InfBO2
© Frank Koch & Thomas Keller

Table of Contents

Objectives

- You should be able to structure data and prepare it for data processing
- You should be able to program and maintain small-sized database applications using MS-Access

Structure and Timetable

1 Organization ~ 1 lesson
2 Data Modeling ~ 15 lessons
3 Data Implementation ~ 12 lessons
4 Group Assignment ~ 8 lessons
5 SQL ~ 4 lessons

Important:

We'll proceed step by step, keeping repetition to a minimum. Don't miss a step because it might become difficult to keep up with the rest of the program.

Study Materials

- This script is not meant to be your only study material! It serves as a framework to support our lessons. Take notes and fill this script with comments.
- You can find this script at \\zeus\Dozenten\kfr\public\InfBO2\InfBO2 Skript.
- You can use „H.R. Hansen / G. Neumann: Wirtschaftsinformatik I, 8. Auflage (2001)“ as reference.
- For MS-Access, desktop-video sequences will be provided.
- Search the Web for additional material!

Group Assignment

- You will form teams of about 4 students each working on your own case. The assignments will correspond to the progress of our lessons and will help you use and verify your knowledge.
- Three deliverables are planned and will be graded:
  1. Documentation of your case containing:
     - Introduction of your case
     - Description of business events
     - Normalized Entity Relationship Model
     - List of Attributes
  2. Presentation of your case based on the above documentation
     - Present you case as well as your ERD
     - 15 Min presentation plus 5 Min Q&A
  3. Presentation of your database solution implementing your case
     - Present you database in Access
     - 15 Min presentation plus 5 Min Q&A

Grading

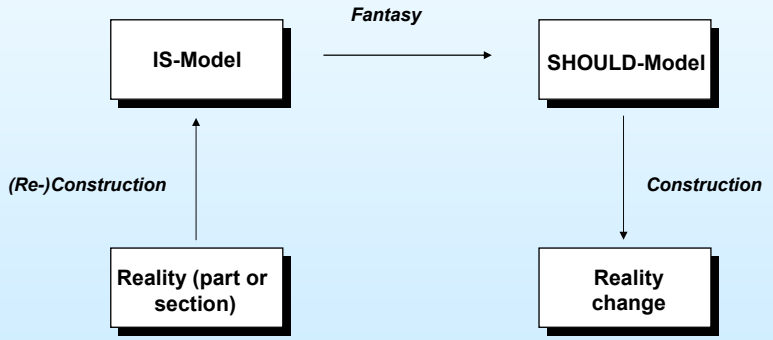
- Your grading will consist out of two points:
  1. Group work (oral presentation and written documentation)
     - Groups of about 4 students
     - Weight: 1/3 of the grade
  2. Written exam
     - Open books
     - Mandatory except with a medical excuse
     - Weight: 2/3 of the grade

Table of Content

Definition of Model

- A model is a concrete or mental image of an existing „thing“ (e.g. model of a vessel)
- Models represent an original based on the purpose of investigation.
- The major types of models being used in IT are semantic models, based on notation and grammar.

Steps of Modelling



Objectives of Data Modeling

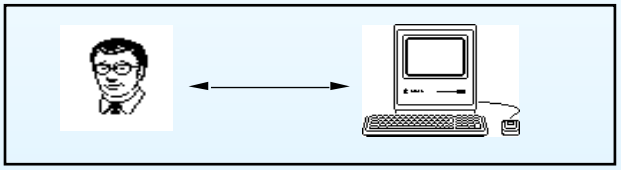
- The data model of an enterprise consists of all necessary information for operational performance in production and marketing.
- Data models represent the information-related aspects of operational events. These show a structured description of information and the relationships among the information.
- A data model is a non-material image of operational object systems. This processed information is in view of the objectives of the information system designers.
- The task of data modeling (or data design) is to define and set the specifically needed
  - information objects (e.g. customers, suppliers)
  - their relationships and dependence
  - and the essential attributes and properties of these information objects (e.g. Name, address, credit limit)

  to make an operational arrangement of tasks.

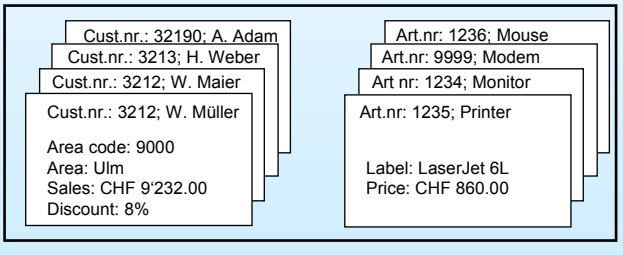
  Data modeling describes the requirements for a realizable organizational solution, from the view of data.

Levels of Data Modeling

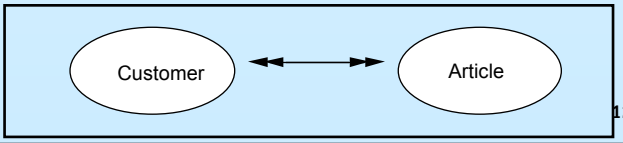
Real World



Quantitatinve Build-up in the Information System

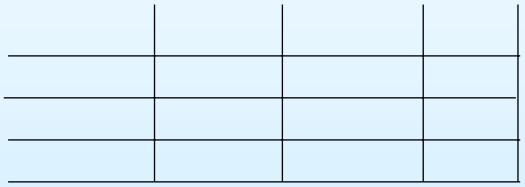


Description of Reality with an ER Diagram



Basic Concepts

- The Concept of Tables („Relation“)



- Data Design Concepts

Real world	Conv. data proces.	Database world
Customers	Data File	Entity Set / Table
CustomerXY	Record, Sentence	Entity
Phone Nr.	Field	Attribute

The Concept of Entity

An *entity* is an individual and identifiable example of a *thing, person* or *idea* in the real or imaginary world, about which we want to collect information.

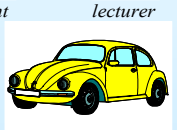
Examples:

An individual

student

lecturer

Some real object



An event *invoice*

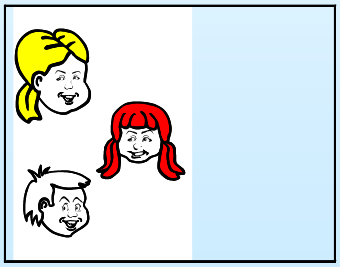
The context must be known in order to know if an idea is a entity or not. For example, the idea *`street'* in salary accounting is more of a description of an employee – and in this context is *not* an entity.

However, in the context of public administration, where information on streets are needed (e.g. construction costs, repairs etc.), a *`street'* is an entity.

The Concept of an Entity Set

An entity set encompasses entities of the same type.

Example:



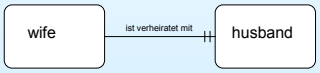
Entity set of students

Types of Relationships

When an entity corresponds exactly to <u>one</u> other entity, this is said to be a simple (i.e., one-to-one) relationship.

Example

A marriage in our legal system represents a one-to-one relationship, when one woman is married exactly to to one man..

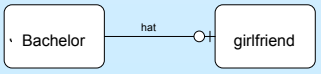


A wife is married to at leaast one, and no more than one husband.'

In a conditional relationship, an entity is related to at most <u>one</u>, or possibly to <u>no</u> entity.

Beispiel:

One is never sure if a bachelor has one girlfriend or none.



‚A bachelor has a minimum of zero, and a maximum of one girlfriend'.

Types of Relationships

A complex relationship is present when one entity is related to at least one, and possibly several entities.

Example:

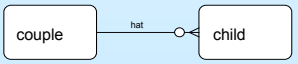
A teacher instructs one or several pupils.

'A teacher instructs *at least one, and a maximum of several* pupils.'

We speak then of a *complex-conditional relationship* if an entity is related to none, one or several entities.

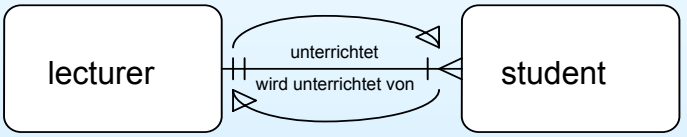
Example:

A couple can have no child, one child or several children.



'A couple has a *minimum of none*, and a *maximum of several* children.'

Reciprocal Relationships



- Relationships between entities are always taken as reciprocal. This is seen in a `*teacher-pupil*´ relationship and a `*pupil-teacher*´ relationship.
- Such diagrams are always read clockwise.
- *`A teacher instructs at least one pupil, and also possibly several pupils. A pupil is taught by exactly one teacher.´*

Exercise 1

Draw an ER-Diagram for the following descriptions:

- One person is allowed to buy exactly one ticket. One ticket is purchased by exactly one person
- Each course is given by exactly one teacher. There teachers who give no or several courses.
- An apartment house comprises one or several apartments. An apartment always belongs to one apartment house.
- A customer has at least one but possibly more addresses. An address belongs to exactly one customer.

Exercise 2

Give some examples for the different relationships according to the table below.

Relationship	One	Condition	Complex	Complex-Condition
One	1:1	1:C	1:1M	1:CM
Condition	C:1	C:C	C:1M	C:CM
Complex	1M:1	1M:C	1M:1M	1M:CM
Complex-Condition	CM:1	CM:C	CM:1M	CM:CM

Exercise 3

a) Define the entities based on the description below.

b) Draw the corresponding ER-Diagram.

- An enterprise organizes courses for internal use (e.g., data modeling, accounting) for different fields (e.g. computer science, economics, etc.)
- Each course is given by one teacher. A teacher can give more than one course.
- Each course is generally taken by more than one student. A student can enroll in more than one course.
- Teachers as well as students are employees of the same company. An employee can at the same time be a teacher in one course and a student in another course.
- Each course needs a course room. A course room can be used by several courses but at different times.
- Courses are often repeated and distinguished by different course dates.

Exercise 4

a) Define the entities based on the description below.

b) Draw the corresponding ER-Diagram.

- A SW company develops applications for customers.
- This company works for several customers. The customers can give one or several orders.
- Orders consist of several different services.
- The SW company has several employees. These employees belong to one of two groups. One group works on the orders (services). These are project employees. The other group is responsible for the administrative tasks (e.g., secretaries, accountants, etc).
- Each order is overseen by a project employee as the project leader.
- Each employee attend courses on a regular basis to maintain and to improve their qualifications.

Exercise 5

a) Define the entities based on the description below.

b) Draw the corresponding ER-Diagram.

- An employee can have several kinds of wages (e.g., monthly pay, overtime pay, allowance for children, AHV, etc)
- Each wage belongs to exactly one type of salary (e.g., monthly base pay, allowances, social security, others, etc).
- An employee belongs to exactly one department. A department has at least one employee.
- One kind of wage can be applied to several employees.
- A type of salary can consist of several kinds of wages.

Exercise 6

a) Define the entities based on the description below.

b) Draw the corresponding ER-Diagram.

- The ZHW offers 3-year courses for economics. Each year there are 8 classes with approx. 20 students.
- A student belongs to one class.
- Instruction is done class by class in different fields (e.g., computer science, math, accounting, etc)
- Each course is given by exactly one teacher. A teacher gives one or more courses.
- Each course needs one, possibly more rooms.

The Concept of Attribute

Attributes identify, classify, characterize respectively an entity.

Example:
For the *entity `Teacher'*, the following *attributes* and *values* are meaningful in the context of salary bookkeeping:

Attribute	*Value*	
Employee Number.	4711	☜ *identify*
First name Family name AHV-Nr. Salary	Boris Becker 513.73.137.114 99'999.95	☜ *describe*
Employee Status	Freelancer	☜ *classify*

Attibutes of entitiy sets are also described in an ER Model as *properties*.

Attribute versus Entity

Some attributes could take the character of entity sets.

•**Student**: Name, Street, **City**, Stud#, AHV#, Sem

but

•**City**: Postal code, Name of area, Canton, Country, Population

Rule: Should further data be needed around an attribute, then this attribute is expanded into an entity set.

Defining the Primary Key

The entity is uniquely identified by a primary key, which is an attribute or one set of attibutes. Convention: The primary key is underlined in modeling.

Lecturer: Name, Address, Salary

National language: Country, Language, Number of Words
Example: {{D, German, ?}, {CH, German, ?},{CH, Italian, ?}}

If the actual attributes are not enough to uniquely identify an entity, then artificial attributes are formed.

Student: Name, Address, Stud#, Sem

As a rule, only one primary key is defined for an entity set; otherwise one speaks of secondary keys. As much as possible, use only attributes that do not change during the life cycle of the entity. Most databases do not allow change of the primary key.

Independent entities show, as a rule, exactly one attribute as a key. Whereas *dependent* and *associative* entities usually form keys out of several attributes.

Defining the Foreign Key

fremdschlüssel definieren

Two entity sets often stand together related. In the case of one-to-many (1:M) relationships, the relationship is technically realized through foreign keys. Convention: the foreign key is encircled.

In the following example, a supplier delivers several articles. Each kind of article is delivered by exactly one supplier.

Supplier: Supplier#, Firm, Street, Area, Tel, Fax, Mail, Homepage

Article: Article#, Supplier#, Article name, Delivery unit, Unit price,Items on stock

In this example, the Supplier# is the foreign key in the entity set Article and the primary key in the entity set Supplier.

Rules:

- A foreign key corresponds always to a primary key in another entity set.
- A pair of primary and foreign keys belong to every 1:M relationship.
- The primary key is found always at the 1 side, while the foreign key is always on the M side.
- Foreign and primary keys could be named differently.

Normalization: Unnormalized form

Statement of the Problem Problemstellung
The following observations were made:

An employee
- has one name (first name and family name)
- draws one salary
- works in one department
- has possibly several children

The children of an employee...
- have one first name

A department...
- has one name
- has one number
- has several employees

P#	VName	NName	Gehalt	A#	Abteilung	AbtBudget	Kind1	K1Alter	Kind2	K2Alter	Kind3	K3Alter
1	Hans	Müller	5425	3	Personal	200'000	David	6	Natalie	4		
2	Rita	Schultz	5744	1	Produktion	1'200'000	Heinz	12				
3	Werner	Meier	4145	1	Produktion	1'200'000	Sabine	19	Rolf	15	Sandra	11
4	Otto	Moser	6724	2	Verkauf	500'000						
5	Maria	Kuntz	8304	2	Verkauf	500'000						
6	Karl	Müller	5478	1	Produktion	1'200'000	Heinz	12				

Note: **P# is the primary key in this entity set!**

Normalization: Unnormalised form Another variation

P#	VName	NName	Gehalt	A#	Abteilung	AbtBudget	Kind
1	Hans	Müller	5425	3	Personal	200'000	David 6, Natalie 4
2	Rita	Schultz	5744	1	Produktion	1'200'000	Heinz 12
3	Werner	Meier	4145	1	Produktion	1'200'000	Sabine 19, Rolf 15, Sandra 11
4	Otto	Moser	6724	2	Verkauf	500'000	
5	Maria	Kuntz	8304	2	Verkauf	500'000	
6	Karl	Müller	5478	1	Produktion	1'200'000	Heinz 12

This table has similar problems as the preceeding variant.

Normalization: First normal form

The first normal form is given when no set of values for an attribute occurs, i.e., no attibute shows several values

P#	VName	NName	Gehalt	A#	Abte	AbtBudget	K#	Kind	KAlter
1	Hans	Müller	5425	3	Pers	200'000	1	David	6
1	Hans	Müller	5425	3	Pers	200'000	2	Natalie	4
2	Rita	Schultz	5744	1	Prod	1'200'000	1	Heinz	12
3	Werner	Meier	4145	1	Prod	1'200'000	1	Sabine	19
3	Werner	Meier	4145	1	Prod	1'200'000	2	Rolf	15
3	Werner	Meier	4145	1	Prod	1'200'000	3	Sandra	11
4	Otto	Moser	6724	2	Verk	500'000			
5	Maria	Kuntz	8304	2	Verk	500'000			
6	Karl	Müller	5478	1	Prod	1'200'000	1	Heinz	12

Note: The primary key must be expanded from „P#“ to „P# and K#“ to keep the entities clearly identifiable!

Normalization: Second normal form

The second normal form is given when the first normal form is complied with and each attribute is dependent on the complete primary key, but not on parts of the primary key only.

P#	VName	NName	Gehalt	A#	Abteilung	AbtBudget
1	Hans	Müller	5425	3	Personal	200'000
2	Rita	Schultz	5744	1	Produktion	1'200'000
3	Werner	Meier	4145	1	Produktion	1'200'000
4	Otto	Moser	6724	2	Verkauf	500'000
5	Maria	Kuntz	8304	2	Verkauf	500'000
6	Karl	Müller	5478	1	Produktion	1'200'000

P#	K#	Kind	KAlter
1	1	David	6
1	2	Natalie	4
2	1	Heinz	12
3	1	Sabine	19
3	2	Rolf	15
3	3	Sandra	11
6	1	Heinz	12

Since the entity „Kind“ can exist only in the context of an employee, it is an attributive or dependent entity. The key of this entity consists of **P# and K#.**

Normalization: Third normal form

A relationship is in the third normal form when the second normal form is complied with and no dependence exists between non-key attibutes i.e., no transitive dependence exist.

P#	VName	NName	Gehalt	A#
1	Hans	Müller	5425	3
2	Rita	Schultz	5744	1
3	Werner	Meier	4145	1
4	Otto	Moser	6724	2
5	Maria	Kuntz	8304	2
6	Karl	Müller	5478	1

Note: A# would be the foreign key in the entity set employee.

A#	Abteilung	AbtBudget
1	Produktion	1'200'000
2	Verkauf	500'000
3	Personal	200'000

P#	K#	Kind	KAlter
1	1	David	6
1	2	Natalie	4
2	1	Heinz	12
3	1	Sabine	19
3	2	Rolf	15
3	3	Sandra	11
6	1	Heinz	12

Exercise 7: Normalization

The target times of certain employees for different projects are presented in the table below:

Pers#	Name	Proj#	ProjName	Dept#	DeptName	TargetTime
10005	Peter	A100	MwSt	A20C	REWE	20
		A110	Stempel	A20C	REWE	10
10006	Otto	A100	MwSt	A20E	ARCH	5
10007	Gustav	A110	Stempel	A20D	FIBU	30
		A120	Liefer	A20D	FIBU	20

- What relationship problems do you see in this table?
- Normalize the relationship and define the primary and foreign keys.
- Goal: 3rd normalized form

Exercise 8: Normalization

Given following relationship:
DELIVERY (Del#, DelName, Prod#, ProdName, Amount, Sup#, SupName)

This is based on the following description:

- *A supplier has exactly one name*
- *A supplier delivers several products*
- *A product has a unique descriptor*
- *A product is delivered by one supplier in a defined amount*
- *The same product is delivered by several suppliers*

Normalize the relation and define primary and foreign keys.
Goal: 3rd normalized form.

Exercise 9: Normalization

- The table below describes the course enrolment of employees.
- Departments with a budget <70‘000 are characterized as projects, otherwise as profit centers.

Pers#	Name	Abt#	AbtBudget	AbtKategorie	Kurs#	KursName	Einschreibung	Durchführung	Raum
425	Meier	15	72'000	Profitcenter	X17	Verkauf	27.04.2003	05.10.2003	A
425	Meier	15	72'000	Profitcenter	L91	Rhetorik	30.05.2003	15.09.2003	B
425	Meier	15	72'000	Profitcenter	L98	Führung	01.11.2003	11.11.2003	A
458	Schulze	18	33'000	Projekt	L91	Rhetorik	20.12.2002	15.09.2003	B
505	Müller	15	72'000	Profitcenter	X17	Verkauf	10.12.2003	01.02.2004	A
703	Schmid	26	80'000	Profitcenter	X17	Verkauf	10.12.2003	12.12.2003	B
703	Schmid	26	80'000	Profitcenter	L31	Analyse	30.09.2003	05.10.2003	C

Goal: Convert this table into the 3rd normalized form!

Optional Homework

The relationship PRODUCTIONTIMES (Emp#, Name, Dep#, Department, P#, ProductName, Time) is based on the following description:

- *An employee has one name*
- *An employee works in one department*
- *The production needs a certain time per product and employee*
- *A department has one name*
- *A product has one name*
- *Several employees belong to a department*
- *The manufacturing of a product needs several employees.*

Questions:

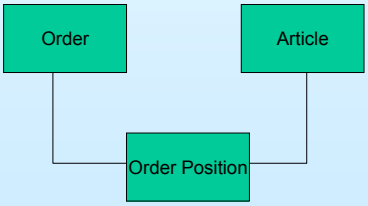
- *What difficulties (anomalies) can be seen in this relationship?*
- *Normalize this relationship and define primary and foreign key(s).*

Exercise 10: Normalization

- The relation **CUSTOMER(K#, Name, Street, PostalCode, Area)** violates the 3rd normalized form! Why?
- What is the disadvantage if you sacrifice normalization due to performance maximization?

Resolving complex relationships

Consider this relationship between the entities *Order* and *Article*: it concerns a complex relationship, which is also known as an M:M relationship. Such relationship cannot be represented in a relational databank system, since it violates the *normal form*. The entity `*Order´* as an example would contain the attribute *Article-Nr.* several times, going against the first normal form.



Therfore such M:M relationships must be resolved by inserting an *association entity*. The key of the entity Order Position consists of *Order-Nr.* and *Article-Nr.*

Types of Entities

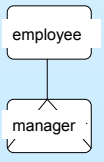
Core entities can exist on their own..

Example:

The entities Customer, Article and Employee can exist independently of another entity. Therefore these are *core entities*.

Dependent entitites are unilaterally (one-sided) dependent on other entites.

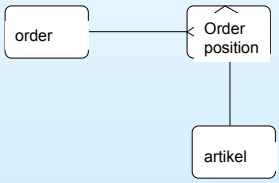
Example:



One can be a *supervisor* in an enterprise only when she or he is an *employee* (company rule).
Thus, the entity Supervisor is *dependent* on the entity Employee.

Associative entities are dependent multilaterally (many-sided) on other entites.

Example



An *Order Position* only exists when there is a corresponding Order and Article. The entity Order Position is dependent on two sides and is therefore an *associative entity*.

Exercise 11

- Draw an ER-Diagram for the examples below and solve the M:M relationships with associative entities.
  - A delivery contains at least one, and possibly more than one article. An article can be part of no or several deliveries.
  - A person can possibly speak several languages. A language is spoken by several persons.
  - Our employees can manufacture several products. Teams of employees often work on a product.
  - In a hierarchical company organization, an employee leads several other employees. An employee is led by only one other employee.

Exersise 12: Public Library

- A public library lends out books on different media (print as well as audio).
- Data is being collected for the library members (clients), the titels, the publishing house, and the author.
- For searching purposes registers are organized by keywords, authors, and categories (i.e. poetry, lyric, novels).
- The library distriguishes titles and their perhaps multiple editions published by the different publishing houses. Furthermore they sometimes buy multiple items of an edition.
- Items are lend out for a fixed 30 days.
- Draw an ER-Diagram and solve possible M:M relationships.
- Define the corresponding entities including their attributes, primary and foreign keys.

Exercise 13

Define the entities including their attributes based on the description below and the two views

Draw an ER-Diagram and solve M:M relationships.

- The company Constructa AG delivers construction material
- A customer gives orders
- An article group contains at least one article
- A supplier can deliver several articles
- An article can be delivered by several suppliers
- An order is given by one customer
- An article can be contained in several orders
- An article belongs to one article group
- An order consists of at least one article

View: Suppliers sorted by article group

article group:	**33 Zubehör Maschinen**	
supplier nbr.	Name	City
4711	Kern GmbH	Winterthur
4712	Baier AG	Zürich
4713	Meier AG	Zürich
article group:	**66 Werkzeuge**	
supplier nbr.	Name	City
4711	Kern GmbH	Winterthur
4713	Meier AG	Zürich
article group:	**88 Bauhilfsmittel**	
supplier nbr.	Name	City
4712	Baier AG	Zürich
4713	Meier AG	Zürich

See next page →

Exercise 13: Order Confirmation

Meier AG
Rigiweg 20
8000 Zürich

Order#: 123456
Customer#: 4713

Date 14.05.2002
Our Contact: P. Müller

Article#	**Description**	**Amount**	**Prize**	**Total**
880596	Gerüstelement	28	589.00	16492.00
660987	Trennscheibe	34	20.00	680.00
339877	Abdeckung	1	700.00	700.00
		Order value excl MWST		17872.00
		MWST		1358.25
		Order Value incl MWST		19230.25

Thank you!

Constructa AG, Everything for constructions, Zürich

← see previous page

Exercise 14a

You received the task to create an ER-Diagram for a grocery chain that has outlets in CH, D and A. Define the entities, including the necessary attributes and relationships between them, for the following statements:

- An employee is identified by her or his personal number and has a name and address
- An outlet is described by an ID and the place where it is located.
- All articles have a number, a name and a price.
- A supplier has a number, a name and an address.
- An outlet employs several persons; a person works exactly at one outlet.
- An outlet sells several articles. The same article can be sold by several outlets.
- A supplier delivers several articles. An article is delivered by exactly one supplier only.

Exercise 14b

Create the database and the entities and add the following data:

- The supplier Maier AG, 8400 Winterthur, delivers washing powder Omo liquid.
- The supplier Close&Fresh GmbH, 8000 Zürich, delivers beer called Schwupps non-alcoholic.
- Mr. Meier, who lives in Zürich, is now employed by outlet Zurich. Mr. Meier replaces Mr. Müller.

Exercise 15

Create an ER-Diagram for the user view „project control“ and „effort employee per project“. Resolve possible M:M relationships and add attributes to the entities.

Remark: Each service code can show up only once

View: Project Control

Customer: **5897 UBS Opfikon, Foreign Exchange**
Project Controller: **Peter Meier PersonalNr: 45**
Project Number: **478**
Project Description: **Anpassung Auftragsabwicklung**

EmpNr	Name	ServCode	ServDesc	AccTyp	Effort
32	Meier Peter	005	Koordination	4	12
56	Müller Hans	010	Analyse	3	58
44	Keller Urs	010	Analyse	2	34
56	Müller Hans	020	Design	1	45
47	Meier Kurt	025	Konstruktion	1	140
47	Meier Kurt	030	Test	4	20
45	Smith Joe	030	Test	4	10
45	Smith Joe	035	Einführung	1	40
			TOTAL		359

Accounting Type: 1 = 100% of the effort is accounted for
2 = 60% of the effort is accounted for
3 = 50% of the effort is accounted for
4 = 0% of the effort is accounted for

View: Man-hours per project

Customer: Peter Meier PersonalNr: 45

Project 478 Anpassung Auftragsabwicklung

Code Service	Hours	Accounting Type
005 Koordination		
010 Analyse		
020 Design		
025 Konstruktion		
030 Test	10	4
035 Einführung	40	1
TOTAL	50	

Project 534 Wartung an Lagerverwaltung

Code Service	Hours	Accounting Type
005 Koordination		
010 Analyse		
020 Design		
025 Konstruktion	25	2
030 Test	5	3
035 Einführung	2	1
TOTAL	32	

Data integrity: Integrity of relationships

Data integrity means the accuracy of data, based on different aspects. The so-called *integrity conditions* can be either directly defined in the databank system or must be programmed.

P#	Vorname	Name	Gehalt	A#
1	Hans	Müller	3425.00	3
2	Rita	Schultz	3744.00	1
3	Werner	Meier	4145.00	1
4	Otto	Moser	4724.00	2
5	Maria	Kuntz	2803.00	2
6	Karl	Müller	5478.00	1

P#	K#	Kind
1	1	David
1	2	Natalie
2	1	Heinz
3	1	Sabine
3	2	Rolf
3	3	Sandra
6	1	Heinz

Data integrity: Integrity of the value range

- Kind of data (e.g. numeric, alphanumeric, time, date etc.)
- Length of the corresponding data field
- If unknown attributes (NULL-value) are allowed to be stored
- Allowable range of values (e.g. only numbers from 1-3, only Codes F and M etc.)
- Unique identification (e.g. the same Personnel-Nr. Us allowed to occur only once.)

These rules are individually controlled through the database system; others must be defined in the application program.

Data integrity: Cross-reference integrity

- Attributes must be monitored in case these exhibit dependence on each other.
- ***Example:*** The *AHV-Nr.* and *Birthday* of an employee are recorded in salary bookkeeping. Since the year of birth (06.02.19**70**) is a part of the AHV-Nr. (513.**70**.137.114), both values must correspond to each other.

Exercise 16: Data Integrity

The following rule applies:

- An employee belongs to a department
- An department consists of at least one employee

Question:

- What kind of rule shall be applied when a department is deleted? Please give some reasoning!

Exercise 17: Reality Analysis

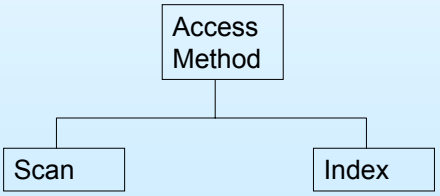
- Identify the entities including their attributes based on the description below and solve M:M relationships. Underline primary keys and encircle foreign keys.
- Draw an ER-Diagram

Description of a Jodler Club:

- The club organizes different activities (e.g., monthly meetings, jodling exercises, management meetings, social events, etc.)
- Per activity and year, several dates are fixed including a list with participants
- The club consists of members who never attend an activity, but also of members who attend several activities.
- The address, the start of the membership and possibly several club functions (e.g., passive member, manager, singer, cashier, etc) of a member are recorded.
- Each club function has its yearly financial contribution. If a member has several club functions, then the lowest yearly contribution is applied.
- Contributions due are computed as of the first of January.
- An invoice is sent at the beginning of each year

Optimizing Access Time

- A relational databank system basically distinguishes two kind of access methods (dependent on the product)



Exercise 18: Video Rental

Design a relational database for the following problem case. Please follow the step-by-step instructions that are given 4 pages further.

The enterprise Otto's Videoplace rents out videos.

As soon as a person rents a video, he or she will be registered as a customer. A customer can rent several videos at the same time. A video can be rented by several customers (of course not at the same time). It is also possible that a video has never been rented.

To improve productivity, the rental system shall be automated. The analysis has shown that the following 3 views (see next pages) need to be implemented:

- Receipt
- Customer Management
- Video Registration

Exercise 18: Receipt

A customer has to sign a receipt for his/her video rentals

```
Otto's Videoplace                                              Page  1
-------------------------------------------------------------------

Frau                                          Customer-Nbr. 55
Heidi Hugentobler
Rütlischwurgrund 52a
9999 Heidiland



Video Rental from 28.10.02 till 29.10.02
---------------------------------------

Video-Nbr.   Title                        Rate              Prize

5219        Blue Velvet                   6.00              12.00
6323        Home alone                    5.00              10.00
                                                            -----
Total                                                       22.00
                                                            =====

Videos received:


_______________
Signature
```


Exercise 18: Customer Mgmt

With help of the following form new customers can be registered, existing customers changed or deleted.

```
28.10.2002             Customer Management                    V001
-------------------------------------------------------------------

       Customer-Nbr.:     55

       Title        :     1    Frau
       First Name   :     Heidi
       Last Name    :     Hugentobler
       Street       :     Rütlischwurgrund 52a
       Place        :     Heidiland



-------------------------------------------------------------------
```

Exercise 18: Video Registration

Following form enables the user to register new videos, to change existing video and to delete old videos.

```
28.10.2002            Video Registration                    V002
-----------------------------------------------------------------

      Video-Nbr.     : 5219

      Title          : Blue Velvet
      Rate           : 6.00
      minimal Age    : 18
      Genre          : TH    Thriller




-----------------------------------------------------------------
```


Step-by-Step Instructions

Step 1: Reality Analysis (e.g., study forms and documents, conduct interviews, analyze existing data) and definine entities.

Step 2: Draw an ER-Diagram including entities‘ attributes, primary and foreign keys.

Step 3: Normalization, Goal: 3rd normalized form

Step 4: Generate relational database model based on the ER-Diagram, resolve M:M relationships

Step 5: Create database and tables

Step 6: Create forms and reports

Step 7: Optimize the performance by introducing additional indexes.

Table of Contents

Learning objectives

- **Indtroduction**
- The Product MS Access RDBMS
- What is an Access database?
- Tables, Queries and Dynasets
- Forms and reports
- First steps

- **Statistical Aspects of Data**
- Creating databases
- Creating tables

- **Dynamic Aspects of Data**
- Data searches
- Creation from queries

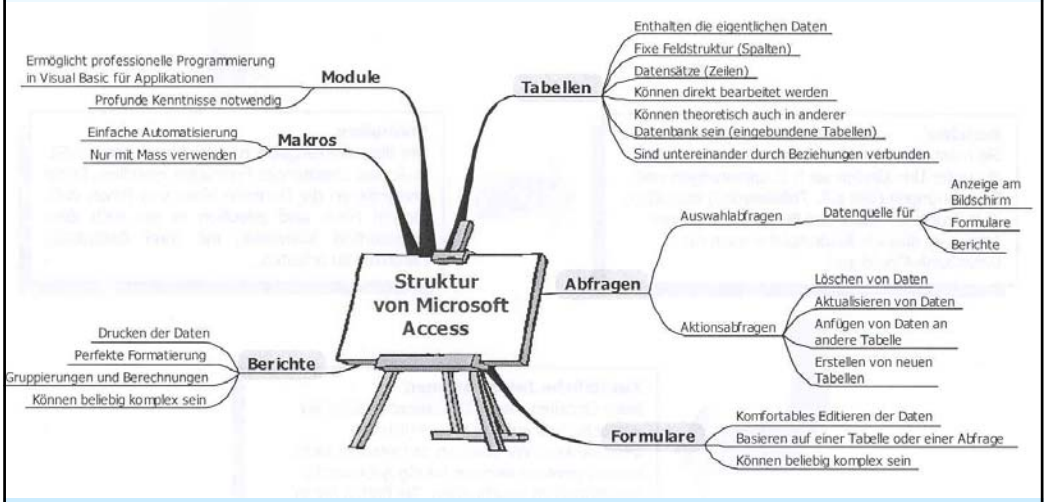
- **Presentation of Data**
- Creating forms
- Creating reports and graphics

- **Structured Query Language**

The Product MS-Access

- Access is an interactive relational DBMS that helps you organise, query, and represent data.
- Access uses the graphics possibilities in Windows to simplify the user dialog and to clearly present data.
- Access offers comfortable data navigation. Query mechanisms enable data to be represented independent of ist location.
- Access can be used to publish data. WYSIWYG helps to produce forms and reports that can be integrated in presentation documents.
- Access provides productivity tools:
  - Macros enable most tasks to be automated without having to program these.
  - In addition, the programming language Access Basic is available for complex or very specialized taks.

Overview of Access Information Objects



Desktop-Video 1-Introduction

- Create your own directory „Access“.
- Copy the database „Nordwind.mdb“ into your Access directory.
- View the Desktop-Video „1-Einführung.avi“.
- Open MS-Access and follow the viewed introduction.

Desktop-Video 2-Schnelleinstieg

- View the Desktop-Video „2-Schnelleinstieg.avi“.
- Open MS-Access and create your own database „Kontaktverwaltung.mdb“.
- Record some data in your „Kontaktverwaltung.mdb“ and utilize them with the information objects generated.
- Also try available features in your „Kontaktinformationen.mdb“ that were not shown in the video.

Desktop-Video 3-Datenbankerstellung

- View the Desktop-Video „3-Datenbankerstellung.avi“.
- Open MS-Access und create the empty database Test.mdb
- Import the Tables Artikel, Kunden, Bestellungen und Bestelldetails from „Nordwind.mdb“ to Test.mdb. How large is Test.mdb?
- Delete the previously imported tables in Test.mdb. How large is Test.mdb now?
- Repair and compress your database. How large is Test.mdb now?
- Link the table Kunde from „Nordwind.mdb“.

Desktop-Video 4-Tabelleneinführung

- View the Desktop-Video „4-Tabelleneinführung.avi“.
- Open MS-Access and create in the database „Test.mdb“ the table „Adressliste Test“.

Desktop-Video 5-Tabellensortieren

1. View the Desktop-Video „5-Tabellensortieren.avi“. The following questions are based on the Nordwind-Database.
2. Select in the table „Kunden“ through „Auswahlbasiertem Filter“ the customers who reside in Switzerland.
3. Select in the table „Kunden“ through „Formularbasiertem Filter“ the customers who reside in the USA.
4. Select in the table „Kunden“ through „Formularbasiertem Filter“ the customers who reside in the USA or in Great Britain.
5. Select in the table „Kunden“ through „Formularbasiertem Filter“ the contacts who reside in the USA or in Great Britain and whose position is „Vertriebsmitarbeiter“.
6. Based on Task 5, you also want to find female „Vertriebsmitarbeiter“. How do you proceed?
7. Select in the Tabelle „Kunden“ through „Formularbasiertem Filter“ the contacts who either reside in the USA or whose position is „Inhaber“.
8. Sort the customers through „Kontaktperson“.

Criteria-Expressions

Feld	Ausdruck	Beschreibung
Textwerte		
Ort	"Bern"	Zeigt die nach Bern versendeten Bestellungen an.
Ort	"Bern" Oder "Zürich"	Verwendet den Operator ***Oder***, um die nach Bern oder Zürich versendeten Bestellungen anzuzeigen.
Versanddatum	Zwischen #5.1.98# Und #10.1.98#	Verwendet den Operator ***Zwischen...Und***, um die Bestellungen anzuzeigen, die zwischen dem 5. und 10. Januar 98 versendet wurden.
Versanddatum	#2.2.98#	Zeigt die Bestellungen an, die am 2. Februar 98 versendet wurden.
Bestimmungsland	In("Kanada"; "UK")	Verwendet die Funktion ***In***, um die nach Kanada oder England versendeten Bestellungen anzuzeigen.
Bestimmungsland	Nicht "USA"	Verwendet den Operator ***Nicht***, um die Bestellungen anzuzeigen, die in andere Länder als die USA versendet werden.
Firma	>="N"	Zeigt die Bestellungen an, die an Firmen mit Anfangsbuchstaben von "N" bis "Z" gesendet werden.
Bestell-Nr.	Rechts([Bestell-Nr]; 2)="99"	Verwendet die Funktion ***Rechts***, um die Bestellungen anzuzeigen, deren Bestell-Nr. mit "99" endet.
Firma	Länge([Firma])>Wert(30)	Verwendet die Funktionen ***Länge*** und ***Wert***, um die Bestellungen derjenigen Firmen anzuzeigen, deren Name länger als 30 Zeichen ist.
Empfänger	Wie "S*"	Bestellungen, die an Kunden gesendet werden, deren Namen mit "S" beginnen.
Empfänger	Wie "*Importe"	Bestellungen, die an Kunden gesendet werden, deren Namen mit "Importe" enden.

Feld	Ausdruck	Beschreibung
Empfänger	Wie "[A-D]*"	Bestellungen, die an Kunden gesendet werden, deren Namen mit den Buchstaben "A" bis "D" beginnen.
Empfänger	Wie "*ar*"	Bestellungen, die an Kunden gesendet werden, deren Namen die Buchstabenfolge "ar" enthalten.
Empfänger	Wie "Maison Dewe?"	Bestellungen, die an einen Kunden gesendet werden, dessen erster Teil des Namens "Maison" lautet, gefolgt von einem zweiten Teil mit fünf Buchstaben, der mit "Dewe" beginnt und dessen letzter Buchstabe unbekannt ist.

Criteria

Datumswerte		
Lieferdatum	Zwischen Datum( ) Und DatAdd("m", 3, Datum( ))	Verwendet den Operator ***Zwischen...Und*** sowie die Funktionen ***DatAdd*** und ***Datum***. Zeigt die Bestellungen an, deren Lieferdatum innerhalb der nächsten drei Monate ab dem heutigen Datum liegt.
Bestelldatum	< Datum( )- 30	Verwendet die Funktion ***Datum***, um die Bestellungen anzuzeigen, die mehr als 30 Tage zurückliegen.
Bestelldatum	Jahr([Bestelldatum])=1996	Verwendet die Funktion ***Jahr***, um die Bestellungen aus dem Jahr 1996 anzuzeigen.
Bestelldatum	DatTeil("q", [Bestelldatum])=4	Verwendet die Funktion ***DatTeil***, um die Bestellungen des vierten Quartals anzuzeigen.
Bestelldatum	DatSeriell(Jahr([Bestelldatum]), Monat([Bestelldatum])+1, 1)-1	Verwendet die Funktionen ***DatSeriell***, ***Jahr*** und ***Monat***, um für die einzelnen Monate jeweils die Bestellungen des letzten Tages anzuzeigen.
Bestelldatum	Jahr([Bestelldatum])= Jahr(Jetzt()) Und Monat([Bestelldatum])= Monat(Jetzt())	Verwendet die Funktionen ***Jahr*** und ***Monat*** sowie den Operator ***Und***, um die Bestellungen des aktuellen Jahres und Monats anzuzeigen.

Behandlung von leeren Feldern		
Region	Ist Null	Zeigt die Bestellungen der Kunden an, für die das Feld Region Null enthält (leer ist).
Region	Ist Nicht Null	Zeigt die Bestellungen der Kunden an, für die das Feld Region einen Wert enthält.
Fax	" " oder Ist Null	Zeigt die Bestellungen der Kunden an, die kein Faxgerät besitzen. Im Feld Telefax ist dann an Stelle entweder ein Nullwert oder eine leere Zeichenfolge enthalten.

Desktop-Video 6-Tabellenerstellung

- View the Desktop-Video „6-Tabellenerstellung.avi“.
- Create your own database „Artikelverwaltung.mdb“ with the table „Artikel“ analogous to the example in the Desktop-Video. See for that purpose the description of the table „Artikel“and also the possible field data types on the next slides.

Table „Artikel“

The table „Artikel“ should

- manage information on articles
- give each article a net selling price
- assign an article description and an entry date to each article
- record the number of items on stock
- categorize articles (e.g. vegetables, fruits...)
- contain additional address information of suppliers for repeat orders

Artikelnummer	Eindeutige Kennzeichnung zur Unterscheidung aller Artikel in der Tabelle
Artikelname	Hauptkriterium zur Sortierung
Verpackungsart	Zur weiteren Unterscheidung, wenn es den gleichen Artikel in unterschiedlichen Verpackungen gibt.
Artikelbeschreibung	Detaillierte Beschreibung des Artikels
Einzelpreis	Preis des Artikels
Kategorie	Zur Gruppierung notwendig
Lagerbestand	Für Lagerwerte und Inventur
Mindestbestand	Wieviel muss im Lager sein?
Aufgenommen am	Datum der Eingabe
Lieferantenname	Wer liefert diesen Artikel?
Strasse und Nr	Lieferanten-Adresse zur Nachbestellung
Postleitzahl	Lieferanten-Adresse zur Nachbestellung
Ort	Lieferanten-Adresse zur Nachbestellung

Data field types

Datentyp	Verwendet für	Grösse
Text (Voreinstellung)	Text oder Kombinationen aus Text und Zahlen und auch Zahlen, die keine Berechnungen erfordern, z.B. Telefonnummern.	Bis zu 255 Zeichen oder die in der Eigenschaft Feldgrösse eingestellte Länge (je nachdem, welcher Wert kleiner ist). Access reserviert keinen Platz für nicht genutzte Teile eines Textfelds.
Memo	Langer Text oder Kombinationen aus Text und Zahlen.	Bis zu 64 KB (65'535 Zeichen). Nützlich für Bemerkungen und ähnliches.
AutoWert	Eine eindeutige, fortlaufende Zahl (die jeweils um 1 hochgezählt wird) oder eine Zufallszahl, die von Access zugewiesen wird, wenn ein neuer Datensatz in eine Tabelle eingetragen wird. Felder vom Typ "AutoWert" können nicht aktualisiert werden.	Normalerweise 4 Bytes
Datum/Zeit	Datums- und Zeitwerte für die Jahre 100 bis 9999.	8 Bytes
Zahl: Byte	Speichert Zahlen von 0 bis 255 (keine Bruchzahlen).	1 Byte
Zahl: Integer	Speichert Zahlen von -32'768 bis 32'767 (keine Bruchzahlen).	2 Bytes
Zahl: Long Integer	(Voreinstellung) Speichert Zahlen von -2'147'483'648 bis 2'147'483'647 (keine Bruchzahlen).	4 Bytes
Zahl: Single	Speichert Zahlen von□-3.402823E38 bis□-1.401298E-45 für negative Werte, und von □1.401298E-45 bis 3.402823E38 für positive Werte.	4 Bytes
Zahl: Double	Speichert Zahlen von□-1.79769313486231E308 bis□-4.94065645841247E-324 für negative Werte, und von 1.79769313486231E308 bis 4.94065645841247E-324 für positive Werte.	8 Bytes
Hyperlink	Text oder Kombinationen aus Text und Zahlen, die als Text abgespeichert und als Hyperlink-Adresse (z.B. E-Mail-Adresse oder WWW-Site) verwendet werden.	Bis zu 8'192 Zeichen.
Nachschlage-Assistent	Erstellt ein Feld, das Ihnen ermöglicht, einen Wert aus einer anderen Tabelle oder aus einer Liste von Werten mit Hilfe eines Listenfelds oder eines Kombinationsfelds nachzuschlagen. Durch das Wählen dieser Option wird der Nachschlage-Assistent gestartet, der ein Nachschlagefeld erstellt (siehe Seite 49).	Die gleiche Grösse wie das Primärschlüsselfeld, das zum Nachschlagen benötigt wird, in der Regel 4 Bytes.

Datumsformate	
Standarddatum	(Standardeinstellung) Enthält der Wert nur ein Datum, so wird keine Uhrzeit angezeigt. Enthält der Wert nur eine Uhrzeit, so wird kein Datum angezeigt. Diese Einstellung ist eine Kombination der Einstellungen ***Datum, kurz*** und ***Zeit, lang***. Beispiele: 03.04.93; 17:34:00; 03.04.93 17:34:00
Datum, lang	Entspricht der Einstellung ***Langes Datumsformat*** im Dialogfeld ***Eigenschaften von Ländereinstellungen*** in der Systemsteuerung von Windows. Beispiel: Samstag, 6. April 1996.
Datum, mittel	Beispiel: 03. Apr. 93.
Datum, kurz	Entspricht der Einstellung ***Kurzes Datumsformat*** im Dialogfeld ***Eigenschaften von Ländereinstellungen*** in der Systemsteuerung von Windows. Beispiel: 06.04.96. Warnung: Bei der Einstellung ***Kurzes Datum*** wird davon ausgegangen, dass Daten zwischen dem 1.1.00 und 31.12.29 Daten des 21. Jahrhunderts darstellen (d.h. dass die Jahre als 2000 und 2029 angesehen werden). Bei Daten zwischen dem 1.1.30 und dem 31.12.99 wird davon ausgegangen, dass es sich um Daten des 20. Jahrhunderts handelt (d.h. dass die Jahre als 1930 und 1999 angesehen werden).
Zeit, lang	Entspricht der Einstellung auf der Registerkarte Zeitformat im Dialogfeld Eigenschaften von Ländereinstellungen in der Systemsteuerung von Windows. Beispiel: 17:34:23.
Zeit, 12Std	Beispiel: 05:34.
Zeit, 24Std	Beispiel: 17:34.

Zahlenformate	
Allgemeine Zahl	(Standardeinstellung) Zeigt die Zahl wie eingegeben an.
Währung	Verwendet das Tausender-Trennzeichen; zeigt negative Zahlen in Klammern an. Die Standardeinstellung der Eigenschaft DecimalPlaces ist 2.
Festkommazahl	Zeigt mindestens eine Ziffer an. Die Standardeinstellung der Eigenschaft DecimalPlaces ist 2.
Standardzahl	Verwendet das Tausender-Trennzeichen. Die Standardeinstellung der Eigenschaft DecimalPlaces ist 2.
Prozentzahl	Multipliziert den Wert mit 100 und fügt ein Prozentzeichen (%) an. Die Standardeinstellung der Eigenschaft DecimalPlaces ist 2.
Exponenzialzahl	Verwendet die Standardschreibweise für Exponenzialzahlen.

Formats

Format	Eingabe	Darstellung
>	Access ACCESS Access	ACCESS ACCESS ACCESS
<	Access ACCESS Access	Access Access Access
@;"Unbekannt"	Access (nichts eingegeben)	Access Unbekannt
"Nr-"@@@\.@@\.@@	1234567 123456	Nr-123.45.67 Nr-12.34.56
!"Nr-"@@@\.@@\.@@	1234567 123456	Nr-123.45.67 Nr-123.45.6
"Nr-"&&@\.@@\.@@	1234567 123456 12345	Nr-123.45.67 Nr-12.34.56 Nr-1.23.45
"Total"*_#'##0	1265.25 0	Total _____ 1'265.25 Total _____ 0
"Nr. "0000	123 12	Nr. 0123 Nr. 0012
0;(0);"Null";"Leer"	123 -123 0 (Nicht eingegeben)	123 (123) Null Leer
0*\	123	123
t. mmmm jjjj	15.4.95	15. März 1995
Tttt, t. mmm. jj*\	15.4.95	Samstag, 15. Mär. 95
ttt, tt.mm.jj hh:nn:ss	15.4.95 10:09:58	Sa, 15.04.95 10:09:58
h:nn:ss	10:09:58	10:09:58
h "Uhr" mm	10:09:58	10 Uhr 09

Desktop-Video 7-Beziehungen

- View the Desktop-Video „7-Beziehungen.avi“.
- Repeat the demonstrated steps.
- Create the new tables „Lieferant“ and „Kategorie“ and link these rationally with relationships.
- Pay attention to referential integrity.
- Use the Lookup-Assistants.

Desktop-Video 8-Abfragenerstellung

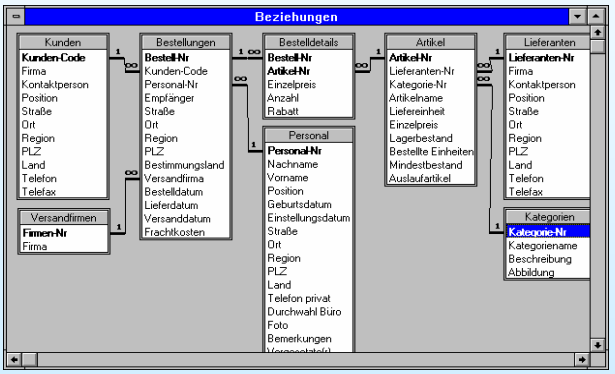
- View the Desktop-Video „ 8-Abfragen.avi“.
- Repeat the demonstrated steps.
- Study the slide „Query Criteria“ and solve the two assignments on slide „Simple Queries“.

Query Criteria

Attribute	Criteria	Result
Stadt	"London"	Bestellungen mit Zielort "London"
Stadt	"London" Oder "Hedge End"	Bestellungen mit Zielort "London" oder "Hedge End"
Versanddatum	=#02.02.1994#	Bestellungen, die am 2. Februar 1994 verschickt wurden
Versanddatum	Zwischen #05. Jan. 94# Und #10. Jan. 94#	Bestellungen, die nicht vor dem 5. Januar 1994 und nicht nach dem 10. Januar 1994 verschickt wurden
Bestimmungsland	In ("Kanada"; "Großbritannien")	Bestellungen mit Bestimmungsland "Kanada" oder "Großbritannien"
Bestimmungsland	Nicht "USA"	Bestellungen, deren Bestimmungsland nicht die "USA" sind
Bestelldatum	< Datum( )- 30	Bestellungen, die mehr als 30 Tage alt sind
Bestelldatum	Jahr([Bestelldatum])=1994	Bestellungen mit Bestelldatum im Jahr 1994
Bestelldatum	DatTeil("q"; [Bestelldatum])=4	Bestellungen aus dem 4. Quartal des laufenden Jahres
Bestelldatum	DatSeriell(Jahr([Bestelldatum]); Monat([Bestelldatum])+1; 1)-1	Bestellungen vom letzten Tag jeden Monats
Bestelldatum	Jahr([Bestelldatum])=Jahr(Jetzt()) Und Monat([Bestelldatum])=Monat(Jetzt())	Bestellungen aus dem laufenden Jahr und Monat
Empfänger	Wie "S*"	Bestellungen für Kunden, deren Namen mit dem Buchstaben "S" beginnen
Empfänger	Wie "*Importe"	Bestellungen für Kunden, deren Namen mit dem Wort "Importe" enden
Empfänger	Wie "[A-D]*"	Bestellungen für Kunden, deren Namen mit den Buchstaben "A" bis "D" (einschließlich) beginnen
Bestell-Nr	Rechts([Bestell-Nr]; 2)="99"	Bestellungen mit Bestellnummern, die mit den Ziffern "99" enden

Simple Queries

1. In Nordwind, there is a corporation based in London whose name starts with letter ‚C'. Find the complete company name.
2. Create a list of customers, who are based in London **and** whose name starts with ‚C' **or** who are based somwhere else starting with ‚Rio'.

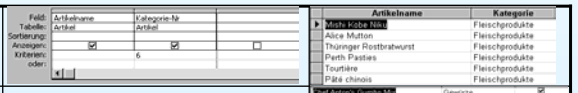
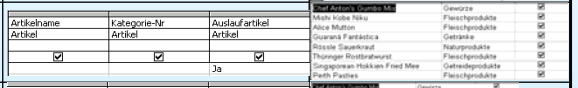
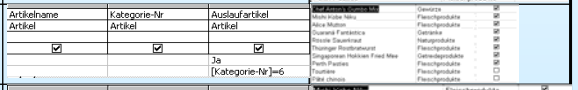
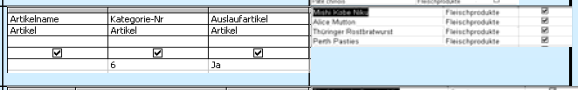


Abfragen - über mehrere Tabellen

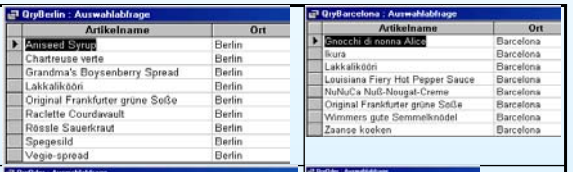
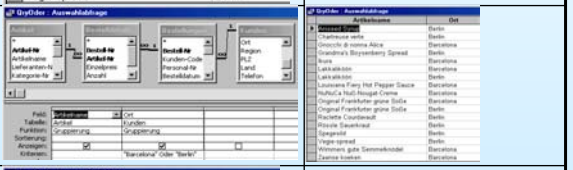
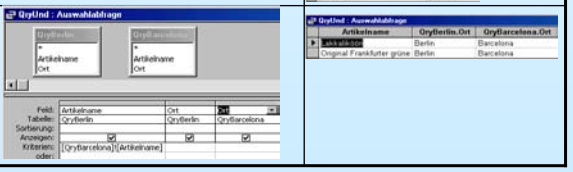
1. Find the items on stock of spices, their quantities and their price. Calculate, too, the worth of stock per spice-article!
2. Find the 5 most expensive spices on stock!
3. Find the suppliers of articles with the highest worth of stock!
4. Find the worth of the stock according to categories!
5. Find the total sales from existing orders, which the individual employees obtained.
6. Find the total sales from existing orders, based on individual articles.
7. Which customer generates the highest turnover?
8. What percentage of number of orders are handled via the shipper „Speedy Express"
9. What percentage of value of orders are handled via the shipper „Speedy Express"
10. What articles are being sold towards Switzerland or Germany

11. Limit the query above to customers from Germany who ordered within the year 1998.
12. Limit your query to customers, who are not based in Germany and who ordered within the year 1998.
13. Limit your query to customers, who are named "Richter Supermarkt„ and who ordered in May 1998.
14. Limit your query to customers, who are either named "Richter Supermarkt„ or who ordered in May 1998.

AND, OR, XOR - One Table

1	Show all articles (Field Artikelname) which are classified as meat products!	
2	Show all articles which are classified as expired products!	
3	Show all articles that are classified as expired products OR as meat products!	
4	Show all articles that are classified as expired products AND meat products!	
5	Show all articles that are classified EITHER as expired products OR as meat products!	

OR - Several Tables

1	Show all articles that are sold in Berlin! → QryBerlin Show all articles that are sold in Barcelona! → QryBarcelona	
2	Show all articles that are sold in Berlin OR Barcelona! → QryOder	
3	Show all articles tha are sold in Berlin AND Barcelona! → QryUnd *Note: QryUnd* consists of *QryBerlin and QryBarcelona! Achtung: QryUnd setzt sich aus QryBerlin und QryBarcelona zusammen!*	

Further Queries Types

- **Update-Query:** Discount on Seafood:
  Your last query showed that the stock of perishable seafood is valued at a significant Fr. 13'010.35. Thus, you decide giving a 10% discount on the unit price of all categories of seafood. Verify the success of your query by running once more the query "Warenwert des Lagers pro Artikelkategorie„.
- **Parametric Query:** Consumption Behavior in Countries:
  Create a query that shows, based on a country given by a user (company headquarters of the customer), what the total sales is in form of existing orders.

Forms 1

- View the following desktop videos:
  - 9-Formulare1.avi
  - 9-Fomulare2.avi
- Create a form for the table „Mitarbeiter“ using the Form-Assistant.

Forms 2

- View the following desktop video:
  - 9-Formulare3.avi
- Repeat the shown examples.
- Create you own example with a main form and a sub-form.
- Which tables in Nordwind can be used as a basis for sub-forms?

Forms 3

- View the following desktop video:
  - 9-Formulare4.avi
- Take your former exercise and change the default layout at your will.

Report 1

- View the following desktop videos:
  - 10-Berichte1.avi
  - 10-Berichte2.avi
- Repeat the shown examples.
- Create your own example within Nordwind.

Report 2

- View the following desktop video:
  - 10-Formulare3.avi
- Repeat the shown examples.
- Try something similar based on your own query.

Report 3

- View the following desktop video:
  - 10-Formulare4.avi
- Repeat the shown examples.
- Create mailing lables for table „Personal“.

ODBC-Connection

- View the following desktop video:
  - 11-odbc1.avi

Table of Content

- Organization
- Data Modeling
- Data Implementation (Access)
- Structured Query Language (SQL)

SQL Demo

- Click on Query
- Select „New“ to create a query.
- Under „View“ select „SQL“.
- You can edit your SQL-code in the given window.
- Select „Data View“ to see the computed result set.
- Select „SQL“ to return to edit mode.

SELECT, ORDER BY, DISTINCT

- SQL Query-Statements have the form:
  `SELECT <attributes> FROM <tables>;`
- Sorting of the result set can be obtained (ASC, DESC) with the add-on ORDER BY.
- If there are rows with equal values in the same column, SELECT DISTINCT limits the result set to one row only.
- SQL doesn‘t recognize the character ‚-‘ (hyphen). Please mask this character in brackets, e.g. [Lieferanten-Nr].
- SQL is not case sensitive.

Exercise: SELECT, ORDER BY, DISTINCT

Based on Table „Artikel“ in „Nordwind“.

Tasks:

a) All rows and columns.

b) All rows for attributes Artikelname, Lagerbestand, and Mindestbestand

c) All rows for attributes Artikelname, Lagerbestand, and Mindestbestand sorted ascending by Artikelname

d) All rows for attributes Artikelname, Lagerbestand, and Mindestbestand sorted descending by Lagerbestand

e) All Lieferanten (Lieferanten-Nr) available in this table. Show each one only once.

Hint:

ORDER BY can be used for more than one attribute

Example:

... ORDER BY [kategorie-nr] ASC, artikelname DESC

WHERE

Tasks:

a) All „Artikel“ of kind „Getränke“ ascending by „Artikelname“

b) Items on stock of Artikel ´Outback Lager´

c) All Artikel with more than with more than 100 units on stock.

d) All Artikel which cost more than 1.- sorted by “Einzelpreis”

Hints: Available operators

=	equal
<>	not equal
>	more than
<	less than
>=	more or equal
<=	less or equal

```
SELECT * FROM
mitarbeiter
WHERE  name = 'Müller'
ORDER  BY vorname ASC;
```


IN, BETWEEN, LIKE

- Always with WHERE
- ... WHERE name IN ('Müller', 'Kuntz');
- ... WHERE lohn BETWEEN 4000 AND 5000
- ... WHERE name LIKE 'M*er'

Exercise IN, BETWEEN, LIKE

Tasks:

a) All “Artikel” which belong to “Kategorie” 1 or 2

b) All „Artikel“ which cost between 10.- and 20.-

c) Names of „Artikel“ whose names start with ‚C’ and end with ‚g’

Hints: IN, BETWEEN, and LIKE allow their negation by using NOT.

example:

```
SELECT *
FROM   mitarbeiter
WHERE  lohn NOT BETWEEN 4000 AND 5000;
```


NULL, AND, OR

- ... WHERE lohn IS NULL;
- ... WHERE lohn IS NOT NULL;
- Expression1 AND Expression2 OR Expression3
- See next slide.

Exercise NULL, AND, OR

Tasks:

a) All „Lieferanten“ who do not have a „Faxnummer“

b) All „Lieferanten“ with known „Homepage“

c) All „Lieferanten“ with known „Homepage“ and known “Faxnummer“

Hints:

The default operators are ranked as follows:

1. NOT
2. AND
3. OR

You can influence the default ranking by setting paratheses.

Example:

(name = 'Müller' OR name = 'Moser') AND NOT lohn < 3000

JOIN Operation

- When two or more tables are combined in an SQL instruction, this is called a JOIN operation.
- The following example illustrates the JOIN operation with the aid of both tables:
  - MITARBEITER: Mitarbeiternr, Name, Vorname, Gehalt, GebDatum
  - KIND: Mitarbeiternr, Kindnr, Vorname, GebDatum, Geschlecht
- SELECT k.vorname, name
  FROM mitarbeiter AS m, kind AS k
  WHERE m.mitarbeiternr = k.mitarbeiternr;
- Result:
  All children are listed with first and family names.
- Explanations
  - If a field occurs in several tables (e.g. VORNAME), it must be clearly identified. This can be attained in the most simple way through the so-called correlation names.
  - The correlation names is an abbreviation, which is written with the command AS in connection with the table names.
  - FROM mitarbeiter AS M, kind AS K
  - Now, each column name can be clearly characterized by placing the correlation name before it.
  - SELECT k.vorname, m.name
  - Two tables can only be connected when both show the same key. In the example, MITARBEITERNR is recorded as the key in the MITARBEITER table and also in the KIND table.

Exercise JOIN Operation

- In the Nordwind database, some customers and suppliers are based in the same „Ort“. Give the respective company names of these customers and suppliers and the „Ort“ concerned.

Column Function

- SQL has the column functions
  - AVG Compute average value of the column
  - SUM Compute sum total of the column
  - COUNT Counts number of selected rows in the column
  - MAX Gives the highest value in the column
  - MIN Gives the lowest value in the column
- These functions can only be written in column lists (SELECT...)

```
SELECT COUNT(*)    AS Anzahl,
       SUM(lohn) AS Summe,
       AVG(lohn) AS Durchschnitt,
       MIN(lohn) AS [Tiefster Lohn],
       MAX(lohn) AS [Höchster Lohn]
FROM   mitarbeiter;
```

- Result
  The above example shows all possible column functions. As a result, you obtain the number of employees, sum of all salaries, average salary, and the lowest and highest salaries With AS ..., the title of the respective function results could be determined.
- Explanations
  - When you expand the SELECT statement into the WHERE clause, you can correspondingly limit the analysis.
  - In the following example, all employess with unknown salaries are counted.
  - SELECT COUNT(*) FROM mitarbeiter WHERE lohn IS NULL;
  - With the functions SUM, AVG, MIN and MAX, fields with NULL values are ignored.
  - MIN and MAX are also applied to non-numeric fields.

Exercise Column Function

- You are interested only in orders that are processed with discounts. Find the lowest, the highest as well as the average discounts.

Forming groups

- GROUP BY makes it possible to make computations on individual groups within a column. Without the GROUP BY, the aggregate functions like SUM will compute all the values in a column.
- As an example, to group the KIND table according to MITARBEITERNR, you would write GROUP BY mitarbeiternr
- You can display the grouped table as:

1	David	30.04.1987	M
1	Natalie	12.02.1990	W
2	Heinz	14.08.1988	M
3	Sabine	13.01.1970	W
3	Rolf	06.12.1973	M
3	Sandra	16.09.1975	W
6	Edwin	14.12.1969	M
6	Olaf	07.06.1973	M
7	Heinz	13.02.1965	M

- Example
  SELECT mitarbeiternr, COUNT(*) FROM kind GROUP BY mitarbeiternr;
- Result
  Number of children per personnel number is specified.

Mitarbeiter-Nr.	Anzahl Kinder
1	2
2	1
3	3
6	2
7	1

Exercise on Forming Groups

- Give the gross value (sum of all „Bestellpositionen“) for each order.

HAVING Clause

- The HAVING clause can further restrict the results of a GROUP BY function. Using the following instructions, only those employees with two of more children will be displayed:

```
SELECT   mitarbeiternr, COUNT(*)
FROM     kind
GROUP BY mitarbeiternr
HAVING   COUNT(*) >= 2;
```

Sub-queries

- SUM, AVG, MIN and MAX can only be specified in column lists. With this limitation, how can we answer the following questions?
- Who among the employees earns more than the average salary?
- Solve the problem with the help of a sub-query. Observe the following example.

1. We first compute the average salary and then save this query under the name LOHN.

```
SELECT   AVG(lohn) AS Durchschnitt
FROM     mitarbeiter;
```

2. Then we use the query LOHN with the average salary as sub-query. Only those employees who earn above the average will now be showed.

```
SELECT   mitarbeiternr, name, vorname, lohn
FROM     mitarbeiter, lohn AS g
WHERE    lohn > g.durchschnitt;
```


Exercise Sub-query

- Compute the average amount of orders of all customers.