weist folgende Nachteile auf:

1. Dieses System ist nur für einen Benutzer geeignet. Das heißt, innerhalb eines Hotels kann es z.B. zur Kontrolle der Kennzahlen, zur Wirtschaftlichkeitsrechnung, zum Vergleich der Ergebnisse genutzt werden. Ein Datenaustausch zwischen mehreren Hotels ist allerdings nicht möglich.
2. Dieses System kann nur unter einem deutschen Operating System (OS) arbeiten. Das heißt, es wird schwer sein, das System z.B. in Asien zu verbreiten. Dieses Problem von Spracheninkompatibilität führt dazu, dass man einige Formulare unter einem chinesischen OS nicht richtig öffnen kann; es treten stets Fehlermeldungen auf.
3. Es kann trotz richtiger Bedingung zu Fehlermeldungen kommen.

## **6. Ausblick**

E-Kuh muss noch weiter entwickelt werden.

Webbasierte Datenbankmanagementsysteme könnten zum Trend in der Zukunft werden. Genaues dazu ist im Kapitel 2 erläutert. Sie ermöglichen eine gute Kommunikation zwischen den Hotels und somit einen Vergleich von Kennzahlen und Verbrauchsergebnissen durch E-Kuh.

Außerdem kann E-Kuh in andere Sprachen übersetzt werden, so dass es in ausländischen Hotels erweitert werden kann.

## **7Webzugriff**

Im Rahmen des Projektes wurde auch eine Internetplattform entwickelt, um das EDV-Tool einer möglichst großen Zahl von Benutzern in der Zielgruppe zur Verfügung zu stellen. Dadurch wird den Benutzern ermöglicht, das komplette Tool auf ihren eigenen Rechner zu installieren und dann zu benutzen.

Diese Internetplattform wurde in HTML realisiert, man kann sie mit allen gängigen Browsern betrachten. Bei dieser Lösung können zusätzliche diverse Inhalte zum Thema eingebunden werden – per Mausklick kann sich der User dann durch das Angebot navigieren. Die Datenbank selber ist layouttechnisch an den Gestaltungsrahmen angepasst und kann per Mausklick aus dem Browser entweder in ihrer Gesamtheit gestartet werden oder für einen Download bereitgestellt werden.

Die Website [www.q-mo.de](http://www.q-mo.de) dient an dieser Stelle zum einen als Rahmen, bzw. Präsentationsplattform für dieses komplette Paket, andererseits übernimmt Q-mo.de die Aufgabe, das EDV-Tool zu veröffentlichen. Durch das Einbinden von Logos und entsprechender Hinweise wird sichergestellt, dass die eigentlichen Initiatoren (DBU, IER, DEHOGA, DQS etc.) auch als solche vorgestellt werden.

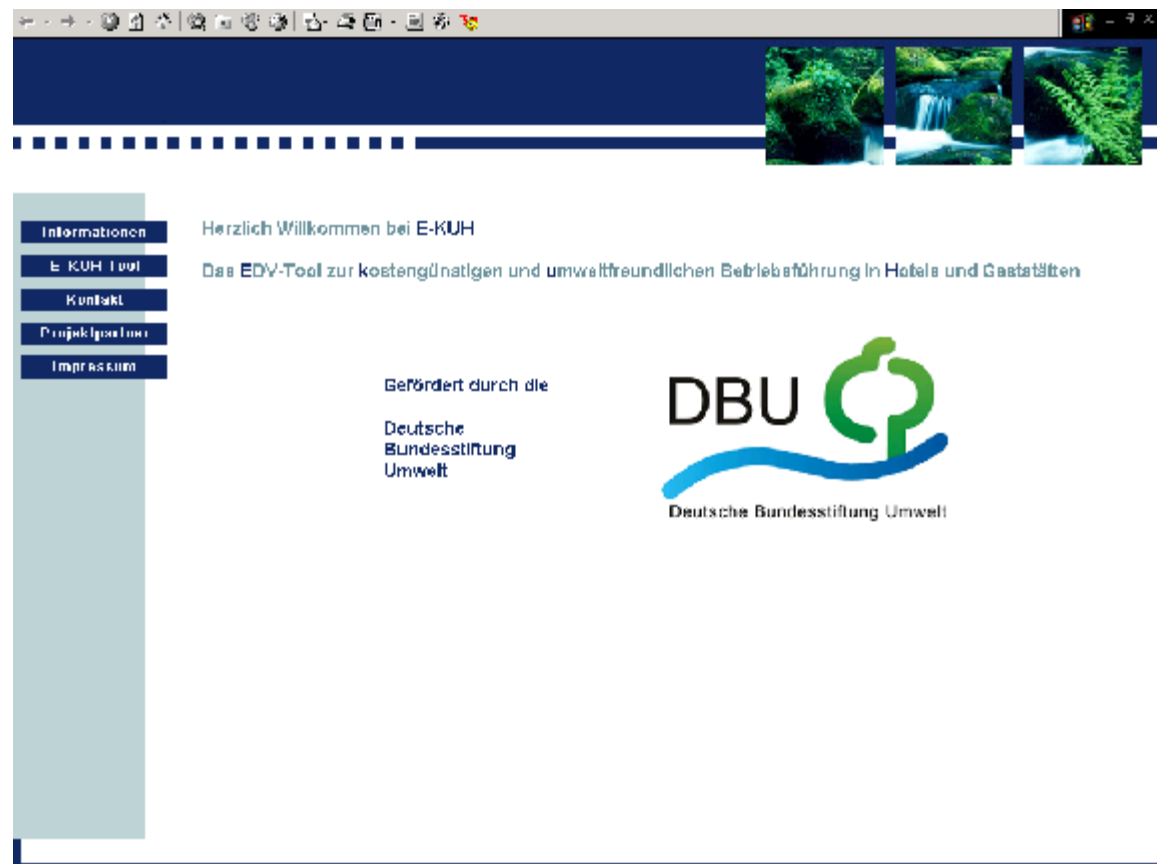
Derzeit umfasst die Internetplattform die folgenden Elemente (vgl. auch beigefügte CD-ROM):

- Startseite
- Informationsseite und Kurzbeschreibung
- Verlinkung zur E-KUH-Datenbank
- Kontakt
- Projektpartner
- Impressum

Auf der Startseite wird insbesondere auf die Förderung durch die DBU verwiesen. Die Informationsseite bietet eine Kurzbeschreibung des EDV-Tools und des Projektes. Über die Verlinkung zur E-KUH-Datenbank wird dem Nutzer die Möglichkeit zum Download geboten. Unter Kontakt wird Hr. Eckardt als Ansprechpartner am IER benannt und unter Projektpartner werden die Partner aufgeführt, einschließlich einer Verlinkung auf deren jeweiligen Internetauftritt. Schließlich werden im Impressum rechtliche Hinweise gegeben, die die Verantwortung der Anbieter beschränkt.

Das für die grafische Aufbereitung erforderliche Bildmaterial wurde sowohl von den

Projektpartnern als auch der Viabono GmbH und dem Naturpark Nördlicher Oberpfälzer Wald e.V. zur Verfügung gestellt.



**Abb. 7-1 Startfenster der Intenetzugang**



Abb.7-2 Vorstellungsf formular von E-KuH

## **8 Datenbankunabhängigkeit**

### **Schnell, zuverlässig, offen**

Datenbanken sind wesentliche Bestandteile einer unternehmensweiten EDV – Komplettlösung. Alle erfassten und aufgezeichneten Informationen werden hier gespeichert und zu einem späteren Zeitpunkt ausgewertet. Die Technologie von E – Kuh steht dafür.

### **Schnelle Transaktionsgeschwindigkeit**

Bei der überaus wichtigen Betriebsdatenerfassung im Produktionsbereich sowie der schnellen Dateneingabe in allen Office – Anwendungen (z.B. der Auftragserfassung) werden hohe Transaktionsgeschwindigkeiten verlangt.

### **Hohe Datensicherheit**

Integriertes Datenbankmanagement mit Transaktionen sichert die volle Wiederherstellbarkeit aller Systemdaten.

### **Skalierbare Lösungen**

Unsere Datenbanklösungen sind skalierbar, vom Kleinbetrieb zum mittelgroßen Hotel. Die Datenbanktechnologie, die geboten wird, passt sowohl zu Unternehmen als auch zu Budget. Abgestufte Lösungen ermöglichen Unternehmenswachstum bei gleichzeitiger Sicherung der Investition.

### **Kosteneffektivität**

Die für die Kunden günstigste Lösung steht im Vordergrund.

## Literaturverzeichnis

- /1/ Kemper A.Eickler.A : Datenbanksysteme – Eine Einführung. 4 Auflage. Oldenbourg (2001)  
ISBN: 3-486-25706-4
  
- /2/ Sven Eckardt & Christoph Weber : E – KUH : EDV – Tpp; für kostengünstige Umweltanalysen in Hotels und Gaststätten (2003)  
Institut für Energiewirtschaft und Rationelle Energieanwendung, Stuttgart
  
- /3/ Silvia Pedote : Maßnahm zur Reduzierung der Umweltbelastungen in Hotels und Gaststättentechnische und ökonomische Analys(2002)  
Institut für Energiewirtschaft und Rationelle Energieanwendung, Stuttgart
  
- /4/ Einführung in PHP und MySQL :  
<http://members.aon.at/heysiflo03/>; aufgerufen am 10.06.2003
  
- /5/ PHP/MySql – Einführung :  
<http://schueler.htldornbirn.vol.at/stmaicla/einf.htm> ; aufgerufen am 10.06.2003
  
- /6/ Kurzbeschreibung des Projekts von Sven Eckardt  
[http://www.ier.uni-stuttgart.de/public/de/organisation/abt/proj\\_desc-deu.php?pid=60](http://www.ier.uni-stuttgart.de/public/de/organisation/abt/proj_desc-deu.php?pid=60)
  
- /7/ Leitfaden zum EG-Umwelt-Audit-System für Hotel- und Gaststättenbetriebe  
herausgegeben von Bundesministerium für Umwelt, Naturschutz und Reaktorsicherheit,  
Umweltbundesamt
  
- /8/ Eigenschaft von MySQL:  
<http://www.mysql.com/doc/de/Features.html>
  
- /9/ Einführung von PHP  
<http://www.php-homepage.de/manual/intro-whatcando.php>

/10/ Grundlagen der Datenbanken I 4. neubearbeitete Auflage Skriptum zur gleichnamigen Vorlesung an der Universität Gesamthochschule Kassel

## Anhang

### Energieverbrauch pro Übernachtung

```

SELECT a.jahr,round(sum(a.menge*b.faktor/x.menge),2) AS kennzahl
FROM W_Daten2 a,def_einheit b ,F_Kennzahlen_Gueter2 c,W_Daten2 x WHERE
a.id_zeitreihe IN
  (SELECT id_zeitreihe
   FROM F_Kennzahlen_Gueter2
   WHERE id_formular=1)
AND a.id_zeitreihe=c.id_zeitreihe
   AND c.id_einheit=b.id_einheit
   AND b.einheit_typ_id=1
   AND x.jahr=a.jahr
   AND x.id_zeitreihe=5
GROUP BY a.jahr ORDER BY a.jahr desc

```

### Abfallmenge pro Übernachtung

```

SELECT a.jahr,round(sum(a.menge*b.faktor/x.menge),2) AS kennzahl
FROM W_Daten2 a,def_einheit b ,F_Kennzahlen_Gueter2 c,W_Daten2 x WHERE
a.id_zeitreihe IN
  (SELECT id_zeitreihe
   FROM F_Kennzahlen_Gueter2
   WHERE id_formular=2)
AND a.id_zeitreihe=c.id_zeitreihe
   AND c.id_einheit=b.id_einheit
   AND b.einheit_typ_id=2
   AND x.jahr=a.jahr
AND x.id_zeitreihe=5

```

```

GROUP BY a.jahr ORDER BY a.jahr desc

```

### Wasserverbrauch pro Übernachtung

```

SELECT a.jahr,round(sum(a.menge*b.faktor/x.menge),2) AS kennzahl
FROM W_Daten2 a,def_einheit b ,F_Kennzahlen_Gueter2 c,W_Daten2 x
WHERE a.id_zeitreihe IN
  (SELECT id_zeitreihe
   FROM F_Kennzahlen_Gueter2
   WHERE id_formular=3)
AND a.id_zeitreihe=c.id_zeitreihe
   AND c.id_einheit=b.id_einheit
   AND b.einheit_typ_id=2
   AND x.jahr=a.jahr
   AND x.id_zeitreihe=5
GROUP BY a.jahr ORDER BY a.jahr desc

```

### Abfrage\_Energie

**SELECT**

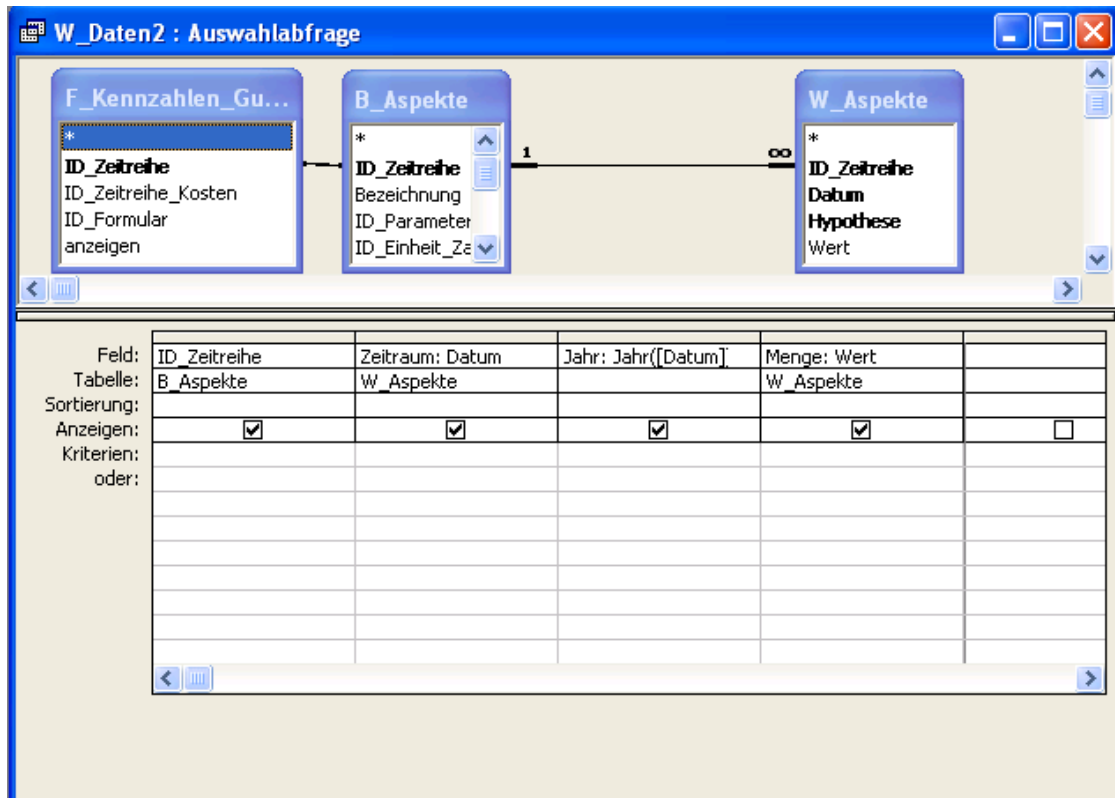
B\_Aspekte.Bezeichnung,  
 W\_Aspekte.Datum,  
 W\_Aspekte.Wert,  
 def\_einheit.einheit,  
 def\_einheit.faktor,  
 [W\_Aspekte.Wert]\*[def\_einheit.faktor] **AS** kWh\_Umrechnung  
**FROM** ((B\_Aspekte **INNER JOIN** F\_Kennzahlen\_Gueter  
**ON** B\_Aspekte.ID\_Zeitreihe = F\_Kennzahlen\_Gueter.ID\_Zeitreihe)  
**INNER JOIN** W\_Aspekte  
**ON** (B\_Aspekte.ID\_Zeitreihe=W\_Aspekte.ID\_Zeitreihe)  
**AND** (F\_Kennzahlen\_Gueter.ID\_Zeitreihe = W\_Aspekte.ID\_Zeitreihe))  
**INNER JOIN** def\_einheit  
**ON** B\_Aspekte.ID\_Einheit\_Zaehler = def\_einheit.id\_einheit  
**WHERE** (((def\_einheit.einheit\_typ\_id)=1));  
**ORDER BY** W\_Aspekte.Datum;

**Abfrage\_Wasser**

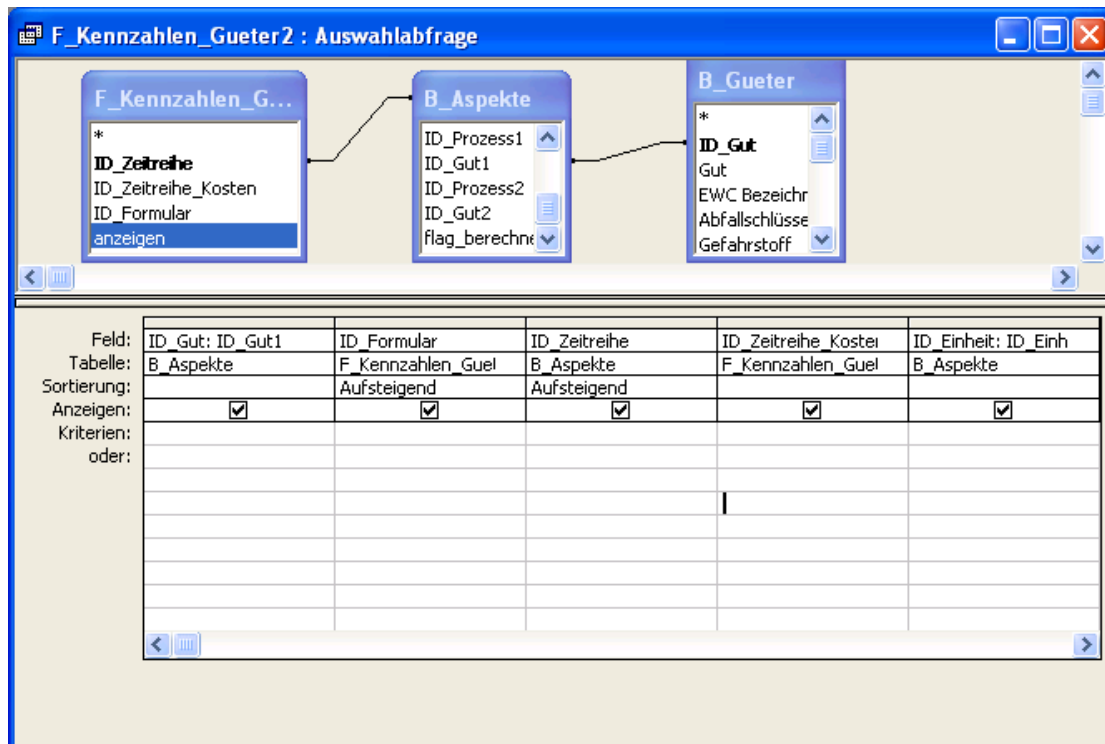
**SELECT** B\_Aspekte.Bezeichnung, W\_Aspekte.Datum, W\_Aspekte.Wert  
**FROM** (W\_Aspekte **INNER JOIN** B\_Aspekte **ON**  
 W\_Aspekte.ID\_Zeitreihe = B\_Aspekte.ID\_Zeitreihe)  
**INNER JOIN** F\_Kennzahlen\_Gueter **ON**  
 W\_Aspekte.ID\_Zeitreihe =F\_Kennzahlen\_Gueter.ID\_Zeitreihe  
**WHERE** (((F\_Kennzahlen\_Gueter.ID\_Formular)=3))  
**ORDER BY** W\_Aspekte.Datum;

**Abfrage\_Abfall**

**SELECT** B\_Aspekte.Bezeichnung, W\_Aspekte.Datum, W\_Aspekte.Wert  
**FROM** (B\_Aspekte **INNER JOIN** W\_Aspekte **ON** \_  
 B\_Aspekte.ID\_Zeitreihe = W\_Aspekte.ID\_Zeitreihe)  
**INNER JOIN** F\_Kennzahlen\_Gueter **ON**  
 B\_Aspekte.ID\_Zeitreihe =F\_Kennzahlen\_Gueter.ID\_Zeitreihe  
**WHERE** (((F\_Kennzahlen\_Gueter.ID\_Formular)=2))  
**ORDER BY** W\_Aspekte.Datum;



Abfrage w\_Daten2



## Abfrage F\_Kennzahlen\_Gueter2

### 'Ausgabe der selbst ermittelten Kennzahlen

```
SQL = "SELECT * FROM b_kennzahl"
```

```
Set tab1 = CurrentDb.OpenRecordset(SQL, dbOpenSnapshot)
```

```
While tab1.EOF = False
```

```
    i = tab1!ID
```

```
    Forms(f("text_wert_" & i) = tab1!Wert
```

```
    Forms(f("einheit_zaeher_" & i) = tab1("einheit_zaeher")
```

```
    Forms(f("einheit_nenner_" & i) = tab1("einheit_nenner")
```

```
    Forms(f("linie_name_" & i).Visible = True
```

```
    Forms(f("linie_einheit_" & i).Visible = True
```

```
    Forms(f("text_nenner_" & i) = tab1!nenner
```

```
    Forms(f("text_zaeher_" & i) = tab1!zaehler
```

```
    Forms(f("jahr_" & i) = "(" & tab1!jahr & ")"
```

```
tab1.MoveNext
```

```
Wend
End Sub
```

```
Sub berechne()
```

```
Dim f, tab1, tab2, SQL
```

```
f = "frm_kennzahl_definiere"
```

```
SQL = "select menge, c.einheit " & _
      "from W_Daten2 a,F_Kennzahlen_Gueter2 d,def_einheit c " & _
      "WHERE d.id_einheit=c.id_einheit and a.id_zeitreihe=d.id_zeitreihe and " & _
      "anzeigen=-1 and jahr=" & Forms(f)("combo_jahr") & _
      " AND a.id_zeitreihe =" & Forms(f)("combo_zaeher")
Set tab1 = CurrentDb.OpenRecordset(SQL, dbOpenSnapshot)
```

```
SQL = "select menge,c.einheit " & _
      "from W_Daten2 a,F_Kennzahlen_Gueter2 d,def_einheit c " & _
      "WHERE d.id_einheit=c.id_einheit and a.id_zeitreihe=d.id_zeitreihe and " & _
      "anzeigen=-1 and jahr=" & Forms(f)("combo_jahr") & _
      " AND a.id_zeitreihe =" & Forms(f)("combo_nenner")
Set tab2 = CurrentDb.OpenRecordset(SQL, dbOpenSnapshot)
```

```
Forms(f)("text_wert").Value = Round(tab1!menge / tab2!menge, 2)
```

```
Forms(f)("einheit_zaeher").Value = tab1!Einheit
```

```
Forms(f)("einheit_nenner").Value = tab2!Einheit
```

```
End Sub
```

```
Sub speichern(g)
```

```
Dim f, tab2, neu, i, tab1, SQL
```

```
f = "frm_kennzahlenberechnen"
```

```
neu = False
```

```
i = 1
```

```
While neu = False
```

```
SQL = "SELECT * FROM b_kennzahl WHERE id=" & i
```

```
Set tab1 = CurrentDb.OpenRecordset(SQL)
```

```
If tab1.EOF = True Then
```

```
    neu = True
```

```
Else: i = i + 1
```

```
End If
```

```
Wend
```

```
Call neue_felder(f, i)
```

```
Set tab1 = CurrentDb.OpenRecordset("b_kennzahl", dbOpenTable)
```

```
tab1.AddNew
```

```

tab1!ID = i

SQL = "SELECT bezeichnung FROM b_aspekte WHERE id_zeitreihe=" & Forms(g)
("combo_nenner")

Set tab2 = CurrentDb.OpenRecordset(SQL, dbOpenSnapshot)
Forms(f)("text_nenner_" & i) = tab2!Bezeichnung 'gut
tab1!nenner = tab2!Bezeichnung 'gut

SQL = "SELECT bezeichnung FROM b_aspekte WHERE id_zeitreihe=" & Forms(g)
("combo_zaeher")
Set tab2 = CurrentDb.OpenRecordset(SQL, dbOpenSnapshot)
Forms(f)("text_zaeher_" & i) = tab2!Bezeichnung 'gut
tab1!zaehler = tab2!Bezeichnung 'gut

Forms(f)("text_wert_" & i) = Forms(g)("text_wert")
Forms(f)("einheit_zaeher_" & i) = Forms(g)("einheit_zaeher")
Forms(f)("einheit_nenner_" & i) = Forms(g)("einheit_nenner")
Forms(f)("linie_name_" & i).Visible = True
Forms(f)("linie_einheit_" & i).Visible = True
Forms(f)("jahr_" & i) = "(" & Forms(g)("combo_jahr") & ")"
Forms(f).Refresh

tab1!einheit_zaeher = Forms(g)("einheit_zaeher")
tab1!einheit_nenner = Forms(g)("einheit_nenner")
tab1!Wert = Forms(g)("text_wert")
tab1!jahr = Forms(g)("combo_jahr")
tab1.Update
DoCmd.Close acForm, g
End Sub

```

### **Abfragen von Wirtschaftlichkeitsrechnung**

#### **Abfrage H\_Wirtschaftlichkeit\_Eingabedaten**

```

SELECT B_Aspekte.flag_berechnet,
        H_Wirtschaftlichkeit_Eingabedaten.ID_Zeitreihe,
        B_Aspekte.Bezeichnung, W_Aspekte.Datum, W_Aspekte.Wert
FROM (H_Wirtschaftlichkeit_Eingabedaten INNER JOIN B_Aspekte ON
        H_Wirtschaftlichkeit_Eingabedaten.ID_Zeitreihe = B_Aspekte.ID_Zeitreihe)
LEFT JOIN W_Aspekte ON B_Aspekte.ID_Zeitreihe = W_Aspekte.
        ID_Zeitreihe
ORDER BY B_Aspekte.flag_berechnet DESC ,
        H_Wirtschaftlichkeit_Eingabedaten.ID_Zeitreihe;

```

#### **Abfrage H\_Wirtschaftlichkeit\_Eingabedaten0**

```

DELETE H_Wirtschaftlichkeit_Eingabedaten.*
FROM H_Wirtschaftlichkeit_Eingabedaten;

```

**Abfrage H\_Kosten\_jährlich**

```

SELECT H_Wirt_Mengen.ID_Massnahme, Sum([Preis]*[vorher]) AS
    Kosten_vorher,
    Sum([preis]*[nachher]) AS Kosten_nachher,
    [Kosten_vorher]-[Kosten_nachher] AS
    Differenz_Kosten, Sum(H_Wirt_Investg.Investitionskosten) AS
    Invest_kosten,
    Sum(H_Wirt_Investg.j_Betrkosten) AS Sj_Betrkosten,
    [Invest_kosten]/([kosten_vorher]-[kosten_nachher]-[Sj_betrkosten]) AS
    Amortzeit,
    Sum(H_Wirt_Investg.annu) AS Sannu,
    [Sannu]+[Kosten_vorher]-[Kosten_nachher] AS
    Annuität_ges, Min(H_Wirt_Investg.Nutzungsdauer) AS MinNutzD,
    Avg(H_Wirt_Investg.kapzins0) AS Mkapzins,
    Sum(IIf(IsNull([barw]),0,[Barw])) AS
    SBarw, -[SBarw]+PV([Mkapzins],[MinNutzD],[Kosten_nachher]
    -[Kosten_vorher]) AS Barwert
FROM H_Wirt_Mengen LEFT JOIN H_Wirt_Investg ON
    H_Wirt_Mengen.ID_Massnahme = H_Wirt_Investg.ID_Massnahme
GROUP BY H_Wirt_Mengen.ID_Massnahme;

```

**Abfrage H\_Massnahmen\_Gueter\_Rechengang**

```

SELECT a.ID_Massnahme, c.ID_Gut, b.Status, d.Gut, b.id_prozess, e.ID_Zeitreihe, f.
    ID_Berechnung, e.ID_Parameter, f.Nr_Rechenschritt
FROM (((B_Massnahmen AS a INNER JOIN Z_Massnahme_Prozess AS b ON a.
    ID_Massnahme = b.ID_Massnahme) INNER JOIN Z_Prozesse_Gueter AS c
ON b.id_prozess = c.ID_Prozess) INNER JOIN B_Gueter AS d ON c.
    ID_Gut = d.ID_Gut) INNER JOIN B_Aspekte AS e ON (d.ID_Gut = e.
    ID_Gut1) AND (c.ID_Prozess = e.ID_Prozess1)) INNER JOIN
    B_Rechengang AS f ON e.ID_Zeitreihe = f.ID_Ergebnis
GROUP BY a.ID_Massnahme, c.ID_Gut, b.Status, d.Gut, b.id_prozess, e.ID_Zeitreihe,
    f.ID_Berechnung, e.ID_Parameter, f.Nr_Rechenschritt, d.Investitionsgut
HAVING (((a.ID_Massnahme)=[Nr_massnahme]) AND ((b.Status)=1 Or (b.Status)=-1)
AND ((e.ID_Parameter)=1) AND ((d.Investitionsgut)=No))
ORDER BY a.ID_Massnahme, c.ID_Gut, b.Status;

```

**Abfrage H\_Rechengang\_Rechenschritte**

```

SELECT B_Rechengang.ID_Berechnung,
    B_Rechengang.Nr_Rechenschritt,
    B_Rechengang.ID_Operator, B_Rechengang.ID_Ergebnis
FROM H_Massnahmen_Gueter_Rechengang INNER JOIN B_Rechengang ON
    H_Massnahmen_Gueter_Rechengang.ID_Berechnung = B_Rechengang.
    ID_Berechnung
WHERE (((B_Rechengang.Nr_Rechenschritt)<=[H_Massnahmen_Gueter_Rechengang]!
    [Nr_Rechenschritt]))
ORDER BY B_Rechengang.ID_Berechnung, B_Rechengang.Nr_Rechenschritt;

```

**Abfrage H\_Rechengang\_Rechenschritte\_Operand**

**SELECT** a.ID\_Berechnung, b.Nr\_Rechenschritt,  
 b.Nr\_Operand, b.ID\_Operand, c.ID\_Zeitreihe, c.Bezeichnung, d.Hypothese,  
 Last(d.Wert) AS relWert  
**FROM** ((H\_Massnahmen\_Gueter\_Rechengang **AS** a **INNER JOIN** [B\_Rechengang  
 Operanden] **AS** b **ON** a.ID\_Berechnung = b.ID\_Berechnung) **INNER JOIN**  
 B\_Aspekte **AS** c **ON** b.ID\_Operand = c.ID\_Zeitreihe) **LEFT JOIN** W\_Aspekte  
**AS** d **ON** c.ID\_Zeitreihe = d.ID\_Zeitreihe  
**GROUP BY** a.ID\_Berechnung, b.Nr\_Rechenschritt, b.Nr\_Operand, b.ID\_Operand, c.  
 ID\_Zeitreihe, c.Bezeichnung, d.Hypothese;

**Abfrage H\_Wirt\_Gueter\_Mengen0**

**DELETE** C\_Wirt\_Investgueter.\*  
**FROM** C\_Wirt\_Investgueter;

**Abfrage H\_Wirt\_Gueter\_Mengen1**

**INSERT INTO** C\_Wirt\_Gueter\_Mengen ( ID\_Massnahme, ID\_Gut, Gut, vorher,  
 nachher )  
**SELECT** H\_Massnahmen\_Gueter\_Rechengang.ID\_Massnahme,  
 H\_Massnahmen\_Gueter\_Rechengang.ID\_Gut,  
 H\_Massnahmen\_Gueter\_Rechengang.Gut, 0 AS vorher, 0 AS nachher  
**FROM** H\_Massnahmen\_Gueter\_Rechengang  
**GROUP BY** H\_Massnahmen\_Gueter\_Rechengang.ID\_Massnahme,  
 H\_Massnahmen\_Gueter\_Rechengang.ID\_Gut,  
 H\_Massnahmen\_Gueter\_Rechengang.Gut, 0, 0;

**Abfrage H\_Wirt\_Gueter\_Mengen2**

**SELECT** C\_Wirt\_Gueter\_Mengen.ID\_Massnahme,  
 C\_Wirt\_Gueter\_Mengen.ID\_Gut,  
 C\_Wirt\_Gueter\_Mengen.Gut, C\_Wirt\_Gueter\_Mengen.ID\_ZR\_vorher,  
 C\_Wirt\_Gueter\_Mengen.ID\_ZR\_nachher, C\_Wirt\_Gueter\_Mengen.vorher,  
 C\_Wirt\_Gueter\_Mengen.nachher  
**FROM** C\_Wirt\_Gueter\_Mengen;

**Abfrage H\_Massnahmen\_InvestGueter\_Rechengang**

**SELECT** a.ID\_Massnahme, c.ID\_Gut, b.Status, d.Gut, b.id\_prozess, e.ID\_Gut1, e.  
 ID\_Zeitreihe, e.ID\_Parameter, f.ID\_Berechnung, f.Nr\_Rechenschritt  
**FROM** (((B\_Massnahmen **AS** a **INNER JOIN** Z\_Massnahme\_Prozess **AS** b **ON** a.  
 ID\_Massnahme = b.ID\_Massnahme) **INNER JOIN** Z\_Prozesse\_Gueter **AS** c  
**ON** b.id\_prozess = c.ID\_Prozess) **INNER JOIN** B\_Gueter **AS** d **ON** c.  
 ID\_Gut = d.ID\_Gut) **INNER JOIN** B\_Aspekte **AS** e **ON** (c.ID\_Prozess = e.  
 ID\_Prozess1) **AND** (d.ID\_Gut = e.ID\_Gut1)) **INNER JOIN** B\_Rechengang  
**AS** f **ON** e.ID\_Zeitreihe = f.ID\_Ergebnis  
**GROUP BY** a.ID\_Massnahme, c.ID\_Gut, b.Status, d.Gut, b.id\_prozess, e.ID\_Gut1, e.  
 ID\_Zeitreihe, e.ID\_Parameter, f.ID\_Berechnung, f.Nr\_Rechenschritt, d.  
 Investitionsgut  
**HAVING** (((a.ID\_Massnahme)=[Nr\_massnahme]) **AND** ((b.Status)=1 **Or** (b.Status)=-1)

**AND** ((e.ID\_Parameter)=2 **Or** (e.ID\_Parameter)=28 **Or** (e.ID\_Parameter)=14 **Or** (e.ID\_Parameter)=29 **Or** (e.ID\_Parameter)=3) **AND** ((d.Investitionsgut=Yes))  
**ORDER BY** a.ID\_Massnahme, c.ID\_Gut, b.Status;

#### Abfrage H\_Rechengang\_Rechenschritte\_InvestGueter

**SELECT** B\_Rechengang.ID\_Berechnung,  
 B\_Rechengang.Nr\_Rechenschritt,  
 B\_Rechengang.ID\_Operator, B\_Rechengang.ID\_Ergebnis  
**FROM** B\_Rechengang **INNER JOIN** H\_Massnahmen\_InvestGueter\_Rechengang **ON**  
 B\_Rechengang.ID\_Berechnung =  
 H\_Massnahmen\_InvestGueter\_Rechengang.ID\_Berechnung  
**WHERE** ( ( ( B \_ R e c h e n g a n g . N r \_ R e c h e n s c h r i t t )  
 <=[H\_Massnahmen\_InvestGueter\_Rechengang][Nr\_Rechenschritt]))  
**ORDER BY** B\_Rechengang.ID\_Berechnung, B\_Rechengang.Nr\_Rechenschritt;

#### Abfrage H\_Rechengang\_Rechenschritte\_Operand\_InvestGueter

**SELECT** a.ID\_Berechnung, b.Nr\_Rechenschritt, b.Nr\_Operand, b.ID\_Operand,  
 c.ID\_Zeitreihe, c.Bezeichnung, d.Hypothese, Last(d.Wert) AS relWert  
**FROM** (B\_Aspekte **AS** c **INNER JOIN** (H\_Massnahmen\_InvestGueter\_Rechengang **AS**  
 a **INNER JOIN** [B\_Rechengang Operanden] **AS** b **ON** a.ID\_Berechnung = b.  
 ID\_Berechnung) **ON** c.ID\_Zeitreihe = b.ID\_Operand) **LEFT JOIN** W\_Aspekte  
**AS** d **ON** c.ID\_Zeitreihe = d.ID\_Zeitreihe  
**GROUP BY** a.ID\_Berechnung, b.Nr\_Rechenschritt, b.Nr\_Operand, b.ID\_Operand,  
 c.ID\_Zeitreihe, c.Bezeichnung, d.Hypothese;

#### Abfrage H\_Wirt\_InvestGueter0

**DELETE** C\_Wirt\_Investgueter.\*  
**FROM** C\_Wirt\_Investgueter;

#### Abfrage H\_Wirt\_InvestGueter1

**INSERT INTO** C\_Wirt\_Investgueter ( ID\_Massnahme, ID\_Gut, Gut, Anzahl,  
 Nutzungsdauer, Investitionskosten, [jährliche Betriebskosten], [spezifischer  
 Kapitalzinssatz] )  
**SELECT** H\_Massnahmen\_InvestGueter.ID\_Massnahme,  
 H\_Massnahmen\_InvestGueter.ID\_Gut,  
 H\_Massnahmen\_InvestGueter.Gut, Last(IIf(IsNull([W\_aspekte\_2][Wert]),1,  
 [W\_aspekte\_2][wert])) **AS** Anzahl, Last(W\_Aspekte\_1.Wert) **AS**  
 Nutzungsdauer,  
 Last([w\_aspekte\_1][Wert]\*IIf(IsNull([w\_aspekte\_2][wert]),1,[w\_aspekte\_2][  
 wert])) **AS** Investkost,  
 Last(IIf(IsNull([w\_aspekte\_2][wert]),1,[w\_aspekte\_2][wert])\*[w\_aspekte\_3][  
 Wert]) **AS** jBetrKost, Last(W\_Aspekte\_4.Wert) **AS** spezZins  
**FROM** (H\_Massnahmen\_InvestGueter **INNER JOIN** (((((B\_Aspekte  
**LEFT JOIN** (B\_Aspekte **AS** B\_Aspekte\_1  
**LEFT JOIN** W\_Aspekte **AS** W\_Aspekte\_1 **ON** B\_Aspekte\_1.ID\_Zeitreihe =

W\_Aspekte\_1.ID\_Zeitreihe) **ON** B\_Aspekte.ID\_Gut1 =  
 B\_Aspekte\_1.ID\_Gut1) **LEFT JOIN** (B\_Aspekte **AS** B\_Aspekte\_2  
**LEFT JOIN** W\_Aspekte **AS** W\_Aspekte\_2 **ON** B\_Aspekte\_2.ID\_Zeitreihe =  
 W\_Aspekte\_2.ID\_Zeitreihe) **ON** B\_Aspekte.ID\_Gut1 =  
 B\_Aspekte\_2.ID\_Gut1) **LEFT JOIN** B\_Aspekte **AS** B\_Aspekte\_3 **ON**  
 B\_Aspekte.ID\_Gut1 = B\_Aspekte\_3.ID\_Gut1)  
**LEFT JOIN** W\_Aspekte **AS** W\_Aspekte\_3 **ON** B\_Aspekte\_3.ID\_Zeitreihe =  
 W\_Aspekte\_3.ID\_Zeitreihe)  
**LEFT JOIN** (B\_Aspekte **AS** B\_Aspekte\_4  
**LEFT JOIN** W\_Aspekte **AS** W\_Aspekte\_4 **ON** B\_Aspekte\_4.ID\_Zeitreihe =  
 W\_Aspekte\_4.ID\_Zeitreihe) **ON** B\_Aspekte.ID\_Gut1 =  
 B\_Aspekte\_4.ID\_Gut1) **ON** H\_Massnahmen\_InvestGueter.ID\_Gut =  
 B\_Aspekte.ID\_Gut1)  
**LEFT JOIN** W\_Aspekte **ON** B\_Aspekte.ID\_Zeitreihe =  
 W\_Aspekte.ID\_Zeitreihe  
**GROUP BY** H\_Massnahmen\_InvestGueter.ID\_Massnahme,  
 H\_Massnahmen\_InvestGueter.ID\_Gut,  
 H\_Massnahmen\_InvestGueter.Gut, H\_Massnahmen\_InvestGueter.Status,  
 B\_Aspekte.ID\_Parameter, B\_Aspekte\_1.ID\_Prozess1,  
 B\_Aspekte\_1.ID\_Parameter,  
 B\_Aspekte.ID\_Prozess1, B\_Aspekte\_2.ID\_Parameter,  
 B\_Aspekte\_3.ID\_Prozess1,  
 B\_Aspekte\_3.ID\_Parameter, B\_Aspekte\_4.ID\_Prozess1,  
 B\_Aspekte\_4.ID\_Parameter  
**HAVING** (((H\_Massnahmen\_InvestGueter.Status)=1) **AND**  
 ((B\_Aspekte.ID\_Parameter)=28  
**Or** (B\_Aspekte.ID\_Parameter)=2) **AND** ((B\_Aspekte\_1.ID\_Prozess1)=-1  
**And**(B\_Aspekte\_1.ID\_Prozess1)=-1) **AND**  
 ((B\_Aspekte\_1.ID\_Parameter)=20) **AND** ((B\_Aspekte.ID\_Prozess1)=-1)  
**AND** ((B\_Aspekte\_2.ID\_Parameter)=3) **AND** .  
 ((B\_Aspekte\_3.ID\_Prozess1)=-1) **AND** ((B\_Aspekte\_3.ID\_Parameter)=31)  
**AND** ((B\_Aspekte\_4.ID\_Prozess1)=-1) **AND**  
 ((B\_Aspekte\_4.ID\_Parameter)=29));

### Abfrage H\_Wirt\_InvestGueter2

**UPDATE** (C\_Wirt\_Investgueter **INNER JOIN** B\_Aspekte **ON**  
 C\_Wirt\_Investgueter.ID\_Gut =  
 B\_Aspekte.ID\_Gut1) **INNER JOIN** W\_Aspekte **ON**  
 B\_Aspekte.ID\_Zeitreihe =  
 W\_Aspekte.ID\_Zeitreihe **SET** C\_Wirt\_Investgueter.Anzahl =  
 [W\_aspekte]![Wert]  
**WHERE** (((B\_Aspekte.ID\_Parameter)=3) **AND** ((B\_Aspekte.ID\_Prozess1)=-1 **Or**  
 (B\_Aspekte.ID\_Prozess1)=[C\_Wirt\_Investgueter]![ID\_Prozess]));

### Abfrage H\_Wirt\_InvestGueter3

**UPDATE** (C\_Wirt\_Investgueter **INNER JOIN** B\_Aspekte **ON**  
 C\_Wirt\_Investgueter.ID\_Gut =  
 B\_Aspekte.ID\_Gut1) **INNER JOIN** W\_Aspekte **ON**

B\_Aspekte.ID\_Zeitreihe =  
W\_Aspekte.ID\_Zeitreihe **SET** C\_Wirt\_Investgueter.Nutzungsdauer =  
[W\_aspekte]![Wert]  
**WHERE** (((B\_Aspekte.ID\_Parameter)=20) **AND** ((B\_Aspekte.ID\_Prozess1)=-1 **Or**  
(B\_Aspekte.ID\_Prozess1)=[C\_Wirt\_Investgueter]![ID\_Prozess]));

#### 4.2.2.3.1.17 Abfrage H\_Wirt\_InvestGueter4

**UPDATE** (C\_Wirt\_Investgueter **INNER JOIN** B\_Aspekte **ON**  
C\_Wirt\_Investgueter.ID\_Gut =  
B\_Aspekte.ID\_Gut1) **INNER JOIN** W\_Aspekte **ON**  
B\_Aspekte.ID\_Zeitreihe =  
W\_Aspekte.ID\_Zeitreihe **SET** C\_Wirt\_Investgueter.[spezifischer  
Kapitalzinssatz] = [W\_aspekte]![Wert]  
**WHERE** (((B\_Aspekte.ID\_Parameter)=29) **AND** ((B\_Aspekte.ID\_Prozess1)=-1 **Or**  
(B\_Aspekte.ID\_Prozess1)=[C\_Wirt\_Investgueter]![ID\_Prozess]));

#### Abfrage H\_Wirt\_InvestGueter5

**UPDATE** (C\_Wirt\_Investgueter **INNER JOIN** B\_Aspekte **ON**  
C\_Wirt\_Investgueter.ID\_Gut =  
B\_Aspekte.ID\_Gut1) **INNER JOIN** W\_Aspekte **ON**  
B\_Aspekte.ID\_Zeitreihe =  
W\_Aspekte.ID\_Zeitreihe **SET** C\_Wirt\_Investgueter.[jährliche  
Betriebskosten] = [W\_aspekte]![Wert]  
**WHERE** (((B\_Aspekte.ID\_Parameter)=31) **AND** ((B\_Aspekte.ID\_Prozess1)=-1 **Or**  
(B\_Aspekte.ID\_Prozess1)=[C\_Wirt\_Investgueter]![ID\_Prozess]));

#### Abfrage H\_Wirt\_InvestGueter6

**UPDATE** (C\_Wirt\_Investgueter **INNER JOIN** B\_Aspekte **ON**  
C\_Wirt\_Investgueter.ID\_Gut =  
B\_Aspekte.ID\_Gut1) **INNER JOIN** W\_Aspekte **ON**  
B\_Aspekte.ID\_Zeitreihe =  
W\_Aspekte.ID\_Zeitreihe **SET** C\_Wirt\_Investgueter.Investitionskosten =  
[W\_aspekte]![Wert]\*[C\_Wirt\_investgueter]![Anzahl]  
**WHERE** (((B\_Aspekte.ID\_Parameter)=2 **Or** (B\_Aspekte.ID\_Parameter)=28) **AND** .  
((B\_Aspekte.ID\_Prozess1)=-1 **Or**  
(B\_Aspekte.ID\_Prozess1)=[C\_Wirt\_Investgueter]![ID\_Prozess]));

#### Abfrage H\_Zeitreihe\_Rechengang

**SELECT** B\_Rechengang.ID\_Ergebnis,  
B\_Rechengang.ID\_Berechnung, B\_Rechengang.Nr\_Rechenschritt  
**FROM** B\_Rechengang  
**GROUP BY** B\_Rechengang.ID\_Ergebnis,  
B\_Rechengang.ID\_Berechnung, B\_Rechengang.Nr\_Rechenschritt  
**HAVING** (((B\_Rechengang.ID\_Ergebnis)=[crit\_erg]));

#### Abfrage H\_Rechengang\_Rechenschritte2

**SELECT** B\_Rechengang.ID\_Berechnung,  
B\_Rechengang.Nr\_Rechenschritt,  
B\_Rechengang.ID\_Operator, B\_Rechengang.ID\_Ergebnis

```

FROM B_Rechengang
WHERE (((B_Rechengang.ID_Berechnung)=[crit_rech]))
ORDER BY B_Rechengang.ID_Berechnung, B_Rechengang.Nr_Rechenschritt;

```

#### **Abfrage H\_Rechengang\_Rechenschritte\_Operand2**

```

SELECT [B_Rechengang Operanden].ID_Berechnung,
        [B_Rechengang Operanden].Nr_Rechenschritt,
        [B_Rechengang Operanden].Nr_Operand,
        [B_Rechengang Operanden].ID_Operand,
        B_Aspekte.ID_Zeitreihe, B_Aspekte.Bezeichnung,
        W_Aspekte.Hypothese, Last(W_Aspekte.Wert) AS relWert
FROM (B_Aspekte INNER JOIN [B_Rechengang Operanden]
ON B_Aspekte.ID_Zeitreihe = [B_Rechengang Operanden].ID_Operand)
INNER JOIN W_Aspekte ON B_Aspekte.ID_Zeitreihe =
        W_Aspekte.ID_Zeitreihe
GROUP BY [B_Rechengang Operanden].ID_Berechnung,
        [B_Rechengang Operanden].Nr_Rechenschritt,
        [B_Rechengang Operanden].Nr_Operand,
        [B_Rechengang Operanden].ID_Operand,
        B_Aspekte.ID_Zeitreihe, B_Aspekte.Bezeichnung, W_Aspekte.Hypothese
HAVING ((([B_Rechengang Operanden].ID_Berechnung)=[crit_rech]));

```

#### **Code von Wirtschaftlichkeitsrechnung**

##### **Hauptprogramm: Wirtschaftlichkeit**

Option Compare Database

Option Explicit

Sub open\_WiRe(ID\_Massnahme, ID\_Umweltauswirkung)

' öffnen Formular frm\_Wirtschaftlichkeitsrechnung1 und Durchführen der entsprechenden Rechnungen

Dim frm, mult, db, query\_del\_input, reccount, Visib, flag\_gesamt

```
frm = "frm_Wirtschaftlichkeitsrechnung1"
```

```
DoCmd.OpenForm frm
```

```
Forms(frm).ID_Umweltauswirkung = ID_Umweltauswirkung
```

```
Forms(frm).ID_Massnahme = ID_Massnahme
```

```
mult = 1440 / 2.54 ' Umrechnungsfaktor cm in Twips (VB Standardeinheit)
```

```
Set db = CurrentDb()
```

```
Set query_del_input = db.QueryDefs("H_Wirtschaftlichkeit_Eingabedaten0")
```

```
query_del_input.Execute
```

```
' Betrachtete Massnahme
```

```
Call Rechnen(ID_Massnahme)
```

```
flag_gesamt = True
```

```
Call Rechnen_InvestGueter(ID_Massnahme, flag_gesamt)
```

```
Forms(frm).RecordSource = "H_Kosten_jährlich"
```

```
Forms(frm).Requery
```

```

If Forms(frm).Recordset.RecordCount > 0 Then
    reccount = Forms(frm).sub
        Frm_Wirt_Investg.Form.Recordset.RecordCount

    If reccount = 0 Then
        Forms(frm).InsideHeight = 8.1 * mult
    Else
        Visib = Forms(frm).txt_annu.Visible
        If Visib = True Then
            Forms(frm).InsideHeight = 14.5 * mult
        Else
            Forms(frm).InsideHeight = 12 * mult
        End If
    End If
End If

End Sub

```

### **Programme: Wirtschaftlichkeit**

#### **Sub Rechnen()**

```

Sub Rechnen(ID_Massnahme)
Dim db, calc, exog, steps, oper, mengen, mengen2, tab_res
Dim rec_calc, rec_exog, rec_steps, rec_oper, rec_mengen, rec_res
Dim ID_Rechengang, Nr_Rechenschritt, ID_Massnahme, i, res, status, id_gut
Dim ID_Zeitreihe

```

#### ' Definition relevante Abfragen

```

Set db = CurrentDb()
Set calc = db.QueryDefs("H_Massnahmen_Gueter_Rechengang")
Set steps = db.QueryDefs("H_Rechengang_Rechenschritte")
Set oper = db.QueryDefs("H_Rechengang_Rechenschritte_Operand")

```

#### ' Ermittlung betrachtete Massnahme

```

ID_Massnahme = Forms("frm_Wirtschaftlichkeitsrechnung1").Controls!ID_Massnahme

```

#### ' Liste durchzuführender Rechengänge

```

calc.Parameters!Nr_massnahme = ID_Massnahme
Set rec_calc = calc.OpenRecordset

```

#### ' Liste der einzelnen Rechenschritte

```

steps.Parameters!Nr_massnahme = ID_Massnahme
Set rec_steps = steps.OpenRecordset

```

#### ' Liste notwendiger Operanden

```

oper.Parameters!Nr_massnahme = ID_Massnahme

```

```
Set rec_oper = oper.OpenRecordset
```

```
' Löschen alte Mengenangaben
```

```
Set mengen = db.QueryDefs("H_Wirt_Gueter_Mengen0")
mengen.Execute
```

```
' Records in Mengentabelle anlegen
```

```
Set mengen = db.QueryDefs("H_Wirt_Gueter_Mengen1")
mengen.Parameters!Nr_massnahme = ID_Massnahme
mengen.Execute
```

```
' Recordset um aktuelle Mengen einzutragen
```

```
Set mengen2 = db.QueryDefs("H_Wirt_Gueter_Mengen2")
Set rec_mengen = mengen2.OpenRecordset
```

```
' Schleife über Rechengänge
```

```
While Not rec_calc.EOF
```

```
  ' Werte zuweisen
```

```
  ID_Rechengang = rec_calc.Fields!ID_Berechnung
  Nr_Rechenschritt = rec_calc.Fields!Nr_Rechenschritt
  id_gut = rec_calc.Fields!id_gut
  status = rec_calc.Fields!status
  ID_Zeitreihe = rec_calc.Fields!ID_Zeitreihe
```

```
  ' Ermittlung des Ergebnisses des Rechengangs
```

```
  res = Rechengang(ID_Rechengang, Nr_Rechenschritt, rec_steps, rec_oper)
```

```
  ' Ergebnis in richtiges Feld schreiben
```

```
  rec_mengen.FindFirst ("[ID_Gut] = " & id_gut)
  rec_mengen.Edit
```

```
  ' Mengen vor Durchführung der Massnahme
```

```
  If status = -1 Then
    rec_mengen.Fields!vorher = res
    rec_mengen.Fields!ID_ZR_vorher = ID_Zeitreihe
```

```
  ' Mengen nach Durchführung der Massnahme
```

```
  Else
    rec_mengen.Fields!nachher = res
    rec_mengen.Fields!ID_ZR_nachher = ID_Zeitreihe
```

```
  End If
```

```
  rec_mengen.Update
```

```
  rec_calc.MoveNext
```

```
Wend
```

```
' Schleife über Operanden - Eintragen der Operanden in Hilfstabelle
```

```
Set rec_listOper =
```

```
db.TableDefs!H_Wirtschaftlichkeit_Eingabedaten.OpenRecordset(dbOpenDynaset)
```

```
If rec_oper.RecordCount > 0 Then
```

```
  rec_oper.MoveFirst
```

```
End If
```

```
While Not rec_oper.EOF
```

```
  oper1 = rec_oper.ID_Operand
```

```

rec_listOper.FindFirst ("[ID_Zeitreihe] = " & oper1)
If rec_listOper.NoMatch = True Then
    rec_listOper.AddNew
    rec_listOper.ID_Zeitreihe = oper1
    rec_listOper.Update
End If
rec_oper.MoveNext
Wend

```

```
End Sub
```

### **Sub Rechnen\_InvestGueter()**

```

Sub Rechnen_InvestGueter(ID_Massnahme, flag_gesamt)
Dim db, calc, exog, steps, oper, mengen, mengen1, tab_res, oper1
Dim rec_calc, rec_exog, rec_steps, rec_oper, rec_mengen, rec_res, rec_listOper
Dim ID_Rechengang, Nr_Rechenschritt, i, res, Status, id_gut
Dim ID_Zeitreihe

```

#### **' Definition relevante Abfragen**

```

Set db = CurrentDb()
Set calc = db.QueryDefs("H_Massnahmen_InvestGueter_Rechengang")
Set steps = db.QueryDefs("H_Rechengang_Rechenschritte_InvestGueter")
Set oper = db.QueryDefs("H_Rechengang_Rechenschritte_Operand_InvestGueter")

```

#### **' Liste durchzuführender Rechengänge**

```

calc.Parameters!Nr_massnahme = ID_Massnahme
Set rec_calc = calc.OpenRecordset

```

#### **' Liste der einzelnen Rechenschritte**

```

steps.Parameters!Nr_massnahme = ID_Massnahme
Set rec_steps = steps.OpenRecordset

```

#### **' Liste notwendiger Operanden**

```

oper.Parameters!Nr_massnahme = ID_Massnahme
Set rec_oper = oper.OpenRecordset

```

```

If flag_gesamt = True Then 'Wirkliche Berechnung - andernfalls nur Ermittlung der
notwendigen Eingabedaten

```

```

' in diesem Fall Ausführung der Abfragen nicht erforderlich

```

```

' Löschen alte Mengenangaben

```

```

Set mengen = db.QueryDefs("H_Wirt_InvestGueter0")
mengen.Execute

```

```
End If
```

#### **' Schleife über Rechengänge**

```
While Not rec_calc.EOF
```

```

' Werte zuweisen

```

```

ID_Rechengang = rec_calc.Fields!ID_Berechnung

```

```

Nr_Rechenschritt = rec_calc.Fields!Nr_Rechenschritt
id_gut = rec_calc.Fields!id_gut
Status = rec_calc.Fields!Status
ID_Zeitreihe = rec_calc.Fields!ID_Zeitreihe

' Ermittlung des Ergebnisses des Rechengangs
res = Rechengang(ID_Rechengang, Nr_Rechenschritt, rec_steps, rec_oper)
' Ergebnis in richtiges Feld schreiben

rec_calc.MoveNext
end

If flag_gesamt = True Then 'Wirkliche Berechnung - andernfalls nur Ermittlung der
notwendigen Eingabedaten
' in diesem Fall Ausführung der Abfragen nicht erforderlich
' Recordset um aktuelle Mengen einzutragen
Set mengen1 = db.QueryDefs("H_Wirt_InvestGueter1")
mengen1.Parameters!Nr_massnahme = ID_Massnahme
mengen1.Execute

' Korrigieren Anzahl
Set mengen1 = db.QueryDefs("H_Wirt_InvestGueter2")
mengen1.Execute

' Korrigieren Nutzungsdauer
Set mengen1 = db.QueryDefs("H_Wirt_InvestGueter3")
mengen1.Execute

' Korrigieren spezifischer Kapitalzinssatz
Set mengen1 = db.QueryDefs("H_Wirt_InvestGueter4")
mengen1.Execute

' Korrigieren jährliche Betriebskosten
Set mengen1 = db.QueryDefs("H_Wirt_InvestGueter5")
mengen1.Execute

' Korrigieren Investitionskosten
Set mengen1 = db.QueryDefs("H_Wirt_InvestGueter6")
mengen1.Execute
End If

' Schleife über Operanden - Eintragen der Operanden in Hilfstabelle
Set rec_listOper =
db.TableDefs!H_Wirtschaftlichkeit_Eingabedaten.OpenRecordset(dbOpenDynaset)
If rec_oper.RecordCount > 0 Then
rec_oper.MoveFirst
End If
While Not rec_oper.EOF

```

```

oper1 = rec_oper.ID_Operand
rec_listOper.FindFirst ("[ID_Zeitreihe] = " & oper1)
If rec_listOper.NoMatch = True Then
    rec_listOper.AddNew
    rec_listOper.ID_Zeitreihe = oper1
    rec_listOper.Update
End If
rec_oper.MoveNext
Wend

```

```
End Sub
```

**Function** Rechengang(ID\_Rechengang, Nr\_Rechenschritt, rec\_steps, rec\_oper)  
'[Ermittelt das Ergebnis für einen einzelnen Rechengang](#)

```

Dim db2, calc2, steps2, oper2, oper1
Dim rec_calc2, rec_exog2, rec_steps2, rec_oper2, rec_listOper
Dim ID_Rechengang2, Nr_Rechenschritt2

```

```
Dim crit1, crit2, i, val_op, res, res2, j_Ber, OPT_Ber, ID_res, heute, rec_values
```

'[Ersten Rechenschritt für Rechengang in der Liste aller Rechenschritte finden](#)

```

crit1 = "[ID_Berechnung] = " & ID_Rechengang
rec_steps.FindFirst (crit1)
res = Null

```

```

Set db2 = CurrentDb()
Set rec_values = db2.TableDefs!W_Aspekte.OpenRecordset(dbOpenDynaset)
Set rec_listOper =
db2.TableDefs!H_Wirtschaftlichkeit_Eingabedaten.OpenRecordset(dbOpenDynaset)

```

'[Schleife über Rechenschritte im Rechengang](#)

```

Do While rec_steps.ID_Berechnung = ID_Rechengang And
rec_steps.Nr_Rechenschritt <= Nr_Rechenschritt

```

[Ersten Operanden für Rechenschritt finden](#)

```

j_Ber = rec_steps.Nr_Rechenschritt
OPT_Ber = rec_steps.ID_Operator
crit2 = crit1 & " AND [Nr_Rechenschritt] = " & j_Ber
rec_oper.FindFirst (crit2)
i = 1

```

'[Schleife über alle Operanden für Rechenschritt](#)

```

Do While (Not rec_oper.EOF) And rec_oper.ID_Berechnung = ID_Rechengang And
    rec_oper.Nr_Rechenschritt = j_Ber
    'Wenn Operand berechneter Wert
    If rec_oper.Fields!flag_berechnet = True Then
        'Ggf. Ergebnis aus vorangegangenen Rechenschritt
        If rec_oper.ID_Operand = ID_res Then

```

```

    val_op = res
' Sonst nach Rechengang mit entsprechendem Ergebnis suchen
Else
    ' Notwendige Abfragen ermitteln, s. o.
    Set db2 = CurrentDb()
    Set calc2 = db2.QueryDefs("H_Zeitreihe_Rechengang")
    calc2.Parameters!crit_erg = rec_oper.ID_Operand
    Set steps2 = db2.QueryDefs("H_Rechengang_Rechenschritte2")
    Set oper2 =
        db2.QueryDefs("H_Rechengang_Rechenschritte_Operand2")

    Set rec_calc2 = calc2.OpenRecordset
    ID_Rechengang2 = rec_calc2.Fields!ID_Berechnung
    Nr_Rechenschritt2 = rec_calc2.Fields!Nr_Rechenschritt

    steps2.Parameters!crit_rech = ID_Rechengang2
    oper2.Parameters!crit_rech = ID_Rechengang2
    Set rec_steps2 = steps2.OpenRecordset
    Set rec_oper2 = oper2.OpenRecordset

'Ergebnis ermitteln
res2 = Rechengang(ID_Rechengang2, Nr_Rechenschritt2,
    rec_steps2, rec_oper2)
val_op = res2

' Schleife über Operanden - Eintragen der Operanden in
    Hilfstabelle
If rec_oper2.RecordCount > 0 Then
    rec_oper2.MoveFirst
End If
rec_oper2.MoveFirst
While Not rec_oper2.EOF
    oper1 = rec_oper2.ID_Operand
    rec_listOper.FindFirst ("[ID_Zeitreihe] = " & oper1)
    If rec_listOper.NoMatch = True Then
        rec_listOper.AddNew
        rec_listOper.ID_Zeitreihe = oper1
        rec_listOper.Update
    End If
    rec_oper2.MoveNext
Wend

End If
'Vorhandenen Wert nehmen
Else
    val_op = rec_oper.relWert
End If

```

```

' Operator anwenden
If i = 1 Then
    res = val_op
Else
    ' Fallunterscheidung nach Operatortyp
    Select Case OPT_Ber
        Case 1
            res = res + val_op
        Case 2
            res = res - val_op
        Case 3
            res = res * val_op
        Case 4
            res = res / val_op
    End Select
End If
ID_res = rec_steps.ID_Ergebnis

i = i + 1
rec_oper.MoveNext
' Schleifenende wenn Ende der Liste erreicht
If rec_oper.EOF Then Exit Do
Loop

heute = Date
rec_values.FindFirst ("[ID_Zeitreihe] = " & ID_res) ' & " AND [Datum] = #" & heute
& "#")
If rec_values.NoMatch = False Then
    rec_values.Edit
    rec_values.Fields!Wert = res
    rec_values.Fields!datum = heute
    rec_values.Update
Else
    rec_values.AddNew
    rec_values.Fields!ID_Zeitreihe = ID_res
    rec_values.Fields!datum = heute
    rec_values.Fields!Wert = res
    rec_values.Update
End If
rec_steps.MoveNext
' Schleifenende wenn Ende der Liste erreicht
If rec_steps.EOF Then Exit Do
Loop

' Ergebnisuweisung
Rechengang = res
End Function

```

**Erklärung**

Ich versichere, dass ich diese Arbeit selbständig verfasst und nur die angegebenen Hilfsmittel verwendet habe.

Feng Ye

**Erklärung**

Ich versichere, dass ich diese Arbeit selbständig verfasst und nur die angegebenen Hilfsmittel verwendet habe.

Yilian Shen