Hochschule Esslingen
University of Applied Sciences

Module “Information Systems (InfoSys)" Organisational Issues

Prof. Dr. Schoop
University of Applied Sciences Esslingen

Winter Term 2016/17

- Information Systems
  - Mandatory module
  - 6th semester SWM, SWT
  - 3 SWS lectures
  - 1 SWS lab (project)
- Lecturer:
  - Prof. Dr. Dominik Schoop
  - Room: F1.354
  - E-Mail: dominik.schoop@hs-esslingen.de
  - Telephone: 0711 397 4467
  - Consultation time: each Wednesday, 12:45 – 13:30

Lab (Project)

- Project work in groups of 4 people
- Regular project meetings of the individual groups (to be arranged by group)
- Regular project meetings with „customer" (= lecturer)
- Task:
  application of the Database System Development Lifecycle (DSDLC) (presented in the lecture) to a real world problem
- Estimated effort: approx. 60 hours (per person)

Lecture and Project Documents

Documents for lecture (slides etc.) and project are available under

W:\Dozenten\Schoop\InfoSys

Exam and Grades

- Written exam (90 minutes) during examination period
- Closed book exam
- Exam language: German and English (bilingual)
- Grading of project according to participation in meetings, project documentation and project results (pass/fail)

Literatur

- Thomas Connolly, Carolyn Begg:
  Database Systems – A Practical Guide to Design, Implementation, and Management, Pearson, 5. Aufl., 2010

  (available in HE library under the key *Ni Con*)

Hochschule Esslingen
University of Applied Sciences

Chapter 10

Database System
Development Lifecycle

Chapter 10 - Objectives

Hochschule Esslingen
University of Applied Sciences

- Main components of an information system.
- Main stages of database system development lifecycle.
- Main phases of database design:
  conceptual, logical, and physical design.
- Benefits of CASE tools.
- How to evaluate and select a DBMS.
- Distinction between data administration and database administration.
- Purpose and tasks associated with data administration and database administration.

Software Depression

- Last few decades have seen proliferation of software applications, many requiring constant maintenance involving:
  - correcting faults,
  - implementing new user requirements,
  - modifying software to run on new or upgraded platforms.
- Effort spent on maintenance began to absorb resources at an alarming rate.
- As a result, many major software projects were
  - late,
  - over budget,
  - unreliable,
  - difficult to maintain,
  - performed poorly.

Software Depression

Hochschule Esslingen
University of Applied Sciences

- 1960s, led to ‘software crisis’, now referred to as the ‘software depression’.
- Major reasons for failure of software projects includes:
  - lack of a complete requirements specification;
  - lack of appropriate development methodology;
  - poor decomposition of design into manageable components.
- Structured approach to development was proposed called Information Systems Lifecycle (ISLC).

Information System (IS)

Resources that enable collection, management, control, and dissemination of information throughout an organization.

- Database is fundamental component of IS, and its development/usage should be viewed from perspective of the wider requirements of the organization.

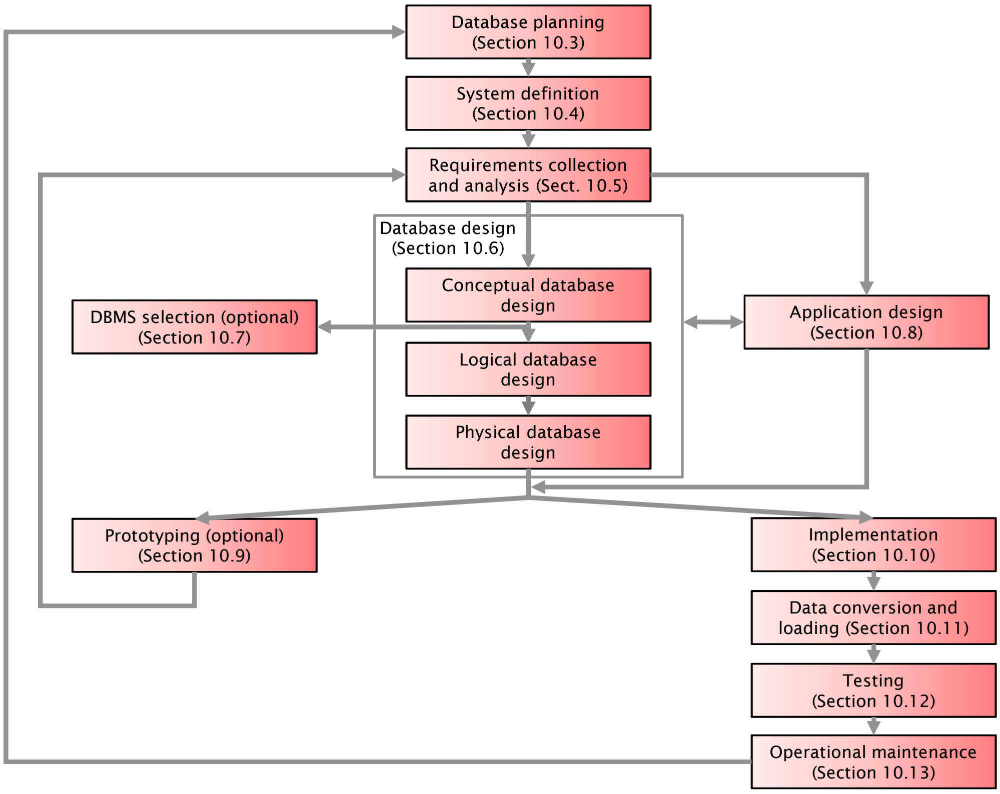
Database System Development Lifecycle

Hochschule Esslingen
University of Applied Sciences

1. Database planning
2. System definition
3. Requirements collection and analysis
4. Database design
5. DBMS selection (optional)
6. Application design
7. Prototyping (optional)
8. Implementation
9. Data conversion and loading
10. Testing
11. Operational maintenance

Stages of the Database System Development Lifecycle

Hochschule Esslingen
University of Applied Sciences



Database Planning

Hochschule Esslingen
University of Applied Sciences

- Management activities that allow stages of database system development lifecycle to be realized as efficiently and effectively as possible.

- Must be integrated with overall IS strategy of the organization.

Database Planning – Mission Statement

- Mission statement for the database project defines major aims of database application.
- Those driving database project normally define the mission statement.
- Mission statement helps clarify purpose of the database project and provides clearer path towards the efficient and effective creation of required database system.

Database Planning – Mission Objectives

Hochschule Esslingen
University of Applied Sciences

- Once mission statement is defined, mission objectives are defined.
- Each objective should identify a particular task that the database must support.
- May be accompanied by some additional information that specifies the work to be done, the resources with which to do it, and the money to pay for it all.

Database Planning – Standards

- Database planning should also include development of standards that govern:
  - how data will be collected,
  - how the format should be specified,
  - what necessary documentation will be needed,
  - how design and implementation should proceed.

System Definition

Hochschule Esslingen
University of Applied Sciences

- Describes scope and boundaries of database system and the major user views.

- User view defines what is required of a database system from perspective of:
  - a particular job role (such as Manager or Supervisor) or
  - enterprise application area (such as marketing, personnel, or stock control).

System Definition

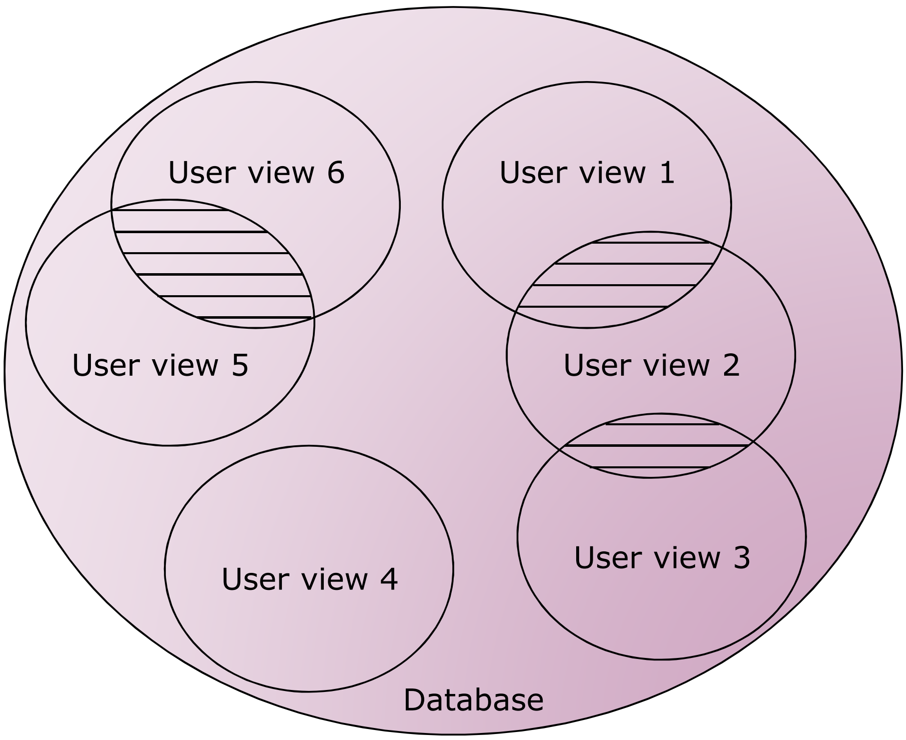
- Database application may have one or more user views.

- Identifying user views helps ensure that no major users of the database are forgotten when developing requirements for new system.

- User views also help in development of complex database system allowing requirements to be broken down into manageable pieces.

Representation of a Database System with Multiple User Views

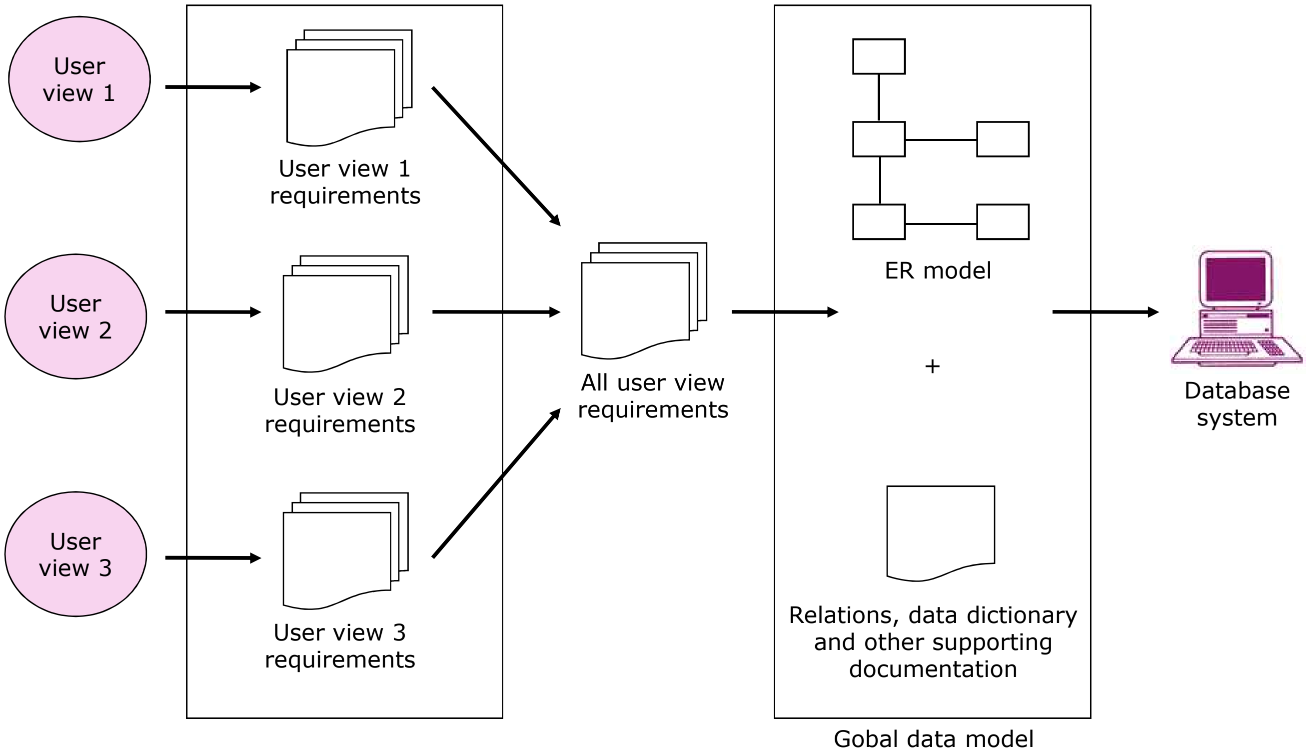
Hochschule Esslingen
University of Applied Sciences



Database system

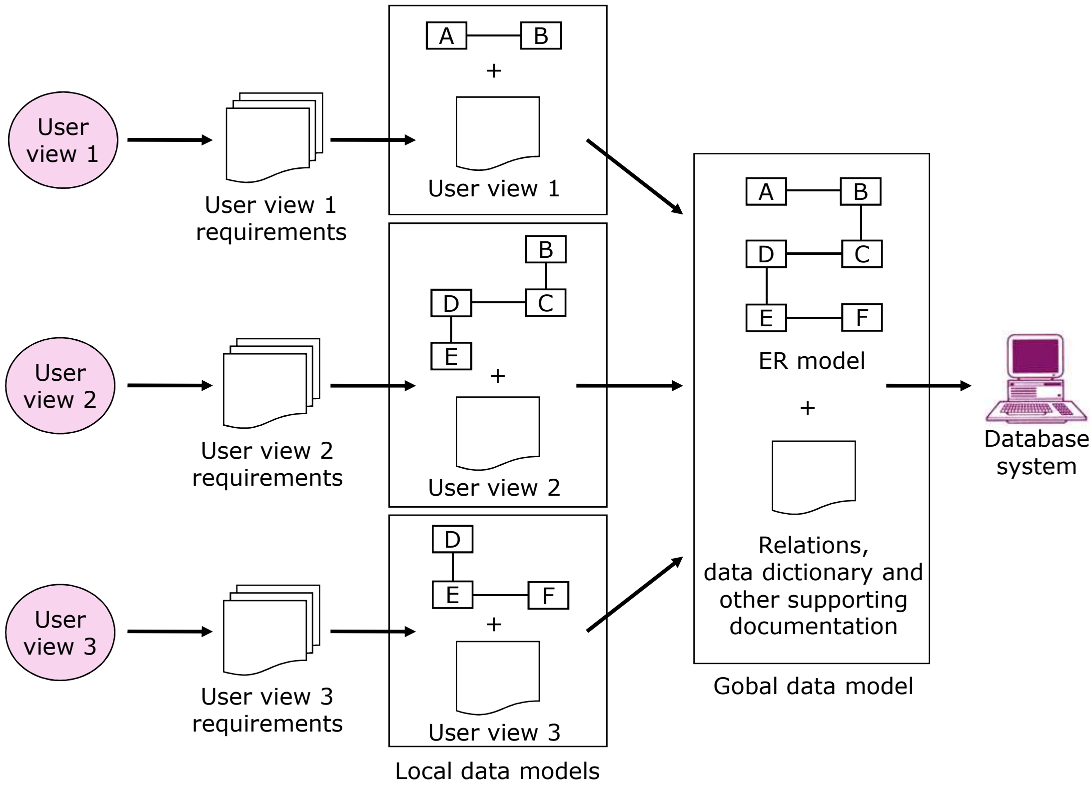
- Process of collecting and analyzing information about the part of organization to be supported by the database system, and using this information to identify users' requirements of new system.
- Information is gathered for each major user view including:
  - a description of data used or generated;
  - details of how data is to be used/generated;
  - any additional requirements for new database system.
- Information is analyzed to identify requirements to be included in new database system. Described in the requirements specification.

- Another important activity is deciding how to manage the requirements for a database system with multiple user views.
- Three main approaches:
  - centralized approach;
  - view integration approach;
  - combination of both approaches.
- *Centralized approach*
  - Requirements for each user view are merged into a single set of requirements.
  - A data model is created representing all user views during the database design stage.



- *View integration approach*
  - Requirements for each user view remain as separate lists.
  - Data models representing each user view are created and then merged later during the database design stage.
- Data model representing single user view (or a subset of all user views) is called a *local data model*.
- Each model includes diagrams and documentation describing requirements for one or more but not all user views of database.
- Local data models are then merged at a later stage during database design to produce a *global data model*, which represents *all* user views for the database.

View Integration Approach to Managing Multiple User Views



Database Design

Hochschule Esslingen
University of Applied Sciences

- Process of creating a design for a database that will support the enterprise's mission statement and mission objectives for the required database system.

- Main approaches include:
  - Top-down
  - Bottom-up
  - Inside-out
  - Mixed

Database Design

- Main purposes of data modeling include:
  - to assist in understanding the meaning (semantics) of the data
  - to facilitate communication about the information requirements.

- Building data model requires answering questions about entities, attributes and relationships.

- A data model ensures we understand:
  - each user's perspective of the data
  - nature of the data itself, independent of its physical representations
  - use of data across user views.

Criteria to Produce an Optimal Data Model

- Structural validity
  - Consistency with the way the enterprise defines and organizes information.

- Simplicity
  - Ease of understanding by IS professionals and non-technical users.

- Expressibility
  - Ability to distinguish between different data, relationships between data, and constraints.

- Nonredundancy
  - Exclusion of extraneous information; in particular, the representation of any one piece of information exactly once.

Criteria to Produce an Optimal Data Model

- Shareability
  - Not specific to any particular application or technology and thereby usable by many.
- Extensibility
  - Ability to evolve to support new requirements with minimal effect on existing users.
- Integrity
  - Consistency with the way the enterprise uses and manages information.
- Diagrammatic representation
  - Ability to represent a model using an easily understood diagrammatic notation.

Hochschule Esslingen
University of Applied Sciences

Database Design

- Three phases of database design:
  - Conceptual database design
  - Logical database design
  - Physical database design.

Conceptual Database Design

- Process of constructing a model of the data used in an enterprise, independent of all physical considerations.

- Data model is built using the information in users' requirements specification.

- Conceptual data model is source of information for logical design phase.

Logical Database Design

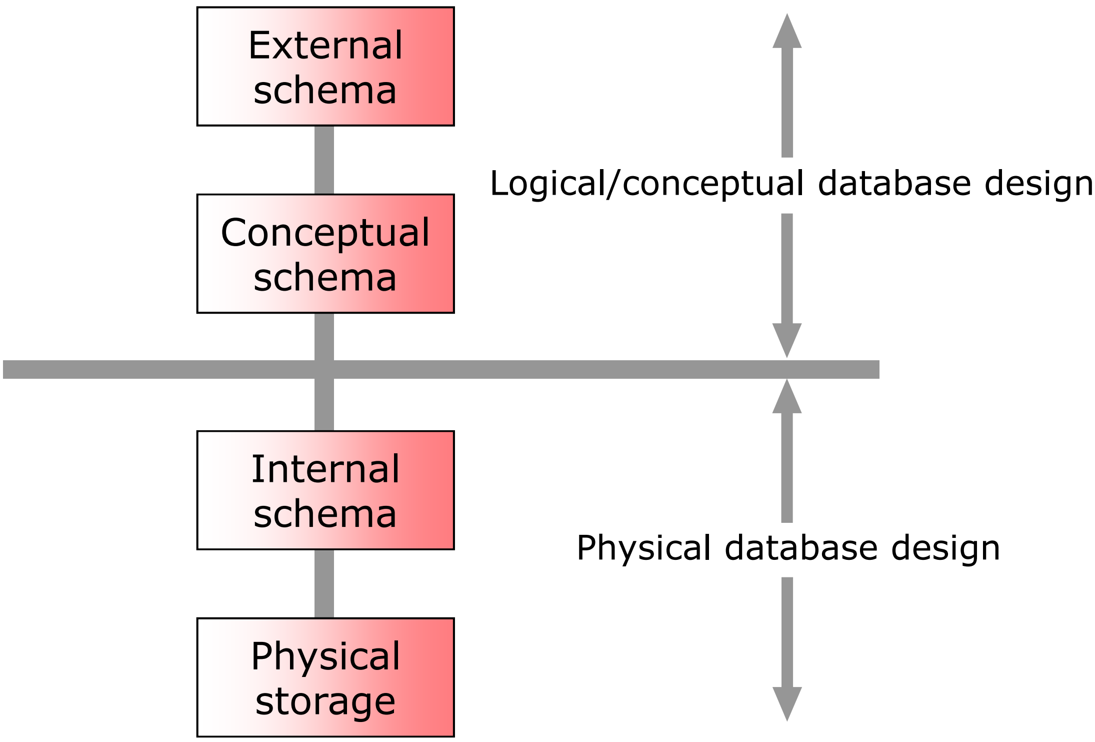
- Process of constructing a model of the data used in an enterprise based on a specific data model (e.g. relational), but independent of a particular DBMS and other physical considerations.

- Conceptual data model is refined and mapped on to a logical data model.

- Process of producing a description of the database implementation on secondary storage.

- Describes base relations, file organizations, and indexes used to achieve efficient access to data. Also describes any associated integrity constraints and security measures.

- Tailored to a specific DBMS system.



DBMS Selection

- Selection of an appropriate DBMS to support the database system.
- Undertaken at any time prior to logical design provided sufficient information is available regarding system requirements.
- Main steps to selecting a DBMS:
  - define Terms of Reference of study;
  - shortlist two or three products;
  - evaluate products;
  - recommend selection and produce report.

DBMS Evaluation Features

Table 9.4 Features for DBMS evaluation.

Data definition	Physical definition
Primary key enforcement	File structures available
Foreign key specification	File structure maintenance
Data types available	Ease of reorganization
Data type extensibility	Indexing
Domain specification	Variable length fields/records
Ease of restructuring	Data compression
Integrity controls	Encryption routines
View mechanism	Memory requirements
Data dictionary	Storage requirements
Data independence	
Underlying data model	
Schema evolution	

DBMS Evaluation Features

Accessibility	Transaction handling
Query language: SQL2/SQL:2003/ODMG compliant	Backup and recovery routines
Interfacing to 3GLs	Checkpointing facility
Multi-user	Logging facility
Security	Granularity of concurrency
– Office Access controls	Deadlock resolution strategy
– Authorization mechanism	Advanced transaction models
	Parallel query processing

Utilities	Development
Performance measuring	4GL/5GL tools
Tuning	CASE tools
Load/unload facilities	Windows capabilities
User usage monitoring	Stored procedures, triggers, and rules
Database administration support	Web development tools

DBMS Evaluation Features

Hochschule Esslingen
University of Applied Sciences

Other features	
Upgradability	Interoperability with other DBMSs and other systems
Vendor stability	Web integration
User base	Replication utilities
Training and user support	Distributed capabilities
Documentation	Portability
Operating system required	Hardware required
Cost	Network support
Online help	Object-oriented capabilities
Standards used	Architecture (2- or 3-tier client/server)
Version management	Performance
Extensibile query optimization	Transaction throughput
Scalability	Maximum number of concurrent users
Support for analytical tools	XML support

Example - Evaluation of DBMS Product

Table 9.5 Analysis of features for DBMS product evaluation.

DBMS: Sample product
Vendor: Sample vendor

Physical Definition Group

Features	Comments	Rating	Weighting	Score
File structures available	Choice of 4	8	0.15	1.2
File structure maintenance	NOT self-regulating	6	0.2	1.2
Ease of reorganization		4	0.25	1.0
Indexing		6	0.15	0.9
Variable length fields/records		6	0.15	0.9
Data compression	Specify with file structure	7	0.05	0.35
Encryption routines	Choice of 2	4	0.05	0.2
Memory requirements		0	0.00	0
Storage requirements		0	0.00	0
Totals		41	1.0	**5.75**
Physical definition group		5.75	0.25	**1.44**

Application Design

- Design of user interface and application programs that use and process the database.
- Database design and application design are parallel activities.
- Includes two important activities:
  - transaction design
  - user interface design.

Application Design - Transactions

- An action, or series of actions, carried out by a single user or application program, which accesses or changes content of the database.
- Should define and document the high-level characteristics of the transactions required.
- Important characteristics of transactions:
  - data to be used by the transaction
  - functional characteristics of the transaction
  - output of the transaction
  - importance to the users
  - expected rate of usage
- Three main types of transactions: retrieval, update, and mixed.

Implementation

- Physical realization of the database and application designs.
  - Use DDL to create database schemas and empty database files.
  - Use DDL to create any specified user views.
  - Use 3GL or 4GL to create the application programs. This will include the database transactions implemented using the DML, possibly embedded in a host programming language.

Data Conversion and Loading

- Transferring any existing data into new database and converting any existing applications to run on new database.

- Only required when new database system is replacing an old system.
  - DBMS normally has utility that loads existing files into new database.

- May be possible to convert and use application programs from old system for use by new system.

Testing

- Process of running the database system with intent of finding errors.

- Use carefully planned test strategies and realistic data.

- Testing cannot show absence of faults; it can show only that software faults are present.

- Demonstrates that database and application programs appear to be working according to requirements.

Testing

- Should also test usability of system.
- Evaluation conducted against a usability specification.
- Examples of criteria include:
  - Learnability
  - Performance
  - Robustness
  - Recoverability
  - Adaptability

Operational Maintenance

Hochschule Esslingen
University of Applied Sciences

- Process of monitoring and maintaining database system following installation.
- Monitoring performance of system.
  - if performance falls, may require tuning or reorganization of the database.
- Maintaining and upgrading database application (when required).
- Incorporating new requirements into database application.

CASE Tools

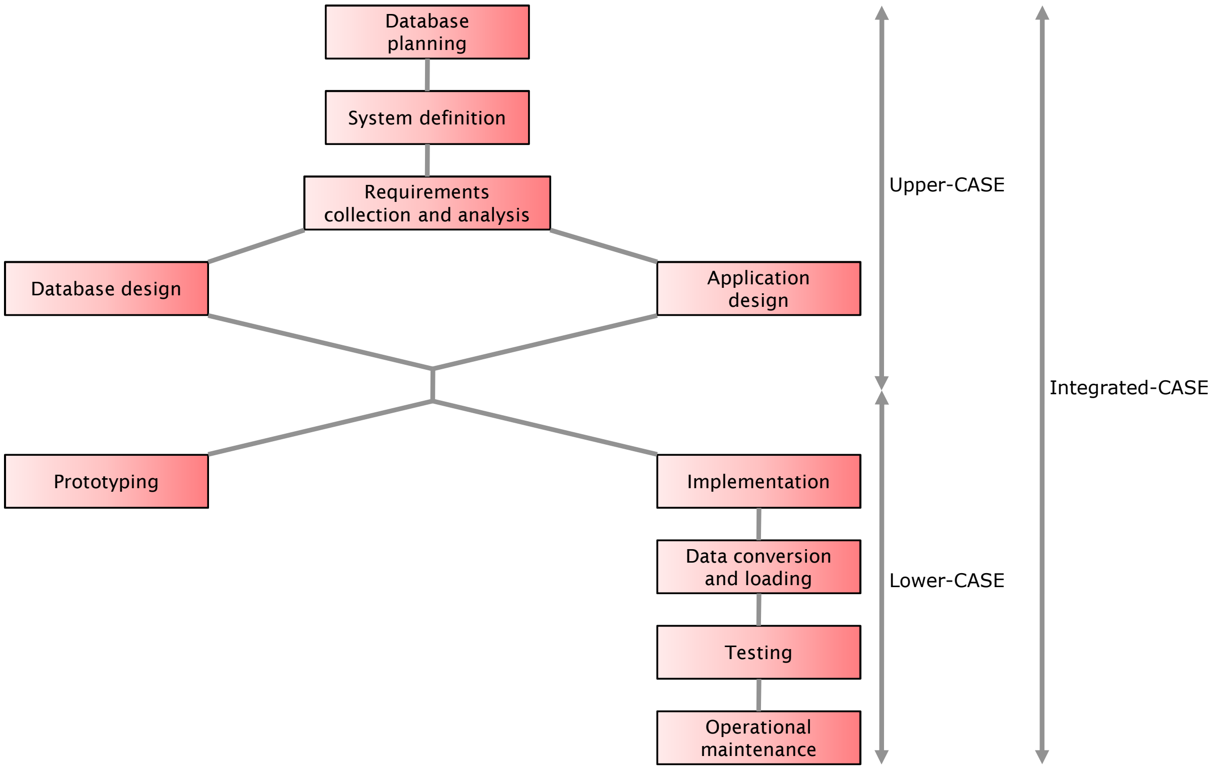
- Support provided by CASE tools include:
  - data dictionary to store information about database system's data
  - design tools to support data analysis
  - tools to permit development of corporate data model, and conceptual and logical data models
  - tools to enable prototyping of applications

CASE Tools

Hochschule Esslingen
University of Applied Sciences

- Provide following benefits:
  - Standards
  - Integration
  - Support for standard methods
  - Consistency
  - Automation

CASE Tools and Database System Development Lifecycle



Data Administration and Database Administration

Hochschule Esslingen
University of Applied Sciences

- The Data Administrator (DA) and Database Administrator (DBA) are responsible for managing and controlling the corporate data and corporate database, respectively.

- DA is more concerned with early stages of database system development lifecycle and DBA is more concerned with later stages.

Data Administration

Hochschule Esslingen
University of Applied Sciences

- Management of data resource including:
  - database planning
  - development and maintenance of standards, policies and procedures, and conceptual and logical database design.

Database Administration

Hochschule Esslingen
University of Applied Sciences

- Management of physical realization of a database system including:
  - physical database design and implementation,
  - setting security and integrity controls,
  - monitoring system performance, and reorganizing the database.

Summary

- The Database System Development Lifecycle (DBSDLC) provides well defined steps to develop a database system, which is the core of an information system.
- A crucial step after the requirements collection and analysis is the database design (i.e. data modelling).
- We distinguish between conceptual, logical and physical database design.
- All steps of the lifecycle are necessary to obtain a working information system.

Hochschule Esslingen
University of Applied Sciences

Chapter 11

Database Analysis and the DreamHome Case Study

Chapter 11 - Objectives

Hochschule Esslingen
University of Applied Sciences

- When fact-finding techniques are used in the database application lifecycle.
- The types of facts collected in each stage of the database application lifecycle.
- The types of documentation produced in each stage of the database application lifecycle.
- The most commonly used fact-finding techniques.
- How to apply fact-finding techniques to the early stages of the database application lifecycle.
- How to use each fact-finding technique and the advantages and disadvantages of each.
- About a property rental company called DreamHome.

Fact-finding techniques

- It is critical to capture the necessary facts to build the required database application.
- These facts are captured using fact-finding techniques.
- The formal process of using techniques such as interviews and questionnaires to collect facts about systems, requirements, and preferences.

When are fact-finding techniques used?

Hochschule Esslingen
University of Applied Sciences

- Fact-finding used throughout the database application lifecycle. Crucial to the early stages including database planning, system definition, and requirements collection and analysis stages.
- Enables developer to learn about the terminology, problems, opportunities, constraints, requirements, and priorities of the organization and the users of the system.

Table 10.1 Examples of the data captured and the documentation produced for each stage of the database system development lifecycle.

Stage of database system development lifecycle	Examples of data captured	Examples of documentation produced
Database planning	Aims and objectives of database project	Mission statement and objectives of database system
System definition	Description of major user views (includes job roles or business application areas)	Definition of scope and boundary of database application; definition of user views to be supported
Requirements collection and analysis	Requirements for user views; systems specifications, including performance and security requirements	Users' and system requirements specifications
Database design	Users' responses to checking the logical database design; functionality provided by target DBMS	Conceptual/logical database design (includes ER model(s), data dictionary, and relational schema); physical database design

Examples of data captured and documentation produced during the database application lifecycle

Hochschule Esslingen
University of Applied Sciences

Stage of database system development lifecycle	Examples of data captured	Examples of documentation produced
Application design	Users' responses to checking interface design	Application design (includes description of programs and user interface)
DBMS selection	Functionality provided by target DBMS	DBMS evaluation and recommendations
Prototyping	Users' responses to prototype	Modified users' requirements and systems specifications
Implementation	Functionality provided by target DBMS	
Data conversion and loading	Format of current data; data import capabilities of target DBMS	
Testing	Test results	Testing strategies used; analysis of test results
Operational maintenance	Performance testing results; new or changing user and system requirements	User manual; analysis of performance results; modified users' requirements and systems specifications

Fact-Finding Techniques

- A database developer normally uses several fact-finding techniques during a single database project including:
  - examining documentation
  - interviewing
  - observing the organization in operation
  - research
  - questionnaires

Examining documentation

- Can be useful
  - to gain some insight as to how the need for a database arose.
  - to identify the part of the organization associated with the problem.
  - To understand the current system.

Examples of types of documentation that should be examined

Table 10.2 Examples of types of documentation that should be examined.

Purpose of documentation	Examples of useful sources
Describes problem and need for database	Internal memos, e-mails, and minutes of meetings Employee/customer complaints, and documents that describe the problem Performance reviews/reports
Describes the part of the enterprise affected by problem	Organizational chart, mission statement, and strategic plan of the enterprise Objectives for the part of the enterprise being studied Task/job descriptions Samples of completed manual forms and reports Samples of completed computerized forms and reports
Describes current system	Various types of flowcharts and diagrams Data dictionary Database system design Program documentation User/training manuals

Interviewing

- Most commonly used, and normally most useful, fact-finding technique. Enables collection of information from individuals face-to-face.
- Objectives include finding out facts, verifying facts, clarifying facts, generating enthusiasm, getting the end-user involved, identifying requirements, and gathering ideas and opinions.
- There are two types of interviews unstructured and structured.
- Open-ended questions allow the interviewee to respond in any way that seems appropriate.
- Closed-ended questions restrict answers to either specific choices or short, direct responses.

Advantages and disadvantages of interviewing

Hochschule Esslingen
University of Applied Sciences

Table 10.3 Advantages and disadvantages of using interviewing as a fact-finding technique.

Advantages	Disadvantages
Allows interviewee to respond freely and openly to questions	Very time-consuming and costly, and therefore may be impractical
Allows interviewee to feel part of project	Success is dependent on communication skills of interviewer
Allows interviewer to follow up on interesting comments made by interviewee	Success can be dependent on willingness of interviewees to participate in interviews
Allows interviewer to adapt or re-word questions during interview	
Allows interviewer to observe interviewee's body language	

Observing the Organization in Operation

Hochschule Esslingen
University of Applied Sciences

- An effective technique for understanding a system.
- Possible to either participate in, or watch, a person perform activities to learn about the system.
- Useful when validity of data collected is in question or when the complexity of certain aspects of the system prevents a clear explanation by the end-users.

Advantages and disadvantages of using observation

Hochschule Esslingen
University of Applied Sciences

Table 10.4 Advantages and disadvantages of using observation as a fact-finding technique.

Advantages	Disadvantages
Allows the validity of facts and data to be checked	People may knowingly or unknowingly perform differently when being observed
Observer can see exactly what is being done	May miss observing tasks involving different levels of difficulty or volume normally experienced during that time period
Observer can also obtain data describing the physical environment of the task	Some tasks may not always be performed in the manner in which they are observed
Relatively inexpensive	May be impractical
Observer can do work measurements	

Research

Hochschule Esslingen
University of Applied Sciences

- Useful to research the application and problem.
- Use computer trade journals, reference books, and the Internet (including user groups and bulletin boards).
- Provide information on how others have solved similar problems, plus whether or not software packages exist to solve or even partially solve the problem.

Advantages and disadvantages of using research

Table 10.5 Advantages and disadvantages of using research as a fact-finding technique.

Advantages	Disadvantages
Can save time if solution already exists	Requires access to appropriate sources of information
Researcher can see how others have solved similar problems or met similar requirements	May ultimately not help in solving problem because problem is not documented elsewhere
Keeps researcher up to date with current developments	

Questionnaires

- Conduct surveys through questionnaires, which are special-purpose documents that allow facts to be gathered from a large number of people while maintaining some control over their responses.

- There are two types of questions, namely free-format and fixed-format.

Advantages and disadvantages of using questionnaires

Table 10.6 Advantages and disadvantages of using questionnaires as a fact-finding technique.

Advantages	Disadvantages
People can complete and return questionnaires at their convenience	Number of respondents can be low, possibly only 5% to 10%
Relatively inexpensive way to gather data from a large number of people	Questionnaires may be returned incomplete
People more likely to provide the real facts as responses can be kept confidential	May not provide an opportunity to adapt or re-word questions that have been misinterpreted
Responses can be tabulated and analyzed quickly	Cannot observe and analyze the respondent's body language

Using Fact-Finding Techniques – A Worked Example

Hochschule Esslingen
University of Applied Sciences

Figure 10.1
The *DreamHome* staff registration form for Susan Brand.

DreamHome
Staff Registration Form

Staff Number SG5
Full Name Susan Brand
Sex F **DOB** 3-Jun-40

Position Manager
Salary 24000

Branch Number B003
Branch Address
163 Main St, Glasgow
Telephone Number(s)
0141-339-2178 / 0141-339-4439

Enter details where applicable
Supervisor Name
Manager Start Date 01-Jun-90
Manager Bonus 2350

Figure 10.2
Example of the first page of a report listing the details of staff working at a *DreamHome* branch office in Glasgow.

DreamHome
Staff Listing

Branch Number B003

Telephone Number(s) 0141-339-2178 / 0141-339-4439

Branch Address
163 Main St, Glasgow
G11 9QX

Staff Number	Name	Position
SG5	Susan Brand	Manager
SG14	David Ford	Supervisor
SG37	Ann Beech	Assistant
SG112	Annet Longhorn	Supervisor
SG126	Chris Lawrence	Assistant
SG132	Sofie Walters	Assistant

Page 1

DreamHome
Property Registration Form

Property Number PG16

Type Flat **Rooms** 4

Rent 450

Address
5 Novar Drive,
Glasgow, G12 9AX

Owner Number C093
(If known)

Person/Business Name
Tony Shaw

Address 12 Park Pl,
Glasgow G4 0QR

Tel No 0141-225-7025

Enter details where applicable

Type of business ________

Contact Name ________

Managed by staff
David Ford

Registered at branch
163 Main St, Glasgow

Figure 10.3
The *DreamHome* property registration form for a property in Glasgow.

Figure 10.4
The *DreamHome* client registration form for Mike Ritchie.

DreamHome
Client Registration Form

Client Number CR74
(Enter if known)

Full Name
Mike Ritchie

Enter property requirements

Type Flat

Max Rent 750

Branch Number B003

Branch Address
163 Main St, Glasgow

Registered By
Ann Beech

Date Registered 16-Nov-02

Using Fact-Finding Techniques –
A Worked Example

Hochschule Esslingen
University of Applied Sciences

Figure 10.5
The first page of the *DreamHome* property for rent report listing property available at a branch in Glasgow.

DreamHome
Property Listing for Week beginning 01/06/04

If you are interested in viewing or renting any of the properties in this l st please contact our branch office as soon as poss ble.

Branch Address
163 Main St, Glasgow
G11 9QX

Telephone Number(s)
0141-339-2178 / 0141-339-4439

Property No	Address	Type	Rooms	Rent
PG4	6 Lawrence St, Glasgow	Flat	3	350
PG36	2 Manor Rd, Glasgow	Flat	3	375
PG21	18 Dale Road, Glasgow	House	5	600
PG16	5 Novar Drive, Glasgow	Flat	4	450
PG77	100A Apple Lane, Glasgow	House	6	560
PG81	781 Greentree Dr, Glasgow	Flat	4	440

Page 1

DreamHome
Property Viewing Report

Property Numner PG4

Type Flat

Rent 350

Property Address 6 Lawrence St, Glasgow

Client No	Name	Date	Comments
CR76	John Kay	20/04/04	Too remote.
CR56	Aline Stewart	26/05/04	
CR74	Mike Ritchie	11/11/04	
CR62	Mary Tregear	11/11/04	OK, but needs redecoration throughout.

Page 1

Figure 10.6 The first page of the *DreamHome* property viewing report for a property in Glasgow.

***DreamHome* Lease**
Number 00345810

Client Number CR74
(Enter if known)

Full Name Mike Ritchie
(Please print)

Client Signature ______________

Property Number PG16

Property Address 5 Novar Dr, Glasgow

Enter payment details

Monthly Rent 450

Payment Method Cheque

Deposit Paid (Y or N) Yes

Rent Start 01/06/04

Rent Finish 31/05/05

Duration 1 year

Figure 10.7 The *DreamHome* lease form for a client called Mike Ritchie renting a property in Glasgow.

Mission Statement for DreamHome Database System

„The purpose of the *DreamHome* database system is to maintain the data that is used and generated to support the property rentals business for our clients and property owners and to facilitate the cooperation and sharing of information between branches."

Mission Objectives for DreamHome Database System

- To
  - maintain (enter, update, and delete) data on
  - perform searches on
  - report on

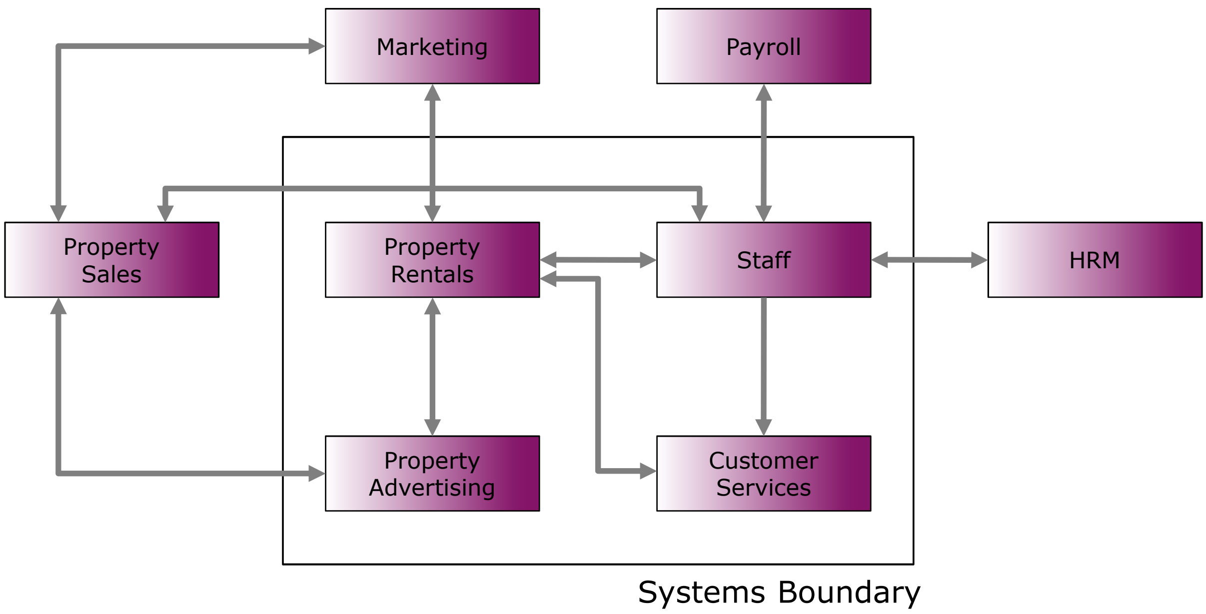
On

- branches
- staff
- properties for rent
- property owners
- clients
- property viewings
- leases
- newspaper adverts

Mission Objectives for DreamHome Database System

- To track the status of
  - properties for rent
  - clients whishing to rent
  - leases

System Boundary for DreamHome Database System



Major User Views for DreamHome Database System

Hochschule Esslingen
University of Applied Sciences

Data	Access Type	Director	Manager	Supervisor	Assistant
All Branches	Maintain				
	Query	X	X		
	Report	X	X		
Single Branch	Maintain		X		
	Query		X		
	Report		X		
All Staff	Maintain				
	Query	X	X		
	Report	X	X		
Branch Staff	Maintain		X		
	Query		X	X	
	Report		X	X	
All Property	Maintain				
	Query	X			
	Report	X	X		
Branch Property	Maintain		X	X	
	Query		X	X	X
	Report		X	X	X
All Owners	Maintain				
	Query	X			
	Report	X	X		
Branch Owners	Maintain		X	X	
	Query		X	X	X
	Report		X		

Figure 10.11 Major user views for the *DreamHome* database system.

Major User Views for DreamHome Database System

Hochschule Esslingen
University of Applied Sciences

Data	Access Type	Director	Manager	Supervisor	Assistant
All Clients	Maintain				
	Query	X			
	Report	X	X		
Branch Clients	Maintain		X	X	
	Query		X	X	X
	Report		X		
All Viewings	Maintain				
	Query				
	Report				
Branch Viewings	Maintain			X	X
	Query			X	X
	Report			X	X
All Leases	Maintain				
	Query	X			
	Report	X	X		
Branch Leases	Maintain		X	X	
	Query		X	X	X
	Report		X	X	
All Newspapers	Maintain				
	Query	X			
	Report	X	X		
Branch Newspapers	Maintain		X		
	Query		X		
	Report		X		

Cross-reference of user views with main types of data used by each

	Director	Manager	Supervisor	Assistant
branch	X	X		
staff	X	X	X	
property for rent	X	X	X	X
owner	X	X	X	X
client	X	X	X	X
property viewing			X	X
lease	X	X	X	X
newspaper	X	X		

Summary

Hochschule Esslingen
University of Applied Sciences

- Fact-finding techniques are crucial to the early stages of the database system development life cycle including database planning, system definition, and requirements collection and analysis stages.
- Fact-finding techniques include
  - examining documentation
  - interviewing
  - observing the organization in operation
  - research
  - questionnaires
- Mission statement, mission objectives, system definition and user views define the scope of the work.

Chapter 12

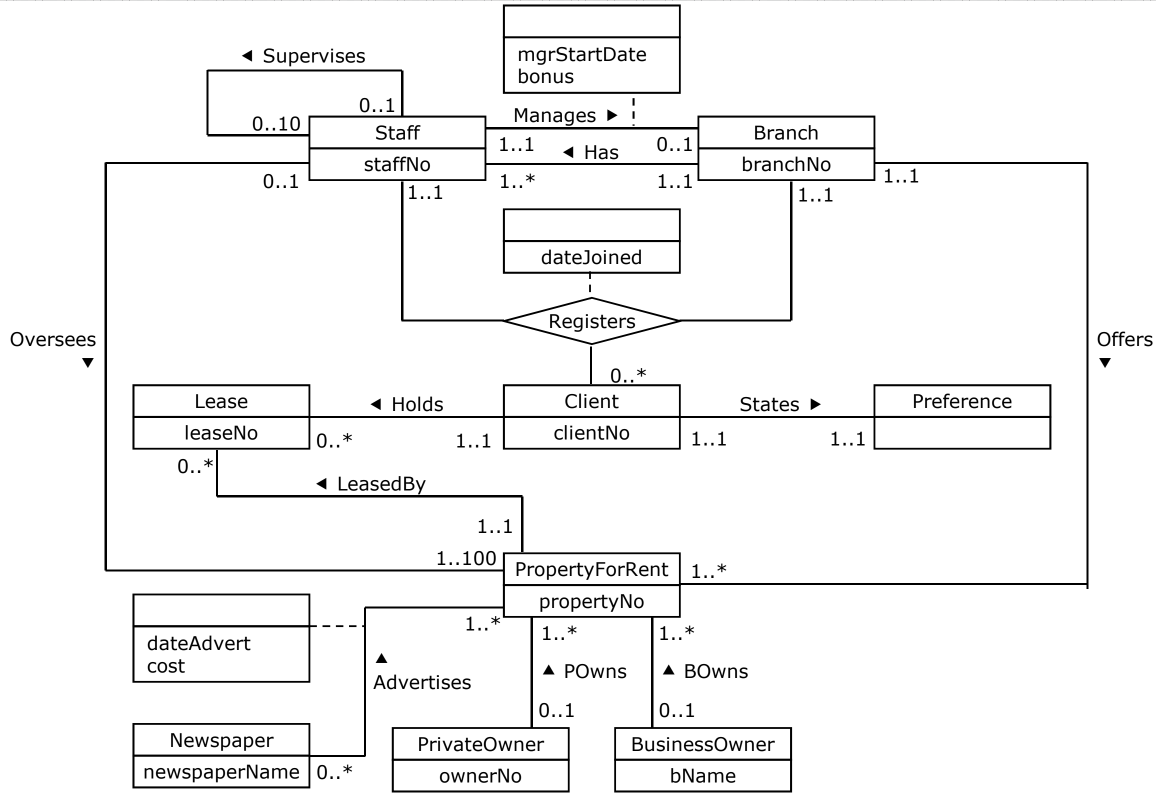
Entity-Relationship Modeling

Chapter 12 - Objectives

Hochschule Esslingen
University of Applied Sciences

- How to use Entity–Relationship (ER) modeling in database design.
- Basic concepts associated with ER model.
- Diagrammatic technique for displaying ER model using Unified Modeling Language (UML).
- How to identify and resolve problems with ER models called connection traps.
- How to build an ER model from a requirements specification.

ER diagram of Branch user views of DreamHome



Concepts of the ER Model

- Entity types
- Relationship types
- Attributes

Entity Type

- Entity type
  - Group of objects with same properties, identified by enterprise as having an independent existence.
- Entity occurrence
  - Uniquely identifiable object of an entity type.

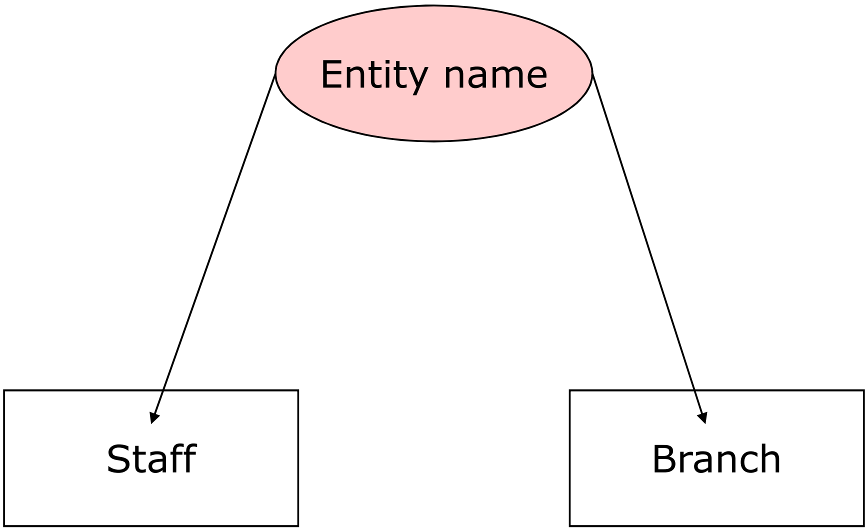
Examples of Entity Types

Hochschule Esslingen
University of Applied Sciences

- Physical existence
  - Staff
  - Property
  - Customer
  - Part
  - Supplier
  - Product
- Conceptual existence
  - Viewing
  - Inspection
  - Sale
  - Work experience

ER diagram of Staff and Branch entity types

Hochschule Esslingen
University of Applied Sciences

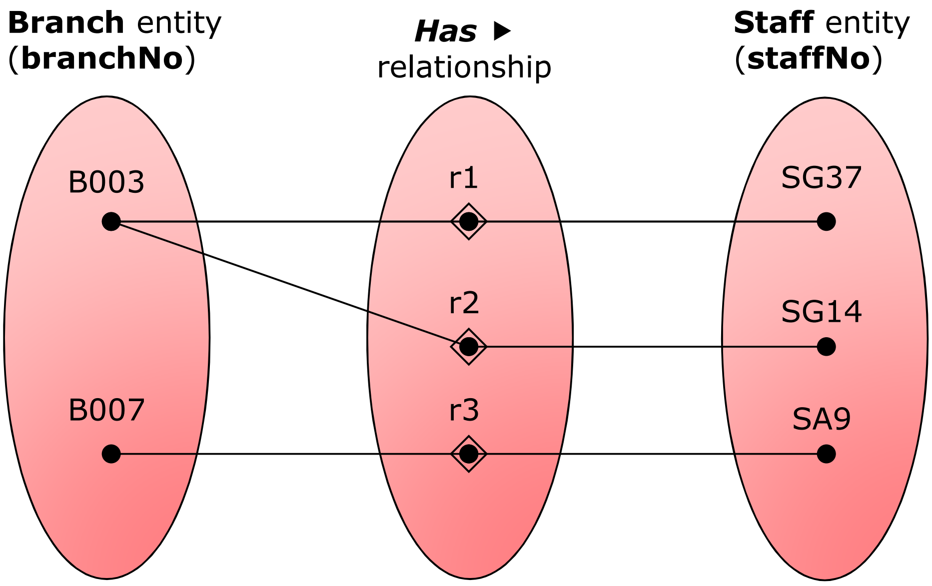


Relationship Types

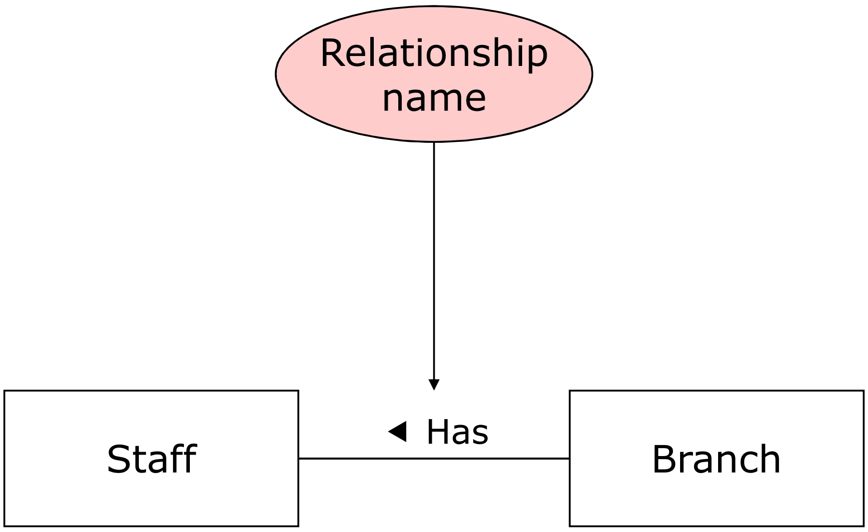
Hochschule Esslingen
University of Applied Sciences

- Relationship type
  - Set of meaningful associations among entity types.
- Relationship occurrence
  - Uniquely identifiable association, which includes one occurrence from each participating entity type.

Semantic net of Has relationship type



ER diagram of Branch Has Staff relationship



'Branch has staff'

Relationship Types

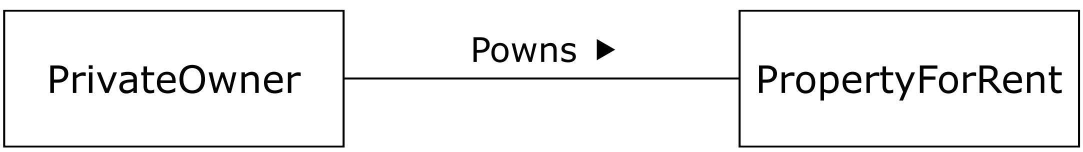
- Degree of a Relationship
  - Number of participating entities in relationship.

- Relationship of degree
  - two is binary
  - three is ternary
  - four is quaternary.

Binary relationship called Powns

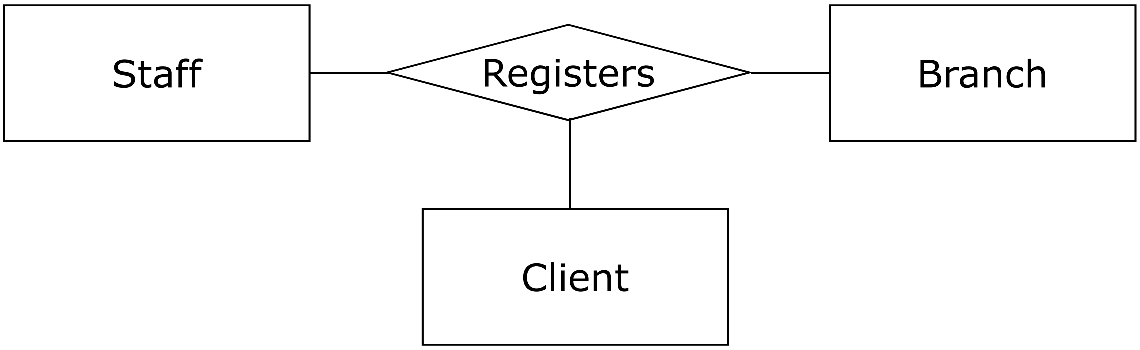
Hochschule Esslingen
University of Applied Sciences

'Private owner owns property for rent'



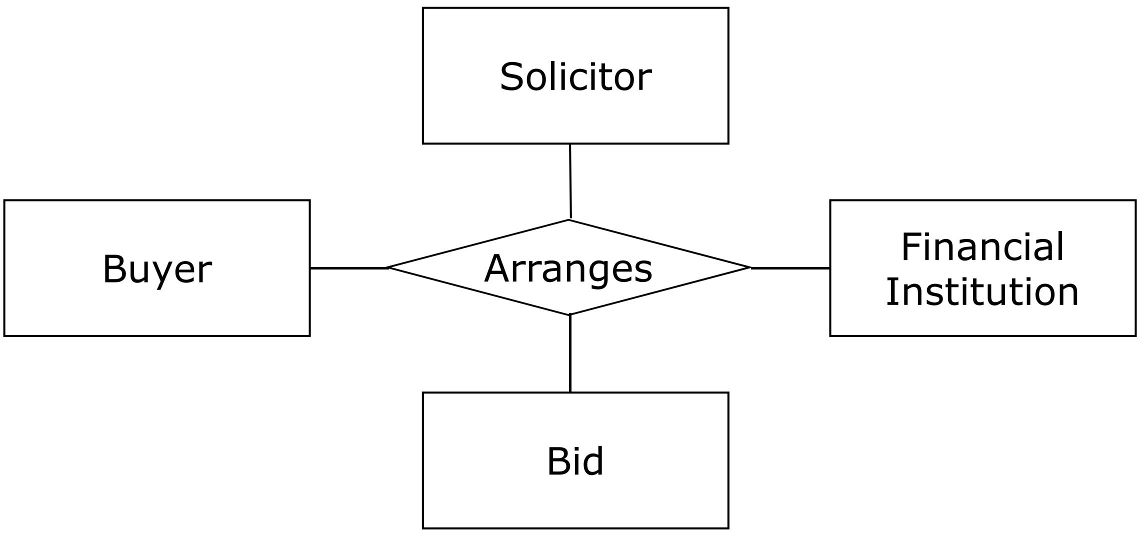
Ternary relationship called Registers

'Staff registers a client at a branch'



Quaternary relationship called Arranges

'A solicitor arranges a bid on behalf of a buyer supported by a financial institution'



Relationship Types

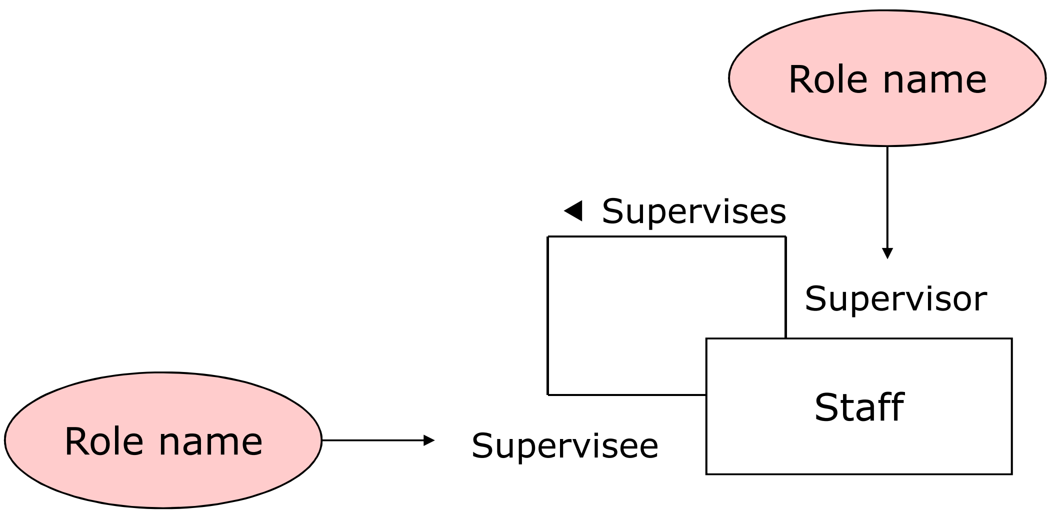
- Recursive Relationship
  - Relationship type where *same* entity type participates more than once in *different roles*.

- Relationships may be given role names to indicate purpose that each participating entity type plays in a relationship.

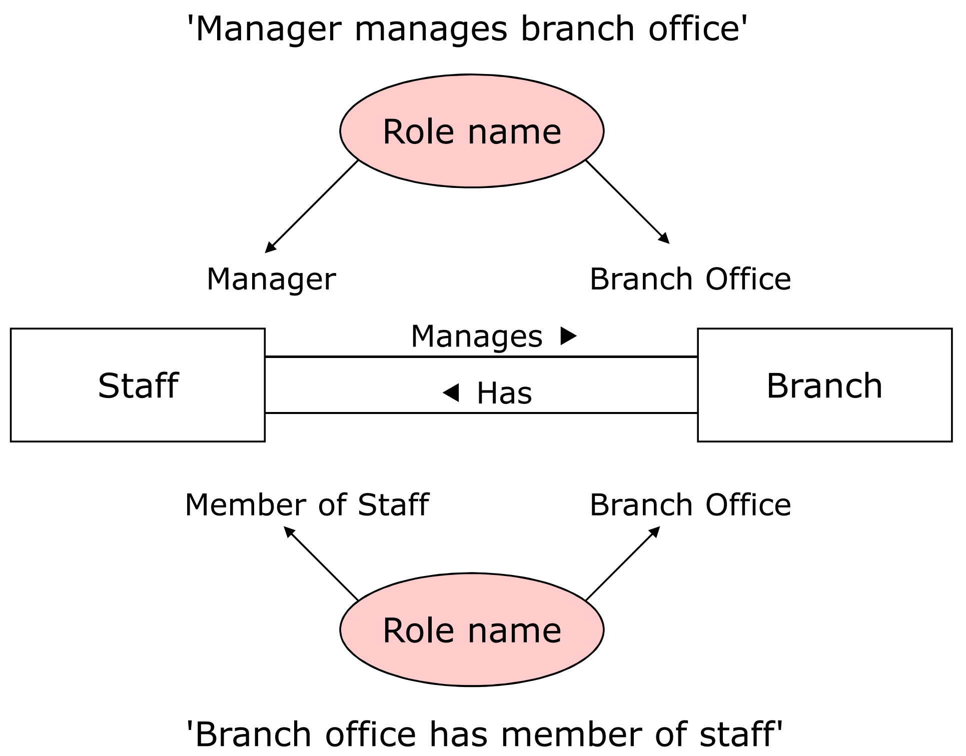
Recursive relationship called Supervises with role names

Hochschule Esslingen
University of Applied Sciences

'Staff (Supervisor) supervises staff (Supervisee)'



Entities associated through two distinct relationships with role names



Attributes

- Attribute
  - Property of an entity or a relationship type.
- Attribute Domain
  - Set of allowable values for one or more attributes.
- Simple Attribute
  - Attribute composed of a single component with an independent existence.
- Composite Attribute
  - Attribute composed of multiple components, each with an independent existence.

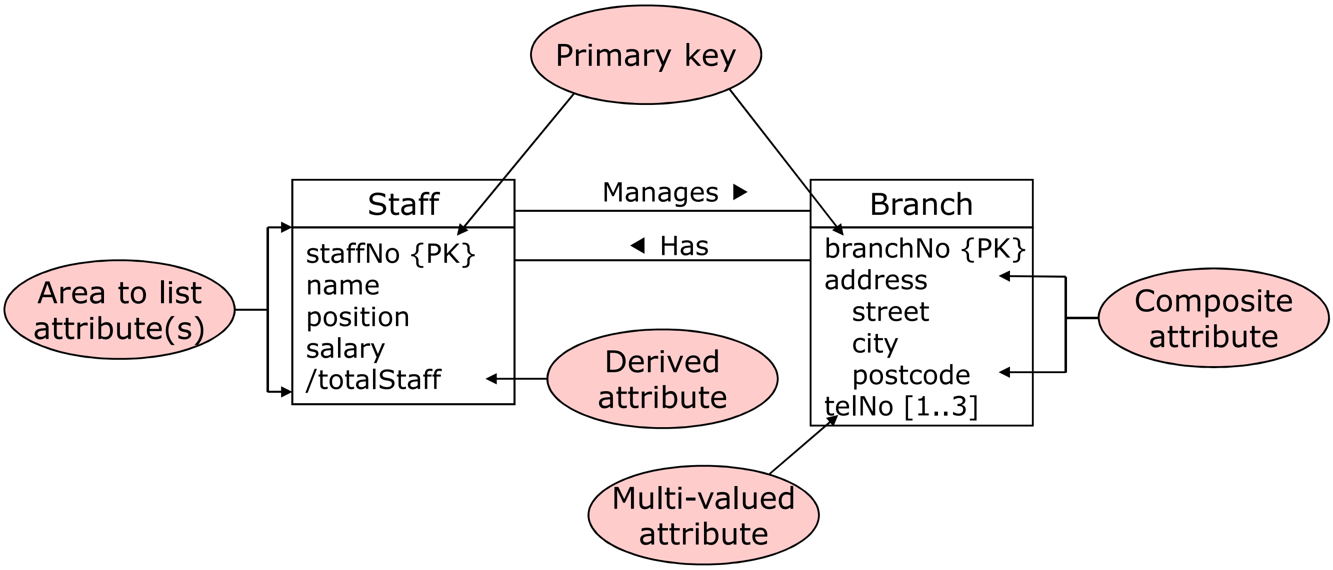
Attributes

- Single-valued Attribute
  - Attribute that holds a single value for each occurrence of an entity type.
- Multi-valued Attribute
  - Attribute that holds multiple values for each occurrence of an entity type.
- Derived Attribute
  - Attribute that represents a value that is derivable from value of a related attribute, or set of attributes, not necessarily in the same entity type.

Keys

- Candidate Key
  - Minimal set of attributes that uniquely identifies each occurrence of an entity type.
- Primary Key
  - Candidate key selected to uniquely identify each occurrence of an entity type.
- Alternate / Secondary Key
  - Candidate key not selected as primary key.
- Composite Key
  - A candidate key that consists of two or more attributes.

ER diagram of Staff and Branch entities and their attributes



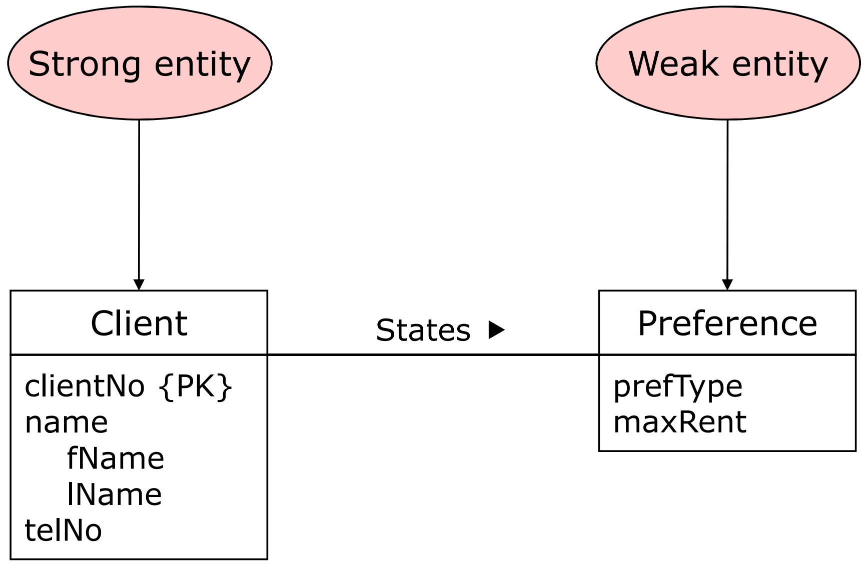
Entity Type

Hochschule Esslingen
University of Applied Sciences

- Strong Entity Type
  - Entity type that is not existence-dependent on some other entity type.

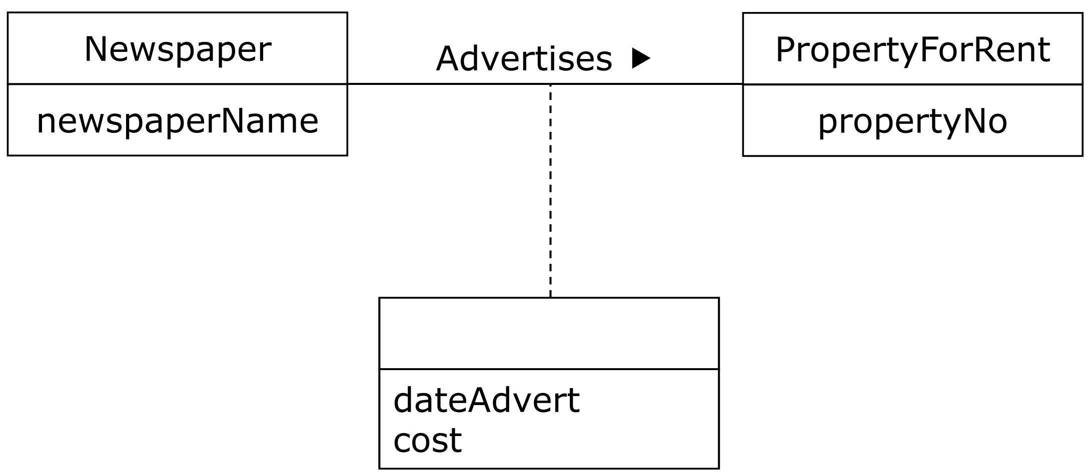
- Weak Entity Type
  - Entity type that is existence-dependent on some other entity type.

Strong entity type called Client and weak entity type called Preference



Relationship called Advertises with attributes

'Newspaper advertises property for rent'



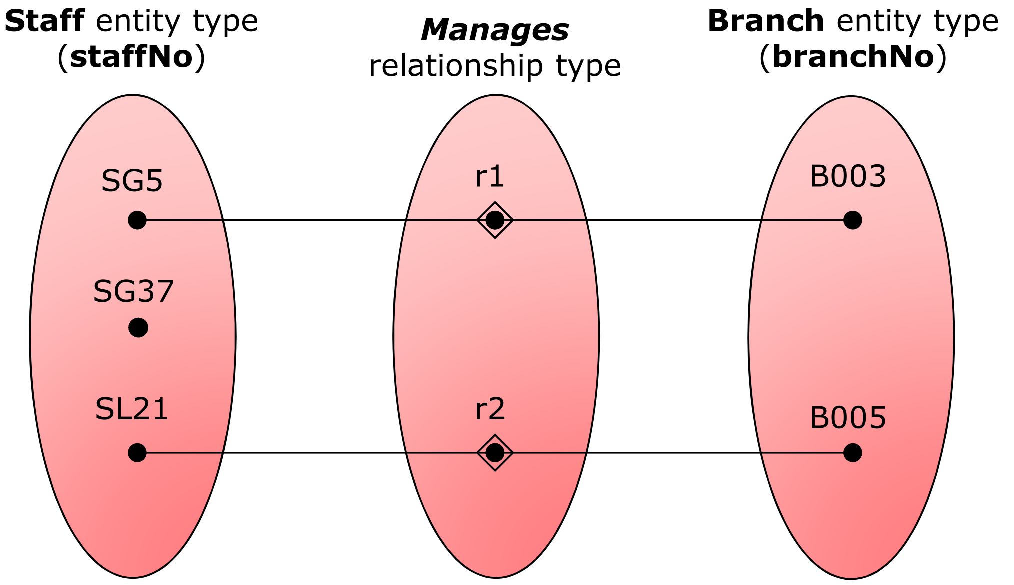
Structural Constraints

- Main type of constraint on relationships is called *multiplicity*.
- Multiplicity - number (or range) of possible occurrences of an entity type that may relate to a **single** occurrence of an associated entity type through a particular relationship.
- Represents policies (called *business rules*) established by user or company.

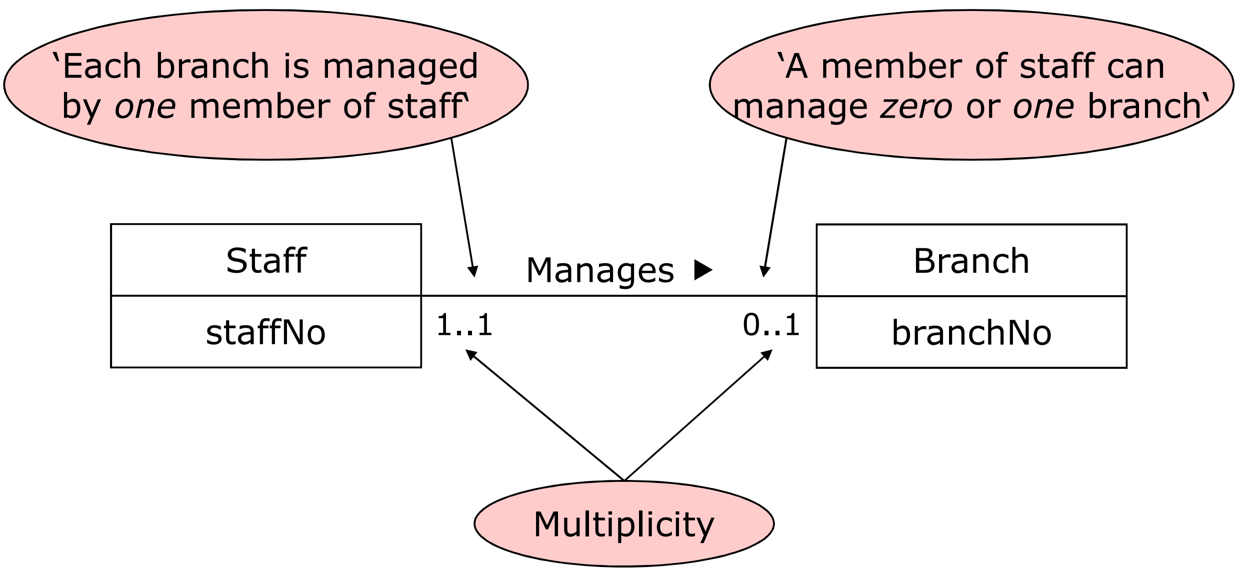
Structural Constraints

- The most common degree for relationships is binary.
- Binary relationships are generally referred to as being:
  - one-to-one (1:1)
  - one-to-many (1:*)
  - many-to-many (*:*)

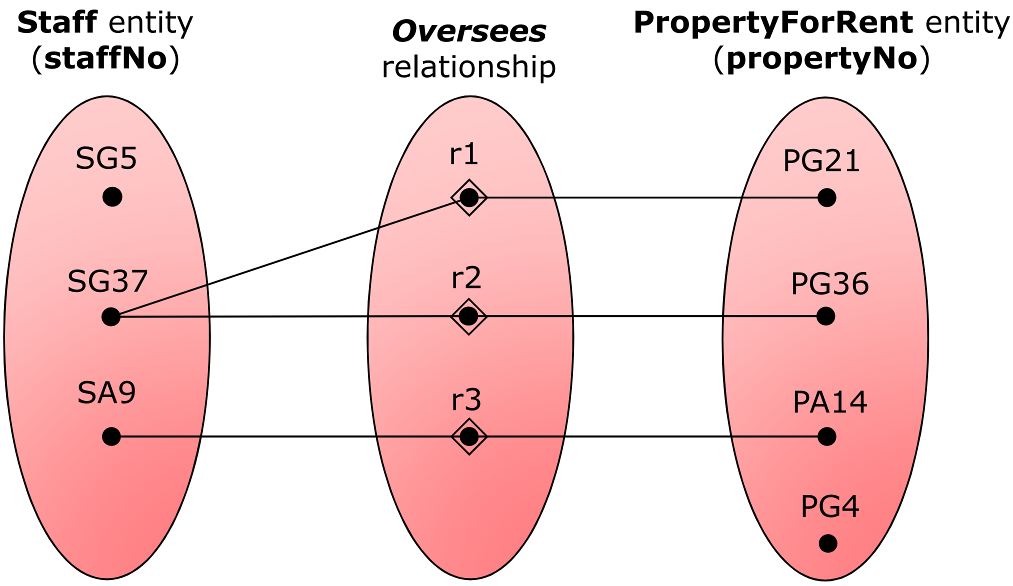
Semantic net of Staff Manages Branch relationship type



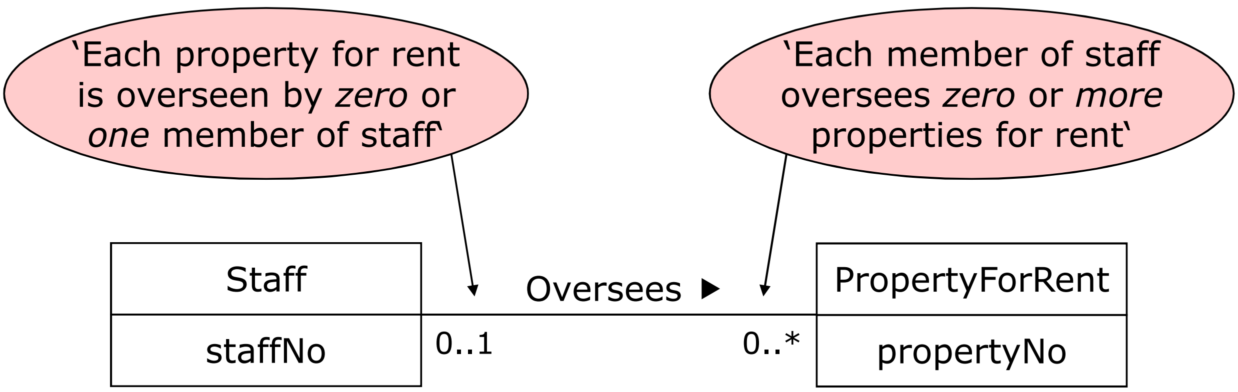
Multiplicity of Staff Manages Branch (1:1) relationship



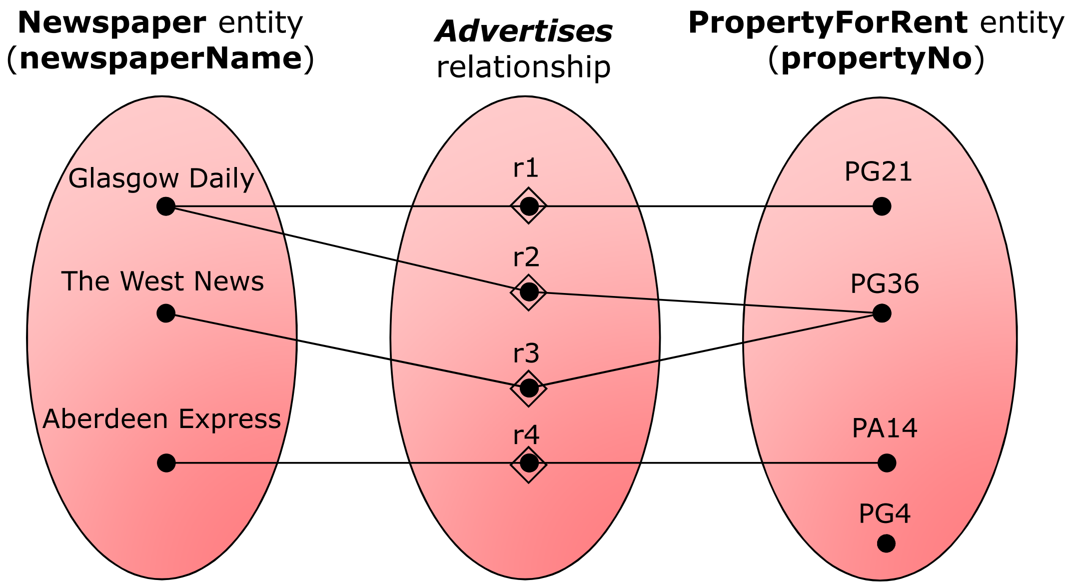
Semantic net of Staff Oversees PropertyForRent relationship type



Multiplicity of Staff Oversees PropertyForRent (1:*) relationship type

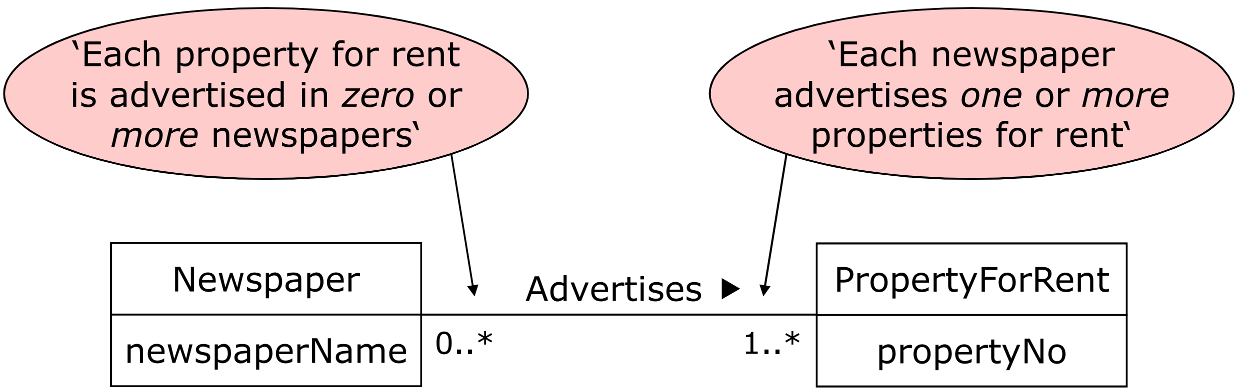


Semantic net of Newspaper Advertises PropertyForRent relationship type

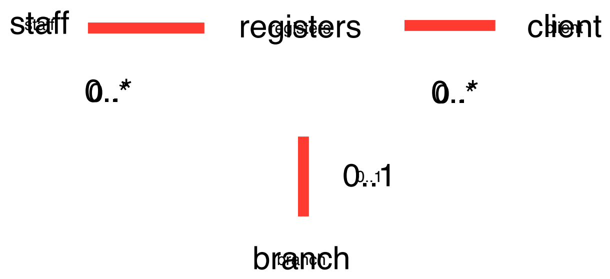


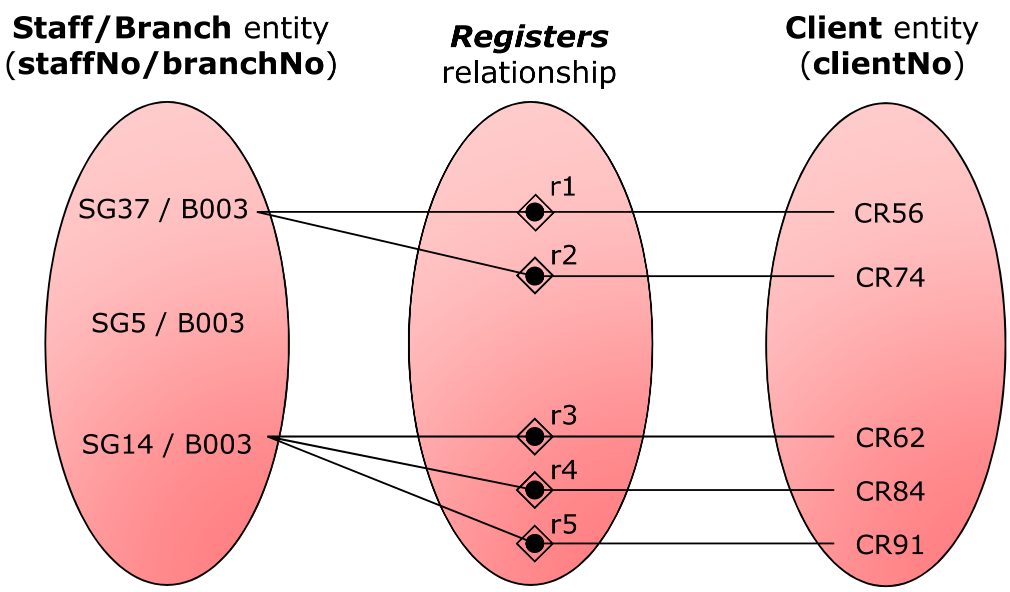
Multiplicity of Newspaper Advertises PropertyForRent (*:*) relationship

Hochschule Esslingen
University of Applied Sciences

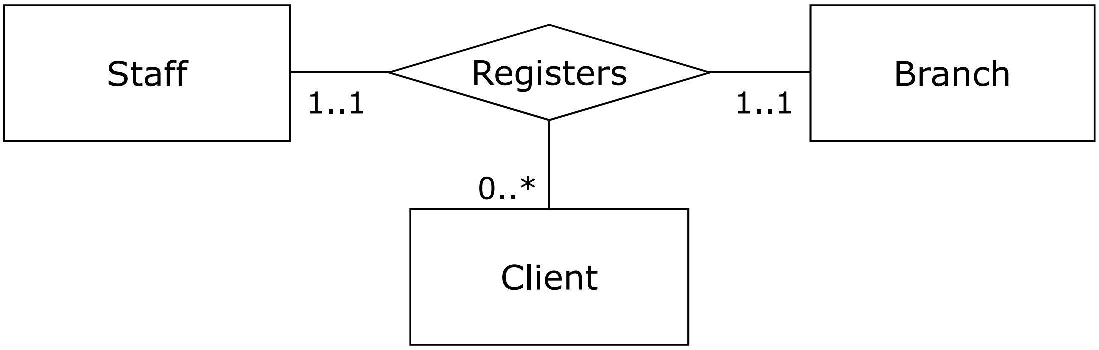


- Multiplicity for complex relationships
  - Number (or range) of possible occurrences of an entity type in an n-ary relationship when other (n-1) values are fixed.





Multiplicity of ternary Registers relationship



Summary of multiplicity constraints

- Alternative way to represent multiplicity constraints

– 0..1	Zero or one entity occurrence
– 1..1 (or just 1)	Exactly one entity occurrence
– 0..* (or just *)	Zero or many entity occurrences
– 1..*	One or many entity occurrences
– 5..10	Minimum of 5 up to a maximum of 10 entity occurrences
– 0, 3, 6-8	Zero or three or six, seven, or eight entity occurrences

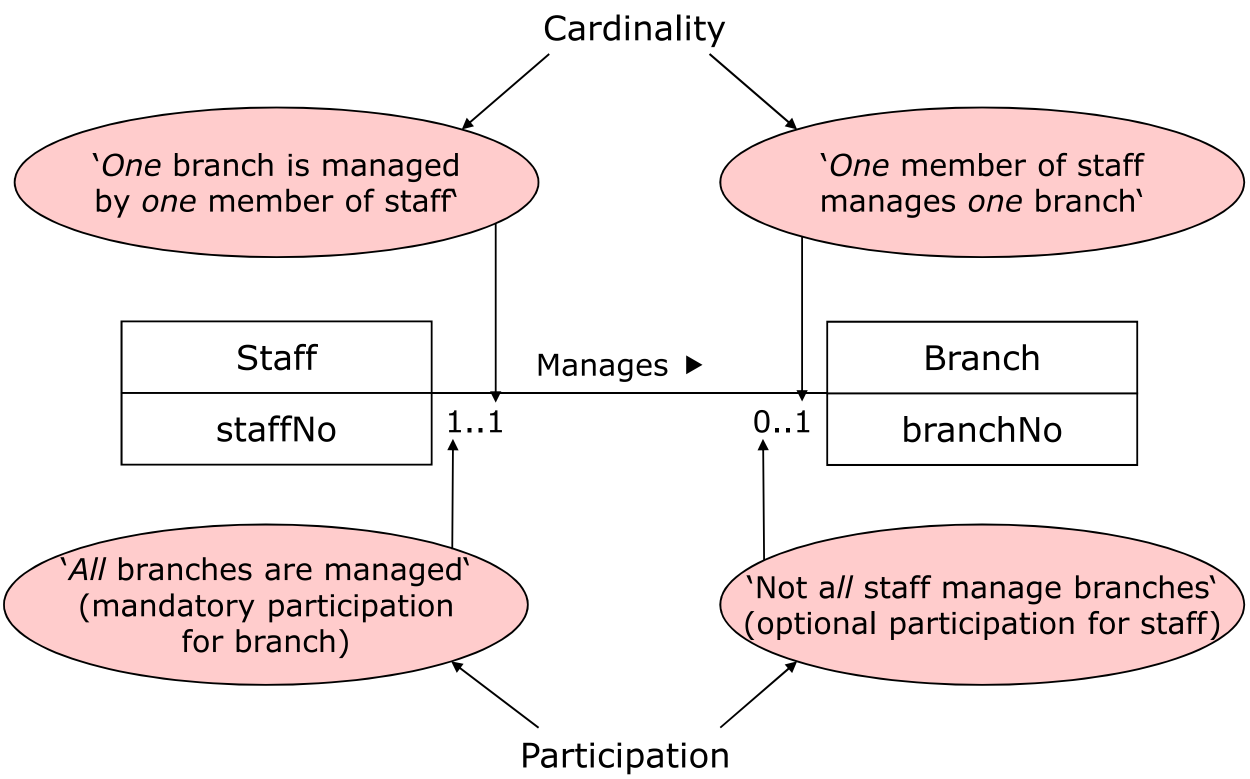
Structural Constraints

- Multiplicity is made up of two types of restrictions on relationships: *cardinality* and *participation*.

- Cardinality
  - Describes maximum number of possible relationship occurrences for an entity participating in a given relationship type.

- Participation
  - Determines whether all or only some entity occurrences participate in a relationship.

Multiplicity as cardinality and participation constraints



Problems with ER Models

- Problems may arise when designing a conceptual data model called *connection traps*.
- Often due to a misinterpretation of the meaning of certain relationships.
- Two main types of connection traps are called *fan traps* and *chasm traps*.

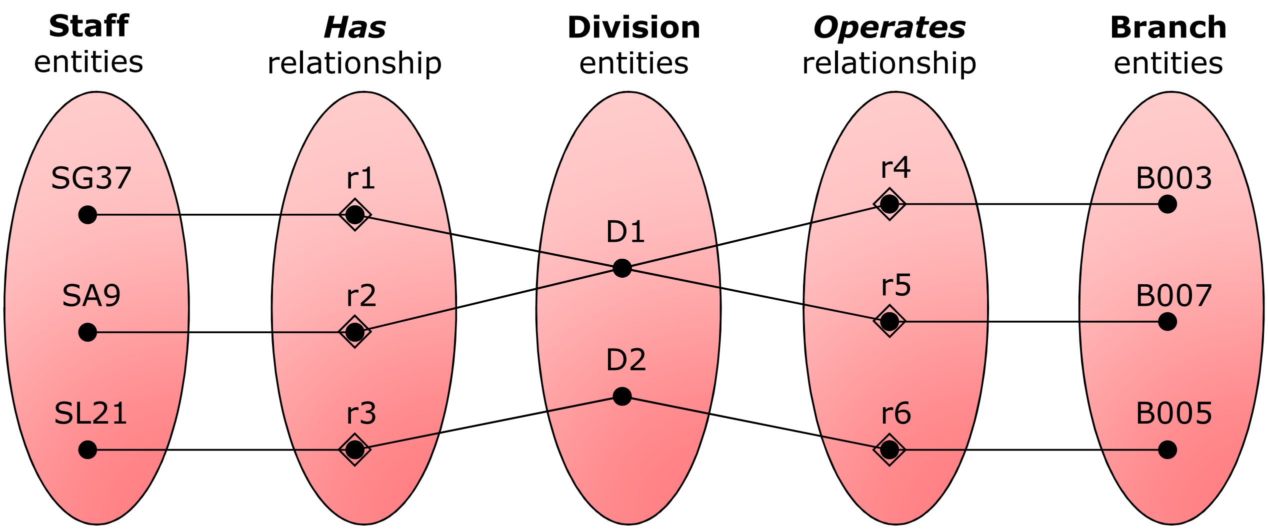
Problems with ER Models

Hochschule Esslingen
University of Applied Sciences

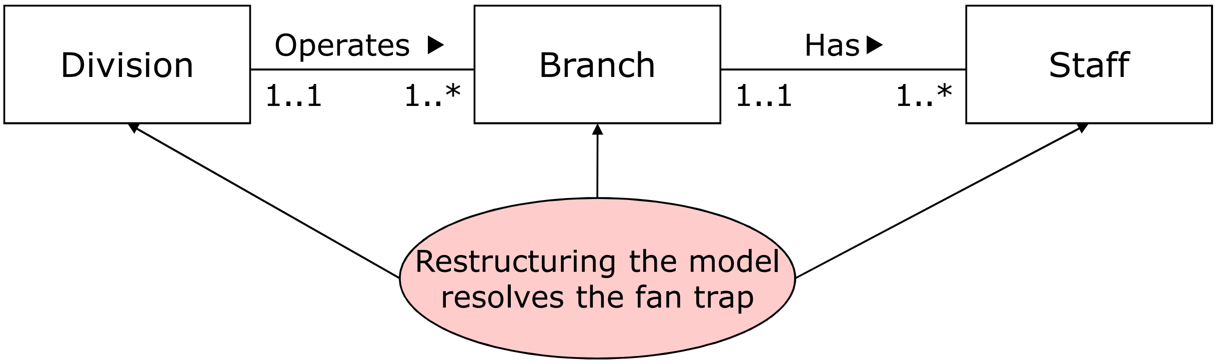
- Fan Trap (Fächerfalle)
  - Where a model represents a relationship between entity types, but pathway between certain entity occurrences is ambiguous.
- Chasm Trap (Schluchtfalle)
  - Where a model suggests the existence of a relationship between entity types, but pathway does not exist between certain entity occurrences.

An Example of a Fan Trap

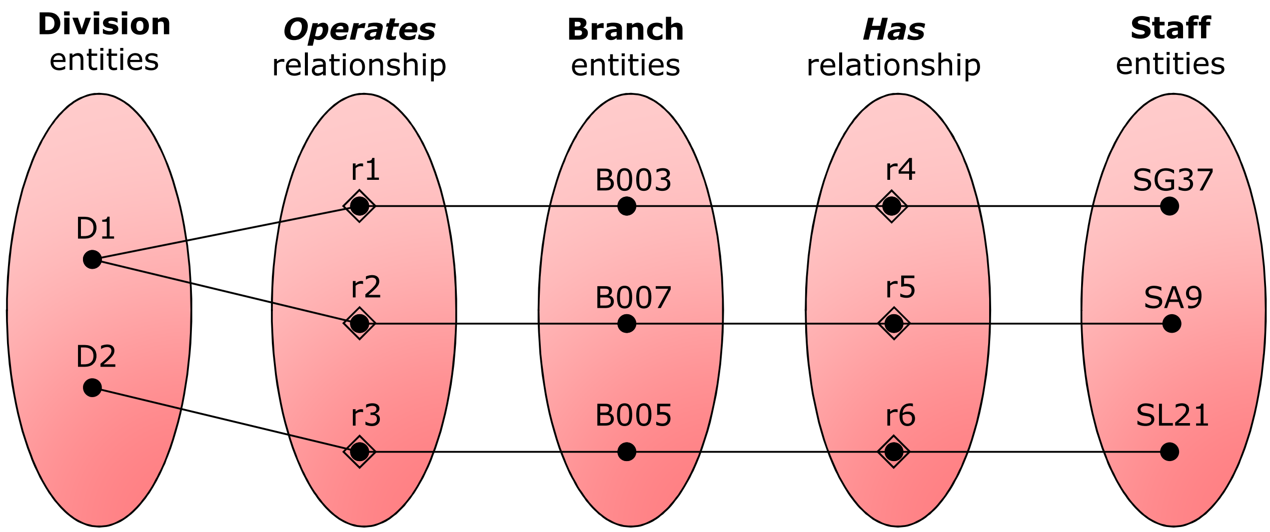
Semantic Net of ER Model with Fan Trap



- At which branch office does staff number SG37 work?



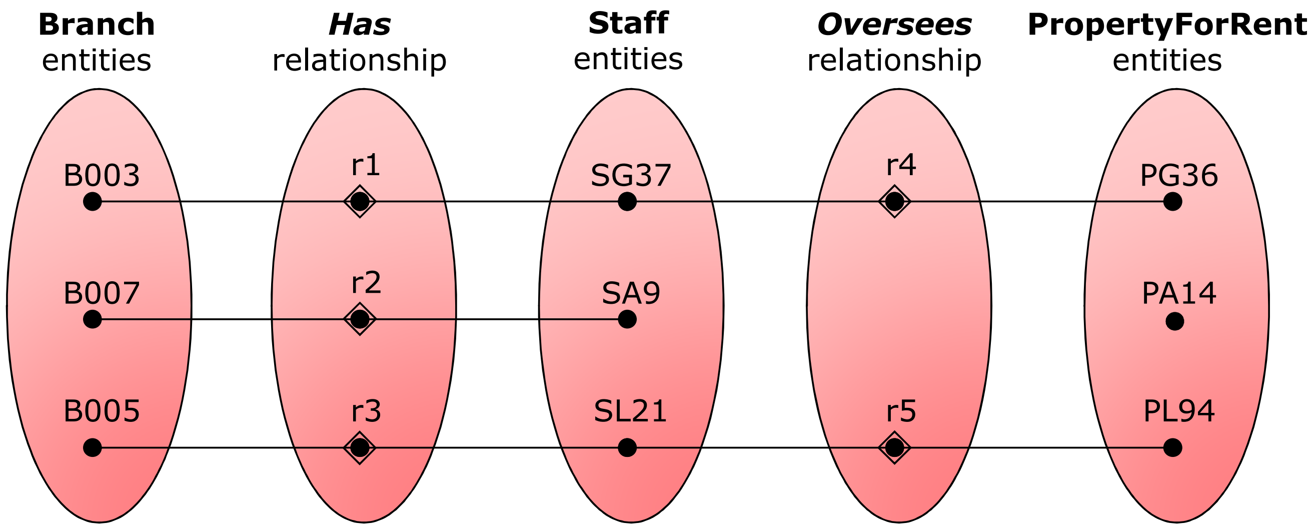
Hochschule Esslingen
University of Applied Sciences



- SG37 works at branch B003.

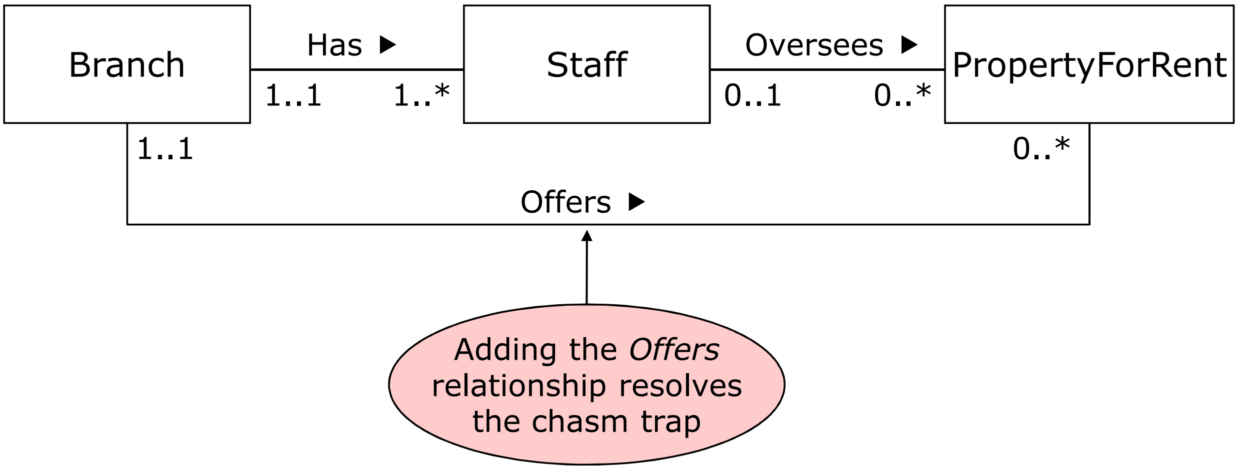
An Example of a Chasm Trap

Semantic Net of ER Model with Chasm Trap



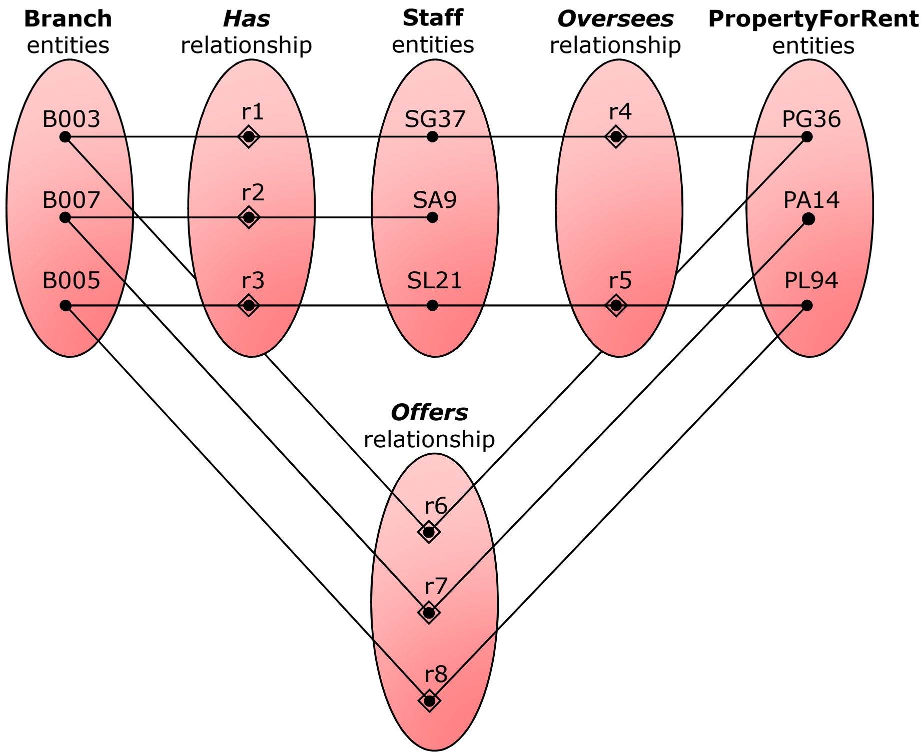
- At which branch office is property PA14 available?

ER Model restructured to remove Chasm Trap

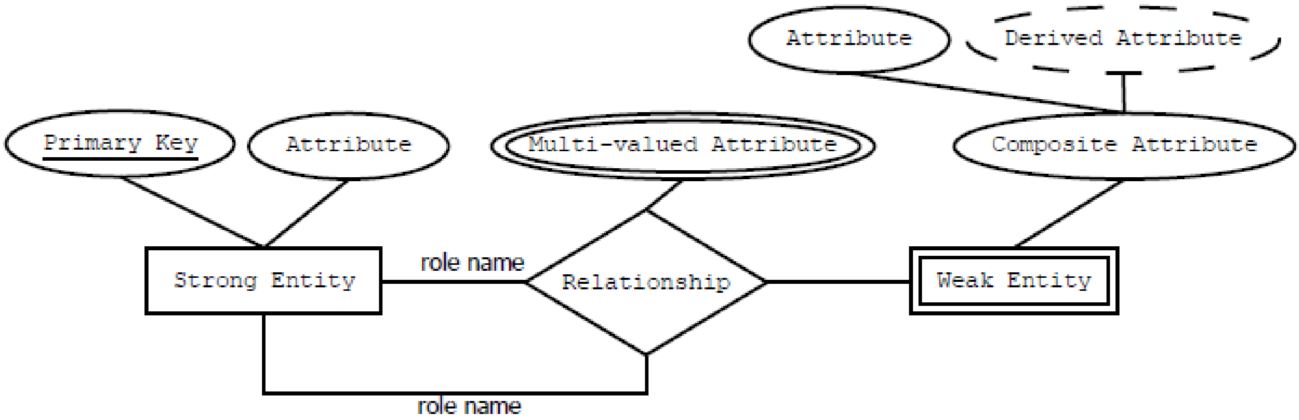


Semantic Net of Restructured ER Model with Chasm Trap Removed

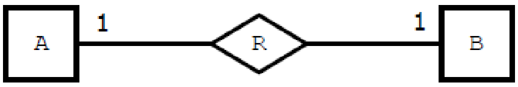
Hochschule Esslingen
University of Applied Sciences



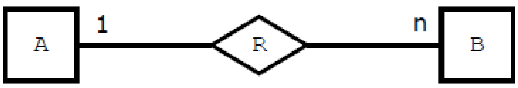
Alternative ER Notations: Chen



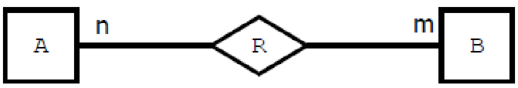
Alternative ER Notations: Chen



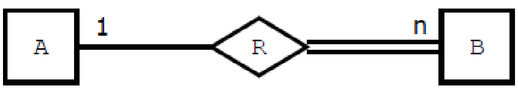
- One-to-one relationship, each A is associated with at most one B and each B is associated with at most one A.



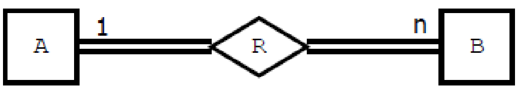
- One-to-many relationship, each A is associated with possibly many Bs and each B is associated with at most one A.



- Many-to-many relationship, each A is associated with possibly many Bs and each B is associated with possibly many As.



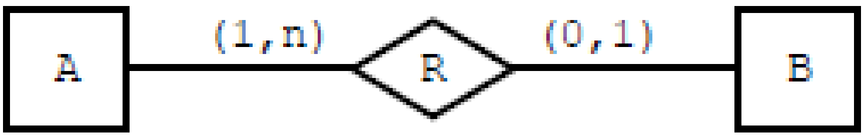
- One-to-many relationship, each A is associated with possibly many Bs and each B is associated with exactly one A (mandatory participation for entity B).



- One-to-many relationship with mandatory participation for both entities A and B.

Alternative ER Notation: Min-Max

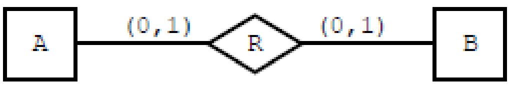
- Notation nearly identical to Chen.
- Cardinality and participation is expressed via minimal and maximal values.
- Note: The position of the number-tags at the relation are **reversed** compared to Chen.



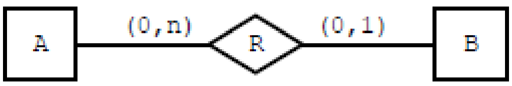
- Each A is associated with at least one B and each B is associated with at most one A. Therefore, A´s participation is mandatory and B´s participation is optional.

Hochschule Esslingen
University of Applied Sciences

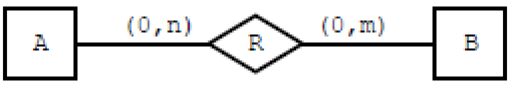
Alternative ER Notation: Min-Max



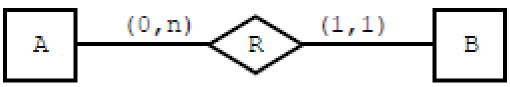
- One-to-one relationship, each A is associated with at most one B and each B is associated with at most one A.



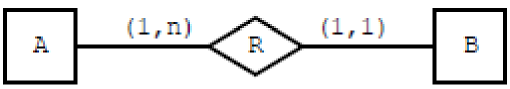
- One-to-many relationship, each A is associated with possibly many Bs and each B is associated with at most one A.



- Many-to-many relationship, each A is associated with possibly many Bs and each B is associated with possibly many As.



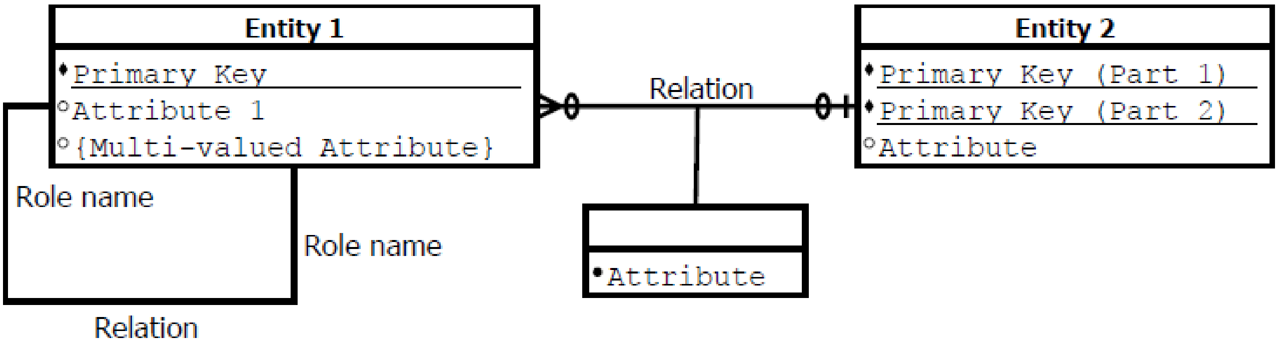
- One-to-many relationship, each A is associated with possibly many Bs and each B is associated with exactly one A (mandatory participation for entity B).

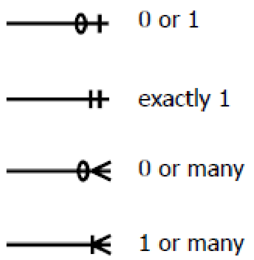


- One-to-many relationship with mandatory participation for both entities A and B.

Alternative ER Notation: Crowfoot (Martin)

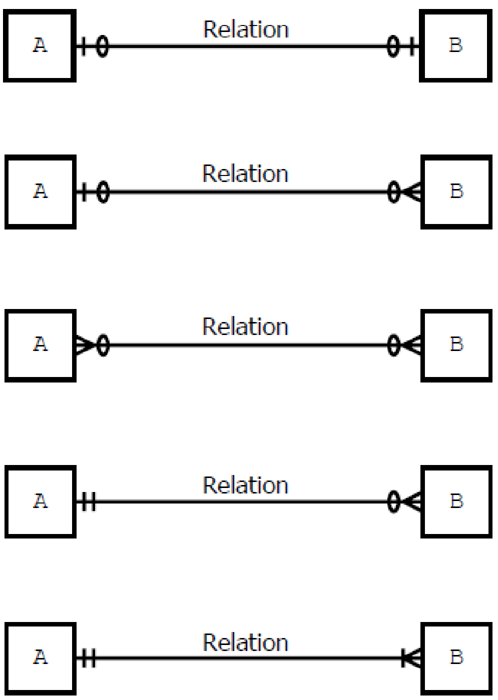
Hochschule Esslingen
University of Applied Sciences





Alternative ER Notations: Crowfoot (Martin)

Hochschule Esslingen
University of Applied Sciences



- One-to-one relationship, each A is associated with at most one B and each B is associated with at most one A.
- One-to-many relationship, each A is associated with possibly many Bs and each B is associated with at most one A.
- Many-to-many relationship, each A is associated with possibly many Bs and each B is associated with possibly many As.
- One-to-many relationship, each A is associated with possibly many Bs and each B is associated with exactly one A (mandatory participation for entity B).
- One-to-many relationship with mandatory participation for both entities A and B.

Comparison of Notations

Feature	UML	Chen	Min-Max	Crowfoot
1:1, 1:n, n:m relations	yes	yes	yes	yes
min-max multiplicity	yes	no	yes	no
detailed cardinality, e.g. 1,3,5	yes	no	no	no
ternary, quaternary, … relations	yes	yes	yes	no
attributes of relations	yes	yes	yes	yes
suitable for model on paper	no	yes	yes	no
suitable for model in CASE-tool	yes	no	no	yes

Summary

- Diagrammatic Entity-Relationship Modelling (ER modelling) is **the** technique for modelling data structures for database systems.
- Main concepts are:
  - entity,
  - relationship,
  - attribute,
  - cardinality
  - participation
- Alternative diagrammatic techniques to UML are:
  - Classical ER modelling (Chen notation)
  - Min-max notation
  - Crowfoot notation (Martin)

Hochschule Esslingen
University of Applied Sciences

Chapter 13

Enhanced Entity-Relationship Modelling

Chapter 13 - Objectives

Hochschule Esslingen
University of Applied Sciences

- Limitations of basic concepts of the ER model and requirements to represent more complex applications using additional data modelling concepts.
- Most useful additional data modelling concept of Enhanced ER (EER) model is called specialization/generalization.
- A diagrammatic technique for displaying specialization/generalization in an EER diagram using UML.

Enhanced Entity-Relationship Model

- Since 1980s there has been an increase in emergence of new database applications with more demanding requirements.
- Basic concepts of ER modelling are not sufficient to represent requirements of newer, more complex applications.
- Response is development of additional 'semantic' modelling concepts.
- Semantic concepts are incorporated into the original ER model and called the Enhanced Entity-Relationship (EER) model.
- Example of additional concept of EER model is called specialization / generalization.

Specialization / Generalization

- Superclass
  - An entity type that includes one or more distinct subgroupings of its occurrences.
- Subclass
  - A distinct subgrouping of occurrences of an entity type.
- Superclass/subclass relationship is one-to-one (1:1).
- Superclass may contain overlapping or distinct subclasses.
- Not all members of a superclass need be a member of a subclass.

Specialization / Generalization

- Attribute Inheritance
  - An entity in a subclass represents same 'real world' object as in superclass, and may possess subclass-specific attributes, as well as those associated with the superclass.
- Specialization
  - Process of maximizing differences between members of an entity by identifying their distinguishing characteristics.
- Generalization
  - Process of minimizing differences between entities by identifying their common characteristics.

Hochschule Esslingen
University of Applied Sciences

AllStaff relation holding details of all staff

Attributes appropriate for all staff (staffNo, name, position, salary)

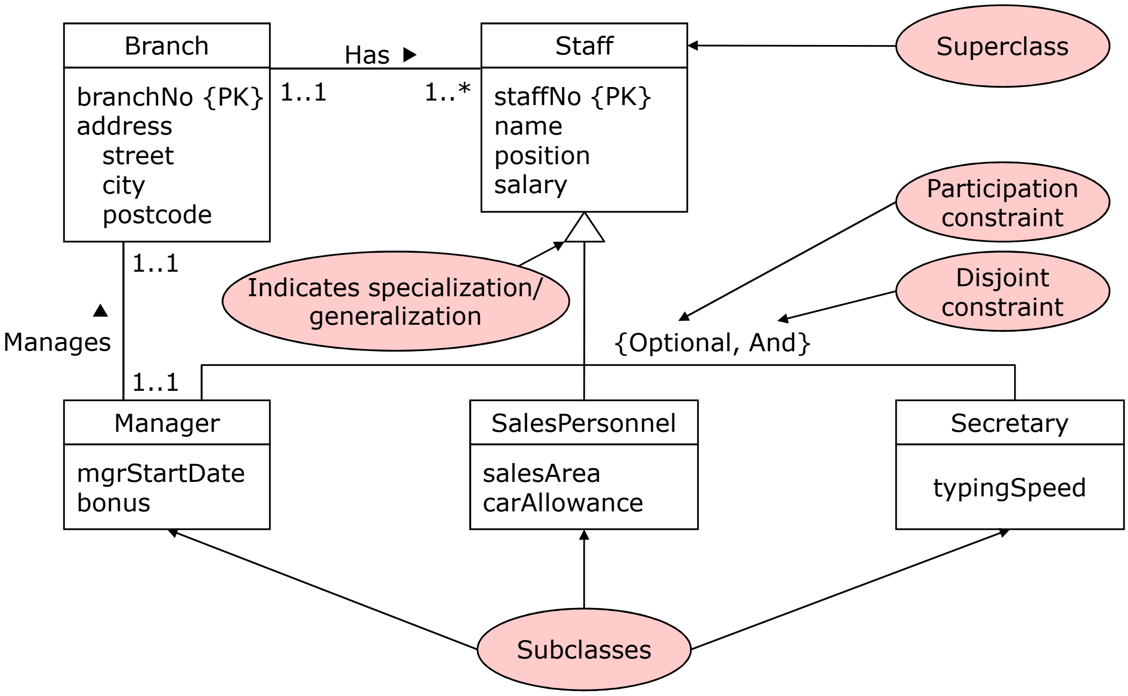
Attributes appropriate for branch Managers (mgrStart Date, bonus)

Attributes appropriate for Sales Personnel (sales Area, car Allowance)

Attributes appropriate for Secretarial staff (typing Speed)

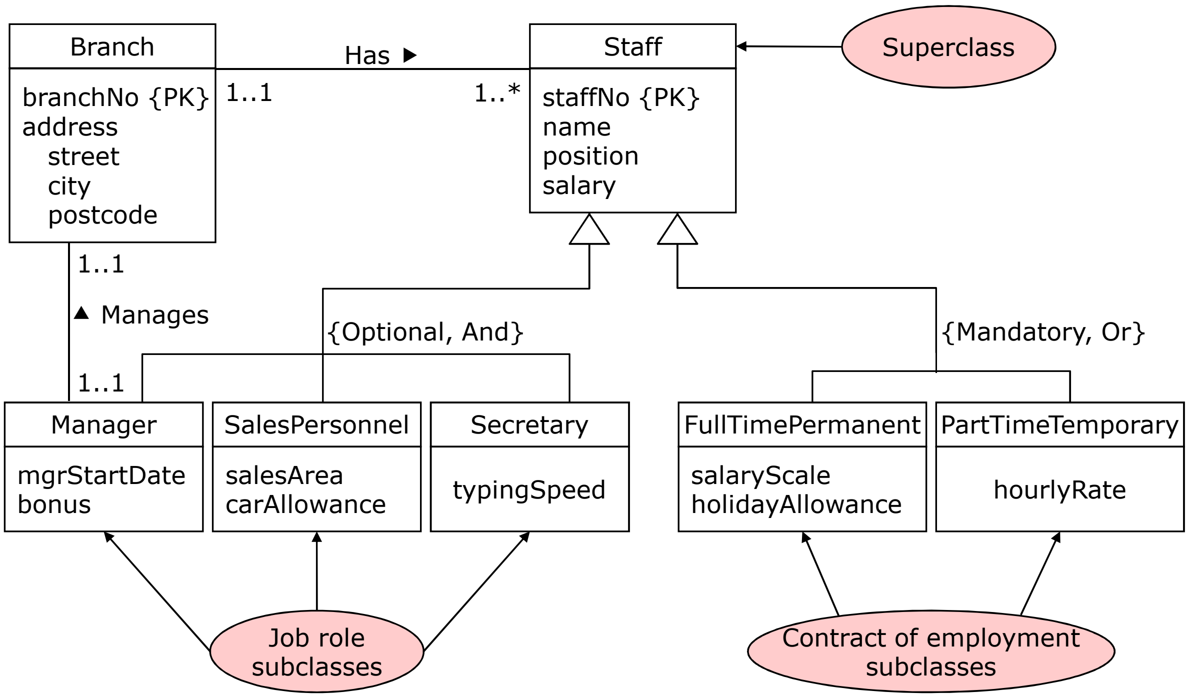
staffNo	name	position	salary	mgrStart Date	bonus	sales Area	car Allowance	typing Speed
SL21	John White	Manager	30000	01/02/95	2000			
SG37	Ann Beech	Assistant	12000					
SG66	Mary Martinez	Sales Manager	27000			SA1A	5000	
SA9	Mary Howe	Assistant	9000					
SL89	Stuart Stern	Secretary	8500					100
SL31	Robert Chin	Snr Sales Asst	17000			SA2B	3700	
SG5	Susan Brand	Manager	24000	01/06/91	2350			

Specialization/generalization of Staff entity into subclasses representing job roles



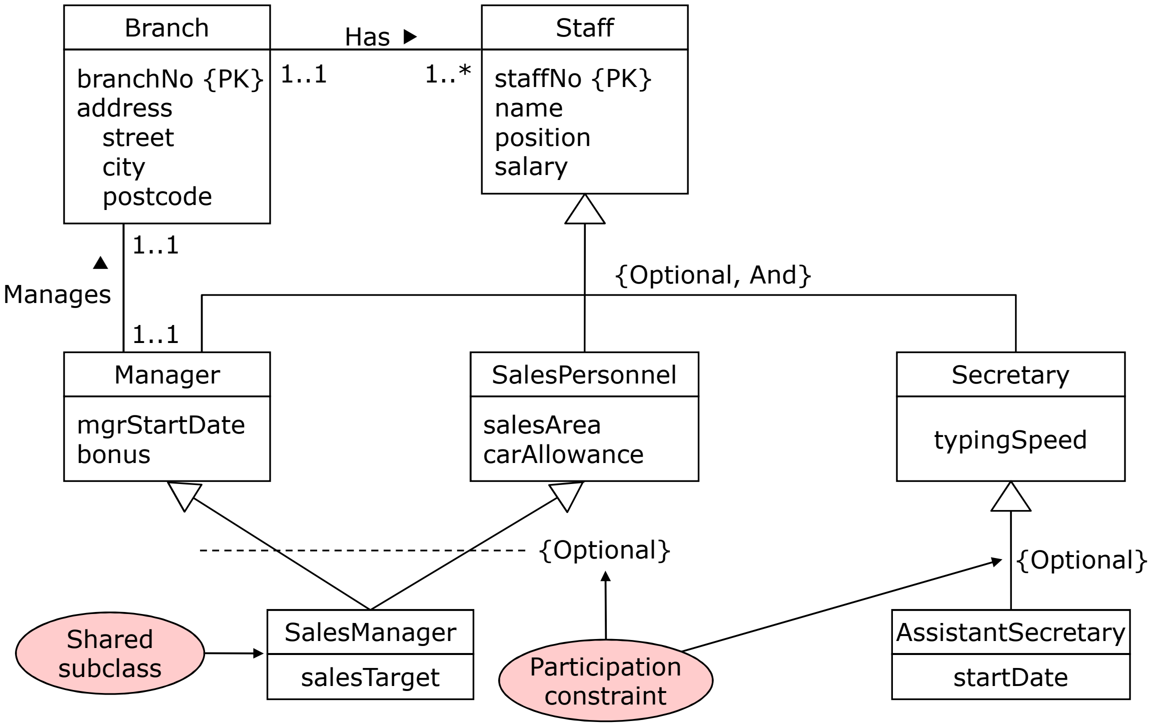
Specialization/generalization of Staff entity into job roles and contracts of employment

Hochschule Esslingen
University of Applied Sciences



EER diagram with shared subclass and subclass with its own subclass

Hochschule Esslingen
University of Applied Sciences



Constraints on Specialization / Generalization

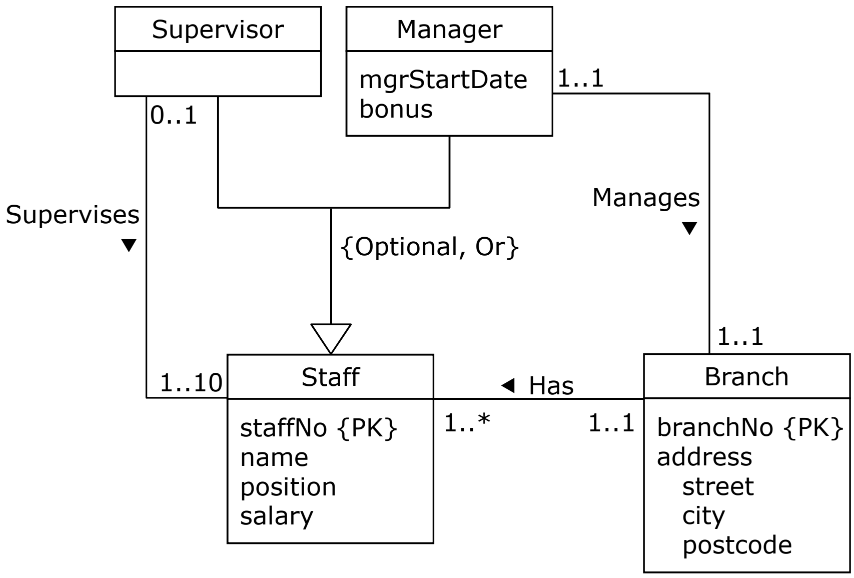
Hochschule Esslingen
University of Applied Sciences

- Two constraints that may apply to a specialization/generalization:
  - participation constraints
  - disjoint constraints.
- Participation constraint
  - Determines whether every member in superclass must participate as a member of a subclass.
  - May be *mandatory* or *optional*.
- Disjoint constraint
  - Describes relationship between members of the subclasses and indicates whether member of a superclass can be a member of one, or more than one, subclass.
  - May be *disjoint (OR)* or *nondisjoint (AND)*.

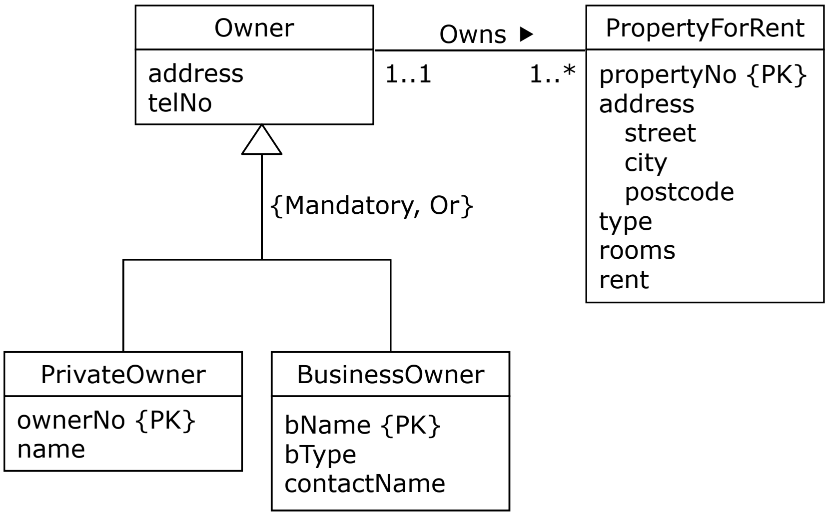
Constraints on Specialization / Generalization

- There are four categories of constraints of specialization and generalization:
  - mandatory and disjoint
  - optional and disjoint
  - mandatory and nondisjoint
  - optional and nondisjoint.

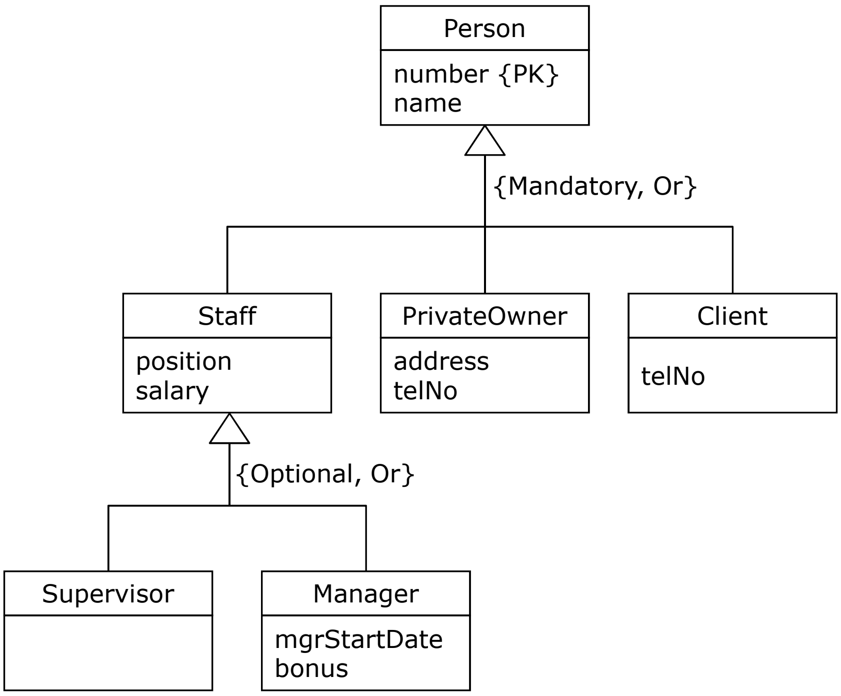
DreamHome worked example - Staff Superclass with Supervisor and Manager subclasses



DreamHome worked example - Owner Superclass with PrivateOwner and BusinessOwner subclasses

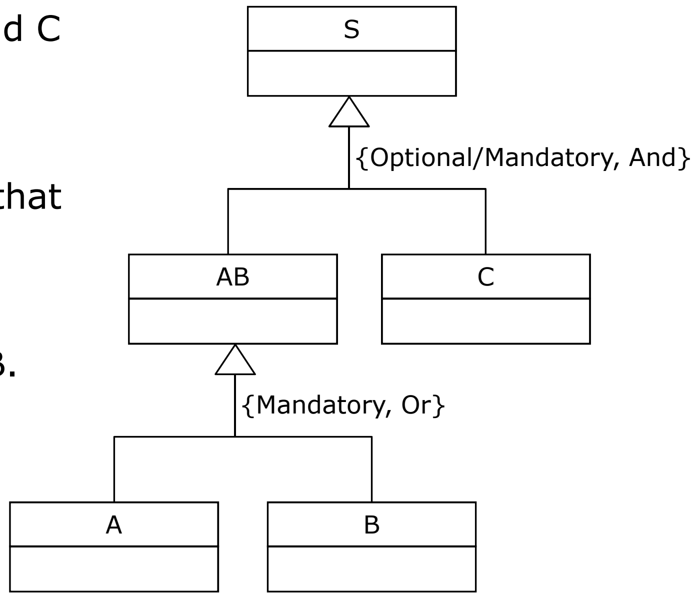


DreamHome worked example - Person superclass with Staff, PrivateOwner, and Client subclasses



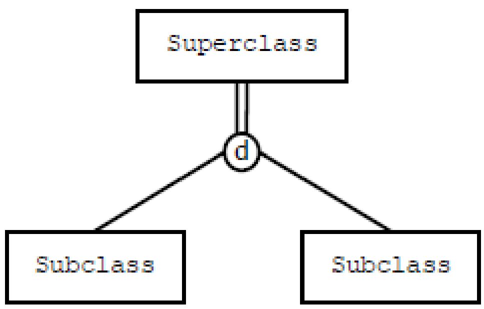
Modelling Non-mutual Disjointness Constraints

- Assume the entities A, B and C are subclasses of entity S, i.e. $A \subseteq S$, $B \subseteq S$, $C \subseteq S$.
- Assume we want to model that A and B must be disjoint, i.e. $A \cap B = \emptyset$.
- C may overlap with A and B.



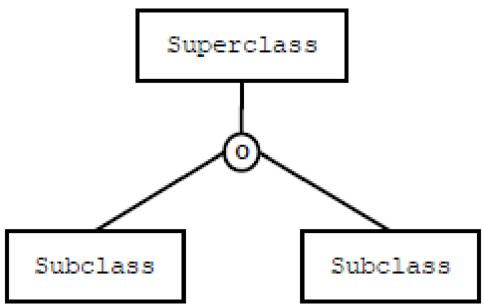
Alternative Notations: Chen, Min-Max

Hochschule Esslingen
University of Applied Sciences



mandatory participation

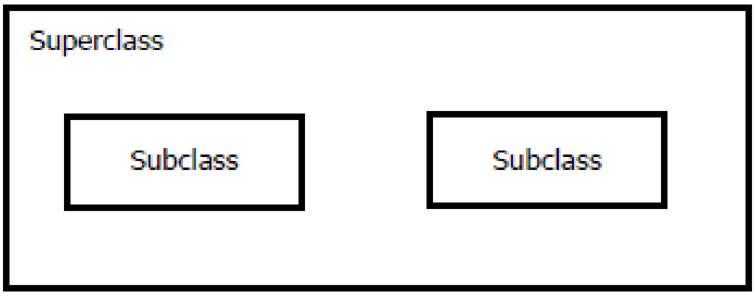
d = disjoint



optional participation

o = overlapping (inclusive or)

Alternative Notations: Crowfoot



Summary

- Enhanced Entity Relationship (EER) modelling extends the ER-modelling with generalisation/spezialisation.
- A superclass (entity) can have several subclasses (entities).
- The subclasses may be mutually disjoint (disjointness constraint).
- Each object of the superclass may be a member of some subclass (participation constraint).

Chapter 14

Normalization

Chapter 14 - Objectives

Hochschule Esslingen
University of Applied Sciences

- The purpose of normalization.
- How normalization can be used when designing a relational database.
- The potential problems associated with redundant data in base relations.
- The concept of functional dependency, which describes the relationship between attributes.
- The characteristics of functional dependencies used in normalization.

Chapter 14 - Objectives

- How to identify functional dependencies for a given relation.
- How functional dependencies identify the primary key for a relation.
- How to undertake the process of normalization.
- How normalization uses functional dependencies to group attributes into relations that are in a known normal form.

Chapter 14 - Objectives

Hochschule Esslingen
University of Applied Sciences

- How to identify the most commonly used normal forms, namely
  - First Normal Form (1NF),
  - Second Normal Form (2NF), and
  - Third Normal Form (3NF).
- The problems associated with relations that break the rules of 1NF, 2NF, or 3NF.
- How to represent attributes shown on a form as 3NF relations using normalization.

Purpose of Normalization

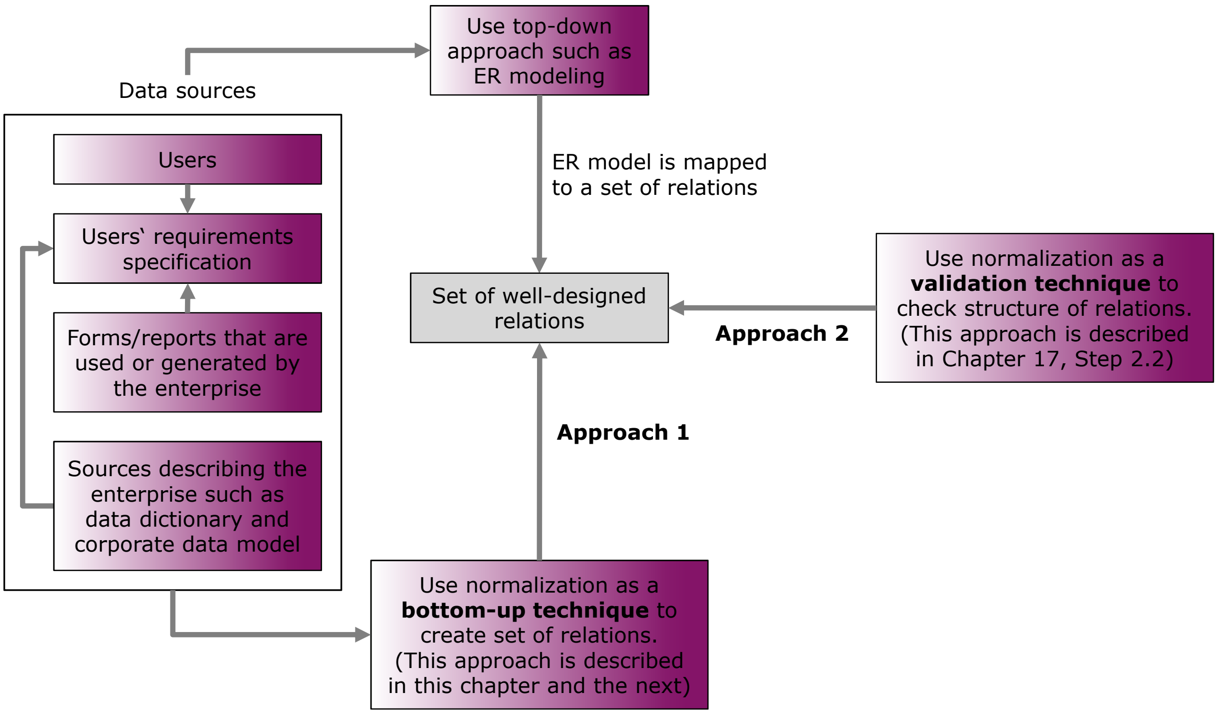
- Normalization is a technique for producing a set of suitable relations that support the data requirements of an enterprise.

- Characteristics of a suitable set of relations include:
  - the minimal number of attributes necessary to support the data requirements of the enterprise
  - attributes with a close logical relationship are found in the same relation
  - minimal redundancy with each attribute represented only once with the important exception of attributes that form all or part of foreign keys.

Purpose of Normalization

- The benefits of using a database that has a suitable set of relations is that the database will be:
  - easier for the user to access and maintain the data
  - take up minimal storage space on the computer.

How Normalization Supports Database Design



Data Redundancy and Update Anomalies

- Major aim of relational database design is to group attributes into relations to minimize data redundancy.
- Potential benefits for implemented database include:
  - Updates to the data stored in the database are achieved with a minimal number of operations thus reducing the opportunities for data inconsistencies.
  - Reduction in the file storage space required by the base relations thus minimizing costs.
- Problems associated with data redundancy are illustrated by comparing the Staff and Branch relations with the StaffBranch relation.

Staff

staffNo	name	position	salary	branchNo
SL21	John White	Manager	30000	B005
SG37	Ann Beech	Assistant	12000	B003
SG14	David Ford	Supervisor	18000	B003
SA9	Mary Howe	Assistant	9000	B007
SG5	Susan Brand	Manager	24000	B003
SL41	Julie Lee	Assistant	9000	B005

Branch

branchNo	bAddress
B005	22 Deer Rd, London
B007	16 Argyll St, Aberdeen
B003	163 Main St, Glasgow

StaffBranch

staffNo	sName	position	salary	branchNo	bAddress
SL21	John White	Manager	30000	B005	22 Deer Rd, London
SG37	Ann Beech	Assistant	12000	B003	163 Main St, Glasgow
SG14	David Ford	Supervisor	18000	B003	163 Main St, Glasgow
SA9	Mary Howe	Assistant	9000	B007	16 Argyll St, Aberdeen
SG5	Susan Brand	Manager	24000	B003	163 Main St, Glasgow
SL41	Julie Lee	Assistant	9000	B005	22 Deer Rd, London

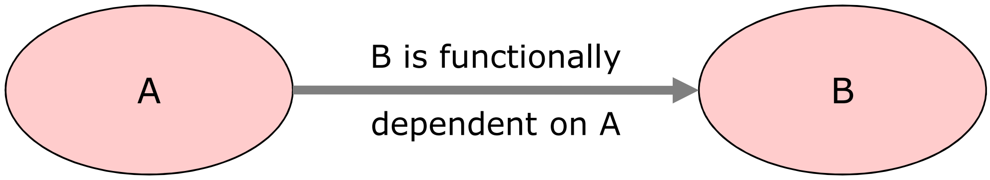
- StaffBranch relation has redundant data; the details of a branch are repeated for every member of staff.
- In contrast, the branch information appears only once for each branch in the Branch relation and only the branch number (branchNo) is repeated in the Staff relation, to represent where each member of staff is located.
- Relations that contain redundant information may potentially suffer from update anomalies.
- Types of update anomalies include
  - Insertion
  - Deletion
  - Modification

Functional Dependencies

- Important concept associated with normalization.
- Functional dependency describes relationship between attributes.
- For example, if A and B are attributes of relation R, B *is functionally dependent on* A (denoted A → B), if each value of A in R is associated with exactly one value of B in R.

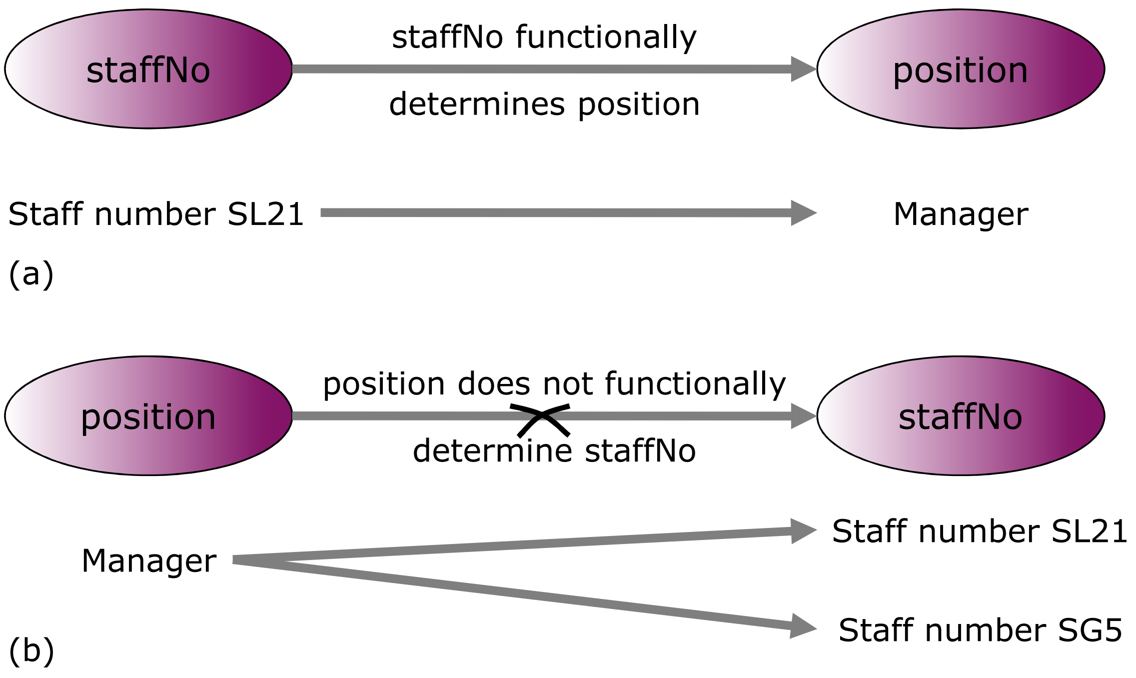
Characteristics of Functional Dependencies

- Property of the meaning or semantics of the attributes in a relation.
- Diagrammatic representation.



- The *determinant* of a functional dependency refers to the attribute or group of attributes on the left-hand side of the arrow.

An Example Functional Dependency



Example Functional Dependency that holds for all Time

- Consider the values shown in staffNo and sName attributes of the Staff relation (see Slide 9).
- Based on sample data, the following functional dependencies appear to hold.

  staffNo → sName
  sName → staffNo

- However, the only functional dependency that remains true for all possible values for the staffNo and sName attributes of the Staff relation is:

  staffNo → sName

Characteristics of Functional Dependencies

- Determinants should have the minimal number of attributes necessary to maintain the functional dependency with the attribute(s) on the right hand-side.

- This requirement is called *full functional dependency*.

- Full functional dependency indicates that if A and B are attributes of a relation:
  B *is fully functionally dependent on* A, if B is functionally dependent on A, but not on any proper subset of A.

Example Full Functional Dependency

- We have in the Staff relation (see Slide 9):

  staffNo, sName → branchNo

- True - each value of (staffNo, sName) is associated with a single value of branchNo.

- However, branchNo is also functionally dependent on a subset of (staffNo, sName), namely staffNo. The example above is a ***partial dependency***.

Transitive Dependencies

- Important to recognize a transitive dependency because its existence in a relation can potentially cause update anomalies.

- Transitive dependency describes a condition where A, B, and C are attributes of a relation such that if A → B and B → C, then C is transitively dependent on A via B (provided that A is not functionally dependent on B or C).

Example Transitive Dependency

- Consider functional dependencies in the StaffBranch relation (see Slide 9).

  staffNo → sName, position, salary, branchNo, bAddress
  branchNo → bAddress

- Transitive dependency, branchNo → bAddress exists on staffNo via branchNo.

The Process of Normalization

- Formal technique for analyzing a relation based on its candidate keys and the functional dependencies between the attributes of that relation.

- Often executed as a series of steps. Each step corresponds to a specific normal form, which has known properties.

Identifying Functional Dependencies

Hochschule Esslingen
University of Applied Sciences

- Identifying all functional dependencies between a set of attributes is relatively simple if the meaning of each attribute and the relationships between the attributes are well understood.

- This information should be provided by the enterprise in the form of discussions with users and/or documentation such as the users' requirements specification.

- However, if the users are unavailable for consultation and/or the documentation is incomplete then depending on the database application it may be necessary for the database designer to use their common sense and/or experience to provide the missing information.

Example - Identifying a set of functional dependencies for the StaffBranch relation

- Examine semantics of attributes in StaffBranch relation (see Slide 9). Assume that position held together with branch determine a member of staff's salary.

- With sufficient information available, identify the functional dependencies for the StaffBranch relation as:

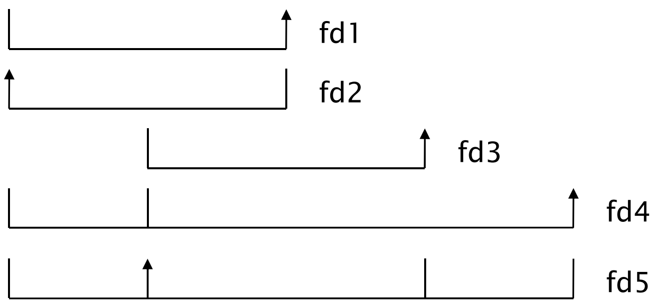
  staffNo → sName, position, salary, branchNo, bAddress
  branchNo → bAddress
  bAddress → branchNo
  branchNo, position → salary
  bAddress, position → salary

Example - Using sample data to identify functional dependencies.

- Assume that sample data values shown in relation are representative of all possible values that can be held by attributes A, B, C, D, and E.

Sample Relation

A	B	C	D	E
a	b	z	w	q
e	b	r	w	p
a	d	z	w	t
e	d	r	w	q
a	f	z	s	t
e	f	r	s	t



Example - Using sample data to identify functional dependencies.

- Functional dependencies between attributes A to E in the Sample relation.

  - $A \rightarrow C$ (fd1)
  - $C \rightarrow A$ (fd2)
  - $B \rightarrow D$ (fd3)
  - $A, B \rightarrow E$ (fd4)
  - $B, C \rightarrow E$ (can be inferred, since $A \rightarrow C$ and $C \rightarrow A$)
  - $A, D, E \rightarrow B$ (fd5)
  - $C, D, E \rightarrow B$ (can be inferred, since $A \rightarrow C$ and $C \rightarrow A$)

Identify the functional dependencies in TEACH (yes or no)

Hochschule Esslingen
University of Applied Sciences

- TEACHER → COURSE
- TEACHER → TEXT
- COURSE → TEACHER
- COURSE → TEXT
- TEXT → TEACHER
- TEXT → COURSE
- TEACHER, COURSE → TEXT
- TEACHER, TEXT → COURSE
- COURSE, TEXT → TEACHER

TEACH relation

Teacher	**Course**	**Text**
Smith	Data Structures	Bartram
Smith	DBMS	AlNour
Hall	Compilers	Hoffman
Brown	Data Bases	Augenthaler

- Main purpose of identifying a set of functional dependencies for a relation is to specify the set of integrity constraints that must hold on a relation.

- An important integrity constraint to consider first is the identification of candidate keys, one of which is selected to be the primary key for the relation.

StaffBranch

staffNo	sName	position	salary	branchNo	bAddress
SL21	John White	Manager	30000	B005	22 Deer Rd, London
SG37	Ann Beech	Assistant	12000	B003	163 Main St, Glasgow
SG14	David Ford	Supervisor	18000	B003	163 Main St, Glasgow
SA9	Mary Howe	Assistant	9000	B007	16 Argyll St, Aberdeen
SG5	Susan Brand	Manager	24000	B003	163 Main St, Glasgow
SL41	Julie Lee	Assistant	9000	B005	22 Deer Rd, London

- StaffBranch relation has five functional dependencies .
- The determinants are
  - staffNo,
  - branchNo,
  - bAddress,
  - (branchNo, position), and
  - (bAddress, position).
- To identify all candidate keys, identify the attribute (or group of attributes) that uniquely identifies each tuple in this relation.

Example - Identifying Primary Key for StaffBranch Relation

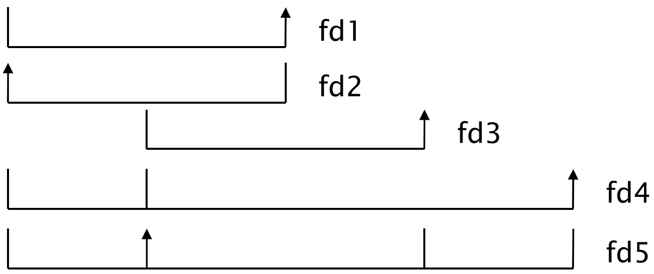
- All attributes that are not part of a candidate key should be functionally dependent on the key.
- The only candidate key and therefore primary key for StaffBranch relation, is staffNo, as *all* other attributes of the relation are functionally dependent on staffNo.

Example - Identifying Primary Key for Sample Relation

Hochschule Esslingen
University of Applied Sciences

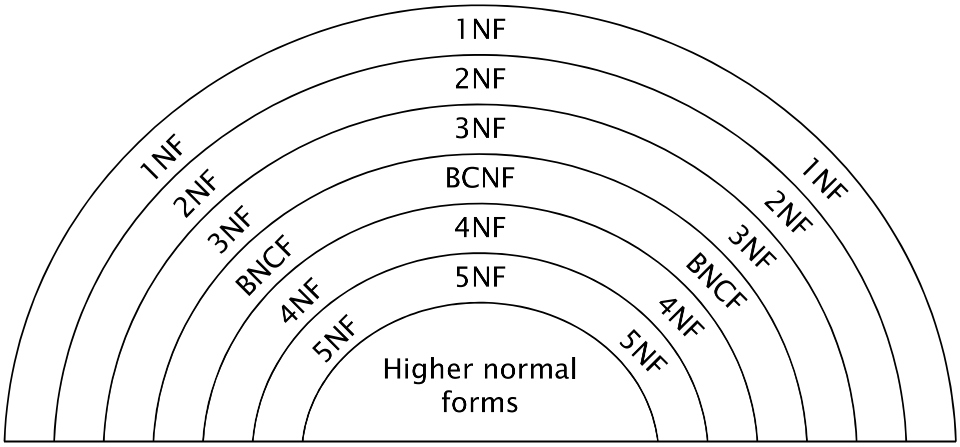
Sample Relation

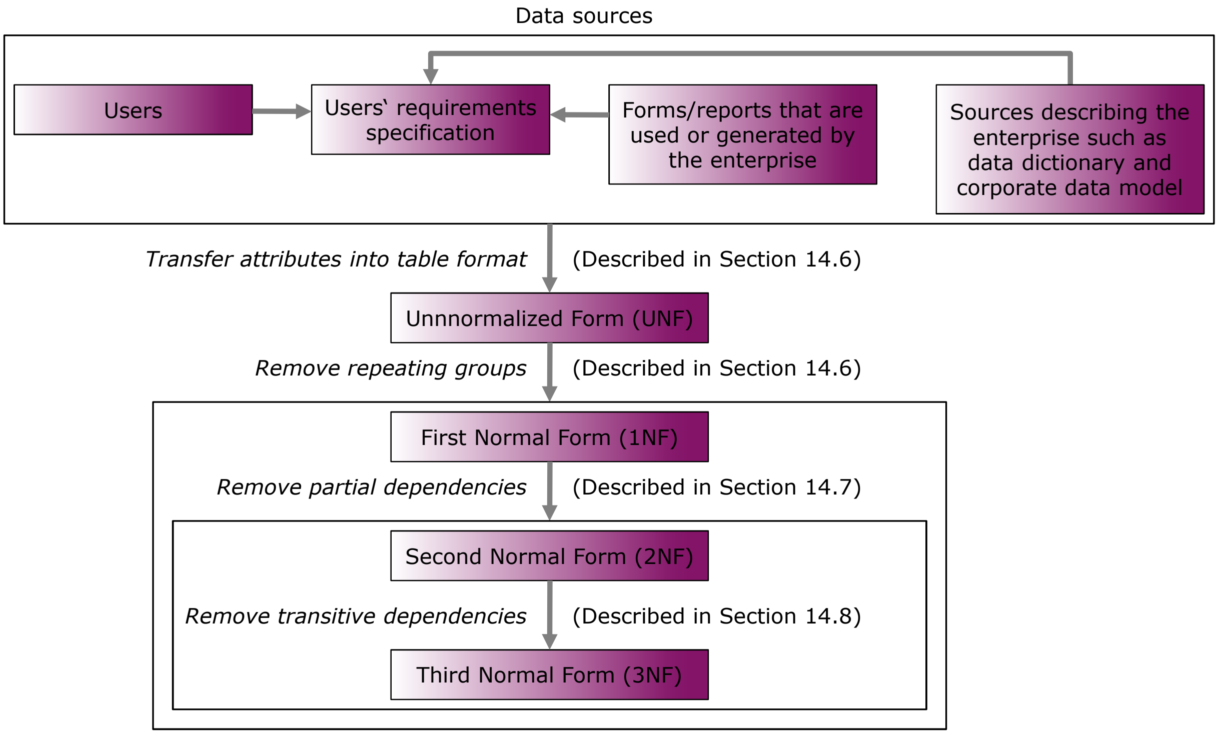
A	B	C	D	E
a	b	z	w	q
e	b	r	w	p
a	d	z	w	t
e	d	r	w	q
a	f	z	s	t
e	f	r	s	t



- The determinants in the Sample relation are:
  - A,
  - B,
  - C,
  - (A, B),
  - (C, B),
  - (A, D, E) and
  - (C, D, E)
- However, the only determinants that functionally determine all the other attributes of the relation are
  - (A, B), (B, C), (A, D, E) and (C, D, E).
- We choose (A, B) as the primary key for this relation.

- As normalization proceeds, the relations become progressively more restricted (stronger) in format and also less vulnerable to update anomalies.





Unnormalized Form (UNF)

- A table that contains one or more repeating groups.
- To create an unnormalized table
  - Transform the data from the information source (e.g. form) into table format with columns and rows.

ClientRental unnormalized table

ClientNo	cName	propertyNo	pAddress	rentStart	rentFinish	rent	ownerNo	oName
CR76	John Kay	PG4	6 lawrence St, Glasgow	1-Jul-00	31-Aug-01	350	CO40	Tina Murphy
		PG16	5 Novar Dr, Glasgow	1-Sep-02	1-Sep-02	450	CO93	Tony Shaw
CR56	Aline Stewart	PG4	6 lawrence St, Glasgow	1-Sep-99	10-Jun-00	350	CO40	Tina Murphy
		PG36	2 Manor Rd, Glasgow	10-Oct-00	1-Dec-01	370	CO93	Tony Shaw
		PG16	5 Novar Dr, Glasgow	1-Nov-02	1-Aug-03	450	CO93	Tony Shaw

First Normal Form (1NF)

- A relation in which the intersection of each row and column contains one and only one value.

- There are two approaches to removing repeating groups from unnormalized tables:

  1. Removes the repeating groups by entering appropriate data in the empty columns of rows containing the repeating data.

  2. Removes the repeating group by placing the repeating data, along with a copy of the original key attribute(s), in a separate relation. A primary key is identified for the new relation.

1NF ClientRental relation with the first approach

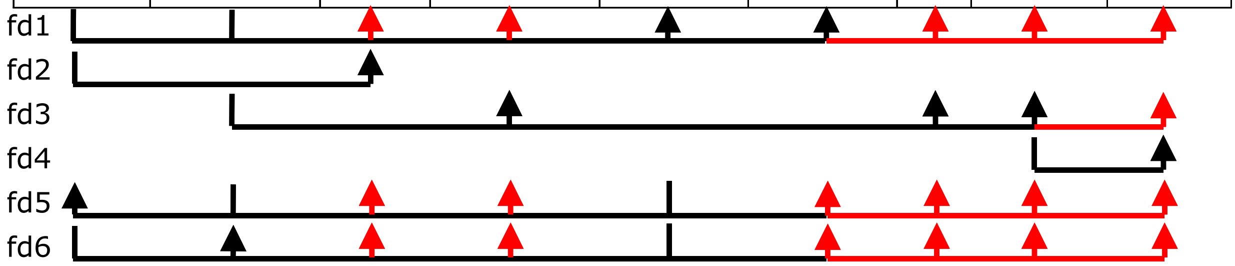
- With the first approach, we remove the repeating group (property rented details) by entering the appropriate client data into each row.

#ClientNo	#propertyNo	cName	pAddress	rentStart	rentFinish	rent	ownerNo	oName
CR76	PG4	John Kay	6 lawrence St, Glasgow	1-Jul-00	31-Aug-01	350	CO40	Tina Murphy
CR76	PG16	John Kay	5 Novar Dr, Glasgow	1-Sep-02	1-Sep-02	450	CO93	Tony Shaw
CR56	PG4	Aline Stewart	6 lawrence St, Glasgow	1-Sep-99	10-Jun-00	350	CO40	Tina Murphy
CR56	PG36	Aline Stewart	2 Manor Rd, Glasgow	10-Oct-00	1-Dec-01	370	CO93	Tony Shaw
CR56	PG16	Aline Stewart	5 Novar Dr, Glasgow	1-Nov-02	1-Aug-03	450	CO93	Tony Shaw

Functional dependencies of the ClientRental relation

Hochschule Esslingen
University of Applied Sciences

#ClientNo	#propertyNo	cName	pAddress	rentStart	rentFinish	rent	ownerNo	oName
CR76	PG4	John Kay	6 lawrence St, Glasgow	1-Jul-00	31-Aug-01	350	CO40	Tina Murphy
CR76	PG16	John Kay	5 Novar Dr, Glasgow	1-Sep-02	1-Sep-02	450	CO93	Tony Shaw
CR56	PG4	Aline Stewart	6 lawrence St, Glasgow	1-Sep-99	10-Jun-00	350	CO40	Tina Murphy
CR56	PG36	Aline Stewart	2 Manor Rd, Glasgow	10-Oct-00	1-Dec-01	370	CO93	Tony Shaw
CR56	PG16	Aline Stewart	5 Novar Dr, Glasgow	1-Nov-02	1-Aug-03	450	CO93	Tony Shaw



Functional dependencies of the ClientRental relation

Hochschule Esslingen
University of Applied Sciences

- fd1 clientNo, propertyNo → cName, pAddress, rentStart, rentFinish, rent, ownerNo, oName (Primary Key)
- fd2 clientNo → cName (Partial dependency)
- fd3 propertyNo → pAddress, rent, ownerNo, oName (Partial dependency)
- fd4 ownerNo → oName (Transitive Dependency)
- fd5 clientNo, rentStart → clientNo, propertyNo, cName, pAddress, rentFinish, rent, ownerNo, oName (Candidate key)
- fd6 propertyNo, rentStart → clientNo, cName, pAddress, rentFinish, rent, ownerNo, oName (Candidate key)

Assumptions: Client does not rent two properties with identical rentStart. No two clients rent a property together.

red = inferred functional dependency

1NF ClientRental relation with the second approach

- With the second approach, we remove the repeating group (property rented details) by placing the repeating data along with a copy of the original key attribute (clientNo) in a seprate relation.

ClientNo	cName
CR76	John Kay
CR56	Aline Stewart

ClientNo	propertyNo	pAddress	rentStart	rentFinish	rent	ownerNo	oName
CR76	PG4	6 lawrence St, Glasgow	1-Jul-00	31-Aug-01	350	CO40	Tina Murphy
CR76	PG16	5 Novar Dr, Glasgow	1-Sep-02	1-Sep-02	450	CO93	Tony Shaw
CR56	PG4	6 lawrence St, Glasgow	1-Sep-99	10-Jun-00	350	CO40	Tina Murphy
CR56	PG36	2 Manor Rd, Glasgow	10-Oct-00	1-Dec-01	370	CO93	Tony Shaw
CR56	PG16	5 Novar Dr, Glasgow	1-Nov-02	1-Aug-03	450	CO93	Tony Shaw

UNF to 1NF

- Nominate an attribute or group of attributes to act as the key for the unnormalized table.
- Identify the repeating group(s) in the unnormalized table which repeats for the key attribute(s).
- Remove the repeating group by
  - Entering appropriate data into the empty columns of rows containing the repeating data (‘flattening’ the table).

    or by
  - Placing the repeating data along with a copy of the original key attribute(s) into a separate relation.

Second Normal Form (2NF)

- Based on the concept of full functional dependency.

- Full functional dependency indicates that if
  - A and B are attributes of a relation,
  - B is fully functionally dependent on A
    if B is functionally dependent on A
    but not on any proper subset of A.

- A relation is in **2NF** if it is in 1NF and every non-candidate-key attribute is fully functionally dependent on any candidate key.

- I.e., there are no partial dependencies of non-candidate-key attributes on candidate keys.

1NF to 2NF

- Identify the candidate keys for the 1NF relation.

- Identify the functional dependencies in the relation.

- If partial dependencies exist on candidate keys remove them by placing then in a new relation along with a copy of their determinant.

Functional dependencies of the ClientRental relation

Hochschule Esslingen
University of Applied Sciences

- fd1 clientNo, propertyNo → cName, pAddress, rentStart, rentFinish, rent, ownerNo, oName (Primary Key)
- fd2 clientNo → cName (Partial dependency)
- fd3 propertyNo → pAddress, rent, ownerNo, oName (Partial dependency)
- fd4 ownerNo → oName (Transitive Dependency)
- fd5 clientNo, rentStart → clientNo, propertyNo, cName, pAddress, rentFinish, rent, ownerNo, oName (Candidate key)
- fd6 propertyNo, rentStart → clientNo, cName, pAddress, rentFinish, rent, ownerNo, oName (Candidate key)

Assumptions: Client does not rent two properties with identical rentStart. No two clients rent a property together.

red = inferred functional dependency

2NF derived from the ClientRental relation

Hochschule Esslingen
University of Applied Sciences

- After removing the partial dependencies, the creation of the three new relations called Client, Rental, and PropertyOwner

ClientNo	cName
CR76	John Kay
CR56	Aline Stewart

ClientNo	propertyNo	rentStart	rentFinish
CR76	PG4	1-Jul-00	31-Aug-01
CR76	PG16	1-Sep-02	1-Sep-02
CR56	PG4	1-Sep-99	10-Jun-00
CR56	PG36	10-Oct-00	1-Dec-01
CR56	PG16	1-Nov-02	1-Aug-03

propertyNo	pAddress	rent	ownerNo	oName
PG4	6 lawrence St, Glasgow	350	CO40	Tina Murphy
PG16	5 Novar Dr, Glasgow	450	CO93	Tony Shaw
PG36	2 Manor Rd, Glasgow	370	CO93	Tony Shaw

Third Normal Form (3NF)

- Based on the concept of transitive dependency.

- Transitive Dependency is a condition where
  - A, B and C are attributes of a relation such that if A → B and B → C,
  - then C is *transitively dependent* on A through B. (Provided that A is not functionally dependent on B or C).

- A relation is in **3NF** if it is in 2NF and no non-candidate-key attribute is transitively dependent on some candidate key.

The functional dependencies for the Client, Rental and PropertyOwner relations

- Client
  - fd2 clientNo → cName (Primary Key)

- Rental
  - fd1 clientNo, propertyNo → rentStart, rentFinish (Primary Key)
  - fd5 clientNo, rentStart → propertyNo, rentFinish (Candidate key)
  - fd6 propertyNo, rentStart → clientNo, rentFinish (Candidate key)

- PropertyOwner
  - fd3 propertyNo → pAddress, rent, ownerNo, oName (Primary Key)
  - fd4 ownerNo → oName (Transitive Dependency)

3NF derived from the PropertyOwner relation

- The normalization of 2NF relations to 3NF involves the removal of transitive dependencies by placing the attribute(s) in a new relation along with a copy of the determinant.

Client

ClientNo	cName
CR76	John Kay
CR56	Aline Stewart

Rental

ClientNo	propertyNo	rentStart	rentFinish
CR76	PG4	1-Jul-00	31-Aug-01
CR76	PG16	1-Sep-02	1-Sep-02
CR56	PG4	1-Sep-99	10-Jun-00
CR56	PG36	10-Oct-00	1-Dec-01
CR56	PG16	1-Nov-02	1-Aug-03

PropertyOwner

propertyNo	pAddress	rent	ownerNo
PG4	6 lawrence St, Glasgow	350	CO40
PG16	5 Novar Dr, Glasgow	450	CO93
PG36	2 Manor Rd, Glasgow	370	CO93

Owner

ownerNo	oName
CO40	Tina Murphy
CO93	Tony Shaw

2NF to 3NF

- Identify the candidate keys in the 2NF relation.
- Identify functional dependencies in the relation.
- If transitive dependencies exist on candidate keys remove them by placing them in a new relation along with a copy of their dominant.

Summary

- The purpose of normalization is the design of suitable set of relations.
- In the bottom-up approach, normalization creates a set of relations from a universal relation.
- In the top-down approach, normalization checks whether the given set of relations is designed well.
- Normalized relations have less update-, delete- and insert-anomalies.
- There are several normal forms: 1NF, 2NF, 3NF, BCNF, 4NF, 5NF
- The stronger the normal form the less redundancy there is; however information might get lost in the transformation process.

Chapter 15

Advanced Normalization

Chapter 15 - Objectives

- How inference rules can identify a set of all functional dependencies for a relation.
- How inference rules called Armstrong's axioms can identify a minimal set of useful functional dependencies from the set of all functional dependencies for a relation.
- Normal forms that go beyond Third Normal Form (3NF), which includes Boyce-Codd Normal Form (BCNF), Fourth Normal Form (4NF), and Fifth Normal Form (5NF).
- How to identify Boyce-Codd Normal Form (BCNF).
- How to represent attributes shown on a report as BCNF relations using normalization.

Chapter 15 - Objectives

- Concept of multi-valued dependencies and Fourth Normal Form (4NF).
- The problems associated with relations that break the rules of 4NF.
- How to create 4NF relations from a relation, which breaks the rules of 4NF.
- Concept of join dependency and Fifth Normal Form (5NF).
- The problems associated with relations that break the rules of 5NF.
- How to create 5NF relations from a relation, which breaks the rules of 5NF.

More on Functional Dependencies

- The complete set of functional dependencies for a given relation can be very large.
- Important to find an approach that can reduce the set to a manageable size.

- Need to identify a set of functional dependencies (represented as X) for a relation that is smaller than the complete set of functional dependencies (represented as Y) for that relation and has the property that every functional dependency in Y is implied by the functional dependencies in X.

- The set of all functional dependencies that are implied by a given set of functional dependencies X is called the *closure of X*, written X^+ .

- A set of inference rules, called *Armstrong's axioms*, specifies how new functional dependencies can be inferred from given ones.

- Let A, B, and C be subsets of the attributes of the relation R. Armstrong's inference rules are as follows:

1. Reflexivity:	If B is a subset of A, then A → B
2. Augmentation:	If A → B, then A,C → B,C
3. Transitivity	If A → B and B → C, then A → C

- Further rules can be derived from the first three rules that simplify the practical task of computing X^+. Let D be another subset of the attributes of relation R, then:

4. Self-determination	A → A
5. Decomposition	If A → B,C, then A → B and A → C
6. Union	If A → B and A → C, then A → B,C
7. Composition	If A → B and C → D then A,C → B,D

Minimal Sets of Functional Dependencies

- A set of functional dependencies Y is *covered* by a set of functional dependencies X, if every functional dependency in Y is also in X^+; that is, every dependency in Y can be inferred from X.

- A set of functional dependencies X is *minimal* if it satisfies the following conditions:
  - Every dependency in X has a single attribute on its right-hand side.
  - We cannot replace any dependency A → B in X with dependency C → B, where C is a proper subset of A, and still have a set of dependencies that is equivalent to X.
  - We cannot remove any dependency from X and still have a set of dependencies that is equivalent to X.

Boyce–Codd Normal Form (BCNF)

- BCNF is based on functional dependencies that take into account all candidate keys in a relation; however, BCNF also has additional constraints compared with the general definition of 3NF.

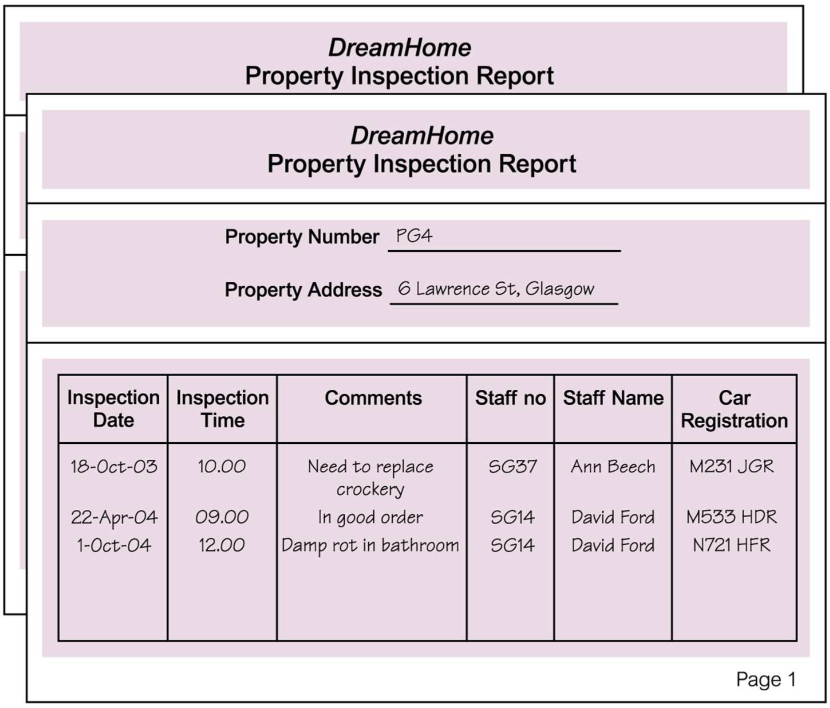
- Boyce–Codd Normal Form (BCNF)
  - A relation is in BCNF if and only if every determinant is a candidate key.

- Difference between 3NF and BCNF is that for a functional dependency A → B, 3NF allows this dependency in a relation if B is a primary-key attribute and A is not a candidate key. Whereas, BCNF insists that for this dependency to remain in a relation, A must be a candidate key.

Boyce–Codd Normal Form (BCNF)

- Every relation in BCNF is also in 3NF. However, a relation in 3NF is not necessarily in BCNF.
- Violation of BCNF is quite rare.
- The potential to violate BCNF may occur in a relation that:
  - contains two (or more) composite candidate keys;
  - the candidate keys overlap, that is have at least one attribute in common.

Review of Normalization (UNF to BCNF)



DreamHome
Property Inspection Report

Property Number PG4

Property Address 6 Lawrence St, Glasgow

Inspection Date	Inspection Time	Comments	Staff no	Staff Name	Car Registration
18-Oct-03	10.00	Need to replace crockery	SG37	Ann Beech	M231 JGR
22-Apr-04	09.00	In good order	SG14	David Ford	M533 HDR
1-Oct-04	12.00	Damp rot in bathroom	SG14	David Ford	N721 HFR

Page 1

Figure 14.3
DreamHome Property Inspection reports.

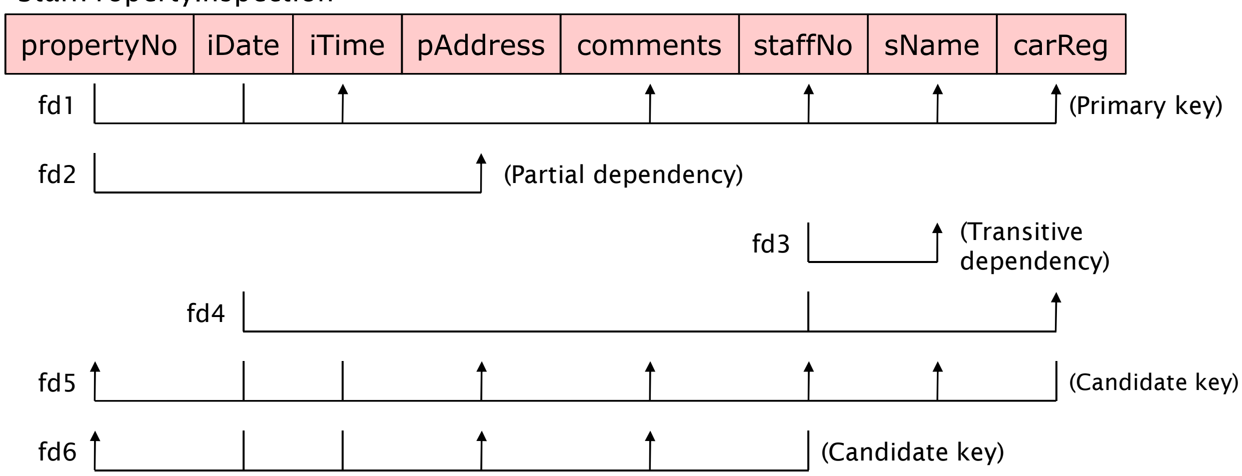
StaffPropertyInspection

propertyNo	pAddress	iDate	iTime	comments	staffNo	sName	carReg
PG4	6 Lawrence St, Glasgow	18-Oct-03	10.00	Need to replace crockery	SG37	Ann Beech	M231 JGR
		22-Apr-04	09.00	In good order	SG14	David Ford	M533 HDR
		1-Oct-04	12.00	Damp rot in bathroom	SG14	David Ford	N721 HFR
PG16	5 Novar Dr, Glasgow	22-Apr-04	13.00	Replace living room carpet	SG14	David Ford	M533 HDR
		24-Oct-04	14.00	Good condition	SG37	Ann Beech	N721 HFR

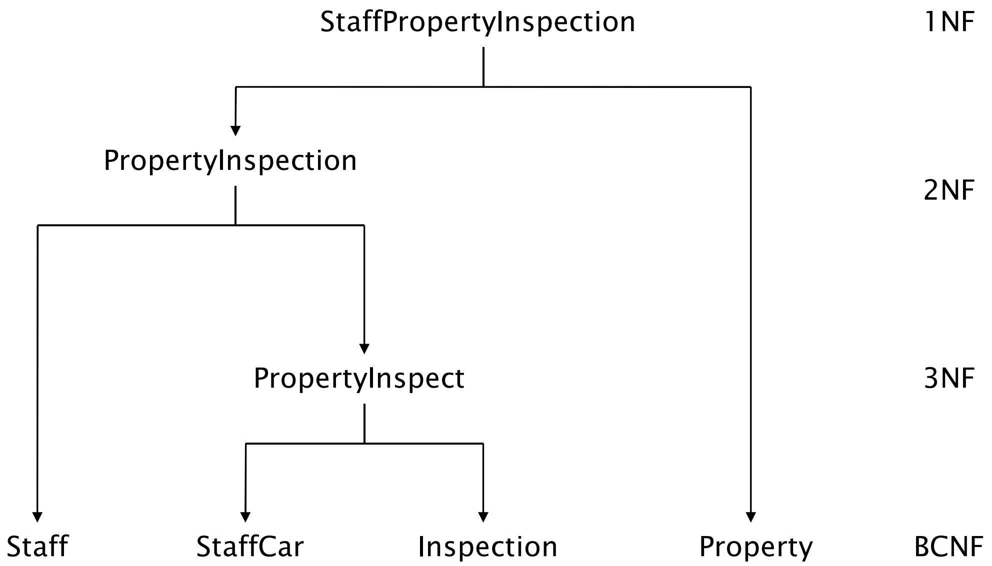
StaffPropertyInspection

propertyNo	iDate	iTime	pAddress	comments	staffNo	sName	carReg
PG4	18-Oct-03	10.00	6 Lawrence St, Glasgow	Need to replace crockery	SG37	Ann Beech	M231 JGR
PG4	22-Apr-04	09.00	6 Lawrence St, Glasgow	In good order	SG14	David Ford	M533 HDR
PG4	1-Oct-04	12.00	6 Lawrence St, Glasgow	Damp rot in bathroom	SG14	David Ford	N721 HFR
PG16	22-Apr-04	13.00	5 Novar Dr, Glasgow	Replace living room carpet	SG14	David Ford	M533 HDR
PG16	24-Oct-04	14.00	5 Novar Dr, Glasgow	Good condition	SG37	Ann Beech	N721 HFR

StaffPropertyInspection



Review of Normalization (UNF to BCNF)



Lossless-join and Dependency Preservation Properties

Hochschule Esslingen
University of Applied Sciences

- Two important properties of decomposition.
  - *Lossless-join property* enables us to find any instance of the original relation from corresponding instances in the smaller relations.
  - *Dependency preservation property* enables us to enforce a constraint on the original relation by enforcing some constraint on each of the smaller relations.

- Transformations into 1NF, 2NF and 3NF have the lossless-join property and the dependency preservation property.

- Transformations into BCNF have the lossless-join property but do not necessarily preserve functional dependencies.

- Transformations into 4NF have the lossless-join property.

Fourth Normal Form (4NF)

- Although BCNF removes anomalies due to functional dependencies, another type of dependency called a multi-valued dependency (MVD) can also cause data redundancy.

- Possible existence of multi-valued dependencies in a relation is due to 1NF and can result in data redundancy.

- Multi-valued Dependency (MVD)
  - Dependency between attributes (for example, A, B, and C) in a relation, such that for each value of A there is a set of values for B and a set of values for C. However, the set of values for B and C are independent of each other.

Hochschule Esslingen
University of Applied Sciences

Fourth Normal Form (4NF)

- MVD between attributes A, B, and C in a relation using the following notation:

  A −>> B

  A −>> C

- MVD can be further defined as being trivial or nontrivial.
  - A MVD A −>> B in relation R is defined as being *trivial* if (a) B is a subset of A or (b) A ∪ B = R.
  - A MVD is defined as being *nontrivial* otherwise
  - A trivial MVD does not specify a constraint on a relation, while a nontrivial MVD does specify a constraint.

- A relation that is in Boyce-Codd Normal Form and contains no nontrivial multi-valued dependencies is in 4NF.

4NF - Example

BranchStaffOwner

branchNo	sName	oName
B003	Ann Beech	Carol Farrel
B003	David Ford	Carol Farrel
B003	Ann Beech	Tina Murphy
B003	David Ford	Tina Murphy

BranchStaff

branchNo	sName
B003	Ann Beech
B003	David Ford

BranchOwner

branchNo	oName
B003	Carol Farrel
B003	Tina Murphy

Fifth Normal Form (5NF)

- A relation decomposed into two relations must have the lossless-join property, which ensures that no spurious tuples are generated when relations are reunited through a natural join operation.
- However, there are requirements to decompose a relation into more than two relations. Although rare, these cases are managed by join dependency and fifth normal form (5NF).
- Defined as a relation that has no join dependency.

5NF - Example

(a) PropertyItemSupplier (illegal state)

propertyNo	itemDescription	supplierNo
PG4	Bed	S1
PG4	Chair	S2
PG16	Bed	S2

When this tuple is added to relation.

(b) PropertyItemSupplier (legal state)

propertyNo	itemDescription	supplierNo
PG4	Bed	S1
PG4	Chair	S2
PG16	Bed	S2
PG4	Bed	S2

This new tuple must also be added to exist in any legal state of the relation.

5NF - Example

PropertyItem

propertyNo	itemDescription
PG4	Bed
PG4	Chair
PG16	Bed

ItemSupplier

itemDescription	supplierNo
Bed	S1
Chair	S2
Bed	S2

PropertySupplier

propertyNo	supplierNo
PG4	S1
PG4	S2
PG16	S2

Denormalisation

- Normalised relational models help to reduce redundancies and inconsistencies (less update-, insert-, delete-anomalies).
- However, a normalised relational model might not be suitable for an implementation which requires fast query responses.
- Example:

Company	City	State	Postcode
StarMedia	Atlanta	GA	30060
DataMax	Ney York	NY	1002
Speedy	Los Angeles	CA	71702

- Relation is not in 3NF since City and State functionally depend on Postcode. However, transforming relation into 3NF will cause many lookups.
- The process of denormalisation, i.e. the joining of tables of a normalised relational model, delivers possibly a more suitable relational model for the problem at hand.

Summary

- Normalisation is used to validate a relational model (analysis) or to buit a relational model from a universal relation (synthesis).
- A normalised relational model reduces redundancies and the problems of update-, insert and delete-anomalies.
- A normalised relational model is not necessarily a good one.
- Normalisation into 2NF or 3NF does not violate the dependency preserving property or the lossless-join property.
- Normalisation into BCNF might destroy a functional dependency.

Chapter 16

Methodology
Conceptual Databases Design

Chapter 16 - Objectives

- The purpose of a design methodology.
- Database design has three main phases: conceptual, logical, and physical design.
- How to decompose the scope of the design into specific views of the enterprise.
- How to use Entity–Relationship (ER) modelling to build a conceptual data model based on the data requirements of an enterprise.

Chapter 16 - Objectives

- How to document the process of conceptual database design.
- Why end-users play an integral role throughout the process of conceptual database design.
- How to validate the resultant conceptual model to ensure it is a true and accurate representation of the data requirements of the enterprise.

Database Design Methodology

- Design Methodology
  - A structured approach that uses procedures, techniques, tools, and documentation aids to support and facilitate the process of design.
- Database Design Methodology has three main phases:
  - Conceptual database design
  - Logical database design
  - Physical database design

Database Design Methodology

- Conceptual database design
  - The process of constructing a model of the data used in an enterprise, independent of all physical considerations.
- Logical database design
  - The process of constructing a model of the data used in an enterprise based on a specific data model (e.g. relational), but independent of a particular DBMS and other physical considerations.
- Physical database design
  - The process of producing a description of the implementation of the database on secondary storage; it describes the base relations, file organizations, and indexes design used to achieve efficient access to the data, and any associated integrity constraints and security measures.

Critical Success Factors in Database Design

- Work interactively with the users as much as possible.
- Follow a structured methodology throughout the data modeling process.
- Employ a data-driven approach.
- Incorporate structural and integrity considerations into the data models.
- Combine conceptualization, normalization, and transaction validation techniques into the data modeling methodology.

Critical Success Factors in Database Design

- Use diagrams to represent as much of the data models as possible.
- Use a Database Design Language (DBDL) to represent additional data semantics.
- Build a data dictionary to supplement the data model diagrams.
- Be willing to repeat steps.

Overview Database Design Methodology

- Conceptual database design
- Step 1 Build conceptual data model
  - Step 1.1 Identify entity types
  - Step 1.2 Identify relationship types
  - Step 1.3 Identify and associate attributes with entity or relationship types
  - Step 1.4 Determine attribute domains
  - Step 1.5 Determine candidate, primary, and alternate (secondary) key attributes
  - Step 1.6 Consider use of enhanced modeling concepts (optional)
  - Step 1.7 Check model for redundancy
  - Step 1.8 Validate conceptual model against user transactions
  - Step 1.9 Review conceptual data model with user

Overview Database Design Methodology

- Logical database design for the relational model

- Step 2 Build and validate logical data model
  - Step 2.1 Derive relations for logical data model
  - Step 2.2 Validate relations using normalization
  - Step 2.3 Validate relations against user transactions
  - Step 2.4 Define integrity constraints
  - Step 2.5 Review logical data model with user
  - Step 2.6 Merge logical data models into global model (optional)
  - Step 2.7 Check for future growth

Overview Database Design Methodology

Hochschule Esslingen
University of Applied Sciences

- Physical database design for relational database

- Step 3 Translate logical data model for target DBMS
  - Step 3.1 Design base relations
  - Step 3.2 Design representation of derived data
  - Step 3.3 Design general constraints

- Step 4 Design file organizations and indexes
  - Step 4.1 Analyze transactions
  - Step 4.2 Choose file organization
  - Step 4.3 Choose indexes
  - Step 4.4 Estimate disk space requirements

Overview Database Design Methodology

Hochschule Esslingen
University of Applied Sciences

- Step 5 Design user views
- Step 6 Design security mechanisms
- Step 7 Consider the introduction of controlled redundancy
- Step 8 Monitor and tune the operational system

Step 1 Build Conceptual Data Model

Hochschule Esslingen
University of Applied Sciences

- To build a conceptual data model of the data requirements of the enterprise.
  - Model comprises entity types, relationship types, attributes and attribute domains, primary and alternate (secondary) keys, and integrity constraints.
- Step 1.1 Identify entity types
  - To identify the required entity types.
- Step 1.2 Identify relationship types
  - To identify the important relationships that exist between the entity types.

- Step 1.3 Identify and associate attributes with entity or relationship types
  - To associate attributes with the appropriate entity or relationship types and document the details of each attribute.
- Step 1.4 Determine attribute domains
  - To determine domains for the attributes in the data model and document the details of each domain.

Step 1 Build Conceptual Data Model

Hochschule Esslingen
University of Applied Sciences

- Step 1.5 Determine candidate, primary, and alternate (secondary) key attributes
  - To identify the candidate key(s) for each entity and if there is more than one candidate key, to choose one to be the primary key and the others as alternate (secondary) keys.
- Step 1.6 Consider use of enhanced modeling concepts (optional)
  - To consider the use of enhanced modeling concepts, such as specialization / generalization, aggregation, and composition.

Step 1 Build Conceptual Data Model

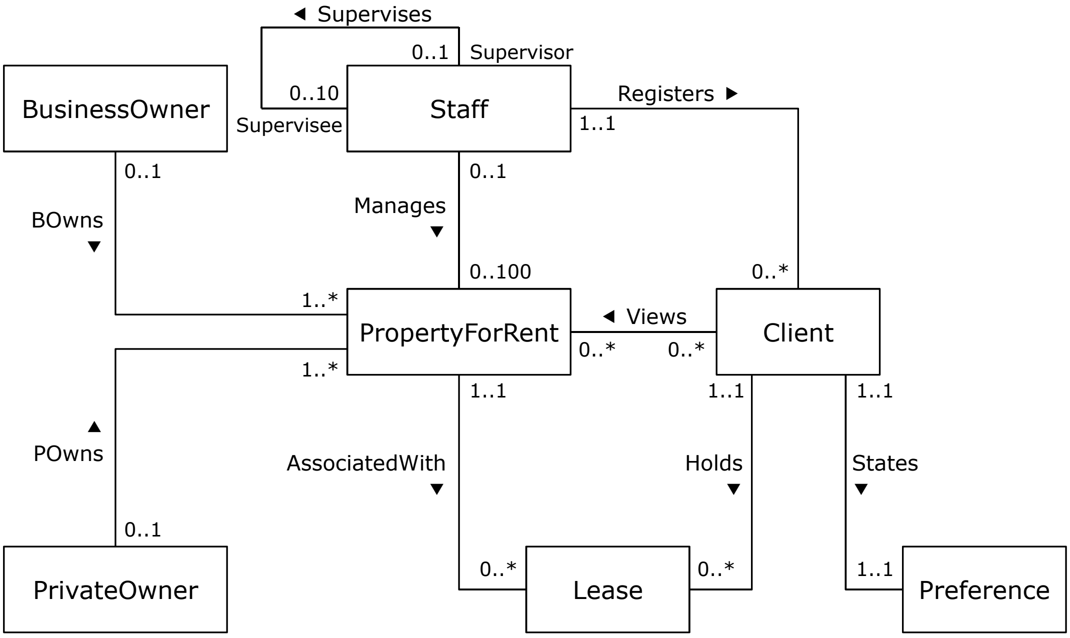
- Step 1.7 Check model for redundancy
  - To check for the presence of any redundancy in the model and to remove any that does exist.
- Step 1.8 Validate conceptual model against user transactions
  - To ensure that the conceptual model supports the required transactions.
- Step1.9 Review conceptual data model with user
  - To review the conceptual data model with the user to ensure that the model is a 'true' representation of the data requirements of the enterprise.

Extract from data dictionary for Staff user views of DreamHome showing description of entities

Entity name	*Description*	*Aliases*	*Occurrence*
Staff	General term descr bing al staff employed by *DreamHome.*	Employee	Each member of staff works at one particular branch.
PropertyForRent	General term descr bing all property for rent.	Property	Each property has a s ngle owner and is available at one specific branch, where the property is managed by one member of staff. A property is viewed by many clients and rented by a single client, at any one time.

Figure 15.1 Extract from the data dictionary for the Staff user views of *DreamHome* showing a description of entities.

First-cut ER diagram for Staff user views of DreamHome



Extract from data dictionary for Staff user views of DreamHome showing description of relationships

Entity name	*Multiplicity*	*Relationship*	*Multiplicity*	*Entity name*
Staff	0..1 0..1	*Manages* *Supervises*	0..100 0..10	**PropertyForRent** **Staff**
PropertyForRent	1..1	*AssociatedWith*	0..*	**Lease**

Figure 15.3 Extract from the data dictionary for the Staff user views of *DreamHome* showing a description of relationships.

Extract from data dictionary for Staff user views of DreamHome showing description of attributes

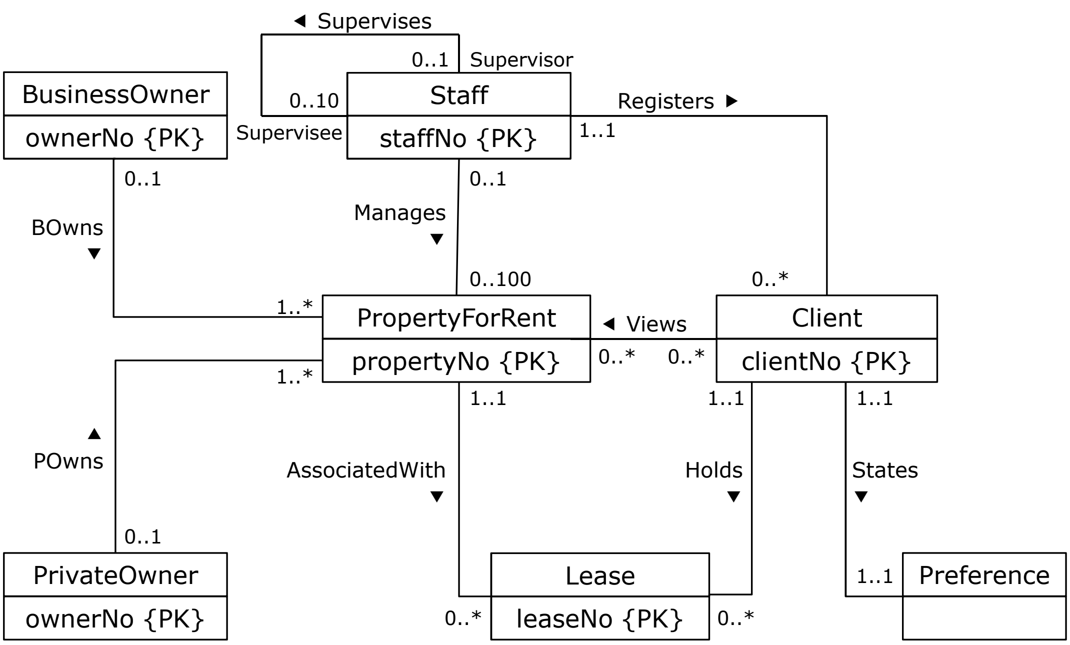
Hochschule Esslingen
University of Applied Sciences

Entity name	*Attributes*	*Description*	*Data Type & Length*	*Nulls*	*Multi-valued*	. . .
Staff	**staffNo**	Unique y identifies a member of staff	5 variable characters	No	No	
	name					
	fName	First name of staff	15 variable characters	No	No	
	lName	Last name of staff	15 variable characters	No	No	
	position	Job title of member of staff	10 variable characters	No	No	
	sex	Gender of member of staff	1 character (M or F)	Yes	No	
	DOB	Date of birth of member of staff	Date	Yes	No	
PropertyForRent	**propertyNo**	Unique y identifies a property for rent	5 variable characters	No	No	

Figure 15.4 Extract from the data dictionary for the Staff user views of *DreamHome* showing a description of attributes.

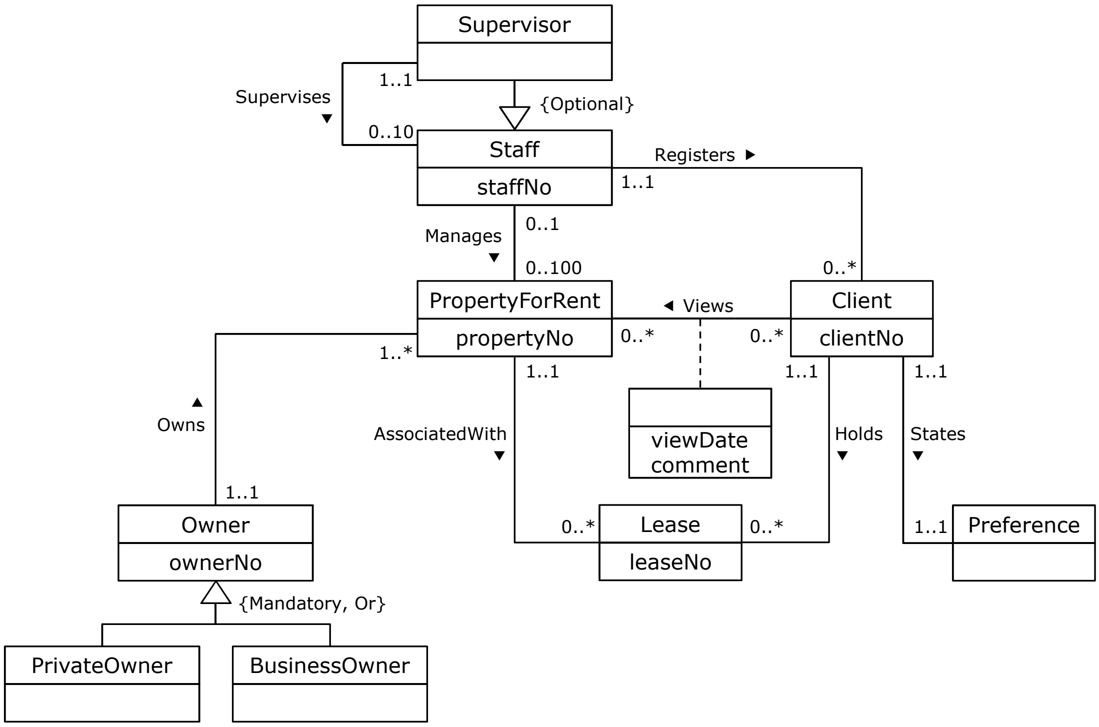
ER diagram for Staff user views of DreamHome with primary keys added

Hochschule Esslingen
University of Applied Sciences



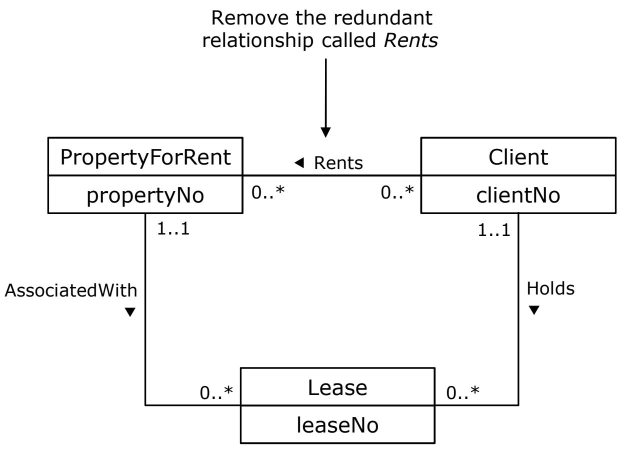
Revised ER diagram for Staff user views of DreamHome with specialization / generalization

Hochschule Esslingen
University of Applied Sciences

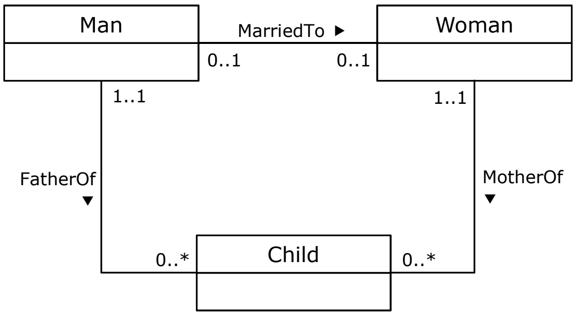


Example of removing a redundant relationship called Rents

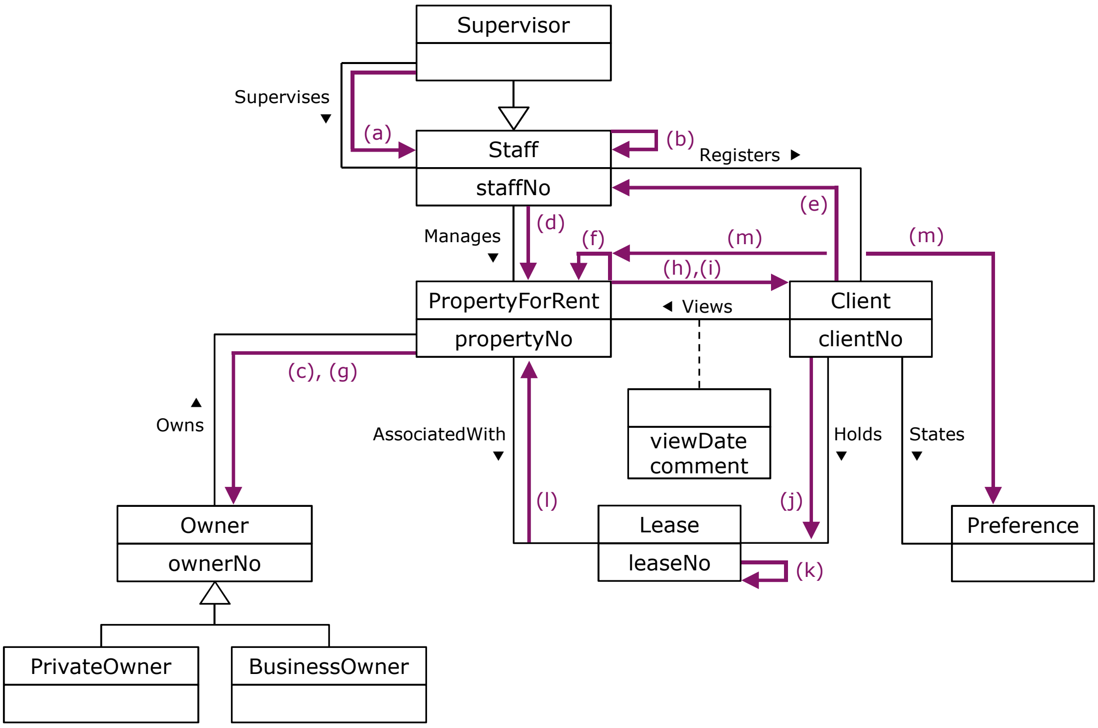
Hochschule Esslingen
University of Applied Sciences



Example of a non-redundant relationship FatherOf



Using pathways to check that the conceptual model supports the user transactions



Using pathways to check that the conceptual model supports the user transactions

a) List details of staff supervised by a named Supervisor at the branch.
b) List details of all Assistants alphabetically by name at the branch.
c) List the details of property (including the rental deposit) available for rent at the branch, along with the owner´s details.
d) List the details of properties managed by a named member of staff at the branch.
e) List the clients registering at the branch and the names of the members of staff who registered the clients.
f) Identify properties located in Glasgow with rents no higher than £450.
g) … (see Connolly, Appendix A.2.2)

Summary

- To ensure the quality of a database system it is important to follow a database design methodology.
- A proved and tested database design methodology consist out of three phases:
  - Conceptual database design → data model independent of any physical considerations
  - Logic database design → data model of a certain type (e.g. relational) independent of DBMS or physical considerations
  - Physical database design → physical implementation of data model (DBMS, files, indexes, …)

Hochschule Esslingen
University of Applied Sciences

Chapter 17

Methodology
Logical Database Design for the Relational Model

Chapter 17 - Objectives

Hochschule Esslingen
University of Applied Sciences

- How to derive a set of relations from a conceptual data model.
- How to validate these relations using the technique of normalization.
- How to validate a logical data model to ensure it supports the required transactions.
- How to merge local logical data models based on one or more user views into a global logical data model that represents all user views.
- How to ensure that the final logical data model is a true and accurate representation of the data requirements of the enterprise.

Step 2
Build and Validate Logical Data Model

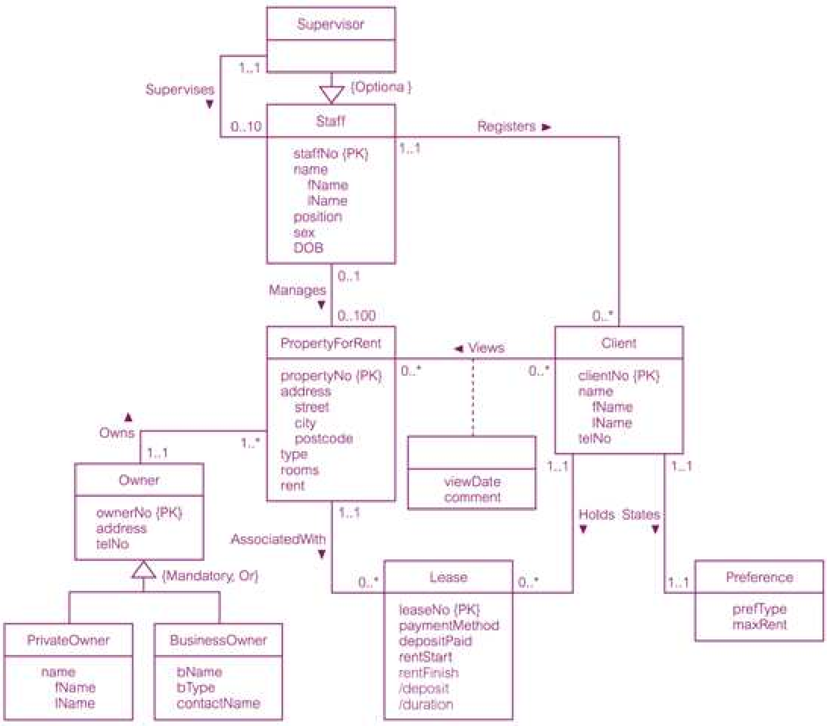
Hochschule Esslingen
University of Applied Sciences

- To translate the conceptual data model into a logical data model and then to validate this model to check that it is structurally correct using normalization and supports the required transactions.

- Step 2.1 Derive relations for logical data model
  - To create relations for the logical data model to represent the entities, relationships, and attributes that have been identified.

Conceptual data model for Staff view showing all attributes

Hochschule Esslingen
University of Applied Sciences



1. Strong entity types

   – For each strong entity in the data model, create a relation that includes all the simple attributes of that entity. For composite attributes, include only the constituent simple attributes.

2. Weak entity types

   – For each weak entity in the data model, create a relation that includes all the simple attributes of that entity. The primary key of a weak entity is partially or fully derived from each owner entity and so the identification of the primary key of a weak entity cannot be made until after all the relationships with the owner entities have been mapped.

Step 2.1
Derive relations for logical data model

Hochschule Esslingen
University of Applied Sciences

3. One-to-many (1:*) binary relationship types

   – For each 1:* binary relationship, the entity on the ‘one side’ of the relationship is designated as the parent entity and the entity on the ‘many side’ is designated as the child entity. To represent this relationship, post a copy of the primary key attribute(s) of parent entity into the relation representing the child entity, to act as a foreign key.

4. One-to-one (1:1) binary relationship types

– Creating relations to represent a 1:1 relationship is more complex as the cardinality cannot be used to identify the parent and child entities in a relationship. Instead, the participation constraints are used to decide whether it is best to represent the relationship by combining the entities involved into one relation or by creating two relations and posting a copy of the primary key from one relation to the other.

– Consider the following
  a) *mandatory* participation on *both* sides of 1:1 relationship
  b) *mandatory* participation on *one* side of 1:1 relationship
  c) *optional* participation on *both* sides of 1:1 relationship.

a) *Mandatory* participation on *both* sides of 1:1 relationship

– Combine entities involved into one relation and choose one of the primary keys of original entities to be primary key of the new relation, while the other (if one exists) is used as an alternate (secondary) key.

b) *Mandatory* participation on *one* side of a 1:1 relationship

– Identify parent and child entities using participation constraints. Entity with optional participation in relationship is designated as parent entity, and entity with mandatory participation is designated as child entity. A copy of primary key of the parent entity is placed in the relation representing the child entity. If the relationship has one or more attributes, these attributes should follow the posting of the primary key to the child relation.

c) *Optional* participation on *both* sides of a 1:1 relationship

- In this case, the designation of the parent and child entities is arbitrary unless we can find out more about the relationship that can help a decision to be made one way or the other.

5. One-to-one (1:1) recursive relationships

- For a 1:1 recursive relationship, follow the rules for participation as described above for a 1:1 relationship.
  - » mandatory participation on both sides, represent the recursive relationship as a single relation with two copies of the primary key.
  - » mandatory participation on only one side, option to create a single relation with two copies of the primary key, or to create a new relation to represent the relationship. The new relation would only have two attributes, both copies of the primary key. As before, the copies of the primary keys act as foreign keys and have to be renamed to indicate the purpose of each in the relation.
  - » optional participation on both sides, again create a new relation as described above.

6. Superclass/subclass relationship types

- Identify superclass entity as parent entity and subclass entity as the child entity. There are various options on how to represent such a relationship as one or more relations.

- The selection of the most appropriate option is dependent on a number of factors such as the disjointness and participation constraints on the superclass/subclass relationship, whether the subclasses are involved in distinct relationships, and the number of participants in the superclass/subclass relationship.

Table 16.1 Guidelines for the representation of a superclass/subclass relationship based on the participation and disjoint constraints.

Participation constraint	Disjoint constraint	Relations required
Mandatory	Nondisjoint {And}	Single relation (with one or more discriminators to distinguish the type of each tuple)
Optional	Nondisjoint {And}	Two relations: one relation for superclass and one relation for all subclasses (with one or more discriminators to distinguish the type of each tuple)
Mandatory	Disjoint {Or}	Many relations: one relation for each combined superclass/subclass
Optional	Disjoint {Or}	Many relations: one relation for superclass and one for each subclass

Representation of superclass / subclass relationship based on participation and disjointness

Option 1 – Mandatory, nondisjoint

AllOwner (ownerNo, address, telNo, fName, lName, bName, bType, contactName, pOwnerFlag, bOwnerFlag)
Primary Key ownerNo

Option 2 – Optional, nondisjoint

Owner (ownerNo, address, telNo)
Primary Key ownerNo

OwnerDetails (ownerNo, fName lName, bName, bType, contactName, pOwnerFlag, bOwnerFlag)
Primary Key ownerNo
Foreign Key ownerNo **references** Owner(ownerNo)

Option 3 – Mandatory, disjoint

PrivateOwner (ownerNo, fName, Name, address, telNo)
Primary Key ownerNo

BusinessOwner (ownerNo, bName, bType, contactName, address, telNo)
Primary Key ownerNo

Option 4 – Optional, disjoint

Owner (ownerNo, address, telNo)
Primary Key ownerNo

PrivateOwner (ownerNo, fName, Name)
Primary Key ownerNo
Foreign Key ownerNo **references** Owner(ownerNo)

BusinessOwner (ownerNo, bName, bType, contactName)
Primary Key ownerNo
Foreign Key ownerNo **references** Owner(ownerNo)

Figure 16.2 Various representations of the Owner superclass/subclass relationship based on the participation and disjointness constraints shown in Table 16.1.

Step 2.1
Derive relations for logical data model

7. Many-to-many (*:*) binary relationship types

   – Create a relation to represent the relationship and include any attributes that are part of the relationship. We post a copy of the primary key attribute(s) of the entities that participate in the relationship into the new relation, to act as foreign keys. These foreign keys will also form the primary key of the new relation, possibly in combination with some of the attributes of the relationship.

8. Complex relationship types

- Create a relation to represent the relationship and include any attributes that are part of the relationship. Post a copy of the primary key attribute(s) of the entities that participate in the complex relationship into the new relation, to act as foreign keys. Any foreign keys that represent a 'many' relationship (for example, 1..*, 0..*) generally will also form the primary key of this new relation, possibly in combination with some of the attributes of the relationship.

9. Multi-valued attributes

- Create a new relation to represent multi-valued attribute and include primary key of entity in new relation, to act as a foreign key. Unless the multi-valued attribute is itself an alternate key of the entity, the primary key of the new relation is the combination of the multi-valued attribute and the primary key of the entity.

Summary of how to map entities and relationships to relations

Entity/Relationship	Mapping
Strong entity	Create relation that includes all simple attributes.
Weak entity	Create relation that includes all simple attributes (primary key still has to be identified after the relationship with each owner entity has been mapped).
1:* binary relationship	Post primary key of entity on 'one' side to act as foreign key in relation representing entity on 'many' side. Any attributes of relationship are also posted to 'many' side.
1:1 binary relationship:	
(a) Mandatory participation on both sides	Combine entities into one relation.
(b) Mandatory participation on one side	Post primary key of entity on 'optional' side to act as foreign key in relation representing entity on 'mandatory' side.
(c) Optional participation on both sides	Arbitrary without further information.
Superclass/subclass relationship	See Table 16.1.
: binary relationship, complex relationship	Create a relation to represent the relationship and include any attributes of the relationship. Post a copy of the primary keys from each of the owner entities into the new relation to act as foreign keys.
Multi-valued attribute	Create a relation to represent the multi-valued attribute and post a copy of the primary key of the owner entity into the new relation to act as a foreign key.

Relations for the Staff user views of DreamHome

Staff (staffNo, fName, lName, position, sex, DOB, supervisorStaffNo) **Primary Key** staffNo **Foreign Key** supervisorStaffNo **references** Staff(staffNo)	**PrivateOwner** (ownerNo, fName, lName, address, telNo) **Primary Key** ownerNo
BusinessOwner (ownerNo, bName, bType, contactName, address, telNo) **Primary Key** ownerNo **Alternate Key** bName **Alternate Key** telNo	**Client** (clientNo, fName, lName, telNo, prefType, maxRent, staffNo) **Primary Key** clientNo **Foreign Key** staffNo **references** Staff(staffNo)
PropertyForRent (propertyNo, street, city, postcode, type, rooms, rent, ownerNo, staffNo) **Primary Key** propertyNo **Foreign Key** ownerNo **references** PrivateOwner(ownerNo) and BusinessOwner(ownerNo) **Foreign Key** staffNo **references** Staff(staffNo)	**Viewing** (clientNo, propertyNo, dateView, comment) **Primary Key** clientNo, propertyNo **Foreign Key** clientNo **references** Cl ent(clientNo) **Foreign Key** propertyNo **references** PropertyForRent(propertyNo)
Lease (leaseNo, paymentMethod, depositPaid, rentStart, rentFinish, clientNo, propertyNo) **Primary Key** leaseNo **Alternate Key** propertyNo, rentStart **Alternate Key** clientNo, rentStart **Foreign Key** clientNo **references** Client(clientNo) **Foreign Key** propertyNo **references** PropertyForRent(propertyNo) **Derived** deposit (PropertyForRent.rent*2) **Derived** duration (rentFinish – rentStart)	

Steps 2.2 and 2.3

Step 2.2 Validate relations using normalization

- To validate the relations in the logical data model using normalization.

Step 2.3 Validate relations against user transactions

- To ensure that the relations in the logical data model support the required transactions.

Step 2.4 Check integrity constraints

- To check integrity constraints are represented in the logical data model. This includes identifying:
  - Required data
  - Attribute domain constraints
  - Multiplicity
  - Entity integrity
  - Referential integrity
  - General constraints

Referential integrity constraints for relations in Staff user views of DreamHome

Staff (staffNo, fName, lName, position, sex, DOB, supervisorStaffNo)
Primary Key staffNo
Foreign Key supervisorStaffNo references Staff(staffNo) ON UPDATE CASCADE ON DELETE SET NULL

Client (clientNo, fName, lName, telNo, prefType, maxRent, staffNo)
Primary Key clientNo
Foreign Key staffNo references Staff(staffNo) ON UPDATE CASCADE ON DELETE NO ACTION

PropertyForRent (propertyNo, street, city, postcode, type, rooms, rent, ownerNo, staffNo)
Primary Key propertyNo
Foreign Key ownerNo references PrivateOwner(ownerNo) and BusinessOwner(ownerNo) ON UPDATE CASCADE ON DELETE NO ACTION
Foreign Key staffNo references Staff(staffNo) ON UPDATE CASCADE ON DELETE SET NULL

Viewing (clientNo, propertyNo, dateView, comment)
Primary Key clientNo, propertyNo
Foreign Key clientNo references Client(clientNo) ON UPDATE CASCADE ON DELETE NO ACTION
Foreign Key propertyNo references PropertyForRent(propertyNo) ON UPDATE CASCADE ON DELETE CASCADE

Lease (leaseNo, paymentMethod, depositPaid, rentStart, rentFinish, cl entNo, propertyNo)
Primary Key leaseNo
Alternate Key propertyNo, rentStart
Alternate Key clientNo, rentStart
Foreign Key clientNo references Client(clientNo) ON UPDATE CASCADE ON DELETE NO ACTION
Foreign Key propertyNo references PropertyForRent(propertyNo) ON UPDATE CASCADE ON DELETE NO ACTION

Figure 16.4 Referential integrity constraints for the relations in the Staff user views of *DreamHome*.

Steps 2.5 and 2.6

Step 2.5 Review logical data model with user

- To review the logical data model with the users to ensure that they consider the model to be a true representation of the data requirements of the enterprise.

Step 2.6 Merge logical data models into global model (optional)

- To merge logical data models into a single global logical data model that represents all user views of a database.

Step 2.6.1 Merge local logical data models into global model

Hochschule Esslingen
University of Applied Sciences

- To merge local logical data model into a single global logical data model.
- This activities in this step include:
  - Step 2.6.1 Merge local logical data models into global model
  - Step 2.6.2 Validate global logical data model
  - Step 2.6.3 Review global logical data model with users

Step 2.6.1 Merge logical data models into a global model

Hochschule Esslingen
University of Applied Sciences

- Task typically includes:

1. Review the names and contents of entities/relations and their candidate keys.
2. Review the names and contents of relationships/foreign keys.
3. Merge entities/relations from the local data models
4. Include (without merging) entities/relations unique to each local data model
5. Merge relationships/foreign keys from the local data models.
6. Include (without merging) relationships/foreign keys unique to each local data model.
7. Check for missing entities/relations and relationships/foreign keys.
8. Check foreign keys.
9. Check Integrity Constraints.
10. Draw the global ER/relation diagram
11. Update the documentation.

Steps 2.6.2 and 2.6.3

- Step 2.6.2 Validate global logical data model:

  To validate the relations created from the global logical data model using the technique of normalization and to ensure they support the required transactions, if necessary.

- Step 2.6.3 Review global logical data model with users:

  To review the global logical data model with the users to ensure that they consider the model to be a true representation of the data requirements of an enterprise.

Relations for the Branch user views of DreamHome

Branch (branchNo, street, c ty, postcode, mgrStaffNo) **Primary Key** branchNo **Alternate Key** postcode **Foreign Key** mgrStaffNo **references** Manager(staffNo)	**Telephone** (telNo, branchNo) **Primary Key** telNo **Foreign Key** branchNo **references** Branch(branchNo)
Staff (staffNo, name, position, salary, supervisorStaffNo, branchNo) **Primary Key** staffNo **Foreign Key** supervisorStaffNo **references** Staff(staffNo) **Foreign Key** branchNo **references** Branch(branchNo)	**Manager** (staffNo, mgrStartDate, bonus) **Primary Key** staffNo **Foreign Key** staffNo **references** Staff(staffNo)
PrivateOwner (ownerNo, name, address, telNo) **Primary Key** ownerNo	**BusinessOwner** (bName, bType, contactName, address, telNo) **Primary Key** bName **Alternate Key** telNo
PropertyForRent (propertyNo, street, city, postcode, type, rooms, rent, ownerNo, staffNo, bName, branchNo) **Primary Key** propertyNo **Foreign Key** ownerNo **references** PrivateOwner(ownerNo) **Foreign Key** bName **references** BusinessOwner(bName) **Foreign Key** staffNo **references** Staff(staffNo) **Foreign Key** branchNo **references** Branch(branchNo)	**Client** (clientNo, name, telNo, prefType, maxRent) **Primary Key** clientNo

Relations for the Branch user views of DreamHome

Hochschule Esslingen
University of Applied Sciences

Lease (leaseNo, paymentMethod, depositPaid, rentStart, rentFinish, clientNo, propertyNo) **Primary Key** leaseNo **Alternate Key** propertyNo, rentStart **Alternate Key** clientNo, rentStart **Foreign Key** clientNo **references** Client(clientNo) **Foreign Key** propertyNo **references** PropertyForRent(propertyNo) **Derived** deposit (PropertyForRent.rent*2) **Derived** duration (rentFinish – rentStart)	**Registration** (clientNo, branchNo, staffNo, dateJoined) **Primary Key** clientNo **Foreign Key** clientNo **references** Client(clientNo) **Foreign Key** branchNo **references** Branch(branchNo) **Foreign Key** staffNo **references** Staff(staffNo)
Advert (propertyNo, newspaperName, dateAdvert, cost) **Primary Key** propertyNo, newspaperName, dateAdvert **Foreign Key** propertyNo **references** PropertyForRent(propertyNo) **Foreign Key** newspaperName **references** Newspaper(newspaperName)	**Newspaper** (newspaperName, address, telNo, contactName) **Primary Key** newspaperName **Alternate Key** telNo

Relations that represent the global logical data model for DreamHome

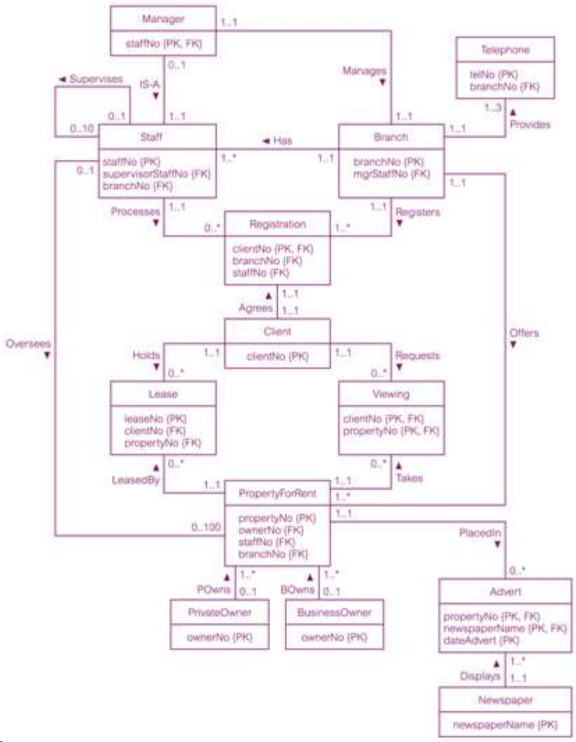
Hochschule Esslingen
University of Applied Sciences

Branch (branchNo, street, city, postcode, mgrStaffNo) **Primary Key** branchNo **Alternate Key** postcode **Foreign Key** mgrStaffNo **references** Manager(staffNo)	**Telephone** (telNo, branchNo) **Primary Key** telNo **Foreign Key** branchNo **references** Branch(branchNo)
Staff (staffNo, fName, lName, position, sex, DOB, salary, supervisorStaffNo, branchNo) **Primary Key** staffNo **Foreign Key** supervisorStaffNo **references** Staff(staffNo) **Foreign Key** branchNo **references** Branch(branchNo)	**Manager** (staffNo, mgrStartDate, bonus) **Primary Key** staffNo **Foreign Key** staffNo **references** Staff(staffNo)
PrivateOwner (ownerNo, fName, lName, address, telNo) **Primary Key** ownerNo	**BusinessOwner** (ownerNo, bName, bType, contactName, address, telNo) **Primary Key** ownerNo **Alternate Key** bName **Alternate Key** telNo
PropertyForRent (propertyNo, street, city, postcode, type, rooms, rent, ownerNo, staffNo, branchNo) **Primary Key** propertyNo **Foreign Key** ownerNo **references** PrivateOwner(ownerNo) and BusinessOwner(ownerNo) **Foreign Key** staffNo **references** Staff(staffNo) **Foreign Key** branchNo **references** Branch(branchNo)	**Viewing** (clientNo, propertyNo, dateView, comment) **Primary Key** clientNo, propertyNo **Foreign Key** clientNo **references** Client(clientNo) **Foreign Key** propertyNo **references** PropertyForRent(propertyNo)

Relations that represent the global logical data model for DreamHome

Client (clientNo, fName, lName, telNo, prefType, maxRent) **Primary Key** clientNo	**Registration** (clientNo, branchNo, staffNo, dateJoined) **Primary Key** clientNo **Foreign Key** clientNo **references** Client(clientNo) **Foreign Key** branchNo **references** Branch(branchNo) **Foreign Key** staffNo **references** Staff(staffNo)
Lease (leaseNo, paymentMethod, depositPaid, rentStart, rentFinish, cl entNo, propertyNo) **Primary Key** easeNo **Alternate Key** propertyNo, rentStart **Alternate Key** c ientNo, rentStart **Foreign Key** clientNo **references** Client(clientNo) **Foreign Key** propertyNo **references** PropertyForRent(propertyNo) **Derived** depos t (PropertyForRent.rent*2) **Derived** duration (rentFinish – rentStart)	**Newspaper** (newspaperName, address, telNo, contactName) **Primary Key** newspaperName **Alternate Key** telNo
Advert (propertyNo, newspaperName, dateAdvert, cost) **Primary Key** propertyNo, newspaperName, dateAdvert **Foreign Key** propertyNo **references** PropertyForRent(propertyNo) **Foreign Key** newspaperName **references** Newspaper(newspaperName)	

Global relation diagram for DreamHome



Summary

- Entities and relations of the conceptual data model are translated into relations of the logical data model in a predefined way.
- Subclasses can be translated into relations in various ways dependent on participation and disjointness constraints and the attributes of subclasses.
- Normalization ensures that redundancies in and between relations are reduced to a minimum.
- User feedback should be used to ensure that the relations hold all necessary data and support all necessary transactions.

Chapter 18

Methodology
Physical Database Design for Relational Databases

Chapter 18 - Objectives

- Purpose of physical database design.
- How to map the logical database design to a physical database design.
- How to design base relations for target DBMS.
- How to design general constraints for target DBMS.
- How to select appropriate file organizations based on analysis of transactions.
- When to use secondary indexes to improve performance.

Chapter 18 - Objectives

- How to estimate the size of the database.
- How to design user views.
- How to design security mechanisms to satisfy user requirements.

Logical v. Physical Database Design

- Sources of information for physical design process includes logical data model and documentation that describes model.
- Logical database design is concerned with the *what*, physical database design is concerned with the *how*.

Physical Database Design

- Process of producing a description of the implementation of the database on secondary storage.
- It describes the base relations, file organizations, and indexes used to achieve efficient access to the data, and any associated integrity constraints and security measures.

Hochschule Esslingen
University of Applied Sciences

Overview of Physical Database Design Methodology

- Step 3 Translate logical data model for target DBMS
  - Step 3.1 Design base relations
  - Step 3.2 Design representation of derived data
  - Step 3.3 Design general constraints
- Step 4 Design file organizations and indexes
  - Step 4.1 Analyze transactions
  - Step 4.2 Choose file organizations
  - Step 4.3 Choose indexes
  - Step 4.4 Estimate disk space requirements

Overview of Physical Database Design Methodology

Hochschule Esslingen
University of Applied Sciences

- Step 5 Design user views
- Step 6 Design security mechanisms
- Step 7 Consider the introduction of controlled redundancy
- Step 8 Monitor and tune operational system

Step 3 Translate Logical Data Model for Target DBMS

Hochschule Esslingen
University of Applied Sciences

- To produce a relational database schema from the logical data model that can be implemented in the target DBMS.
- Need to know functionality of target DBMS such as how to create base relations and whether the system supports the definition of:
  - PKs, FKs, and AKs (alternate keys = secondary keys)
  - required data – i.e. whether system supports NOT NULL
  - domains
  - relational integrity constraints
  - general constraints.

Step 3.1 Design base relations

- To decide how to represent base relations identified in logical model in target DBMS.

- For each relation, need to define:
  - the name of the relation
  - a list of simple attributes in brackets
  - the PK and, where appropriate, AKs and FKs
  - referential integrity constraints for any FKs identified

- From data dictionary, we have for each attribute:
  - its domain, consisting of a data type, length, and any constraints on the domain
  - an optional default value for the attribute
  - whether it can hold nulls
  - whether it is derived, and if so, how it should be computed.

DDL for the PropertyForRent Relation

```
    Domain PropertyNumber:      variable length character string, length 5
    Domain Street:              variable length character string, length 25
    Domain City:                variable length character string, length 15
    Domain Postcode:            variable length character string, length 8
    Domain PropertyType:        single character, must be one of 'B', 'C', 'D', 'E', 'F', 'H', 'M', 'S'
    Domain PropertyRooms:       integer, in the range 1–15
    Domain PropertyRent:        monetary value, in the range 0.00–9999.99
    Domain OwnerNumber:         variable length character string, length 5
    Domain StaffNumber:         variable length character string, length 5
    Domain BranchNumber:        fixed length character string, length 4
PropertyForRent(
    propertyNo   PropertyNumber   NOT NULL,
    street       Street           NOT NULL,
    city         City             NOT NULL,
    postcode     Postcode,
    type         PropertyType     NOT NULL DEFAULT 'F',
    rooms        PropertyRooms    NOT NULL DEFAULT 4,
    rent         PropertyRent     NOT NULL DEFAULT 600,
    ownerNo      OwnerNumber      NOT NULL,
    staffNo      StaffNumber,
    branchNo     BranchNumber     NOT NULL,
    PRIMARY KEY (propertyNo),
    FOREIGN KEY (staffNo) REFERENCES Staff(staffNo) ON UPDATE CASCADE ON DELETE SET NULL,
    FOREIGN KEY (ownerNo) REFERENCES PrivateOwner(ownerNo) and BusinessOwner(ownerNo)
                              ON UPDATE CASCADE ON DELETE NO ACTION,
    FOREIGN KEY (branchNo) REFERENCES Branch(branchNo)
                              ON UPDATE CASCADE ON DELETE NO ACTION);
```

Step 3.2 Design representation of derived data

- To decide how to represent any derived data present in logical data model in target DBMS.
- Examine logical data model and data dictionary, and produce list of all derived attributes.
- Derived attribute can be stored in database or calculated every time it is needed.
- Option selected is based on:
  - additional cost to store the derived data and keep it consistent with operational data from which it is derived
  - cost to calculate it each time it is required.
- Less expensive option is chosen subject to performance constraints.

PropertyforRent Relation and Staff Relation with Derived Attribute noOfProperties

PropertyForRent

propertyNo	street	city	postcode	type	rooms	rent	ownerNo	staffNo	branchNo
PA14	16 Holhead	Aberdeen	AB7 5SU	House	6	650	CO46	SA9	B007
PL94	6 Argyll St	London	NW2	Flat	4	400	CO87	SL41	B005
PG4	6 Lawrence St	Glasgow	G11 9QX	Flat	3	350	CO40		B003
PG36	2 Manor Rd	Glasgow	G32 4QX	Flat	3	375	CO93	SG37	B003
PG21	18 Dale Rd	Glasgow	G12	House	5	600	CO87	SG37	B003
PG16	5 Novar Dr	Glasgow	G12 9AX	Flat	4	450	CO93	SG14	B003

Staff

staffNo	fName	lName	branchNo	noOfProperties
SL21	John	White	B005	0
SG37	Ann	Beech	B003	2
SG14	David	Ford	B003	1
SA9	Mary	Howe	B007	1
SG5	Susan	Brand	B003	0
SL41	Julie	Lee	B005	1

Step 3.3 Design general constraints

To design the general constraints for target DBMS

- Some DBMS provide more facilities than others for defining enterprise constraints.

  Example:

```
CONSTRAINT StaffNotHandlingTooMuch
      CHECK (NOT EXISTS (
              SELECT staffNo
              FROM PropertyForRent
              GROUP BY staffNo
              HAVING COUNT(*) > 100))
```


Step 4 Design File Organizations and Indexes

To determine optimal file organizations to store the base relations and the indexes that are required to achieve acceptable performance; that is, the way in which relations and tuples will be held on secondary storage

- Must understand the typical *workload* that database must support.

Step 4.1 Analyze transactions

To understand the functionality of the transactions that will run on the database and to analyze the important transactions

- Attempt to identify performance criteria, such as:
  - transactions that run frequently and will have a significant impact on performance
  - transactions that are critical to the business
  - times during the day/week when there will be a high demand made on the database (called the *peak load*).
- Use this information to identify the parts of the database that may cause performance problems.

Hochschule Esslingen
University of Applied Sciences

Step 4.1 Analyze transactions

- Also need to know high-level functionality of the transactions, such as:
  - attributes that are updated;
  - search criteria used in a query.
- Often not possible to analyze all transactions, so investigate most 'important' ones.
- To help identify these can use:
  - *transaction/relation cross-reference matrix*, showing relations that each transaction accesses, and/or
  - *transaction usage map*, indicating which relations are potentially heavily used.

Step 4.1 Analyze transactions

- To focus on areas that may be problematic:

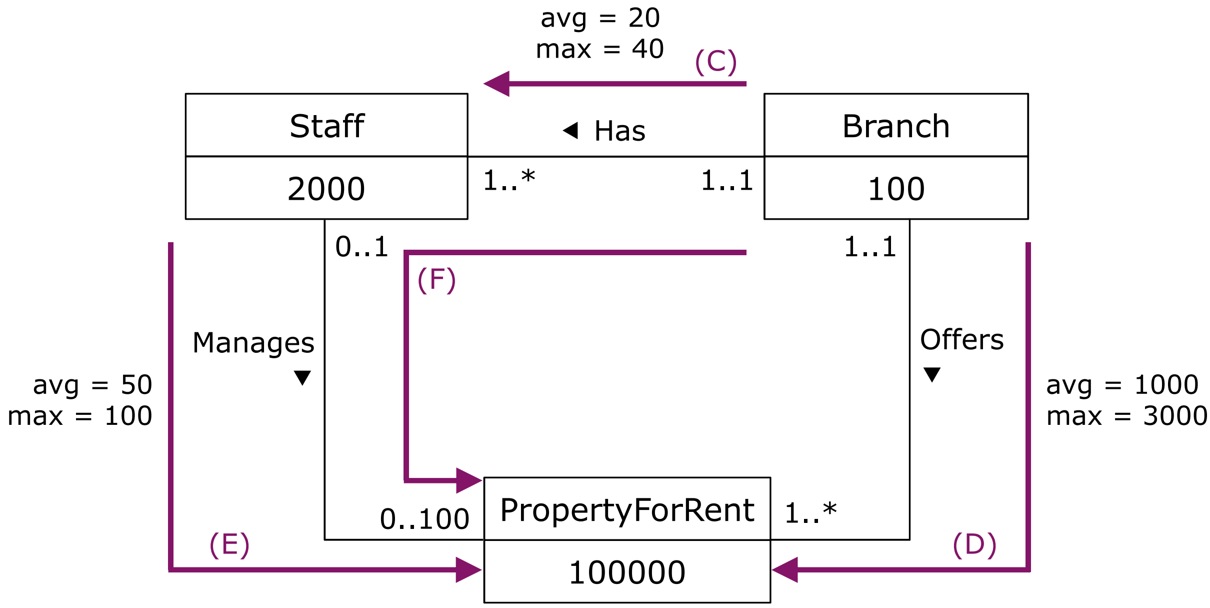
  1. Map all transaction paths to relations.
  2. Determine which relations are most frequently accessed by transactions.
  3. Analyze the data usage of selected transactions that involve these relations.

Cross-referencing transactions and relations

Transaction/ Relation	(A)				(B)				(C)				(D)				(E)				(F)			
	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D
Branch										X				X								X		
Telephone																								
Staff		X				X				X								X				X		
Manager																								
PrivateOwner	X																							
BusinessOwner	X																							
PropertyForRent	X					X	X	X						X				X				X		
Viewing																								
Client																								
Registration																								
Lease																								
Newspaper																								
Advert																								

I = Insert; R = Read; U = Update; D = Delete

Example Transaction Usage Map



Example Transaction Analysis Form

Transaction Analysis Form **1-Sept-2004**

Transaction (D) List the property number, address, type, and rent of all properties in Glasgow, ordered by rent

Transaction volume

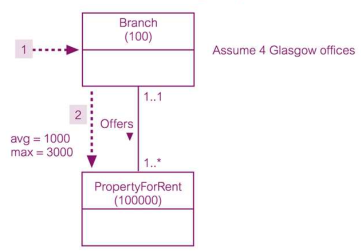
Average: 50 per hour

Peak: 100 per hour (between 17.00 and 19.00 Monday–Saturday)

```
SELECT propertyNo, p.street, p.postcode, type, rent
FROM Branch b INNER JOIN PropertyForRent p ON
        b.branchNo = p.branchNo
WHERE p.city = 'Glasgow'
ORDER BY rent;
```

Predicate: p.city = 'Glasgow'
Join attributes: b.branchNo = p.branchNo
Ordering attribute: rent
Grouping attribute: none
Built-in functions: none
Attributes updated: none

Transaction usage map



Access	Entity	Type of Access	No. of References		
			Per Transaction	Avg Per Hour	Peak Per Hour
1	**Branch** (entry)	R	100	5000	10000
2	**PropertyForRent**	R	4000–12000	200000–600000	400000–1200000
Total References			**4100–12100**	**205000–605000**	**410000–1210000**

Figure 17.4 Example transaction analysis form.

Step 4.2 Choose file organizations

To determine an efficient file organization for each base relation

- File organizations include Heap, Hash, Indexed Sequential Access Method (ISAM), B+-Tree, and Clusters.
- Some DBMSs may not allow selection of file organizations.

Step 4.3 Choose indexes

To determine whether adding indexes will improve the performance of the system

- One approach is to keep tuples unordered and create as many <u>secondary indexes</u> as necessary.
- Another approach is to order tuples in the relation by specifying a *primary* or clustering *index*.
- In this case, choose the attribute for ordering or clustering the tuples as:
  – attribute that is used most often for join operations - this makes join operation more efficient, or
  – attribute that is used most often to access the tuples in a relation in order of that attribute.

Step 4.3 Choose indexes

- If ordering attribute chosen is key of relation, index will be a primary index; otherwise, index will be a clustering index.

- Each relation can only have either a primary index or a clustering index.

- Secondary indexes provide a mechanism for specifying an additional key for a base relation that can be used to retrieve data more efficiently.

Step 4.3 Choose indexes

Hochschule Esslingen
University of Applied Sciences

- Have to balance overhead involved in maintenance and use of secondary indexes against performance improvement gained when retrieving data.

- This includes:
  - adding an index record to every secondary index whenever tuple is inserted
  - updating secondary index when corresponding tuple updated
  - increase in disk space needed to store secondary index
  - possible performance degradation during query optimization to consider all secondary indexes.

Step 4.3 Choose indexes – Guidelines for choosing 'wish-list'

1. Do not index small relations.
2. Index PK of a relation if it is not a key of the file organization.
3. Add secondary index to a FK if it is frequently accessed.
4. Add secondary index to any attribute heavily used as a secondary key.
5. Add secondary index on attributes involved in: selection or join criteria; ORDER BY; GROUP BY; and other operations involving sorting (such as UNION or DISTINCT).

Step 4.3 Choose indexes – Guidelines for choosing 'wish-list'

6. Add secondary index on attributes involved in built-in functions.
7. Add secondary index on attributes that could result in an index-only plan.
8. Avoid indexing an attribute or relation that is frequently updated.
9. Avoid indexing an attribute if the query will retrieve a significant proportion of the relation.
10. Avoid indexing attributes that consist of long character strings.

Step 4.4 Estimate disk space requirements

To estimate the amount of disk space that will be required by the database.

Step 5 Design User Views

To design the user views that were identified during the Requirements Collection and Analysis stage of the database system development lifecycle.

Step 6 Design Security Measures

To design the security measures for the database as specified by the users.

- Use GRANT & REVOKE to assign only the necessary object and system privileges to roles (not to users)

- Assign users to necessary roles only.

- Use views in combination with reduced privileges for access control

- Consider label-based security where access to data is granted dependent on the content and type of the data (mandatory access control)

Hochschule Esslingen
University of Applied Sciences

Module "Information Systems" – Recap from DB1: Relational Data Model

Prof. Dr. Schoop
University of Applied Sciences Esslingen

Hochschule Esslingen
University of Applied Sciences

Relational Database Model

- Introduced in 1970 by E.F. Codd
- Originally as theory for database models
- Today: type of database model
- Standard in professional database systems, e.g.
  - Oracle
  - DB2
  - Informix
  - MS SQL Server
  - My SQL
- Alternative models:
  hierarchical, network, object-oriented

Structure of a Relational Database Model

- Data stored in relations
- Operations
  - selection of relations (= selecting rows)
  - reduction of relations (= selecting columns = projection)
  - combination of relations (union, intersection, join, …)

Converting a Table to a Relation

Student		
MatrNo	**Name**	**Subject**
1001	Marc	SWB
1002	Carmen	KTB

Student ´ =
{
(1001, Marc, SWB),
(1002, Carmen, KTB)
}

Column headings are lost! ☹

Converting a Table to a Relation

Student		
MatrNo	**Name**	**Subject**
1001	Marc	SWB
1002	Carmen	KTB

Solution: assigning values to column headings (= attribute name)

Student ´ =
{
(1001, Marc, SWB),
(1002, Carmen, KTB)
}

Definitions: Tuple and Relation

For each **attribute** A (= table column heading), there is a set dom(A) of possible values that can be assigned to the attribute A. The set dom(A) is called the **domain of A**.

Let X = $\{A_1, \ldots, A_k\}$ be a set of attributes.

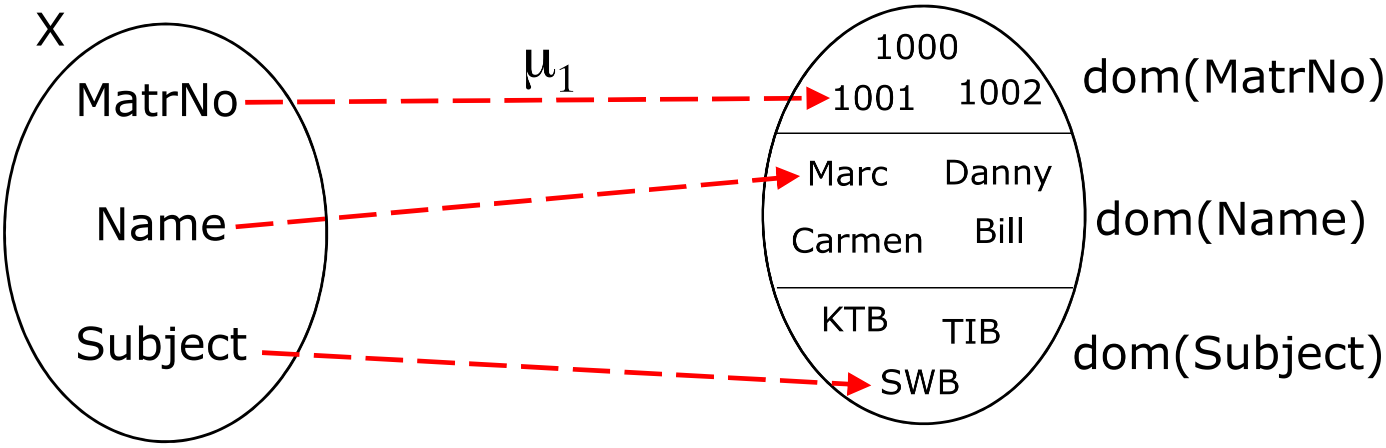
Then dom(X) $= \bigcup_{i=1}^{k} \mathrm{dom}(A_i) = \mathrm{dom}(A_1) \cup \ldots \cup \mathrm{dom}(A_k)$.

A **tuple** of X is defined as a function μ: X→dom(X) such that for each A∈X: $\mu(A) \in \mathrm{dom}(A)$.

A **relation** r over X is a set of tuples of X.

Converting a Table to a Relation

Student		
MatrNo	**Name**	**Subject**
1001	Marc	SWB
1002	Carmen	KTB



Hochschule Esslingen
University of Applied Sciences

Converting a Table to a Relation

Student		
MatrNo	**Name**	**Subject**
1001	Marc	SWB
1002	Carmen	KTB

X = { MatrNo, Name, Subject }

μ_1, μ_2 : X→dom(X)

μ_1(MatrNo)=1001, μ_1(Name)=Marc, μ_1(Subject)=SWB

μ_2(MatrNo)=1002, μ_2(Name)=Carmen, μ_2(Subject)=KTB

Student =

{ μ_1,

μ_2

}

Alternative Notations for Tuples

- X_1= {MatNr, Name, Subject}
  - μ_1=

MatrNo	Name	Subject
1001	Marc	SWB

  - μ_2=

MatrNo	Name	Subject
1002	Carmen	KTB

- Alternatively:
  - μ_1= [MatrNo/1001 | Name/Marc | Subject/SWB]
  - μ_2= [MatrNo/1002 | Name/Carmen | Subject/KTB]

Definition: Tup(X)

- Given a set X of attributes (with their domains) we use **Tup(X)** to denote the set of all possible tuples of X.
- Example:
  X = { A, B }
  dom(A) = { a, b }
  dom(B) = { 1, 2 }
  Tup(X) = { t_1, t_2, t_3, t_4 }
  = { [A/a | B/1], [A/a | B/2], [A/b | B/1], [A/b | B/2] }
- Then a relation r over X is simply a subset of Tup(X), i.e. $r \subseteq$ Tup(X).

Definition: Rel(X)

- To denote the set of all possible relations over X we use **Rel(X)** = { r | r ⊆ Tup(X) and r ≠ ø }.
- Example for Tup(X) of previous slide:
  Rel(X) =

$$
\{\;
\{t_1\}, \{t_2\}, \{t_3\}, \{t_4\},\;
\{t_1,t_2\}, \{t_1,t_3\}, \{t_1,t_4\}, \{t_2,t_3\}, \{t_2,t_4\}, \{t_3,t_4\}\;
\{t_1,t_2,t_3\}, \{t_1,t_2,t_4\}, \{t_1,t_3,t_4\}, \{t_2,t_3,t_4\},\;
\{t_1,t_2,t_3,t_4\}
\;\}
$$

Linking Relations

- How to refer from one tuple (row) in a relation (table) to some other tuple (row) in some other relation (table)?
- Example:

Student

MatrNo	Name	Subject
1001	Marc	SWB
1002	Carmen	KTB
1003	Benny	SWB

Subject

Subject	Name
KTB	Kommunikations-technik
SWB	Softwaretechnik
TI	Technische Informatik

- Solution: keys

Definition: Key

- Let $K \subseteq X$ be a set of attributes and r a relation over X.
- We denote the **reduction** from $\mu \in r$ onto a non-empty set $K \subseteq X$ by $\mu[K]$,
  i.e. $\mu[K]: K \rightarrow dom(K)$ such that
  for all $A \in K$, $\mu[K](A) = \mu(A)$.
- K is called a **key** of r if and only if

(i) uniqueness:
for all $\mu,\nu \in r$: $\mu[K]=\nu[K]$ implies $\mu=\nu$

(ii) minimality:
There is no non-empty $K' \subset K$ such that (i) holds

Example - Keys

What are the keys?

Example 1

A	B	C
1	A	B
2	B	B
3	B	C

Example 2

A	B	C
1	A	B
2	B	B
2	B	C

Example 3

A	B	C
1	A	B
1	B	B
1	B	C
2	B	C

Example 4

A	B	C
1	A	B
1	A	B
2	B	C

Solution:

Keys

- All keys as defined above are called **candidate keys**.
- One of the candidate keys is chosen as the **primary key** to identify a tuple uniquely.
- A candidate key which is not chosen as primary key is called **secondary key** or **alternate key**.

- A key K is called a **composite key** if it consists of more than one attribute, e.g. $K = \{ A_1, A_2 \}$.

Example Tables

Student

MatrNo	Surname	First name
1001	Allerton	Andy
1002	Meyers	Maisie
1003	Allerton	Maisie

Lecturer

ID	Name
1111	Nonnast
2222	Schoop
3333	Roessler

Course

ID	Name	Lecturer
101	Databases 1	2222
102	Databases 1	2222
303	Computer Science 3	3333
304	Computer Science 3	2222

Enrolment

MatrNo	Course
1001	101
1001	303
1002	303

Example Tables – Candidate Keys

Student		
MatrNo	Surname	First name
1001	Allerton	Andy
1002	Meyers	Maisie
1003	Allerton	Maisie

Lecturer	
ID	Name
1111	Nonnast
2222	Schoop
3333	Roessler

Course		
ID	Name	Lecturer
101	Databases 1	2222
102	Databases 1	2222
303	Computer Science 3	3333
304	Computer Science 3	2222

Enrolment	
MatrNo	Course
1001	101
1001	303
1002	303

Example Tables – Candidate Keys

Student		
MatrNo	Surname	First name
1001	Allerton	Andy
1002		
1003		

Lecturer	
ID	Name
1111	Nonnast

Candidate keys in the present state of the tables. However, considering future data inserts they may not stay keys.

Course		
ID	Name	Lecturer
101	Databases 1	2222
102	Databases 1	2222
303	Computer Science 3	3333
304	Computer Science 3	2222

Enrolment	
MatrNo	Course
1001	101
1001	303
1002	303

Intra-relational Data Dependencies

- Constraints that concern exactly one relation:
  - constraints regarding key
  - constraints regarding attributes
- Examples:
  - attribute must have properties of key, i.e. uniqueness and minimality
  - an attribute is dependent on other attributes
- Notation example:
  - Assuming X = {A, B, C, D, E}
  - A → B, C ; A → X ; A, B → C

Relational Schema

- Given a set X of attributes and a set of intra-relational data dependencies Σ_X the pair

$$\mathbf{R=(X, \Sigma_X)}$$

is a **relational schema**.

Example – Relational Schema

Student		
MatrNo	**Surname**	**First name**
1001	Allerton	Andy
1002	Meyers	Maisie
1003	Allerton	Maisie

$X_{Student}$ = { MatrNo, Surname, First name }

$\Sigma_{X\_Student}$ = { MatrNo → $X_{Student}$ }

$R_{Student}$ = ($X_{Student}$, $\Sigma_{X\_Student}$)

Example – Relational Schema

Enrolment	
MatrNo	**Course**
1001	101
1001	303
1002	303

$X_{Enrolment}$ = { MatrNo, Course }

$\Sigma_{X\_Enrolment}$ = { MatrNo, Course → $X_{Enrolment}$ }

$R_{Enrolment}$ = ($X_{Enrolment}$, $\Sigma_{X\_Enrolment}$)

Relationships between Relations

- An attribute (or a set of attributes) in a relation (table) r whose values are those of a candidate key in some other relation is called a **foreign key** of r.
- A foreign key defines a logical relationship between (two tuples (rows) of) two relations.
- A foreign key is a composite key if the candidate key it is associated with is a composite key.

- We distinguish between
  - one-to-one relationships (1:1)
  - one-to-many relationships (1:n)
  - many-to-many relationships (n:m)

  between relations (tables).

Example Tables – Foreign Keys

Student

MatrNo	Surname	First name
1001	Allerton	Andy
1002	Meyers	Maisie
1003	Allerton	Maisie

Lecturer

ID	Name
1111	Nonnast
2222	Schoop
3333	Roessler

Course

ID	Name	Lecturer
101	Databases 1	2222
102	Databases 1	2222
303	Computer Science 3	3333
304	Computer Science 3	2222

Enrolment

MatrNo	Course
1001	101
1001	303
1002	303

Example Tables – Foreign Keys

Student		
MatrNo	**Surname**	**First name**
1001	Allerton	Andy
1002	Meyers	Maisie
1003	Allerton	Maisie

Lecturer	
ID	**Name**
1111	Nonnast
2222	Schoop
3333	Roessler

Course		
ID	**Name**	**Lecturer**
101	Databases 1	2222
102	Databases 1	2222
303	Computer Science 3	3333
304	Computer Science 3	2222

Enrolment	
MatrNo	**Course**
1001	101
1001	303
1002	303

Hochschule Esslingen
University of Applied Sciences

Example Tables – Foreign Keys

Student		
MatrNo	**Surname**	**First name**
1001	Allerton	Andy
1002	Meyers	Maisie
1003	Allerton	Maisie

Lecturer	
ID	**Name**
1111	Nonnast
2222	Schoop
3333	Roessler

Course		
ID	**Name**	**Lecturer**
101	Databases 1	2222
102	Databases 1	2222
303	Computer Science 3	3333
304	Computer Science 3	2222

Enrolment	
MatrNo	**Course**
1001	101
1001	303
1002	303

One-to-one Relationships

- One-to-one relations usually do not occur in a relational database when the data model is optimized (i.e. without redundancies).
- Unique information is gathered together into one table. One-to-one relations are implemented as an (additional) column.
- Example: each student has a unique passport ID (ppID) and a unique immatriculation number (MatrNo), i.e.
  a passport ID is related to exactly **one** MatNo and
  a MatNo is related to exactly **one** passport ID

Passport	
ppID	**MatrNo**
a1234	1001
b4321	1002

Student			
MatrNo	**FName**	**LName**	**ppID**
1001	Allerton	Andy	a1234
1002	Meyers	Maisie	b4321

One-to-many Relationships

- Relations between tables via primary key as foreign key (one-to-many, 1:n)
- Most common relation type
- Name of primary key and name of foreign key can be different. Rows from master table are joined with rows form referenced table by looking for identical attribute values.
- Examples:
  - A course is taught by **one** lecturer and
    a lecturer teaches **many** courses.

Example Tables – 1:n-Relationship

Student		
MatrNo	**Surname**	**First name**
1001	Allerton	Andy
1002	Meyers	Maisie
1003	Allerton	Maisie

Lecturer	
ID	**Name**
1111	Nonnast
2222	Schoop
3333	Roessler

Course		
ID	**Name**	**Lecturer**
101	Databases 1	2222
102	Databases 1	2222
303	Computer Science 3	3333
304	Computer Science 3	2222

Enrolment	
MatrNo	**Course**
1001	101
1001	303
1002	303

Many-to-many Relationships

- Example: A student takes **many** courses and a course is taken by **many** students

Student		
MatrNo	**LName**	**FName**
1001	Allerton	Andy
1002	Meyers	Maisie

a student takes **many** courses

a course is taken by **many** students

Course		
ID	**Name**	**Lecturer**
101	Databases 1	1111
303	Computer Science 3	2222

- Many-to-many relations are implemented in the relational model as individual relations (tables) because of (space) optimization.
- Consequently, the relation representing the relationship has two foreign keys which serve as the composite primary key of the relation.

Example Tables – n:m-Relationship

Student		
MatrNo	**Surname**	**First name**
1001	Allerton	Andy
1002	Meyers	Maisie
1003	Allerton	Maisie

Lecturer	
ID	**Name**
1111	Nonnast
2222	Schoop
3333	Roessler

Course		
ID	**Name**	**Lecturer**
101	Databases 1	2222
102	Databases 1	2222
303	Computer Science 3	3333
304	Computer Science 3	2222

Enrolment	
MatrNo	**Course**
1001	101
1001	303
1002	303

Inter-relational Data Dependencies

- Inter-relational data dependencies concern several relations and express:
  - Foreign keys
  - Inclusion dependencies ("X is a Y")
  - Exclusion dependencies (disjointness constraint) ("disjointness: X is not a Y")
  - Covering dependencies (participation constraint) ("totality: X is either a Y or a Z")

Hochschule Esslingen
University of Applied Sciences

Inter-relational Data Dependencies – Example – Foreign Keys

Student		
MatrNo	**Surname**	**First name**
1001	Allerton	Andy
1002	Meyers	Maisie
1003	Allerton	Maisie

Lecturer	
ID	**Name**
1111	Nonnast
2222	Schoop
3333	Roessler

Course		
ID	**Name**	**Lecturer**
101	Databases 1	2222
102	Databases 1	2222
303	Computer Science 3	3333
304	Computer Science 3	2222

Enrolment	
MatrNo	**Course**
1001	101
1001	303
1002	303

Hochschule Esslingen
University of Applied Sciences

Inter-relational Data Dependencies – Example – Foreign Keys

- Notation for foreign keys:

  Course[Lecturer] ⊆ Lecturer[ID]
  Enrolment[MatrNo] ⊆ Student[MatrNo]
  Enrolment[Course] ⊆ Course[ID]

Person	
ID	**Age**
1001	19
1002	20
1111	34
2222	33
3333	32

Lecturer	
ID	**Name**
1111	Nonnast
2222	Schoop
3333	Roessler

Student		
MatrNo	**LName**	**FName**
1001	Allerton	Andy
1002	Meyers	Maisie

- Inclusion:
  Each lecturer is a person.
  Notation: Lecturer[ID] ⊆ Person[ID]
  Each student is a person.
  Notation: Student[MatrNo] ⊆ Person[ID]
- Exclusion:
  A lecturer is no student and vice versa.
  Notation: Lecturer[ID] ∩ Student[MatrNo] = ∅
- Covering:
  Every person is either a student or a lecturer.
  Notation:
  Lecturer[ID] ∪ Student[MatrNo] = Person[ID]

Relational Database Schema

- Let $\mathfrak{R}$ be a set of relational schemata and $\Sigma_{\mathfrak{R}}$ a set of inter-relational dependencies.

- A **relational database schema** is defined as a pair $\mathbf{D=(\mathfrak{R}, \Sigma_{\mathfrak{R}})}$.

Example – Relational Database Schema

Student

MatrNo	Surname	First name
1001	Allerton	Andy
1002	Meyers	Maisie
1003	Allerton	Maisie

Lecturer

ID	Name
1111	Nonnast
2222	Schoop
3333	Roessler

Course

ID	Name	Lecturer
101	Databases 1	2222
102	Databases 1	2222
303	Computer Science 3	3333
304	Computer Science 3	2222

Enrolment

MatrNo	Course
1001	101
1001	303
1002	303

Relational Database Schema

- $R_{Student} = (X_{Student} , \Sigma_{X\_Student})$ (see Slide 24)
- $R_{Enrolment} = (X_{Enrolment} , \Sigma_{X\_Enrolment})$ (see Slide 25)

- $X_{Lecturer} = \{ID, Name\}$
  $\Sigma_{X\_Lecturer} = \{ID \rightarrow X_{Lecturer}\}$
  $R_{Lecturer} = (X_{Lecturer} , \Sigma_{X\_Lecturer})$

- $X_{Course} = \{ID, Name, Lecturer\}$
  $\Sigma_{X\_Course} = \{ID \rightarrow X_{Course}\}$
  $R_{Course} = (X_{Course} , \Sigma_{X\_Course})$

Relational Database Schema

- $\mathfrak{R} = \{R_{Student}, R_{Lecturer}, R_{Course}, R_{Enrolment} \}$

- $\Sigma_{\mathfrak{R}} = \{$ Course[Lecturer] $\subseteq$ Lecturer[ID],
  Enrolment[MatrNo] $\subseteq$ Student[MatrNo],
  Enrolment[Course] $\subseteq$ Course[ID] $\}$

- $D=(\mathfrak{R}, \Sigma_{\mathfrak{R}})$

Summary

- Data models: structures, operations, constraints
- Relational data model consists of definitions of relations that are represented as tables
- The columns of tables are associated to attributes
- The rows of tables are identified by primary keys
- The tables are interrelated via foreign keys
- One-to-many relationships are most common
- Many-to-many relationships are realised as individual tables

Exercise: Conceptual Data Model for Car Sharing Community

October 19, 2010

1 Data Model Version 1

The following requirements have been extracted from interviews with the Car Sharing Community (CSC).

1. CSC owns several vehicles.
2. Each vehicle has a registration number (e.g. ES – T 123) and a number of seats (7 max).
3. Each vehicle is parked at one of the parking lots rented by CSC. A parking lot is identified by a street name. Each parking lot has a certain number of places. There cannot be more cars parked in a parking lot than there are places.
4. CSC has several members. Each member has a member identification number (mid), a last name (Familienname), possibly several first names (Vornamen) and an address, where she or he lives (street, number, postcode, city).
5. Members can book vehicles at a certain time for a certain period of time. However, a member can book only one vehicle at a time.

2 Data Model Version 2

The data model needs to be extended and/or modified to cover the following requirements.

1. CSC has employees who look after the vehicles (cleaning, fuelling, inspection etc.).
2. The employees are grouped into teams.
3. Each team looks after a defined set of vehicles.
4. Each team has a team-leader.

3 Data Model Version 3

The data model needs to be extended and/or modified to cover the following requirements.

1. There are two kinds of vehicles: cars (PKWs) and lorries (LKWs).
2. Lorries have storage space defined by length, width and height.
3. Only members with a suitable driving licence can book lorries.
4. Each member has a preferred parking lot which is close to her/his living address.

Exercise: Conceptual Data Models from Connolly, Chapter 12

October 19, 2010

1 Data model for company specializing in IT training

Create an ER model of the data requirements for a company that specializes in IT training.

The company has 30 instructors and can handle up to 100 trainees per training session. The company offers five advanced technology courses, each of which is taught by a teaching team of two or more instructors. Each instructor is assigned to a maximum of two teaching teams or may be assigned to do research. Each trainee undertakes one advanced technology course per training session.

2 Data model for rental DVD rental company

The following describes the data requirements of a DVD rental company. Develop an ER model.

The DVD rental company has several branches throughout Germany. The data held on each branch is the branch address made up from street, post code, city and telephone number. Each branch is given a branch number which is unique throughout the company. Each branch is allocated staff which includes a manager. The manager is responsible for the day-to-day running of a given branch. The data held on a member of staff is his or her name, position and salary. Each member of staff is given a staff number, which is unique throughout the company. Each branch has a stock of DVDs. The data held on a DVD is the catalog number, DVD number, title, category, daily rental, cost, status and the names of the main actors and the director. The catalog number uniquely identifies a DVD. However, in most cases there are several copies of each DVD at a branch and the individual copies are identified by using the DVD number. A DVD is given a category such as Action, Children, Drama, Horror or Sci-Fi. The status indicates whether a specific copy of a DVD is available for rent. Before borrowing a DVD from the company, a customer must first register as a member of a local branch. The data held on a member is the first and last name, address and the date that the member registered at a branch. Each member is given a member number, which is unique throughout all branches of the company. Once registered, a member is free to rent DVDs, up to a maximum of ten at any one time. The data held on each DVD rented is the rental number, the full name and number of the member, the DVD number, title and daily rental, and the dates the DVD is rented out and returned. The DVD number is unique throughout the company.

Exercise: Conceptual Data Models from Connolly, Chapter 12

October 19, 2010

1 Data model for company specializing in IT training

Create an ER model of the data requirements for a company that specializes in IT training.

The company has 30 instructors and can handle up to 100 trainees per training session. The company offers five advanced technology courses, each of which is taught by a teaching team of two or more instructors. Each instructor is assigned to a maximum of two teaching teams or may be assigned to do research. Each trainee undertakes one advanced technology course per training session.

2 Data model for rental DVD rental company

The following describes the data requirements of a DVD rental company. Develop an ER model.

The DVD rental company has several branches throughout Germany. The data held on each branch is the branch address made up from street, post code, city and telephone number. Each branch is given a branch number which is unique throughout the company. Each branch is allocated staff which includes a manager. The manager is responsible for the day-to-day running of a given branch. The data held on a member of staff is his or her name, position and salary. Each member of staff is given a staff number, which is unique throughout the company. Each branch has a stock of DVDs. The data held on a DVD is the catalog number, DVD number, title, category, daily rental, cost, status and the names of the main actors and the director. The catalog number uniquely identifies a DVD. However, in most cases there are several copies of each DVD at a branch and the individual copies are identified by using the DVD number. A DVD is given a category such as Action, Children, Drama, Horror or Sci-Fi. The status indicates whether a specific copy of a DVD is available for rent. Before borrowing a DVD from the company, a customer must first register as a member of a local branch. The data held on a member is the first and last name, address and the date that the member registered at a branch. Each member is given a member number, which is unique throughout all branches of the company. Once registered, a member is free to rent DVDs, up to a maximum of ten at any one time. The data held on each DVD rented is the rental number, the full name and number of the member, the DVD number, title and daily rental, and the dates the DVD is rented out and returned. The DVD number is unique throughout the company.

Module “Information Systems (InfoSys)" Information regarding Project

Prof. Dr. Schoop
University of Applied Sciences Esslingen
Winter Term 2016/17

Learning Objectives

- To apply and train project management techniques and social skills
- To get familiar with the Database System Development Lifecycle (DSDLC) by
  - analysing a given application area
  - collecting and analysing requirements from a “customer”
  - designing a conceptual, logical and physical data model
  - implementing a data model, constraints and business rules
  - designing and implementing a user interface
  - testing the application
- To get familiar with CASE-tools (Innovator, Toad)
- To experience advantages and disadvantages of CASE-tools
- To get experience with rapid prototyping by using MS Access

Project Goal

- The goal of the project is

„A database application for the planning and reporting of teaching at the IT department of Esslingen University"

- Motivation:
  Each semester somebody has to plan and report about which lecturer is teaching which lecture, lab, … taking a large number of constraints into account.

Organisational Issues

- Project work in teams of 3 – 4 students
- Regular project meetings of the individual groups (to be arranged by group)
- 4 milestone meetings of "supplier" (i.e. you) with "customer" (i.e. lecturer) to enable structured information exchange between supplier and customer
- Estimated effort for supplier: approx. 60 hours of work (per person), i.e. product should be produced within 180-240 working hours.

Project Management

- A team of students (supplier) has to define
  - A team leader (project manager)
  - A sub-project leader for each milestone
- The work may be devided between the team members.
- Each team member has to keep a project manual documenting meetings, progress, next steps, …
- A time schedule has to be created and updated regularly according to progress and changes in estimation of efforts.

DSDLC Project Steps (1/6)

1. Database planning
   a) Tasks:
      i. organise your team
      ii. define project leader and task managers
      iii. estimate effort
   b) Deliverables:
      i. documentation of group structure
      ii. mission statement & mission objectives
      iii. time schedule
2. System definition
   a) Tasks:
      i. describe users of system
      ii. define user views
   b) Deliverables:
      i. user views

DSDLC Project Steps (2/6)

3. Requirements collection and analysis
   a) Tasks:
      i. use fact-finding techniques to identify all necessary requirements
      ii. identify system areas where there are no requirements
      iii. document design decisions regarding those system areas
   b) Deliverables:
      i. list of requirements (structured, sorted, numbered)

DSDLC Project Steps (3/6)

4. Database design
   a) Tasks:
      i. identify entities, attributes and relationships
      ii. design conceptual data model on paper (Chen notation)
      iii. associate modelling elements to user views
      iv. implement conceptual data model in Innovator
      v. implement logical data model in Innovator
      vi. bring all relations in BCNF (where useful)
   b) Deliverables:
      i. conceptual data model on paper (Chen notation)
      ii. list of all relevant entities and relationships for each user view
      iii. conceptual data model in CASE-tool
      iv. logical data model
      v. proof that all relations are in BCNF or an explanation why not

DSDLC Project Steps (4/6)

5. DBMS selection
   a) Tasks: none (IBM DB2 has been preselected)
   b) Deliverables: none
6. Application design
   a) Tasks:
      i. database application: define any constraints and business functions you want to implement
      ii. GUI: define functional sequences of windows
      iii. GUI: make sketches of windows for data input and queries
   b) Deliverables: none
7. Prototyping (we will do the implemention directly)
   a) Tasks: none
   b) Deliverables: none

DSDLC Project Steps (5/6)

8. Implementation
   a) Tasks:
      i. export DDL from Innovator
      ii. define constraints, triggers, procedures in SQL
      iii. implement GUI with MS Access
      iv. pre-testing with simple data
   b) Deliverables:
      i. all SQL code
      ii. MS Access project
9. Data conversion and loading
   a) Tasks:
      i. convert the provided testing data into SQL statements which insert the data into your tables
   b) Deliverables:
      i. SQL-code for data loading

DSDLC Project Steps (6/6)

10. Testing
    a) Tasks:
        i. carry out all defined use cases with various parameters (test cases)
        ii. let the system be tested by the customer (product acceptance test)
    b) Deliverables:
        i. list of positive and failed test cases
        ii. product with required functionality and a minimality of bugs
11. Operational maintenance (out of scope)
    a) Tasks: none
    b) Deliverables: none

Milestone Meetings (1/4)

- There are 4 milestone meetings at which the supplier has to present their deliverables to the customer. There is an additional meeting if deliverables were not acceptable previously.

- M1 (after "Requirement collection and analysis"):

1. documentation of group structure
2. mission statement & mission objectives
3. time schedule
4. user views (use cases)
5. list of requirements (structured, sorted, numbered)

Milestone Meetings (2/4)

- M2 (during “Database design”):

1. conceptual data model on paper (Chen notation)
2. list of all relevant entities and relationships for each user view

- M3 (after “Database design”):

1. conceptual data model in CASE-tool
2. logical data model
3. proof that all relations are in BCNF or explanation why not

Milestone Meetings (3/4)

- M4 (after “Testing”):

1. all SQL code
2. MS Access project
3. SQL-code for data loading
4. list of successful and failed test cases

- MF (if product acceptance failed in M4)

1. See M4

Milestone Meetings (4/4)

- Milestone meetings will take place in the database lab (room F1.410) at the large computer screen.
- A separate time schedule will be provided for the meetings.
- All documents and presentations must be accessible from the computer with the large screen. Alternatively, you can connect your computer to the large screen.

- **Rehearse your presentation and technical set-up with the large screen. The presentation time is limited.**

Main Requirements

1. Strict application of the Database System Development Lifecycle to ensure efficiency of development and to avoid data redundancy.
2. Strict separation of database, database application and GUI
3. All data, constraints and business rules are stored in the database (application).
4. The GUI provides means for data access and input only.
5. Implementation of the following business processes.

Main Requirements – Business Process „Lecture Planning“

6. The examination regulations define which modules have to be offered. Note that not all lectures are defined in the examination regulations (e.g. Wahlfach X, elective X).
7. Modules can consist out of various elements (lectures, labs, projects, …) having a certain number of weekly semester hours (SWS).
8. Each module element is taught by professors and/or external lecturers (Lehrbeauftragte).
9. A semester group may be split into various teaching groups (e.g. IT3A, IT3B) and sub-groups (e.g. lab groups 1 and 2 of IT3A).
10. Two groups may share the same lecture (e.g. SWT, SWM share Databases 2) having different names for the groups.

Main Requirements – Business Process „Lecture Planning“

11. A lecturer cannot be professor and external lecturer at the same time. However, an external lecturer can become a professor and a professor can become an external lecturer after reaching his pension age.
12. An external lecturer is associated to a specific department (Fakultät).
13. A module element has a certain number of SWS assigned in the examination regulations. The number of hours in the time schedule and the number of SWS as work-load of the lecturer might be different to those, e.g. DB2 Project: Student SWS = 2, Lecturer SWS = 1, hours in time schedule = 0.

Main Requirements – Business Process „Work Load Documentation"

14. Professors have to work a certain amount of SWS each semester. The work can be teaching or assigned tasks (dean, running a lab room, research, …).
15. Currently each full time professor has to work 18 SWS on average (required work load = Deputat). The assigned number of SWS per semester might be higher or lower but not below 9 SWS.
16. Professors might work part-time, be ill for a longer time or be on sabbatical.
17. External lecturers do not have a required work load and cannot take other assigned tasks.
18. The application has to document the work load of each professor over the years.

Main Requirements – Business Process „Report Generation"

The application must be able to generate a number of reports:

19. List of each task of each professor for a selected semester also giving the total work load balance (Stundenkontostand – accumulated real work load versus required work load (Deputat))
20. List of module elements offered in a selected academic half year for a selected degree (Studiengang)
21. List of external lecturers, their SWS for a selected academic half year and their addresses

Main Requirements – Business Process „Report Generation"

22. List of services provided, i.e. list of module elements taught by IT professors for a different department (name of module element, name of the lecturer, SWS, department which the service is provided for)
23. List of services used, i.e. list of module elements taught to IT students by a lecturer of another department (name of module element, name of the lecturer, SWS, department which the service is provided by)

Main Requirements

24. The GUI, to be designed with MS Access, serves the only purpose to insert, update, access and delete data.
25. The GUI must not implement any data constraints.
26. The data model, constraints and business logic in the database must be designed to forbid any inconsistent data.
27. The data model has to be implemented with the modelling tool Innovator.

Main Requirments

28. Each modelling element (entity, role, constraint) has to have a description defining the meaning of the entity precisely to avoid misunderstandings.

    Example: Possible definitions for entity „Semester"

    1. „Identification of a specific academic half year, e.g. winter semester 2012/13."
    2. „Identification of a generic academic half year and all its modules according to the examination regulations, e.g. 6th semester of Softwaretechnik (SWT6)."
    3. „Group of students having to take the same lectures as stipulated in the examination regulations (e.g. student group SWB6)."

29. The language of the application (data model, GUI) is German.

Optional Main Requirements

- The students who are fiddling their thumbs out of boredom are invited to extend their application with the following requirements:

30. Business process „schedule planning": The application can be used to plan the time and room to be reserved for a module element.
31. Business process „schedule reporting": For a given week, a schedule for a room or a group of students is presented.

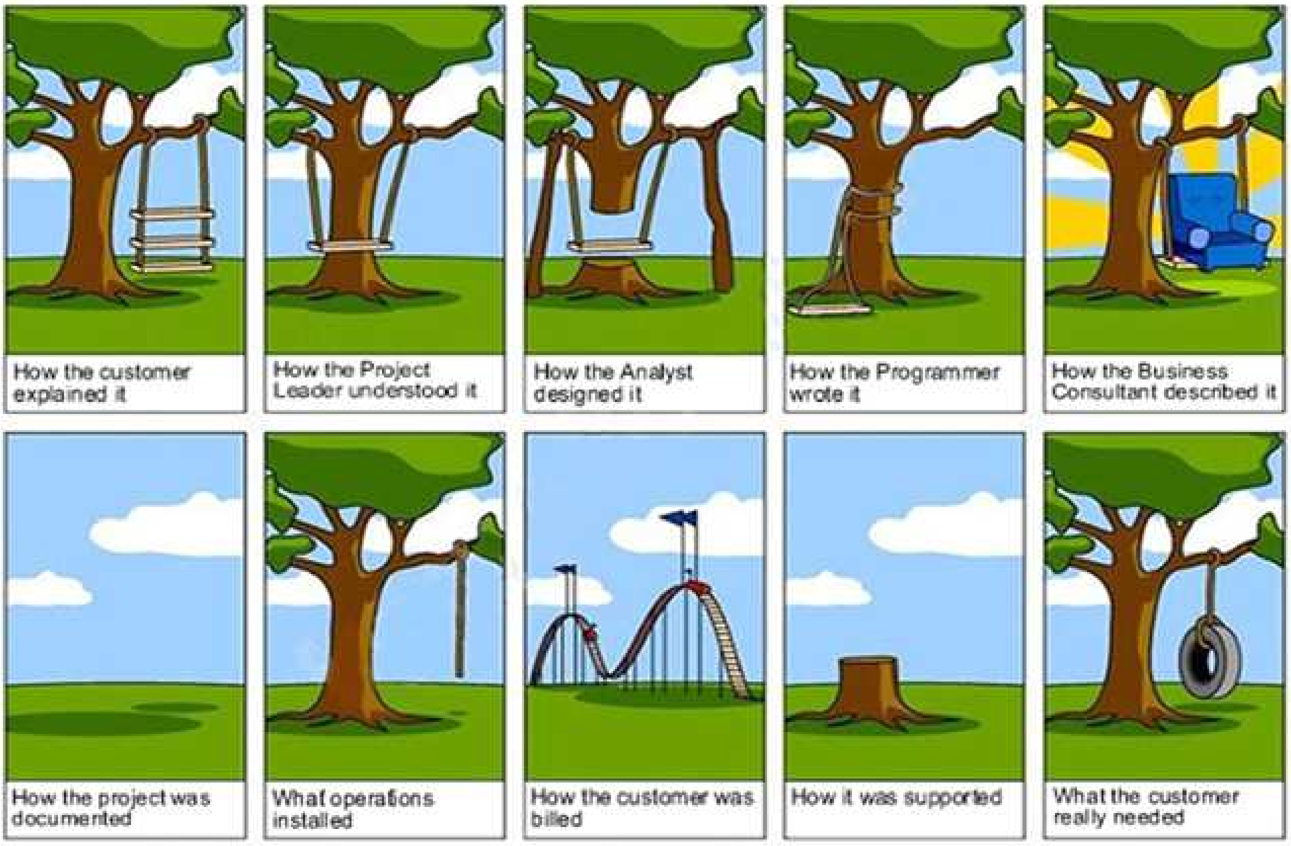
Available Information

The following documents providing useful information are available in Moodle.

- Examination regulations (KTB, SWB, TIB)
- List of all module elements for SS11 (POG)
- Time schedules
- Example reports
- Test data
- HOWTO Innovator
- HOWTO MS Access

For other detailed information you have to contact the customer (i.e. the lecturer).

What we do not want to have ...



STRONG ADVISE !!!

- The requirements in these slides are not sufficient to build a useful application.
- Therefore, use the fact-finding techniques to find all requirements so that the customer (i.e. the grading lecturer) gets what he needs and not what you think he described.

Grading the Project Results

- The customer (i.e. the lecturer) will carry out a product acceptance test at the Milestone Meeting 4.
- The first test of the acceptance test is a complete deletion and reload of the database application.
- If the acceptance test should fail there is a second and final chance at the Finale Milestone Meeting.
- After a positive acceptance test the supplier (i.e. you) has to provide a ZIP-file with all deliverables. In addition to the acceptance test in the presence of the supplier, the customer will carry out more tests in private and review the deliverables.

Grading the Project Results

- The acceptance test will evaluate whether the application is functional and functions correctly and efficiently.
- The review of the deliverables evaluates whether the documentation is complete, of sufficient quality and consistent with the implementation.
- You have passed the project if

  the product is ready for productive operation

POG für Studiengangleiter

Studiengang: TIB Stand WS2008/09 20.09.2008 Technische Informatik BACHELOR

pvers	pnr	pfsem	pdtxt	part	pdauer	ppflich	pminsem	phoesem	angebote	pform	Credits	psws	nachname	partngb	Gruppe
1	1011	1	Mathematik 1 Mathematics 1	L	150	P	1	4	J	KL	10	10	Ulmet	C	A
1	1011	1	Mathematik 1 Mathematics 1	L	150	P	1	4	J	KL	10	10	Plappert	C	B
1	1011	1	Mathematik 1 Mathematics 1	L	150	P	1	4	J	KL	10	10	Sigg	C	C
1	1021	1	Physik 1 Physics 1	L	90	P	1	4	J	KL	5	5	Coenning	C	A
1	1021	1	Physik 1 Physics 1	L	90	P	1	4	J	KL	5	5	Strobel	C	B
1	1021	1	Physik 1 Physics 1	L	90	P	1	4	J	KL	5	5	Coening	C	C
1	1031	1	Elektrotechnik 1 Electrical Engineering 1	L	90	P	1	4	J	KL	4	4	Gündner	C	A
1	1031	1	Elektrotechnik 1 Electrical Engineering 1	L	90	P	1	4	J	KL	4	4	Malz	C	B
1	1031	1	Elektrotechnik 1 Electrical Engineering 1	L	90	P	1	4	J	KL	4	4	Malz	C	C
1	1041	1	Labor Elektrotechnik 1 Lab Electrical Engineering 1	S		P	1	4	J	BE	1	1	LB Schulz	U	A
1	1041	1	Labor Elektrotechnik 1 Lab Electrical Engineering 1	S		P	1	4	J	BE	1	1	LB Haußer	U	B
1	1041	1	Labor Elektrotechnik 1 Lab Electrical Engineering 1	S		P	1	4	J	BE	1	1	LB Kähler	U	C
1	1051	1	Informatik 1 Computer Science 1	L	90	P	1	4	J	KL	3	3	Kappen	C	A
1	1051	1	Informatik 1 Computer Science 1	L	90	P	1	4	J	KL	3	3	Warendorf	C	B
1	1051	1	Informatik 1 Computer Science 1	L	90	P	1	4	J	KL	3	3	Weber	C	C
1	1061	1	Labor Informatik 1 Lab Computer Science 1	S		P	1	4	J	BE	2	2	LB Müller, Martin	U	A
1	1061	1	Labor Informatik 1 Lab Computer Science 1	S		P	1	4	J	BE	2	2	Warendorf	U	B
1	1061	1	Labor Informatik 1 Lab Computer Science 1	S		P	1	4	J	BE	2	2	Weber	U	C

POG für Studiengangleiter

Studiengang: TIB Stand WS2008/09 20.09.2008 Technische Informatik BACHELOR

pvers	pnr	pfsem	pdtxt	part	pdauer	ppflich	pminsem	phoesem	angebote	pform	Credits	psws	nachname	partngb	Gruppe
1	1071	1	Technisches Englisch Technical English	L	60	P	1	4	J	KL	2	2	Warendorf, Eve	C	A
1	1071	1	Technisches Englisch Technical English	L	60	P	1	4	J	KL	2	2	Warendorf, Eve	C	B
1	1071	1	Technisches Englisch Technical English	L	60	P	1	4	J	KL	2	2	Warendorf, Eve	C	C
1	1081	1	Persönlichkeitsentwicklung Personality Development	S		P	1	4	J	TE	3	3	Väterlein	C	A
1	1081	1	Persönlichkeitsentwicklung Personality Development	S		P	1	4	J	TE	3	3	Väterlein	C	B
1	1081	1	Persönlichkeitsentwicklung Personality Development	S		P	1	4	J	TE	3	3	Väterlein	C	C

POG für Studiengangleiter

Studiengang: TIB Stand WS2008/09 20.09.2008 Technische Informatik BACHELOR

pvers	pnr	pfsem	pdtxt	part	pdauer	ppflich	pminsem	phoesem	angebote	pform	Credits	psws	nachname	partngb	Gruppe
1	2011	2	Mathematik 2 Mathematics 2	L	90	P	1	4	J	KL	4	4	Ulmet	C	A
1	2011	2	Mathematik 2 Mathematics 2	L	90	P	1	4	J	KL	4	4	Koch	C	B
1	2011	2	Mathematik 2 Mathematics 2	L	90	P	1	4	J	KL	4	4		C	
1	2021	2	Labor Mathematik 2 Lab Mathematics 2	S		P	1	4	J	TE	1	1	Ulmet	U	A
1	2021	2	Labor Mathematik 2 Lab Mathematics 2	S		P	1	4	J	TE	1	1	Koch	U	B
1	2021	2	Labor Mathematik 2 Lab Mathematics 2	S		P	1	4	J	TE	1	1		U	
1	2031	2	Physik 2 Physics 2	L	90	P	1	4	J	KL	4	4	Martin	C	A
1	2031	2	Physik 2 Physics 2	L	90	P	1	4	J	KL	4	4	Coenning	C	B
1	2031	2	Physik 2 Physics 2	L	90	P	1	4	J	KL	4	4		C	
1	2041	2	Labor Physik 2 Lab Physics 2	S		P	1	4	J	BE	1	1	Strobel	U	A
1	2041	2	Labor Physik 2 Lab Physics 2	S		P	1	4	J	BE	1	1	Coenning	U	B
1	2041	2	Labor Physik 2 Lab Physics 2	S		P	1	4	J	BE	1	1		U	
1	2051	2	Elektrotechnik 2 Electrical Engineering 2	L	90	P	1	4	J	KL	4	4	Höfer	C	A
1	2051	2	Elektrotechnik 2 Electrical Engineering 2	L	90	P	1	4	J	KL	4	4	Höfer	C	B
1	2051	2	Elektrotechnik 2 Electrical Engineering 2	L	90	P	1	4	J	KL	4	4		C	
1	2061	2	Labor Elektrotechnik 2 Lab Electrical Engineering 2	S		P	1	4	J	BE	1	1	LB Hehl	U	A
1	2061	2	Labor Elektrotechnik 2 Lab Electrical Engineering 2	S		P	1	4	J	BE	1	1	LB Rauschnabel	U	B
1	2061	2	Labor Elektrotechnik 2 Lab Electrical Engineering 2	S		P	1	4	J	BE	1	1		U	

POG für Studiengangleiter

Studiengang: TIB Stand WS2008/09 20.09.2008 Technische Informatik BACHELOR

pvers	pnr	pfsem	pdtxt	part	pdauer	ppflich	pminsem	phoesem	angebote	pform	Credits	psws	nachname	partngb	Gruppe
1	2071	2	Elektronik Electronics	L	90	P	1	4	J	KL	3	3	Gündner	C	A
1	2071	2	Elektronik Electronics	L	90	P	1	4	J	KL	3	3	Buck	C	B
1	2071	2	Elektronik Electronics	L	90	P	1	4	J	KL	3	3		C	
1	2081	2	Projekt Elektronik Project Electronics	S		P	1	4	J	PA	2	2	Gündner	U	A
1	2081	2	Projekt Elektronik Project Electronics	S		P	1	4	J	PA	2	2	Buck	U	B
1	2081	2	Projekt Elektronik Project Electronics	S		P	1	4	J	PA	2	2		U	
1	2091	2	Informatik 2 Computer Science 2	L	90	P	1	4	J	KL	3	3	Beck	C	A
1	2091	2	Informatik 2 Computer Science 2	L	90	P	1	4	J	KL	3	3	Warendorf	C	B
1	2091	2	Informatik 2 Computer Science 2	L	90	P	1	4	J	KL	3	3		C	
1	2101	2	Projekt Informatik 2 Project Computer Science 2	S		P	1	4	J	PA	2	2	Beck	U	A
1	2101	2	Projekt Informatik 2 Project Computer Science 2	S		P	1	4	J	PA	2	2	Warendorf	U	B
1	2101	2	Projekt Informatik 2 Project Computer Science 2	S		P	1	4	J	PA	2	2		U	
1	2111	2	Computerarchitektur 1 Computer Architecture 1	L	90	P	1	4	J	KL	4	4	Lindermeir	C	A
1	2111	2	Computerarchitektur 1 Computer Architecture 1	L	90	P	1	4	J	KL	4	4	Keller	C	B
1	2111	2	Computerarchitektur 1 Computer Architecture 1	L	90	P	1	4	J	KL	4	4		C	
1	2121	2	Labor Computerarchitektur 1 Lab Computer Architecture 1	S		P	1	4	J	BE	1	1	Lindermeir	U	A
1	2121	2	Labor Computerarchitektur 1 Lab Computer Architecture 1	S		P	1	4	J	BE	1	1	Keller	U	B
1	2121	2	Labor Computerarchitektur 1 Lab Computer Architecture 1	S		P	1	4	J	BE	1	1		U	

POG für Studiengangleiter

Studiengang: TIB Stand WS2008/09 20.09.2008 Technische Informatik BACHELOR

pvers	pnr	pfsem	pdtxt	part	pdauer	ppflich	pminsem	phoesem	angebote	pform	Credits	psws	nachname	partngb	Gruppe
1	3001	3	Mathematik 3 Mathematics 3	L	90	P	3	10	J	KL	5	5	Sigg	C	A
1	3001	3	Mathematik 3 Mathematics 3	L	90	P	3	10	J	KL	5	5	Glatz	C	B
1	3011	3	Informatik 3 Computer Science 3	L	90	P	3	10	J	KL	3	3	Schoop	C	A
1	3011	3	Informatik 3 Computer Science 3	L	90	P	3	10	J	KL	3	3	Rößler	C	B
1	3021	3	Labor Informatik 3 Lab Computer Science 3	S		P	3	10	J	BE	2	2	Schoop	U	A
1	3021	3	Labor Informatik 3 Lab Computer Science 3	S		P	3	10	J	BE	2	2	Rößler	U	B
1	3031	3	Computerarchitektur 2 Computer Architectur 2	L	90	P	3	10	J	KL	4	4	Lindermeir	C	A
1	3031	3	Computerarchitektur 2 Computer Architectur 2	L	90	P	3	10	J	KL	4	4	Keller	C	B
1	3041	3	Labor Computerarchitektur 2 Labor Computer Architecture 2	S		P	3	10	J	BE	1	1	Lindermeir	U	A
1	3041	3	Labor Computerarchitektur 2 Labor Computer Architecture 2	S		P	3	10	J	BE	1	1	Keller	U	B
1	3051	3	Signale und Systeme Signals and Systems	L	90	P	3	10	J	KL	4	4	Doster	C	A
1	3051	3	Signale und Systeme Signals and Systems	L	90	P	3	10	J	KL	4	4	Doster	C	B
1	3061	3	Labor Signale und Systeme Lab Signals and Systems	S		P	3	10	J	BE	1	1	Doster	U	A
1	3061	3	Labor Signale und Systeme Lab Signals and Systems	S		P	3	10	J	BE	1	1	Doster	U	B
1	3071	3	Betriebssyteme Operating Systems	L	90	P	3	10	J	KL	4	4	Väterlein	C	A
1	3071	3	Betriebssyteme Operating Systems	L	90	P	3	10	J	KL	4	4	Weber	C	B
1	3081	3	Labor Betriebssyteme Lab Operating Systems	S		P	3	10	J	BE	1	1	Väterlein	U	A
1	3081	3	Labor Betriebssyteme Lab Operating Systems	S		P	3	10	J	BE	1	1	Weber	U	B

POG für Studiengangleiter

Studiengang: TIB Stand WS2008/09 20.09.2008 Technische Informatik BACHELOR

pvers	pnr	pfsem	pdtxt	part	pdauer	ppflich	pminsem	phoesem	angebote	pform	Credits	psws	nachname	partngb	Gruppe
1	3091	3	Datenbanken 1 Database Systems 1	L	90	P	3	10	J	KL	4	4	Nonnast	C	A
1	3091	3	Datenbanken 1 Database Systems 1	L	90	P	3	10	J	KL	4	4	Schoop	C	B
1	3101	3	Labor Datenbanken 1 Lab Database Systems 1	S		P	3	10	J	BE	1	1	Nonnast	U	A
1	3101	3	Labor Datenbanken 1 Lab Database Systems 1	S		P	3	10	J	BE	1	1	Schoop	U	B

POG für Studiengangleiter

Studiengang: TIB Stand WS2008/09 20.09.2008 Technische Informatik BACHELOR

pvers	pnr	pfsem	pdtxt	part	pdauer	ppflich	pminsem	phoesem	angebote	pform	Credits	psws	nachname	partngb	Gruppe
1	4001	4	Betriebswirtschaft Business Economics	L	90	P	3	10	J	KL	4	4	LB Feil	U	A
1	4001	4	Betriebswirtschaft Business Economics	L	90	P	3	10	J	KL	4	4	LB Feil	U	B
1	4011	4	Projektmanagement Project Management	S		P	3	10	J	BE	1	1	LB Stumpp	C	A
1	4011	4	Projektmanagement Project Management	S		P	3	10	J	BE	1	1	LB Stumpp	C	B
1	4021	4	Computerarchitektur 3 Computer Architekture 3	L	90	P	3	10	J	KL	4	4	Friedrich	U	A
1	4021	4	Computerarchitektur 3 Computer Architekture 3	L	90	P	3	10	J	KL	4	4	Friedrich	U	B
1	4031	4	Labor Computerarchitektur 3 Lab Computer Architecture	S		P	3	10	J	BE	1	1	Friedrich	C	A
1	4031	4	Labor Computerarchitektur 3 Lab Computer Architecture	S		P	3	10	J	BE	1	1	Friedrich	C	B
1	4041	4	Systemtechnik 1 System Design 1	L	90	P	3	10	J	KL	4	4	Zimmermann	U	A
1	4041	4	Systemtechnik 1 System Design 1	L	90	P	3	10	J	KL	4	4	Zimmermann	U	B
1	4051	4	Labor Systemtechnik 1 Lab System Design 1	S		P	3	10	J	BE	1	1	Zimmermann	C	A
1	4051	4	Labor Systemtechnik 1 Lab System Design 1	S		P	3	10	J	BE	1	1	Zimmermann	C	B
1	4061	4	Echtzeitsysteme Real Time Systems	L	90	P	3	10	J	KL	4	4	Friedrich	U	A
1	4061	4	Echtzeitsysteme Real Time Systems	L	90	P	3	10	J	KL	4	4	Friedrich	U	B
1	4071	4	Labor Echtzeitsysteme Lab Real Time Systems	S		P	3	10	J	BE	1	1	Friedrich	C	A
1	4071	4	Labor Echtzeitsysteme Lab Real Time Systems	S		P	3	10	J	BE	1	1	Friedrich	C	B
1	4081	4	Rechnernetze 1 Computer Networks 1	L	90	P	3	10	J	KL	4	4	Wiese	U	A
1	4081	4	Rechnernetze 1 Computer Networks 1	L	90	P	3	10	J	KL	4	4	Wiese	U	B

POG für Studiengangleiter

Studiengang: TIB Stand WS2008/09 20.09.2008 Technische Informatik BACHELOR

pvers	pnr	pfsem	pdtxt	part	pdauer	ppflich	pminsem	phoesem	angebote	pform	Credits	psws	nachname	partngb	Gruppe
1	4091	4	Labor Rechnernetze 1 Lab Computer Networks 1	S		P	3	10	J	BE	1	1	Wiese	C	A
1	4091	4	Labor Rechnernetze 1 Lab Computer Networks 1	S		P	3	10	J	BE	1	1	Wiese	C	B
1	4101	4	Objektorientierte Systeme 1 Object Oriented Systems 1	L	90	P	3	10	J	KL	3	3	Dausmann	U	A
1	4101	4	Objektorientierte Systeme 1 Object Oriented Systems 1	L	90	P	3	10	J	KL	3	3	Dausmann	U	B
1	4111	4	Labor Objektorientierte Systeme 1 Lab Object Oriented Systems 1	S		P	3	10	J	BE	2	2	Dausmann	C	A
1	4111	4	Labor Objektorientierte Systeme 1 Lab Object Oriented Systems 1	S		P	3	10	J	BE	2	2	Dausmann	C	B

POG für Studiengangleiter

Studiengang: TIB Stand WS2008/09 20.09.2008 Technische Informatik BACHELOR

pvers	pnr	pfsem	pdtxt	part	pdauer	ppflich	pminsem	phoesem	angebote	pform	Credits	psws	nachname	partngb	Gruppe
1	5011	5	Ingenieurmethodiken 2 Methodologies for Engineers 2	L		P	5	10	J	TE	4	3	Gündner	C	
1	5021	5	Betriebliche Praxis Company Internship	S		P	5	10	J	BE	26	0	Doster	U	

POG für Studiengangleiter

Studiengang: TIB Stand WS2008/09 20.09.2008 Technische Informatik BACHELOR

pvers	pnr	pfsem	pdtxt	part	pdauer	ppflich	pminsem	phoesem	angebote	pform	Credits	psws	nachname	partngb	Gruppe
1	6011	6	Studienarbeit Application Project	L		P	6	10	N	RE	5	5	it professoren	C	
1	6021	6	Labor Digitale Signalverarbeitung Lab Digital Signal Processing	S		P	3	10	N	BE	1	1	Höfer	U	
1	6031	6	Digitale Signalverarbeitung Digital Signal Processing	L	90	P	3	10	N	KL	4	4	Höfer	C	
1	6041	6	Labor Systemtechnik 2 Lab System Design 2	S		P	3	10	N	BE	1	1	Kull	U	
1	6051	6	Systemtechnik 2 System Design	L	90	P	3	10	N	KL	4	4	Kull	C	
1	6061	6	Labor Maschinelles Sehen Lab Machine Vision	S		P	3	10	N	BE	1	1	Malz	U	
1	6071	6	Maschinelles Sehen Machine Vision	L	90	P	3	10	N	KL	4	4	Malz	C	
1	6081	6	Labor Embedded Systems Software Lab Embedded Systems Software	S		P	3	10	N	BE	1	1	Kappen	U	
1	6091	6	Embedded Systems Software Embedded Systems Software	L	90	P	3	10	N	KL	4	4	Kappen	C	
1	6101	6	Labor Bussysteme Lab Bus Systems	S		P	3	10	N	BE	1	1	Wiese	U	
1	6111	6	Bussysteme Bus Systems	L	90	P	3	10	N	KL	4	4	Wiese	C	

POG für Studiengangleiter

Studiengang: TIB Stand WS2008/09 20.09.2008 Technische Informatik BACHELOR

pvers	pnr	pfsem	pdtxt	part	pdauer	ppflich	pminsem	phoesem	angebote	pform	Credits	psws	nachname	partngb	Gruppe
1	7000	7	Bachelorarbeit und Kolloquium Bachelor Thesis and Defense	P		P	6	10	J	BA	15	15	it professoren	C	
1	7011	7	Wissenschaftliche Vertiefung Scientific Research	L		P	6	10	J	ML	9	9	it professoren	C	

POG für Studiengangleiter

Studiengang: TIB Stand WS2008/09 20.09.2008 Technische Informatik BACHELOR

pvers	pnr	pfsem	pdtxt	part	pdauer	ppflich	pminsem	phoesem	angebote	pform	Credits	psws	nachname	partngb	Gruppe
			WAHLFÄCHER TIB												
1	8601		Kfz-Steuergeräte-Design Automotive-ECU-Design	L		W	4	99	N	ML	2	2		C	
1	8811	6	Algorithmen Algorithms	L		W	4	99	J	ML	2	2	Koch	C	
1	8821	6	Fundamentals of Optoelectronics Fundamentals of Optoelectronics	L		W	4	99	N	ML	2	2		C	
1	8831	6	IT-Sicherheit IT-Security	L		W	4	99	J	ML	2	2	Schoop	C	
1	8841	6	Fernsehtechnik Digital Video Broadcasting	L		W	4	99	N	ML	2	2		C	
1	8851	6	Technischer Vertrieb Sales and Marketing	L		W	4	99	N	ML	2	2		C	
1	8861	6	Programmieren in C# Introduction to C#	L		W	4	99	N	ML	2	2		C	
1	8871	6	Introduction to eCommerce Introduction to eCommerce	L		W	4	99	N	ML	2	2		C	
1	8881	6	Systemarchitekturen mit .NET Software Systems Architectures with .NET	L		W	4	99	J	ML	2	2	LB Erath	C	
1	8891		Kfz-Steuergeräte-Design 2 Automotive-ECU-Design 2	L		W	4	99	N	ML	2	2		C	
			neue WAHLFÄCHER TIB												
1	neu	6	Vertrags- und Internet-Recht Contract and Media Law	L		W	4	99	J	ML	2	2	LB Schließ	C	
1	neu	6	Methoden der künstlichen Intelligenz Artificial Intelligence	L		W	4	99	J	ML	2	2	Weber	C	
1	neu	6	Einführung in CAD Introduction to CAD	L		W	4	99	J	ML	2	2	Koch	C	
1	neu	6	Kfz-Systeme Automotive Systems Engineering	L		W	4	99	J	ML	2	2	LB Marchthaler	C	
1	neu	6	Qualitätstechniken Quality Engineering	L		W	4	99	J	ML	2	2	LB Grübel	C	
1	neu	6	Optische Nachrichtenübertragung Optical Communications Engineering	L		W	4	99	J	ML	2	2	LB Khakzar	C	
1	neu	6	Optische Nachrichtenübertragung 2 Optical Communications Engineering 2	L		W	4	99	J	ML	2	2	LB Khakzar	C	

POG für Studiengangleiter

Studiengang: TIB Stand WS2008/09 20.09.2008 Technische Informatik BACHELOR

pvers	pnr	pfsem	pdtxt	part	pdauer	ppflich	pminsem	phoesem	angebote	pform	Credits	psws	nachname	partngb	Gruppe
			neue Zusatzfächer TIB												
1	neu	6	Blockseminar XML Summer School Course XML	S		Z	4	99	J	TE			Dausmann	U	
1	neu	6	Blockseminar C# Summer School Course C#	S		Z	4	99	J	TE			Melcher	U	

POG für Studiengangleiter

Studiengang: SWB Stand WS2008 20.09.2008 Softwaretechnik und Medieninformatik BACHELOR

pvers	vert	pnr	pfsem	pdtxt	part	pdauer	ppflich	pminsem	phoesem	angebote	pform	psws	Credits	nachname	partngb	Gruppe
1		1011	1	Mathematik 1 Mathematics 1	L	150	P	1	4	J	KL	10	10	Ulmet	C	A
1		1011	1	Mathematik 1 Mathematics 1	L	150	P	1	4	J	KL	10	10	Plappert	C	B
1		1011	1	Mathematik 1 Mathematics 1	L	150	P	1	4	J	KL	10	10	Sigg	C	C
1		1021	1	Physik 1 Physics 1	L	90	P	1	4	J	KL	5	5	Coenning	C	A
1		1021	1	Physik 1 Physics 1	L	90	P	1	4	J	KL	5	5	Strobel	C	B
1		1021	1	Physik 1 Physics 1	L	90	P	1	4	J	KL	5	5	Coening	C	C
1		1031	1	Elektrotechnik 1 Electrical Engineering 1	L	90	P	1	4	J	KL	4	4	Gündner	C	A
1		1031	1	Elektrotechnik 1 Electrical Engineering 1	L	90	P	1	4	J	KL	4	4	Malz	C	B
1		1031	1	Elektrotechnik 1 Electrical Engineering 1	L	90	P	1	4	J	KL	4	4	Malz	C	C
1		1041	1	Labor Elektrotechnik 1 Lab Electrical Engineering 1	S		P	1	4	J	BE	1	1	LB Schulz	U	A
1		1041	1	Labor Elektrotechnik 1 Lab Electrical Engineering 1	S		P	1	4	J	BE	1	1	LB Haußer	U	B
1		1041	1	Labor Elektrotechnik 1 Lab Electrical Engineering 1	S		P	1	4	J	BE	1	1	LB Kähler	U	C
1		1051	1	Informatik 1 Computer Science 1	L	90	P	1	4	J	KL	3	3	Kappen	C	A
1		1051	1	Informatik 1 Computer Science 1	L	90	P	1	4	J	KL	3	3	Warendorf	C	B
1		1051	1	Informatik 1 Computer Science 1	L	90	P	1	4	J	KL	3	3	Weber	C	C
1		1061	1	Labor Informatik 1 Lab Computer Science 1	S		P	1	4	J	BE	2	2	LB Müller, Martin	U	A
1		1061	1	Labor Informatik 1 Lab Computer Science 1	S		P	1	4	J	BE	2	2	Warendorf	U	B
1		1061	1	Labor Informatik 1 Lab Computer Science 1	S		P	1	4	J	BE	2	2	Weber	U	C

POG für Studiengangleiter

Studiengang: SWB Stand WS2008 20.09.2008 Softwaretechnik und Medieninformatik BACHELOR

pvers	vert	pnr	pfsem	pdtxt	part	pdauer	ppflich	pminsem	phoesem	angebote	pform	psws	Credits	nachname	partngb	Gruppe
1		1071	1	Technisches Englisch Technical English	L	60	P	1	4	J	KL	2	2	Warendorf, Eve	C	A
1		1071	1	Technisches Englisch Technical English	L	60	P	1	4	J	KL	2	2	Warendorf, Eve	C	B
1		1071	1	Technisches Englisch Technical English	L	60	P	1	4	J	KL	2	2	Warendorf, Eve	C	C
1		1081	1	Persönlichkeitsentwicklung Personality Development	S		P	1	4	J	TE	3	3	Väterlein	U	A
1		1081	1	Persönlichkeitsentwicklung Personality Development	S		P	1	4	J	TE	3	3	Väterlein	U	B
1		1081	1	Persönlichkeitsentwicklung Personality Development	S		P	1	4	J	TE	3	3	Väterlein	U	C

POG für Studiengangleiter

Studiengang: SWB Stand WS2008 20.09.2008 Softwaretechnik und Medieninformatik BACHELOR

pvers	vert	pnr	pfsem	pdtxt	part	pdauer	ppflich	pminsem	phoesem	angebote	pform	psws	Credits	nachname	partngb	Gruppe
1		2011	2	Mathematik 2 Mathematics 2	L	90	P	1	4	J	KL	4	4	Ulmet	C	A
1		2011	2	Mathematik 2 Mathematics 2	L	90	P	1	4	J	KL	4	4	Koch	C	B
1		2011	2	Mathematik 2 Mathematics 2	L	90	P	1	4	J	KL	4	4		C	
1		2021	2	Labor Mathematik 2 Lab Mathematics 2	S		P	1	4	J	TE	1	1	Ulmet	U	A
1		2021	2	Labor Mathematik 2 Lab Mathematics 2	S		P	1	4	J	TE	1	1	Koch	U	B
1		2021	2	Labor Mathematik 2 Lab Mathematics 2	S		P	1	4	J	TE	1	1		U	
1		2031	2	Physik 2 Physics 2	L	90	P	1	4	J	KL	4	4	Martin	C	A
1		2031	2	Physik 2 Physics 2	L	90	P	1	4	J	KL	4	4	Coenning	C	B
1		2031	2	Physik 2 Physics 2	L	90	P	1	4	J	KL	4	4		C	
1		2041	2	Labor Physik 2 Lab Physics 2	S		P	1	4	J	BE	1	1	Strobel	U	A
1		2041	2	Labor Physik 2 Lab Physics 2	S		P	1	4	J	BE	1	1	Coenning	U	B
1		2041	2	Labor Physik 2 Lab Physics 2	S		P	1	4	J	BE	1	1		U	
1		2051	2	Elektrotechnik 2 Electrical Engineering 2	L	90	P	1	4	J	KL	4	4	Höfer	C	A
1		2051	2	Elektrotechnik 2 Electrical Engineering 2	L	90	P	1	4	J	KL	4	4	Höfer	C	B
1		2051	2	Elektrotechnik 2 Electrical Engineering 2	L	90	P	1	4	J	KL	4	4		C	
1		2061	2	Labor Elektrotechnik 2 Lab Electrical Engineering 2	S		P	1	4	J	BE	1	1	LB Hehl	U	A
1		2061	2	Labor Elektrotechnik 2 Lab Electrical Engineering 2	S		P	1	4	J	BE	1	1	LB Rauschnabel	U	B
1		2061	2	Labor Elektrotechnik 2 Lab Electrical Engineering 2	S		P	1	4	J	BE	1	1		U	

POG für Studiengangleiter

Studiengang: SWB Stand WS2008 20.09.2008 Softwaretechnik und Medieninformatik BACHELOR

pvers	vert	pnr	pfsem	pdtxt	part	pdauer	ppflich	pminsem	phoesem	angebote	pform	psws	Credits	nachname	partngb	Gruppe
1		2071	2	Elektronik Electronics	L	90	P	1	4	J	KL	3	3	Gündner	C	A
1		2071	2	Elektronik Electronics	L	90	P	1	4	J	KL	3	3	Buck	C	B
1		2071	2	Elektronik Electronics	L	90	P	1	4	J	KL	3	3		C	
1		2081	2	Projekt Elektronik Project Electronics	S		P	1	4	J	PA	2	2	Gündner	U	A
1		2081	2	Projekt Elektronik Project Electronics	S		P	1	4	J	PA	2	2	Buck	U	B
1		2081	2	Projekt Elektronik Project Electronics	S		P	1	4	J	PA	2	2		U	
1		2091	2	Informatik 2 Computer Science 2	L	90	P	1	4	J	KL	3	3	Beck	C	A
1		2091	2	Informatik 2 Computer Science 2	L	90	P	1	4	J	KL	3	3	Warendorf	C	B
1		2091	2	Informatik 2 Computer Science 2	L	90	P	1	4	J	KL	3	3		C	
1		2101	2	Projekt Informatik 2 Project Computer Science 2	S		P	1	4	J	PA	2	2	Beck	U	A
1		2101	2	Projekt Informatik 2 Project Computer Science 2	S		P	1	4	J	PA	2	2	Warendorf	U	B
1		2101	2	Projekt Informatik 2 Project Computer Science 2	S		P	1	4	J	PA	2	2		U	
1		2111	2	Computerarchitektur 1 Computer Architecture 1	L	90	P	1	4	J	KL	4	4	Lindermeir	C	A
1		2111	2	Computerarchitektur 1 Computer Architecture 1	L	90	P	1	4	J	KL	4	4	Keller	C	B
1		2111	2	Computerarchitektur 1 Computer Architecture 1	L	90	P	1	4	J	KL	4	4		C	
1		2121	2	Labor Computerarchitektur 1 Lab Computer Architecture 1	S		P	1	4	J	BE	1	1	Lindermeir	U	A
1		2121	2	Labor Computerarchitektur 1 Lab Computer Architecture 1	S		P	1	4	J	BE	1	1	Keller	U	B
1		2121	2	Labor Computerarchitektur 1 Lab Computer Architecture 1	S		P	1	4	J	BE	1	1		U	

POG für Studiengangleiter

Studiengang: SWB Stand WS2008 20.09.2008 Softwaretechnik und Medieninformatik BACHELOR

pvers	vert	pnr	pfsem	pdtxt	part	pdauer	ppflich	pminsem	phoesem	angebote	pform	psws	Credits	nachname	partngb	Gruppe
1		3001	3	Mathematik 3 Mathematics 3	L	90	P	3	10	J	KL	5	5	Sigg	C	A
1		3001	3	Mathematik 3 Mathematics 3	L	90	P	3	10	J	KL	5	5	Glatz	C	B
1		3011	3	Informatik 3 Computer Science 3	L	90	P	3	10	J	KL	3	3	Schoop	C	A
1		3011	3	Informatik 3 Computer Science 3	L	90	P	3	10	J	KL	3	3	Rößler	C	B
1		3021	3	Labor Informatik 3 Lab Computer Science 3	S		P	3	10	J	BE	2	2	Schoop	U	A
1		3021	3	Labor Informatik 3 Lab Computer Science 3	S		P	3	10	J	BE	2	2	Rößler	U	B
1		3031	3	Computerarchitektur 2 Computer Architectur 2	L	90	P	3	10	J	KL	4	4	Lindermeir	C	A
1		3031	3	Computerarchitektur 2 Computer Architectur 2	L	90	P	3	10	J	KL	4	4	Keller	C	B
1		3041	3	Labor Computerarchitektur 2 Labor Computer Architecture 2	S		P	3	10	J	BE	1	1	Lindermeir	U	A
1		3041	3	Labor Computerarchitektur 2 Labor Computer Architecture 2	S		P	3	10	J	BE	1	1	Keller	U	B
1		3051	3	Signale und Systeme Signals and Systems	L	90	P	3	10	J	KL	4	4	Doster	C	A
1		3051	3	Signale und Systeme Signals and Systems	L	90	P	3	10	J	KL	4	4	Doster	C	B
1		3061	3	Labor Signale und Systeme Lab Signals and Systems	S		P	3	10	J	BE	1	1	Doster	U	A
1		3061	3	Labor Signale und Systeme Lab Signals and Systems	S		P	3	10	J	BE	1	1	Doster	U	B
1		3071	3	Betriebssyteme Operating Systems	L	90	P	3	10	J	KL	4	4	Väterlein	C	A
1		3071	3	Betriebssyteme Operating Systems	L	90	P	3	10	J	KL	4	4	Weber	C	B
1		3081	3	Labor Betriebssyteme Lab Operating Systems	S		P	3	10	J	BE	1	1	Väterlein	U	A
1		3081	3	Labor Betriebssyteme Lab Operating Systems	S		P	3	10	J	BE	1	1	Weber	U	B

POG für Studiengangleiter

Studiengang: SWB Stand WS2008 20.09.2008 Softwaretechnik und Medieninformatik BACHELOR

pvers	vert	pnr	pfsem	pdtxt	part	pdauer	ppflich	pminsem	phoesem	angebote	pform	psws	Credits	nachname	partngb	Gruppe
1		3091	3	Datenbanken 1 Database Systems 1	L	90	P	3	10	J	KL	4	4	Nonnast	C	A
1		3091	3	Datenbanken 1 Database Systems 1	L	90	P	3	10	J	KL	4	4	Schoop	C	B
1		3101	3	Labor Datenbanken 1 Lab Database Systems 1	S		P	3	10	J	BE	1	1	Nonnast	U	A
1		3101	3	Labor Datenbanken 1 Lab Database Systems 1	S		P	3	10	J	BE	1	1	Schoop	U	B

POG für Studiengangleiter

Studiengang: SWB Stand WS2008 20.09.2008 Softwaretechnik und Medieninformatik BACHELOR

pvers	vert	pnr	pfsem	pdtxt	part	pdauer	ppflich	pminsem	phoesem	angebote	pform	psws	Credits	nachname	partngb	Gruppe
1		4001	4	Betriebswirtschaft Business Economics	L	90	P	3	10	J	KL	4	4	LB Feil	U	A
1		4001	4	Betriebswirtschaft Business Economics	L	90	P	3	10	J	KL	4	4	LB Feil	U	B
1		4011	4	Projektmanagement Project Management	S		P	3	10	J	BE	1	1	LB Stumpp	C	A
1		4011	4	Projektmanagement Project Management	S		P	3	10	J	BE	1	1	LB Stumpp	C	B
1		4021	4	Computerarchitektur 3 Computer Architekture 3	L	90	P	3	10	J	KL	4	4	Friedrich	U	A
1		4021	4	Computerarchitektur 3 Computer Architekture 3	L	90	P	3	10	J	KL	4	4	Friedrich	U	B
1		4031	4	Labor Computerarchitektur 3 Lab Computer Architecture	S		P	3	10	J	BE	1	1	Friedrich	C	A
1		4031	4	Labor Computerarchitektur 3 Lab Computer Architecture	S		P	3	10	J	BE	1	1	Friedrich	C	B
1		4041	4	Systemtechnik 1 System Design 1	L	90	P	3	10	J	KL	4	4	Zimmermann	U	A
1		4041	4	Systemtechnik 1 System Design 1	L	90	P	3	10	J	KL	4	4	Zimmermann	U	B
1		4051	4	Labor Systemtechnik 1 Lab System Design 1	S		P	3	10	J	BE	1	1	Zimmermann	C	A
1		4051	4	Labor Systemtechnik 1 Lab System Design 1	S		P	3	10	J	BE	1	1	Zimmermann	C	B
1		4061	4	Echtzeitsysteme Real Time Systems	L	90	P	3	10	J	KL	4	4	Friedrich	U	A
1		4061	4	Echtzeitsysteme Real Time Systems	L	90	P	3	10	J	KL	4	4	Friedrich	U	B
1		4071	4	Labor Echtzeitsysteme Lab Real Time Systems	S		P	3	10	J	BE	1	1	Friedrich	C	A
1		4071	4	Labor Echtzeitsysteme Lab Real Time Systems	S		P	3	10	J	BE	1	1	Friedrich	C	B
1		4081	4	Rechnernetze 1 Computer Networks 1	L	90	P	3	10	J	KL	4	4	Wiese	U	A
1		4081	4	Rechnernetze 1 Computer Networks 1	L	90	P	3	10	J	KL	4	4	Wiese	U	B
1		4091	4	Labor Rechnernetze 1 Lab Computer Networks 1	S		P	3	10	J	BE	1	1	Wiese	C	A
1		4091	4	Labor Rechnernetze 1 Lab Computer Networks 1	S		P	3	10	J	BE	1	1	Wiese	C	B

POG für Studiengangleiter

Studiengang: SWB Stand WS2008 20.09.2008 Softwaretechnik und Medieninformatik BACHELOR

pvers	vert	pnr	pfsem	pdtxt	part	pdauer	ppflich	pminsem	phoesem	angebote	pform	psws	Credits	nachname	partngb	Gruppe
1		4101	4	Objektorientierte Systeme 1 Object Oriented Systems 1	L	90	P	3	10	J	KL	3	3	Dausmann	U	A
1		4101	4	Objektorientierte Systeme 1 Object Oriented Systems 1	L	90	P	3	10	J	KL	3	3	Dausmann	U	B
1		4111	4	Labor Objektorientierte Systeme 1 Lab Object Oriented Systems 1	S		P	3	10	J	BE	2	2	Dausmann	C	A
1		4111	4	Labor Objektorientierte Systeme 1 Lab Object Oriented Systems 1	S		P	3	10	J	BE	2	2	Dausmann	C	B

POG für Studiengangleiter

Studiengang: SWB Stand WS2008 20.09.2008 Softwaretechnik und Medieninformatik BACHELOR

pvers	vert	pnr	pfsem	pdtxt	part	pdauer	ppflich	pminsem	phoesem	angebote	pform	psws	Credits	nachname	partngb	Gruppe
1		5011	5	Ingenieurmethodiken 2 Methodologies for Engineers 2	L		P	3	10	J	TE	4	3	Gündner	C	
1		5021	5	Betriebliche Praxis Company Internship	S		P	3	10	J	BE	26	0	Doster	U	

POG für Studiengangleiter

Studiengang: SWB Stand WS2008 20.09.2008 Softwaretechnik und Medieninformatik BACHELOR

pvers	vert	pnr	pfsem	pdtxt	part	pdauer	ppflich	pminsem	phoesem	angebote	pform	psws	Credits	nachname	partngb	Gruppe
1		6011	6	Studienarbeit Application Project	L		P	6	10	J	RE	5	0	it-professoren	C	
1		6021	6	Projekt Datenbanken 2 Project Database Systems 2	S		P	3	10	J	BE	2	2	Nonnast	U	
1		6031	6	Datenbanken 2 Database Systems 2	L	90	P	3	10	J	KL	3	3	Nonnast	C	
1		6081	6	Labor Grafische Benutzungsoberflächen Lab Grafical User Interfaces	S		P	3	10	J	BE	1	1	Rößler	U	
1		6091	6	Grafische Benutzungsoberflächen Grafical User Interfaces	L	90	P	3	10	J	KL	4	4	Rößler	C	

POG für Studiengangleiter

Studiengang: SWB Stand WS2008 20.09.2008 Softwaretechnik und Medieninformatik BACHELOR

pvers	vert	pnr	pfsem	pdtxt	part	pdauer	ppflich	pminsem	phoesem	angebote	pform	psws	Credits	nachname	partngb	Gruppe
1		7000	7	Bachelorarbeit und Kolloquium Bachelor Thesis and Defense	P		P	6	10	J	BA	15	15	it professoren	C	
1		7011	7	Wissenschaftliche Vertiefung Scientific Research	L		P	6	10	J	ML	9	9	it professoren	C	

POG für Studiengangleiter

Studiengang: SWB Stand WS2008 20.09.2008 Softwaretechnik und Medieninformatik BACHELOR

pvers	vert	pnr	pfsem	pdtxt	part	pdauer	ppflich	pminsem	phoesem	angebote	pform	psws	Credits	nachname	partngb	Gruppe
				Medientechnik												
1	SWM	6041	6	Labor Digitale Medien Lab Digital Media	S		P	3	10	J	BE	1	1	Beck	U	
1	SWM	6051	6	Digitale Medien Digital Media	L	90	P	3	10	J	KL	4	4	Schmidt	C	
1	SWM	6061	6	Labor Virtuelle Realität Lab Virtual Reality	S		P	3	10	J	BE	1	1	Schmidt	U	
1	SWM	6071	6	Virtuelle Realität Virtual Reality	L	90	P	3	10	J	KL	4	4	Schmidt	C	
1	SWM	6081	6	Labor Interaktive Systeme Lab Interactive Systems	S		P	3	10	J	BE	1	1	Beck	U	
1	SWM	6111	6	Interaktive Systeme Interactive Systems	L	90	P	3	10	J	KL	4	4	Beck	C	

POG für Studiengangleiter

Studiengang: SWB Stand WS2008 20.09.2008 Softwaretechnik und Medieninformatik BACHELOR

pvers	vert	pnr	pfsem	pdtxt	part	pdauer	ppflich	pminsem	phoesem	angebote	pform	psws	Credits	nachname	partngb	Gruppe
				Softwaretechnik												
1	SWT	6041	6	Labor Objektorientierte Systeme 2 Lab Object Oriented Systems 2	S		P	3	10	J	A	BE	1	Dausmann	U	
1	SWT	6051	6	Objektorientierte Systeme 2 Object Oriented Systems 2	L	90	P	3	10	J	A	KL	4	Dausmann	C	
1	SWT	6061	6	Labor Softwarearchitektur Lab Software Architecture	S		P	3	10	J	A	BE	1	LB Müller-Hofmann	U	
1	SWT	6071	6	Softwarearchitektur Software Architecture	L	90	P	3	10	J	A	KL	4	LB Müller-Hofmann	C	
1	SWT	6101	6	Labor Rechnerbetrieb Lab Operation of Computer Systems	S		P	3	10	J	A	BE	1	Weber	U	
1	SWT	6111	6	Rechnerbetrieb Computer Systems Operating	L	90	P	3	10	J	A	KL	4	Weber	C	

POG für Studiengangleiter

Studiengang: SWB Stand WS2008 20.09.2008 Softwaretechnik und Medieninformatik BACHELOR

pvers	vert	pnr	pfsem	pdtxt	part	pdauer	ppflich	pminsem	phoesem	angebote	pform	psws	Credits	nachname	partngb	Gruppe
				WAHLFÄCHER SWB												
1		8811	6	Algorithmen Algorithms	L		W	4	99	J	ML	2	2	Koch	C	
1		8821	6	Fundamentals of Optoelectronics Fundamentals of Optoelectronics	L		W	4	99	N	ML	2	2		C	
1		8831	6	IT-Sicherheit IT-Security	L		W	4	99	J	ML	2	2	Schoop	C	
1		8841	6	Fernsehtechnik Digital Video Broadcasting	L		W	4	99	N	ML	2	2		C	
1		8851	6	Technischer Vertrieb Sales and Marketing	L		W	4	99	N	ML	2	2		C	
1		8861	6	Programmieren in C# Introduction to C#	L		W	4	99	N	ML	2	2		C	
1		8871	6	Introduction to eCommerce Introduction to eCommerce	L		W	4	99	N	ML	2	2		C	
1		8881	6	Systemarchitekturen mit .NET Software Systems Architectues with .NET	L		W	4	99	J	ML	2	2	LB Erath	C	
				neue WAHLFÄCHER SWB												
1	neu		6	Vertrags- und Internet-Recht Contract and Media Law	L		W	4	99	J	ML	2	2	LB Schließ	C	
1	neu		6	Methoden der künstlichen Intelligenz Artificial Intelligence	L		W	4	99	J	ML	2	2	Weber	C	
1	neu		6	Einführung in CAD Introduction to CAD	L		W	4	99	J	ML	2	2	Koch	C	
1	neu		6	Kfz-Systeme Automotive Systems Engineering	L		W	4	99	J	ML	2	2	LB Marchthaler	C	
1	neu		6	Qualitätstechniken Quality Engineering	L		W	4	99	J	ML	2	2	LB Grübel	C	
1	neu		6	Optische Nachrichtenübertragung Optical Communications Engineering	L		W	4	99	J	ML	2	2	LB Khakzar	C	

POG für Studiengangleiter

Studiengang: SWB Stand WS2008 20.09.2008 Softwaretechnik und Medieninformatik BACHELOR

pvers	vert	pnr	pfsem	pdtxt	part	pdauer	ppflich	pminsem	phoesem	angebote	pform	psws	Credits	nachname	partngb	Gruppe
				neue Zusatzfächer TIB												
1		neu	6	Blockseminar XML Summer School Course XML	S		Z	4	99	J	TE			Dausmann	U	
1		neu	6	Blockseminar C# Summer School Course C#	S		Z	4	99	J	TE			Melcher	U	

HOCHSCHULE ESSLINGEN

Service-Anforderung der Fakultät IT an das Institut für Fremdsprachen IFS

Fachbereich: **Informationstechnik** **SS10** **Datum:** **20.11.2009**

Fach	Fach-Nr.	P W Z	Stud. Woh V	Stud. Woh Ü	an FB-Sem.	Doz. Woh	Dozent	gekoppelt mit	Bemerkungen
									2 Züge im SS BA-StuPO 1. Semester
Technisches Englisch	1072	P	2	0	IT1A	4			BA-StuPO ca. 35 Studenten 2 x2 Stunden, Halbierung des Semesters
Technisches Englisch	1072	P	2	0	IT1B	4			BA-StuPO ca. 35 Studenten 2 x2 Stunden, Halbierung des Semesters
Database Systems 1	3092	P	0	2	IT3A/B	4			Unterstützung Datase Systems
						16			

HOCHSCHULE ESSLINGEN

Service-Anforderung der Fakultät IT an die Fakultät G

Fachbereich: **Informationstechnik** **SS10** **Datum:** **20.11.2009**

Fach	Fach-Nr.	P W Z	Stud. Woh V	Stud. Woh Ü	an FB-Sem.	Doz. Woh	Dozent	gekoppelt mit	Bemerkungen
Mathematik 1	1012	P	10	0	IT1A	10			**2 Züge im SS BA-StuPO 1. Semester**
Mathematik 1	1012	P	10	0	IT1B	10			
Zusätzliche Übungen Mathematik 1	ohne	Z	0	2	IT1A	2			
Zusätzliche Übungen Mathematik 1	ohne	Z	0	2	IT1B	2			
Tutorium Mathematik 1	ohne	Z	0	2	IT1A	0			Tutoren
Tutorium Mathematik 1	ohne	Z	0	2	IT1B	0			Tutoren
						24			

HOCHSCHULE ESSLINGEN

Service-Anforderung der Fakultät IT an die Fakultät G

Fachbereich: **Informationstechnik** **SS10** **Datum:** **20.11.2009**

Fach	Fach-Nr.	P W Z	Stud. Woh V	Stud. Woh Ü	An FB-Sem.	Doz. Woh	Dozent	gekoppelt mit	Bemerkungen
Übertrag						**24**			**2 Züge im SS BA-StuPO 1. Semester**
Physik 1	1022	P	5	0	IT1A	5			
Physik 1	1022	P	5	0	IT1B	5			
Zusätzliche Übungen Physik 1	ohne	Z	0	2	IT1A	2			LB aus Studiengebühren G
Zusätzliche Übungen Physik 1	ohne	Z	0	2	IT1B	2			LB aus Studiengebühren G
Physik Tutorium	ohne	Z	0	2	IT1A	0			Tutoren
Physik Tutorium	ohne	Z	0	2	IT1B	0			Tutoren
						38			

HOCHSCHULE ESSLINGEN

Service-Anforderung der Fakultät IT an die Fakultät G

Fachbereich: **Informationstechnik** **SS10** **Datum:** **20.11.2009**

Fach	Fach-Nr.	P W Z	Stud. Woh V	Stud. Woh Ü	An FB-Sem.	Doz. Woh	Dozent	gekoppelt mit	Bemerkungen
Übertrag						**38**			**4 Züge im SS BA-StuPO 2. Semester**
Mathematik 2	2012	P	4	0	IT2A	4			
Mathematik 2	2012	P	4	0	IT2B	4			
Mathematik 2	2012	P	4	0	IT2C	4			
Mathematik 2	2012	P	4	0	IT2D	0			Höfer IT
Labor Mathematik 2	2022	P	1	0	IT2A	1			
Labor Mathematik 2	2022	P	1	0	IT2B	1			
Labor Mathematik 2	2022	P	1	0	IT2C	1			
Labor Mathematik 2	2022	P	1	0	IT2D	0			Höfer IT
Mathematik 2 Zusatzüb.	ohne	Z	0	2	IT2A	1			1 Stunde davon aus Studiengebühren G
Mathematik 2 Zusatzüb.	ohne	Z	0	2	IT2B	1			1 Stunde davon aus Studiengebühren G
Mathematik 2 Zusatzüb.	ohne	Z	0	2	IT2C	1			1 Stunde davon aus Studiengebühren G
Mathematik 2 Zusatzüb.	ohne	Z	0	2	IT2D	1			1 Stunde davon aus Studiengebühren G
						57			

HOCHSCHULE ESSLINGEN

Service-Anforderung der Fakultät IT an die Fakultät G

Fachbereich: **Informationstechnik** **SS10** **Datum:** **20.11.2009**

Fach	Fach-Nr.	P W Z	Stud. Woh V	Stud. Woh Ü	An FB-Sem.	Doz. Woh	Dozent	gekoppelt mit	Bemerkungen
Übertrag						**57**			**2 Züge im WS BA-StuPO 2. Semester**
Physik 2	2032	P	4	0	IT2A	4			
Physik 2	2032	P	4	0	IT2B	4			
Physik 2	2032	P	4	0	IT2C	0			Väterlein IT
Physik 2	2032	P	4	0	IT2D	4			
Labor Physik 2	2042	Z	0	1	IT2A	2			Halbierung der Kurse
Labor Physik 2	2042	Z	0	1	IT2B	2			Halbierung der Kurse
Labor Physik 2	2042	Z	0	1	IT2A	2			Halbierung der Kurse
Labor Physik 2	2042	Z	0	1	IT2B	2			Halbierung der Kurse
						77			

HOCHSCHULE ESSLINGEN

Service-Anforderung der Fakultät IT an die Fakultät G

Fachbereich: **Informationstechnik** **SS10** **Datum:** **20.11.2009**

Fach	Fach-Nr.	P W Z	Stud. Woh V	Stud. Woh Ü	an FB-Sem.	Doz. Woh	Dozent	gekoppelt mit	Bemerkungen
Übertrag						**77**			**2 Züge BA-StuPO 3. Semester**
Elektronik	2072	P		4	IT2A	4			
Elektronik	2072	P		4	IT2B	1			Gekoppelt IT2A / IT2B
Projekt Elektronik	2082	P		1	IT2A	2			
Projekt Elektronik	2082	P		1	IT2B	2			
Mathematik 3	3002	P	5	0	IT3A	5			
Mathematik 3	3002	P	5	0	IT3B	5			
						96			

HOCHSCHULE ESSLINGEN

Service-Anforderung der Fakultät IT an die Fakultät G

Fachbereich: **Informationstechnik** **SS10** **Datum:** **20.11.2009**

Fach	Fach-Nr.	P W Z	Stud. Woh V	Stud. Woh Ü	an FB-Sem.	Doz. Woh	Dozent	gekoppelt mit	Bemerkungen
						96			
Numerische Methoden		W	2	0	IT	2			
						98			

HOCHSCHULE ESSLINGEN

Service-Anforderung der Fakultät IT an das Institut für Fremdsprachen IFS

Fachbereich: Informationstechnik **SS10** **Datum: 20.11.2009**

Fach	Fach-Nr.	P W Z	Stud. Woh V	Stud. Woh Ü	an FB-Sem.	Doz. Woh	Dozent	gekoppelt mit	Bemerkungen
									2 Züge im SS BA-StuPO 1. Semester
Technisches Englisch	1072	P	2	0	IT1A	4			BA-StuPO ca. 35 Studenten 2 x2 Stunden, Halbierung des Semesters
Technisches Englisch	1072	P	2	0	IT1B	4			BA-StuPO ca. 35 Studenten 2 x2 Stunden, Halbierung des Semesters
Database Systems 1	3092	P	0	2	IT3A/B	4			Unterstützung Datase Systems
						16			

HOCHSCHULE ESSLINGEN

Service-Anforderung der Fakultät IT an die Fakultät G

Fachbereich: **Informationstechnik** **SS10** **Datum:** **20.11.2009**

Fach	Fach-Nr.	P W Z	Stud. Woh V	Stud. Woh Ü	an FB-Sem.	Doz. Woh	Dozent	gekoppelt mit	Bemerkungen
Mathematik 1	1012	P	10	0	IT1A	10			**2 Züge im SS BA-StuPO 1. Semester**
Mathematik 1	1012	P	10	0	IT1B	10			
Zusätzliche Übungen Mathematik 1	ohne	Z	0	2	IT1A	2			
Zusätzliche Übungen Mathematik 1	ohne	Z	0	2	IT1B	2			
Tutorium Mathematik 1	ohne	Z	0	2	IT1A	0			Tutoren
Tutorium Mathematik 1	ohne	Z	0	2	IT1B	0			Tutoren
						24			

HOCHSCHULE ESSLINGEN

Service-Anforderung der Fakultät IT an die Fakultät G

Fachbereich: **Informationstechnik** **SS10** **Datum:** **20.11.2009**

Fach	Fach-Nr.	P W Z	Stud. Woh V	Stud. Woh Ü	An FB-Sem.	Doz. Woh	Dozent	gekoppelt mit	Bemerkungen
Übertrag						**24**			**2 Züge im SS BA-StuPO 1. Semester**
Physik 1	1022	P	5	0	IT1A	5			
Physik 1	1022	P	5	0	IT1B	5			
Zusätzliche Übungen Physik 1	ohne	Z	0	2	IT1A	2			LB aus Studiengebühren G
Zusätzliche Übungen Physik 1	ohne	Z	0	2	IT1B	2			LB aus Studiengebühren G
Physik Tutorium	ohne	Z	0	2	IT1A	0			Tutoren
Physik Tutorium	ohne	Z	0	2	IT1B	0			Tutoren
						38			

HOCHSCHULE ESSLINGEN

Service-Anforderung der Fakultät IT an die Fakultät G

Fachbereich: **Informationstechnik** **SS10** **Datum:** **20.11.2009**

Fach	Fach-Nr.	P W Z	Stud. Woh V	Stud. Woh Ü	An FB-Sem.	Doz. Woh	Dozent	gekoppelt mit	Bemerkungen
Übertrag						**38**			**4 Züge im SS BA-StuPO 2. Semester**
Mathematik 2	2012	P	4	0	IT2A	4			
Mathematik 2	2012	P	4	0	IT2B	4			
Mathematik 2	2012	P	4	0	IT2C	4			
Mathematik 2	2012	P	4	0	IT2D	0			Höfer IT
Labor Mathematik 2	2022	P	1	0	IT2A	1			
Labor Mathematik 2	2022	P	1	0	IT2B	1			
Labor Mathematik 2	2022	P	1	0	IT2C	1			
Labor Mathematik 2	2022	P	1	0	IT2D	0			Höfer IT
Mathematik 2 Zusatzüb.	ohne	Z	0	2	IT2A	1			1 Stunde davon aus Studiengebühren G
Mathematik 2 Zusatzüb.	ohne	Z	0	2	IT2B	1			1 Stunde davon aus Studiengebühren G
Mathematik 2 Zusatzüb.	ohne	Z	0	2	IT2C	1			1 Stunde davon aus Studiengebühren G
Mathematik 2 Zusatzüb.	ohne	Z	0	2	IT2D	1			1 Stunde davon aus Studiengebühren G
						57			

HOCHSCHULE ESSLINGEN

Service-Anforderung der Fakultät IT an die Fakultät G

Fachbereich: **Informationstechnik** **SS10** **Datum:** **20.11.2009**

Fach	Fach-Nr.	P W Z	Stud. Woh V	Stud. Woh Ü	An FB-Sem.	Doz. Woh	Dozent	gekoppelt mit	Bemerkungen
Übertrag						**57**			**2 Züge im WS BA-StuPO 2. Semester**
Physik 2	2032	P	4	0	IT2A	4			
Physik 2	2032	P	4	0	IT2B	4			
Physik 2	2032	P	4	0	IT2C	0			Väterlein IT
Physik 2	2032	P	4	0	IT2D	4			
Labor Physik 2	2042	Z	0	1	IT2A	2			Halbierung der Kurse
Labor Physik 2	2042	Z	0	1	IT2B	2			Halbierung der Kurse
Labor Physik 2	2042	Z	0	1	IT2A	2			Halbierung der Kurse
Labor Physik 2	2042	Z	0	1	IT2B	2			Halbierung der Kurse
						77			

HOCHSCHULE ESSLINGEN

Service-Anforderung der Fakultät IT an die Fakultät G

Fachbereich: **Informationstechnik** **SS10** **Datum:** **20.11.2009**

Fach	Fach-Nr.	P W Z	Stud. Woh V	Stud. Woh Ü	an FB-Sem.	Doz. Woh	Dozent	gekoppelt mit	Bemerkungen
Übertrag						**77**			**2 Züge BA-StuPO 3. Semester**
Elektronik	2072	P		4	IT2A	4			
Elektronik	2072	P		4	IT2B	1			Gekoppelt IT2A / IT2B
Projekt Elektronik	2082	P		1	IT2A	2			
Projekt Elektronik	2082	P		1	IT2B	2			
Mathematik 3	3002	P	5	0	IT3A	5			
Mathematik 3	3002	P	5	0	IT3B	5			
						96			

HOCHSCHULE ESSLINGEN

Service-Anforderung der Fakultät IT an die Fakultät G

Fachbereich: **Informationstechnik** **SS10** **Datum:** **20.11.2009**

Fach	Fach-Nr.	P W Z	Stud. Woh V	Stud. Woh Ü	an FB-Sem.	Doz. Woh	Dozent	gekoppelt mit	Bemerkungen
						96			
Numerische Methoden		W	2	0	IT	2			
						98			

Hochschule Esslingen

Service Anforderung der Fakultät GS an IT

Fakultät: **GS** **Studiengang:** **Automotive Systems (ASM)** **Semester:** **ASM 2** **SS 2010** **Datum: 25.09.09**

Fach	Fach-Nr.	P WZ	Stud. - Woh lt. Stundenplan SWS	Stud. - Woh ***	Serv vom FB	Doz. Woh DWS	Dozent	gekoppelt mit Sem. Fach-Nr.	Bemerkungen
Wireless and Wired On-board and Offboard Communication Systems		P	4	4	**IT**	4	Zieher 2 DWS Melcher 2 DWS		
Man-Machine-Interactions		P	4	4	**IT**	4	Rößler 2 DWS Beck 1 DWS		LB Meroth 1 DWS
Safety and Security		P	4	4	**IT**	2	Kull 2 DWS		Schoop GS 2 DWS
Selected Topics on Real-Time-Systems		P	4	4	**IT**	4	Friedrich 4 DWS		
Team Project		P	3	8	**IT**	6	Friedrich 2 DWS		Schoop GS 2 DWS
Summe DWS						20			

*** Tatsächlich im Stundenplan einzugebende Stundenzahl, durch 2 teilbar!

Hochschule Esslingen

Service-Angebot der Fakultät Informationstechnik an AN (BT)

Fakultät: **Informationstechnik** **SS10** **Datum:** **20.11. 2009**

Fach	Fach-Nr.	P W Z	Stud. Woh V	Stud. Woh Ü	an Fakultät Sem.	Doz. Woh	Dozent	gekoppelt mit	Bemerkungen
Bioinformatik		P	2	0	BT7	**1**	Melcher		Montag 5. Stunde

Hochschule Esslingen

Service Angebot der Fakultät IT an GS

Fakultät: **IT** **Studiengang:** **Automotive Systems (ASM)** **Semester:** **ASM 2** **SS 2010** **Datum: 25.09.09**

Fach	Fach-Nr.	P WZ	Stud.-Woh lt. Stundenplan SWS	Stud.-Woh ***	Serv vom FB	Doz. Woh DWS	Dozent	gekoppelt mit Sem. Fach-Nr.	Bemerkungen
Wireless and Wired Onboard and Offboard Communication Systems		P	4	4	**IT**	4	Zieher IT 2 DWS Melcher IT 2 DWS		
Man-Machine-Interactions		P	4	4	**IT**	4	Rößler IT 2 DWS Beck IT 1 DWS		LB Meroth 1 DWS
Safety and Security		P	4	4	**IT**	0			Schoop GS 4 DWS
Selected Topics on Real-Time-Systems		P	4	4	**IT**	4	Friedrich IT 4 DWS		
Team Project		P	3	8	**IT**	6	Friedrich IT 2 DWS		Schoop GS 2 DWS
Summe DWS						18			

*** Tatsächlich im Stundenplan einzugebende Stundenzahl, durch 2 teilbar!

Informationstechnik

Bachelor – Studiengänge

HOCHSCHULE ESSLINGEN **IT1A**

Lehrangebot der Fakultät Informationstechnik **BA-StuPO**

Fakultät: **IT** **Studiengang:** **Informationstechnik** **Semester:** **IT1A** **SS10** **Datum:** **29.11.09**

Fach	Fach-Nr.	P W Z	Stud. – Woh lt. Stundenplan	***	Serv vom FB	Doz. Woh	Dozent	gekoppelt mit Semester Fach-Nr.	Bemerkungen
Mathematik 1	1012	P	10	10	G	10	Service		
Mathematik 1 Zusatzübungen	Ohne	Z	0	2	G	2	Service		
Mathematik 1 Tutorium	Ohne	Z	0	2	G	0	Service		Tutoren
Physik 1	1022	P	5	6	G	5	Service		
Physik Tutorium	Ohne	Z	0	2	G	0	Service		Tutoren
Zusätzliche Übungen Physik	Ohne	Z	0	2	G	2	Service		
Elektrotechnik 1	1032	P	4	4	IT	4	Malz		
Labor Elektrotechnik 1	1042	P	1	4	IT	2	LB Kaehler		**Di 6.+7. Stunde** 4h F1.308
Informatik 1	1052	P	3	4	IT	3	Zieher		
Labor Informatik 1	1062	P	2	4	IT	4	Zieher		4h PC-Pool
Ingenieurmethodik 1 Persönlichkeitsentwicklung	1082	P	3	4	IT	3	Väterlein und diverse LBs	IT1A/B	Blockseminar in der ersten Woche
Technisches Englisch	1072	P	2	4	IFS	4	Service Eve Warendorf		2 x 2h **F1.311** Halbierung des Semesters
						0			
		Summe Woh:	30	48		39			

*** Tatsächlich im Stundenplan einzugebende Stundenzahl, durch 2 teilbar!

HOCHSCHULE ESSLINGEN **IT1B**

Lehrangebot der Fakultät Informationstechnik **BA-StuPO**

Fakultät: IT Studiengang: Informationstechnik Semester: IT1B SS10 Datum: 29.11.09

Fach	Fach-Nr.	P W Z	Stud. – Woh lt. Stun-denplan	***	Serv vom FB	Doz. Woh	Dozent	gekoppelt mit Semester Fach-Nr.	Bemerkungen
Mathematik 1	1012	P	10	10	G	10	Service		
Mathematik 1 Zusatzübungen	Ohne	Z	0	2	G	2	Service		
Mathematik 1 Tutorium	Ohne	Z	0	2	G	0	Service		Tutoren
Physik 1	1022	P	5	6	G	5	Service		
Physik Tutorium	Ohne	Z	0	2	G	0	Service		Tutoren
Zusätzliche Übungen Physik	Ohne	Z	0	2	G	2	Service		
Elektrotechnik 1	1032	P	4	4	IT	4	Melcher		
Labor Elektrotechnik 1	1042	P	1	4	IT	2	2h LB Haußer		**Mi 6.+7. Stunde** 4h F1.308
Informatik 1	1052	P	3	4	IT	3	Beck		
Labor Informatik 1	1062	P	2	4	IT	4	LB M. Müller		**Mo 1.+2. Stunde** 4h PC-Pool
Ingenieurmethodik 1 Persönlichkeitsentwicklung	1082	P	3	4	IT	3	diverse LBs	IT1A/B	Blockseminar in der ersten Woche
Technisches Englisch	1072	P	2	4	IFS	4	Service Eve Warendorf		2 x 2h **F1.311** Halbierung des Semsters
Summe Woh:			30	48		39			

*** Tatsächlich im Stundenplan einzugebende Stundenzahl, durch 2 teilbar!

HOCHSCHULE ESSLINGEN **IT2A**

Lehrangebot der Fakultät Informationstechnik **BA-StuPO**

Fakultät: **IT** **Studiengang:** **Informationstechnik** **Semester:** **IT2A** **SS10** **Datum:** **29.11.09**

Fach	Fach-Nr.	P W Z	Stud. - Woh lt. Stundenplan	Stud. - Woh ***	Serv vom FB	Doz. Woh	Dozent	Gekoppelt mit Semester Fach-Nr.	Bemerkungen
Mathematik 2	2012	P	4	4	G	4	Service		
Labor Mathematik 2	2022	P	1	2	G	2	Service		
Mathematik 2 Zusatzübungen	Ohne	Z	0	2	G	0	Service		
Physik 2	2032	P	4	4	G	4	Service		
Labor Physik 2	2042	P	1	4	G	3	Service		
Physik 2 Zusatzübungen	Ohne	Z	0	2	G	2	Service		aus Studiengehühren G
Elektrotechnik 2	2052	P	4	4	IT	4	Malz	**IT2A / IT2B**	
Labor Elektrotechnik 2	2062	P	1	4	IT	2	LB Rauschnabel		**Di 1.+2. Stunde** 4h F1.409
Elektronik	2072	P	3	4	IT	4	Buck	**IT2A / IT2B**	Hörsaal
Projekt Elektronik	2082	P	2	4	IT	4	Buck		4h F1.409
Informatik 2	2092	P	3	4	IT	3	Beck		Hörsaal mit Beamer
Projekt Informatik 2	2102	P	2	4	IT	4	Beck		4h PC-Pool
Computerarchitektur 1	2112	P	4	4	IT	4	Keller	**IT2A / IT2B**	Hörsaal
Labor Computerarchitektur 1	2122	P	1	4	IT	2	Keller		4h F1.304a
		Summe Woh:	30	50		42			

*** Tatsächlich im Stundenplan einzugebende Stundenzahl, durch 2 teilbar!

HOCHSCHULE ESSLINGEN **IT2B**

Lehrangebot der Fakultät Informationstechnik **BA-StuPO**

Fakultät: **IT** **Studiengang:** ______ **Semester:** **IT2B** **SS10** ______ **Datum:** **29.11.09**

Fach	Fach-Nr.	P W Z	Stud. - Woh lt. Stun-denplan	***	Serv vom FB	Doz. Woh	Dozent	Gekoppelt mit Semester Fach-Nr.	Bemerkungen
Mathematik 2	2012	P	4	4	G	4	Service		
Labor Mathematik 2	2022	P	1	2	G	2	Service		
Mathematik 2 Zusatzübungen	Ohne	Z	0	2	G	0	Service		
Physik 2	2032	P	4	4	G	4	Service		
Labor Physik 2	2042	P	1	4	G	3	Service		
Physik 2 Zusatzübungen	Ohne	Z	0	2	G	2	Service		Studiengehühren G
Elektrotechnik 2	2052	P	4	4	IT	1	Malz	**IT2A / IT2B**	
Labor Elektrotechnik 2	2062	P	1	4	IT	2	2h LB Hehl		**Mo 1.+2. Stunde** 4h F1.409
Elektronik	2072	P	3	4	IT	1	Buck	**IT2A / IT2B**	
Projekt Elektronik	2082	P	2	4	IT	4	Buck		4h F1.409
Informatik 2	2092	P	3	4	IT	3	Warendorf	**IT2D / IT2D**	
Projekt Informatik 2	2102	P	2	4	IT	4	Warendorf		4h PC-Pool
Computerarchitektur 1	2112	P	4	4	IT	1	Keller		
Labor Computerarchitektur 1	2122	P	1	4	IT	2	LB N.N.		4h F1.306
		Summe Woh:	30	50		33			

*** Tatsächlich im Stundenplan einzugebende Stundenzahl, durch 2 teilbar!

HOCHSCHULE ESSLINGEN **IT2C**

Lehrangebot der Fakultät Informationstechnik **BA-StuPO**

Fakultät: IT **Studiengang:** Informationstechnik **Semester:** IT2C SS10 **Datum:** 29.11.09

Fach	Fach-Nr.	P W Z	Stud. - Woh lt. Stundenplan	***	Serv vom FB	Doz. Woh	Dozent	Gekoppelt mit Semester Fach-Nr.	Bemerkungen
Mathematik 2	2012	P	4	4	G	4	Service		
Labor Mathematik 2	2022	P	1	2	G	2	Service		
Mathematik 2 Zusatzübungen	Ohne	Z	0	2	G	0	Service		
Physik 2	2032	P	4	4	G	4	Service		
Labor Physik 2	2042	P	1	4	G	3	Service		
Physik 2 Zusatzübungen	Ohne	Z	0	2	G	2	Service		aus Studiengehühren G
Elektrotechnik 2	2052	P	4	4	IT	4	Doster	**IT2C / IT2D**	
Labor Elektrotechnik 2	2062	P	1	4	IT	2	LB Schulz		**Mo 5.+6. Stunde** 4h F1.409
Elektronik	2072	P	3	4	G	4	Coenning	**IT2C / IT2D**	Seminarraum mit Beamer
Projekt Elektronik	2082	P	2	4	G	4	Coenning		4h F1.409
Informatik 2	2092	P	3	4	IT	3	Beck		Hörsaal mit Beamer
Projekt Informatik 2	2102	P	2	4	IT	4	Beck		4h PC-Pool
Computerarchitektur 1	2112	P	4	4	IT	4	Lindermeir		Hörsaal mit Beamer
Labor Computerarchitektur 1	2122	P	1	4	IT	2	Lindermeir		4h F1.306
		Summe Woh:	30	50		42			

*** Tatsächlich im Stundenplan einzugebende Stundenzahl, durch 2 teilbar!

HOCHSCHULE ESSLINGEN **IT2D**

Lehrangebot der Fakultät Informationstechnik **BA-StuPO**

Fakultät: IT **Studiengang:** ______ **Semester:** IT2D **SS10** ______ **Datum:** 29.11.09

Fach	Fach-Nr.	P W Z	Stud. - Woh lt. Stun-denplan	***	Serv vom FB	Doz. Woh	Dozent	Gekoppelt mit Semester Fach-Nr.	Bemerkungen
Mathematik 2	2012	P	4	4	G	4	Service		
Labor Mathematik 2	2022	P	1	2	G	2	Service		
Mathematik 2 Zusatzübungen	Ohne	Z	0	2	G	0	Service		
Physik 2	2032	P	4	4	G	4	Service		
Labor Physik 2	2042	P	1	4	G	3	Service		
Physik 2 Zusatzübungen	Ohne	Z	0	2	G	2	Service		Studiengehühren G
Elektrotechnik 2	2052	P	4	4	IT	1	Doster	**IT2C / IT2D**	
Labor Elektrotechnik 2	2062	P	1	4	IT	2	2h LB Schulz		**Di 5.+6. Stunde** 4h F1.409
Elektronik	2072	P	3	4	G	1	Coenning	**IT2C / IT2D**	
Projekt Elektronik	2082	P	2	4	G	4	Coenning		4h F1.409
Informatik 2	2092	P	3	4	IT	1	Warendorf	**IT2B / IT2D**	
Projekt Informatik 2	2102	P	2	4	IT	4	Warendorf		4h PC-Pool
Computerarchitektur 1	2112	P	4	4	IT	1	Lindermeir		
Labor Computerarchitektur 1	2122	P	1	4	IT	2	Lindermeir		4h F1.304a
Summe Woh:			30	50		31			

*** Tatsächlich im Stundenplan einzugebende Stundenzahl, durch 2 teilbar!

HOCHSCHULE ESSLINGEN **IT3A**

Lehrangebot der Fakultät Informationstechnik **BA-StuPO**

Fakultät: **IT** **Studiengang:** **Informationstechnik** **Semester:** **IT3A** **SS10** **Datum:** **29.11.09**

Fach	Fach-Nr.	P W Z	Stud. - Woh lt. Stun-denplan	***	Serv vom FB	Doz. Woh	Dozent	Gekoppelt mit Semester Fach-Nr.	Bemerkungen
Mathematik 3	3002	P	5	6	G	5	Service		
Informatik 3	3012	P	3	4	IT	3	Kappen		
Labor Informatik 3	3022	P	2	4	IT	4	Kappen		4h PC-Pool
Betriebssysteme	3072	P	4	4	IT	4	LB Seiffert		
Labor Betriebssysteme	3082	P	1	4	IT	2	Reber		4h F1.305
Signale und Systeme	3052	P	4	4	IT	4	Doster		
Labor Signale und Systeme	3062	P	1	4	IT	2	Doster		4h F1.403
Computerarchitektur 2	3032	P	4	4	IT	4	Lindermeir		
Labor Computerarchitektur 2	3042	P	1	4	IT	2	Lindermeir		4h F1.304a
Datenbanken 1	3092	P	4	4	IT	4	Nonnast	**IT3A/ IT3B**	4h Seminarraum und F1.410 Do 1.+2.Stunde
Datenbanken 1	3092	P	1	2	IT	4	Schoop	**IT3A /IT3B**	4h Seminarraum, und F1.410 und PC-Pool Mi 1.+2. Stunde
Datenbanken 1	3092	P	1	2	IT	4	Nonnast LB Reiser Warendorf, Eve		4h Seminarraum und F1.410 Do 5.+6.Stunde
Summe Woh:			31	46		42			

*** Tatsächlich im Stundenplan einzugebende Stundenzahl, durch 2 teilbar!

HOCHSCHULE ESSLINGEN **IT3B**

Lehrangebot der Fakultät Informationstechnik **BA-StuPO**

Fakultät: IT **Studiengang:** Informationstechnik **Semester:** IT3B **SS10** **Datum:** 29.11.09

Fach	Fach-Nr.	P W Z	Stud. - Woh lt. Stundenplan	Stud. - Woh ***	Serv vom FB	Doz. Woh	Dozent	Gekoppelt mit Semester Fach-Nr.	Bemerkungen
Mathematik 3	3002	P	5	6	G	5	Service		
Informatik 3	3012	P	3	4	IT	3	Rößler		
Labor Informatik 3	3022	P	2	4	IT	4	Rößler		4h PC-Pool
Betriebssysteme	3072	P	4	4	IT	0	LB Seiffert		
Labor Betriebssysteme	3082	P	1	4	IT	2	Reber		4h F1.305
Signale und Systeme	3052	P	4	4	IT	4	Höfer		
Labor Signale und Systeme	3062	P	1	4	IT	2	Höfer		4h F1.403
Computerarchitektur 2	3032	P	4	4	IT	4	Keller		
Labor Computerarchitektur 2	3042	P	1	4	IT	2	Keller		4h F1.304a
Datenbanken 1	3092	P	4	4	IT	0	Nonnast	**IT3A/ IT3B**	4h Seminarraum und F1.410 Do 1.+1.Stunde
Datenbanken 1	3092	P	1	2	IT	0	Schoop	**IT3A /IT3B**	4h Seminarraum, und F1.410 und PC-Pool Mi 1.+2. Stunde
Datenbanken 1	3092	P	1	2	IT	0	Nonnast LB Reiser Warendorf, Eve		4h Seminarraum und F1.410 Fr 1.+2.Stunde
	Summe Woh:		31	46		38			

*** Tatsächlich im Stundenplan einzugebende Stundenzahl, durch 2 teilbar!

HOCHSCHULE ESSLINGEN **IT4A**

Lehrangebot der Fakultät Informationstechnik **BA-StuPO**

Fakultät: **IT** **Studiengang:** ______ **Semester:** **IT4A** **SS10** ______ **Datum:** **29.11.09**

Fach	Fach-Nr.	P W Z	Stud. - Woh lt. Stun-denplan	Stud. - Woh ***	Serv vom FB	Doz. Woh	Dozent	gekoppelt mit Semester Fach-Nr.	Bemerkungen
Betriebswirtschaft	4001	P	4	4	IT	4	LB Bühler		Seminarraum mit Beamer Do 6.+7. Stunde
Projektmanagement	4011	P	1	0	IT	1	LB Stumpp	**IT4A / IT4B**	Blockseminar am Ende des Semesters
Computerarchitektur 3	4021	P	4	4	IT	4	Zimmermann		
Labor Computerarchitektur 3	4031	P	1	4	IT	2	1h LB Haag 1h LB Redmer		4h F1.307
Systemtechnik 1	4041	P	4	4	IT	4	Zimmermann		
Labor Systemtechnik 1	4051	P	1	4	IT	2	Zimmermann		4h F1.307
Echtzeitsysteme	4061	P	4	4	IT	4	Friedrich	**IT4A / IT4B**	
Labor Echtzeitsysteme	4071	P	1	4	IT	2	Friedrich		4h F1.301
Rechnernetze 1	4081	P	4	4	IT	4	Wiese		
Labor Rechnernetze 1	4091	P	1	4	IT	2	Wiese		4h F1.401
Objektorientierte Systeme 1	4101	P	3	2	IT	3	Warendorf		
Lab. Objektorientierte Systeme 1	4111	P	2	4	IT	4	Warendorf		4h PC-Pool
Ingenieurmethodiken 2		P	4	4	IT	1	Doster div. LBs	**IT4A / IT4B**	Fr 5.+6. Stunde
		Summe Woh:	34	46		37			

*** Tatsächlich im Stundenplan einzugebende Stundenzahl, durch 2 teilbar!

HOCHSCHULE ESSLINGEN **IT4B**

Lehrangebot der Fakultät Informationstechnik **BA-StuPO**

Fakultät: **IT** **Studiengang:** **Semester:** **IT4B** **SS10** **Datum:** **29.11.09**

Fach	Fach-Nr.	P W Z	Stud. - Woh lt. Stun-denplan	***	Serv vom FB	Doz. Woh	Dozent	gekoppelt mit Semester Fach-Nr.	Bemerkungen
Betriebswirtschaft	4001	P	4	4	IT	4	LB Feil		Seminarraum mit Beamer Mi 6.+7. Stunde
Projektmanagement	4011	P	1	0	IT	1	LB Stumpp	**IT4A IT4B**	Blockseminar am Ende des Semesters
Computerarchitektur 3	4021	P	4	4	IT	1	Zimmermann		
Labor Computerarchitektur 3	4031	P	1	4	IT	2	Zimmermann		4h F1.307
Systemtechnik 1	4041	P	4	4	IT	4	Kull		
Labor Systemtechnik 1	4051	P	1	4	IT	2	Kull		4h F1.307
Echtzeitsysteme	4061	P	4	4	IT	1	Friedrich	**IT4A IT4B**	
Labor Echtzeitsysteme	4071	P	1	4	IT	2	Friedrich		4h F1.301
Rechnernetze 1	4081	P	4	4	IT	4	Zieher		
Labor Rechnernetze 1	4091	P	1	4	IT	2	Zieher		4h F1.401
Objektorientierte Systeme 1	4101	P	3	4	IT	3	Dausmann		
Lab. Objektorientierte Systeme 1	4111	P	2	2	IT	4	2h LB Sari 2h LB Hiller		4h PC-Pool
Ingenieurmethodiken 2		P	4	4	IT	1	Doster div. LBs	**IT4A IT4B**	Fr 5.+6. Stunde
	Summe Woh:		34	46		31			

*** Tatsächlich im Stundenplan einzugebende Stundenzahl, durch 2 teilbar!

HOCHSCHULE ESSLINGEN
Lehrangebot der Fakultät Informationstechnik

IT5 A/B
BA-StuPO

Fakultät: IT **Studiengang:** Kommunikationstechnik **Semester:** IT5A SS10 **Datum:** 29.11.09

PRAXISSEMESTER

Fach	Fach-Nr.	P W Z	Stud. - Woh lt. Stun-denplan	***	Serv vom FB	Doz. Woh	Dozent	gekoppelt mit Semester Fach-Nr.	Bemerkungen
Praxisbetreuung		P		0	IT	1	Beck		
Praxisbetreuung		P		0	IT	1	Dausmann		
Praxisbetreuung		P		0	IT	1	Keller		
Praxisbetreuung		P		0	IT	1	Rößler		
		Summe Woh:	0	0		4			

*** Tatsächlich im Stundenplan einzugebende Stundenzahl, durch 2 teilbar!

HOCHSCHULE ESSLINGEN

Lehrangebot der Fakultät Informationstechnik **Bachelor-StuPO KTB6**

Fakultät: IT **Studiengang:** Kommunikationstechnik **Semester:** KTB6 SS10 ______ **Datum:** 29.11.09

Fach	Fach-Nr.	P W Z	Stud. - Woh lt. Stun-denplan	***	Serv vom FB	Doz. Woh	Dozent	gekoppelt mit Semester Fach-Nr.	Bemerkungen
Festnetze	6101	P	4	4	IT	4	Melcher		
Labor Festnetze	6111	P	1	4	IT	1	Melcher	**Labor Funknetze**	4h F1.403
Funknetze	6091	P	4	4	IT	4	Melcher / Buck		
Labor Funknetze	6081	P	1	4	IT	1	Melcher	**Labor Festnetze**	4h F1.403
Digitale Signalverarbeitung	6031	P	4	4	IT	4	Höfer	**KTB6 / TIB6**	
Labor Digitale Signalverarbeitung	6021	P	1	4	IT	1	Höfer		F1.403
Übertragungsmedien	6071	P	4	4	IT	4	Buck		Seminarraum
Labor Übertragungsmedien	6061	P	1	4	IT	1	Buck		F1.405
Rechnernetze 2	6051	P	4	4	IT GS	4	2h Zieher 2h Schoop		
Labor Rechnernetze 2	6041	P	1	4	IT GS	1	0,5h Zieher 0,5h Schoop		F1.401
Summe Woh:			25	40		25			

*** Tatsächlich im Stundenplan einzugebende Stundenzahl, durch 2 teilbar!

HOCHSCHULE ESSLINGEN

Lehrangebot der Fakultät Informationstechnik **Bachelor-StuPO SWM6**

Fakultät: IT Studiengang: Softwaretechnik und Medieninf. Semester: SWM6 SS10 Datum: 29.11.09

SCHWERPUNKT Medientechnik

Pflichtfächer

Fach	Fach-Nr.	P W Z	Stud. - Woh lt. Stundenplan	***	Serv vom FB	Doz. Woh	Dozent	gekoppelt mit Semester Fach-Nr.	Bemerkungen
Virtuelle Realität	6071	P	4	4	IT	4	2h Schmidt 2h Rößler		
Labor Virtuelle Realität	6061	P	1	4	IT	1	Schmidt		4h F1.303
Digitale Medien	6051	P	4	4	IT	4	Schmidt		
Labor Digitale Medien	6041	P	1	4	IT	1	Schmidt		4h F1.303
Grafische Benutzungsoberflächen	SWB_6091	P	4	4	IT	0	Rößler		
Labor Grafische Benutzungsoberflächen	SWB_6081	P	1	4	IT	1	Rößler		4h F1.303
Datenbanken 2	6031	P	3	4	IT	0	Nonnast		Seminarraum und F1.410
Projekt Datenbanken 2	6021	P	2	2	IT	0	Nonnast		Seminarraum und F1.410
Interaktive Systeme	6091	P	4	4	IT	4	Beck		Seminarraum
Labor Interaktive Systeme	6081	P	1	4	IT	1	Beck		4h F1.303
		Summe Woh:	25	38		16			

HOCHSCHULE ESSLINGEN

Lehrangebot der Fakultät Informationstechnik **Bachelor-StuPO SWT6**

Fakultät: **IT** **Studiengang:** **Softwaretechnik und Medieninf.** **Semester:** **SWT6** **SS10** **Datum:** **29.11.09**

SCHWERPUNKT Softwaretechnik

Pflichtfächer

Fach	Fach-Nr.	P W Z	Stud. - Woh lt. Stundenplan	Stud. - Woh ***	Serv vom FB	Doz. Woh	Dozent	gekoppelt mit Semester Fach-Nr.	Bemerkungen
Objektorientierte Systeme 2	6051	P	4	4	IT	4	Dausmann		Seminarraum
Labor Objektorientierte Systeme 2	6041	P	1	4	IT	2	Dausmann		4h PC-Pool
Softwarearchitekturen	6071	P	4	4	IT	4	Goll		Sem.raum F1.402
Labor Softwarearchitekturen	6061	P	1	4	IT	2	LB Müller-Hofmann		4h Sem.raum F1.402 PC-Pool
Grafische Benutzungsoberflächen	6091	P	4	4	IT	4	Rößler		Seminarraum
Labor Grafische Benutzungsoberflächen	6081	P	1	4	IT	2	Rößler		F1.303
Datenbanken 2	6021	P	3	4	IT	3	Nonnast	**SWM6 SWT6**	4h Sem.raum F1.410
Projekt Datenbanken 2	6031	P	2	2	IT	2	Nonnast	**SWM6 SWT6**	2h Sem.raum F1.410
Rechnerbetrieb	6111	P	4	4	IT	4	Väterlein		Seminarraum
Labor Rechnerbetrieb	6101	P	1	4	IT	2	Väterlein		F1.305
		Summe Woh:	25	38		29			

*** Tatsächlich im Stundenplan einzugebende Stundenzahl, durch 2 teilbar!

HOCHSCHULE ESSLINGEN

Lehrangebot der Fakultät Informationstechnik **Bachelor-StuPO TIB6**

Fakultät: **IT** **Studiengang:** **Technische Informatik** **Semester:** **TIB6** **SS10** **Datum:** **29.11.09**

Fach	Fach-Nr.	P W Z	Stud. - Woh lt. Stundenplan	Stud. - Woh ***	Serv vom FB	Doz. Woh	Dozent	gekoppelt mit Semester Fach-Nr.	Bemerkungen
Systemtechnik 2	6051	P	4	4	IT	4	Kull		Seminarraum
Labor Systemtechnik 2	6041	P	1	4	IT	1	Kull		F1.307
Signalverarbeitung	6031	P	4	4	IT	0	Höfer	**KTB6, TIB6**	Seminarraum
Labor Signalverarbeitung	6021	P	1	4	IT	0	Höfer	**KTB6 / TIB6**	
Bussysteme	6111	P	4	4	IT	4	Wiese		F1.403
Labor Bussysteme	6101	P	1	4	IT	2	Wiese		F1.304a
Embedded Systems	6091	P	4	4	IT	4	Kappen		Seminarraum
Labor Embedded Systems	6081	P	1	4	IT	2	1h Linkohr 1h Klenk		F1.304a
Maschinelles Sehen	6071	P	4	4	IT	4	Malz		Seminarraum
Labor Maschinelles Sehen	6061	P	1	4	IT	1	Malz		PC-Pool
Summe Woh:			25	40		22			

*** Tatsächlich im Stundenplan einzugebende Stundenzahl, durch 2 teilbar!

HOCHSCHULE ESSLINGEN

Lehrangebot der Fakultät Informationstechnik **Bachleor-StuPO KTB7**

Fakultät: **IT** **Studiengang:** **Softwaretechnik und Medieninf.** **Semester:** **SWB7** **SS10** **Datum:** **29.11.09**

Fach	Fach-Nr.	P W Z	Stud. - Woh lt. Stundenplan	***	Serv vom FB	Doz. Woh	Dozent	gekoppelt mit Semester Fach-Nr.	Bemerkungen
IT-Sicherheit		W	2	2	GS	0	Schoop	**KBT SWB7 TIB7**	
Kryptologie		W	2	2	IT	0	Schmidt	**KBT SWB7 TIB7**	
Fernsehtechnik		W	2	2	IT	0	LB Schulz	**KBT SWB7 TIB7**	
Kfz-Steuergeräte-Design		W	2	2	IT	0	Kull	**KBT SWB7 TIB7**	
Paralleles Rechnen		W	2	2	IT	0	Väterlein	**KBT SWB7 TIB7**	
Numerische Methoden		W	2	2	G	0	Service G	**KBT SWB7 TIB7**	
		Summe Woh:	12	12		0			

*** Tatsächlich im Stundenplan einzugebende Stundenzahl, durch 2 teilbar!

HOCHSCHULE ESSLINGEN

Lehrangebot der Fakultät Informationstechnik **Bachleor-StuPO SWB7**

Fakultät: **IT** **Studiengang:** **Softwaretechnik und Medieninf.** **Semester:** **SWB7** **SS10** **Datum:** **29.11.09**

Fach	Fach-Nr.	P W Z	Stud. - Woh lt. Stundenplan	Stud. - Woh ***	Serv vom FB	Doz. Woh	Dozent	gekoppelt mit Semester Fach-Nr.	Bemerkungen
IT-Sicherheit		W	2	2	GS	2	Schoop	**KBT SWB7 TIB7**	
Kryptologie		W	2	2	IT	2	Schmidt	**KBT SWB7 TIB7**	
Fernsehtechnik		W	2	2	IT	2	LB Schulz	**KBT SWB7 TIB7**	
Kfz-Steuergeräte-Design		W	2	2	IT	2	Kull	**KBT SWB7 TIB7**	
Paralleles Rechnen		W	2	2	IT	2	Väterlein	**KBT SWB7 TIB7**	
Numerische Methoden		W	2	2	G	2	Service G	**KBT SWB7 TIB7**	
Summe Woh:			12	12		12			

*** Tatsächlich im Stundenplan einzugebende Stundenzahl, durch 2 teilbar!

HOCHSCHULE ESSLINGEN

Lehrangebot der Fakultät Informationstechnik

Bachleor-StuPO TIB7

Fakultät: IT **Studiengang:** Softwaretechnik und Medieninf. **Semester:** TIB7 SS10 **Datum:** 29.11.09

Fach	Fach-Nr.	P W Z	Stud. - Woh lt. Stundenplan	Stud. - Woh ***	Serv vom FB	Doz. Woh	Dozent	gekoppelt mit Semester Fach-Nr.	Bemerkungen
IT-Sicherheit		W	2	2	GS	0	Schoop	**KBT SWB7 TIB7**	
Kryptologie		W	2	2	IT	0	Schmidt	**KBT SWB7 TIB7**	
Fernsehtechnik		W	2	2	IT	0	LB Schulz	**KBT SWB7 TIB7**	
Kfz-Steuergeräte-Design		W	2	2	IT	0	Kull	**KBT SWB7 TIB7**	
Paralleles Rechnen		W	2	2	IT	0	Väterlein	**KBT SWB7 TIB7**	
Numerische Methoden		W	2	2	G	0	Service G	**KBT SWB7 TIB7**	
Summe Woh:			12	12		0			

*** Tatsächlich im Stundenplan einzugebende Stundenzahl, durch 2 teilbar!

POG für Studiengangleiter

Studiengang: KTB Stand WS 2008/09 20.09.2008 Kommunikationstechnik BACHELOR

pvers	vert	pnr	pfsem	pdtxt	part	pdauer	ppflich	pminsem	phoesem	angebote	pform	Credits	psws	nachname	partngb	Gruppe
1		1011	1	Mathematik 1 Mathematics 1	L	150	P	1	4	J	KL	10	10	Ulmet	C	A
1		1011	1	Mathematik 1 Mathematics 1	L	150	P	1	4	J	KL	10	10	Plappert	C	B
1		1011	1	Mathematik 1 Mathematics 1	L	150	P	1	4	J	KL	10	10	Sigg	C	C
1		1021	1	Physik 1 Physics 1	L	90	P	1	4	J	KL	5	5	Coenning	C	A
1		1021	1	Physik 1 Physics 1	L	90	P	1	4	J	KL	5	5	Strobel	C	B
1		1021	1	Physik 1 Physics 1	L	90	P	1	4	J	KL	5	5	Coening	C	C
1		1031	1	Elektrotechnik 1 Electrical Engineering 1	L	90	P	1	4	J	KL	4	4	Gündner	C	A
1		1031	1	Elektrotechnik 1 Electrical Engineering 1	L	90	P	1	4	J	KL	4	4	Malz	C	B
1		1031	1	Elektrotechnik 1 Electrical Engineering 1	L	90	P	1	4	J	KL	4	4	Malz	C	C
1		1041	1	Labor Elektrotechnik 1 Lab Electrical Engineering 1	S		P	1	4	J	BE	1	1	LB Schulz	U	A
1		1041	1	Labor Elektrotechnik 1 Lab Electrical Engineering 1	S		P	1	4	J	BE	1	1	LB Haußer	U	B
1		1041	1	Labor Elektrotechnik 1 Lab Electrical Engineering 1	S		P	1	4	J	BE	1	1	LB Kähler	U	C
1		1051	1	Informatik 1 Computer Science 1	L	90	P	1	4	J	KL	3	3	Kappen	C	A
1		1051	1	Informatik 1 Computer Science 1	L	90	P	1	4	J	KL	3	3	Warendorf	C	B
1		1051	1	Informatik 1 Lab Computer Science 1	L	90	P	1	4	J	KL	3	3	Weber	C	C
1		1061	1	Labor Informatik 1 Lab Computer Science 1	S		P	1	4	J	BE	2	2	LB Müller, Martin	U	A
1		1061	1	Labor Informatik 1 Lab Computer Science 1	S		P	1	4	J	BE	2	2	Warendorf	U	B
1		1061	1	Labor Informatik 1 Lab Computer Science 1	S		P	1	4	J	BE	2	2	Weber	U	C
1		1071	1	Technisches Englisch Technical English	L	60	P	1	4	J	KL	2	2	Warendorf, Eve	C	A
1		1071	1	Technisches Englisch Technical English	L	60	P	1	4	J	KL	2	2	Warendorf, Eve	C	B
1		1071	1	Technisches Englisch Technical English	L	60	P	1	4	J	KL	2	2	Warendorf, Eve	C	C

POG für Studiengangleiter

Studiengang: KTB Stand WS 2008/09 20.09.2008 Kommunikationstechnik BACHELOR

pvers	vert	pnr	pfsem	pdtxt	part	pdauer	ppflich	pminsem	phoesem	angebote	pform	Credits	psws	nachname	partngb	Gruppe
1		1081	1	Persönlichkeitsentwicklung Personality Development	S		P	1	4	J	RE	3	3	Väterlein	U	A
1		1081	1	Persönlichkeitsentwicklung Personality Development	S		P	1	4	J	RE	3	3	Väterlein	U	B
1		1081	1	Persönlichkeitsentwicklung Personality Development	S		P	1	4	J	RE	3	3	Väterlein	U	C

POG für Studiengangleiter

Studiengang: KTB Stand WS 2008/09 20.09.2008 Kommunikationstechnik BACHELOR

pvers	vert	pnr	pfsem	pdtxt	part	pdauer	ppflich	pminsem	phoesem	angebote	pform	Credits	psws	nachname	partngb	Gruppe
1		2011	2	Mathematik 2 Mathematics 2	L	90	P	1	4	J	KL	4	4	Ulmet	C	A
1		2011	2	Mathematik 2 Mathematics 2	L	90	P	1	4	J	KL	4	4	Koch	C	B
1		2011	2	Mathematik 2 Mathematics 2	L	90	P	1	4	J	KL	4	4		C	
1		2021	2	Labor Mathematik 2 Lab Mathematics 2	S		P	1	4	J	TE	1	1	Ulmet	U	A
1		2021	2	Labor Mathematik 2 Lab Mathematics 2	S		P	1	4	J	TE	1	1	Koch	U	B
1		2021	2	Labor Mathematik 2 Lab Mathematics 2	S		P	1	4	J	TE	1	1		U	
1		2031	2	Physik 2 Physics 2	L	90	P	1	4	J	KL	4	4	Martin	C	A
1		2031	2	Physik 2 Physics 2	L	90	P	1	4	J	KL	4	4	Coenning	C	B
1		2031	2	Physik 2 Physics 2	L	90	P	1	4	J	KL	4	4		C	
1		2041	2	Labor Physik 2 Lab Physics 2	S		P	1	4	J	BE	1	1	Strobel	U	A
1		2041	2	Labor Physik 2 Lab Physics 2	S		P	1	4	J	BE	1	1	Coenning	U	B
1		2041	2	Labor Physik 2 Lab Physics 2	S		P	1	4	J	BE	1	1		U	
1		2051	2	Elektrotechnik 2 Electrical Engineering 2	L	90	P	1	4	J	KL	4	4	Höfer	C	A
1		2051	2	Elektrotechnik 2 Electrical Engineering 2	L	90	P	1	4	J	KL	4	4	Höfer	C	B
1		2051	2	Elektrotechnik 2 Electrical Engineering 2	L	90	P	1	4	J	KL	4	4		C	
1		2061	2	Labor Elektrotechnik 2 Lab Electrical Engineering 2	S		P	1	4	J	BE	1	1	LB Hehl	U	A
1		2061	2	Labor Elektrotechnik 2 Lab Electrical Engineering 2	S		P	1	4	J	BE	1	1	LB Rauschnabel	U	B
1		2061	2	Labor Elektrotechnik 2 Lab Electrical Engineering 2	S		P	1	4	J	BE	1	1		U	
1		2071	2	Elektronik Electronics	L	90	P	1	4	J	KL	3	3	Gündner	C	A
1		2071	2	Elektronik Electronics	L	90	P	1	4	J	KL	3	3	Buck	C	B
1		2071	2	Elektronik Electronics	L	90	P	1	4	J	KL	3	3		C	

POG für Studiengangleiter

Studiengang: KTB Stand WS 2008/09 20.09.2008 Kommunikationstechnik BACHELOR

pvers	vert	pnr	pfsem	pdtxt	part	pdauer	ppflich	pminsem	phoesem	angebote	pform	Credits	psws	nachname	partngb	Gruppe
1		2081	2	Projekt Elektronik Project Electronics	S		P	1	4	J	PA	2	2	Gündner	U	A
1		2081	2	Projekt Elektronik Project Electronics	S		P	1	4	J	PA	2	2	Buck	U	B
1		2081	2	Projekt Elektronik Project Electronics	S		P	1	4	J	PA	2	2		U	
1		2091	2	Informatik 2 Computer Science 2	L	90	P	1	4	J	KL	3	3	Beck	C	A
1		2091	2	Informatik 2 Computer Science 2	L	90	P	1	4	J	KL	3	3	Warendorf	C	B
1		2091	2	Informatik 2 Computer Science 2	L	90	P	1	4	J	KL	3	3		C	
1		2101	2	Projekt Informatik 2 Project Computer Science 2	S		P	1	4	J	PA	2	2	Beck	U	A
1		2101	2	Projekt Informatik 2 Project Computer Science 2	S		P	1	4	J	PA	2	2	Warendorf	U	B
1		2101	2	Projekt Informatik 2 Project Computer Science 2	S		P	1	4	J	PA	2	2		U	
1		2111	2	Computerarchitektur 1 Computer Architecture 1	L	90	P	1	4	J	KL	4	4	Lindermeir	C	A
1		2111	2	Computerarchitektur 1 Computer Architecture 1	L	90	P	1	4	J	KL	4	4	Keller	C	B
1		2111	2	Computerarchitektur 1 Computer Architecture 1	L	90	P	1	4	J	KL	4	4		C	
1		2121	2	Labor Computerarchitektur 1 Lab Computer Architecture 1	S		P	1	4	J	BE	1	1	Lindermeir	U	A
1		2121	2	Labor Computerarchitektur 1 Lab Computer Architecture 1	S		P	1	4	J	BE	1	1	Keller	U	B
1		2121	2	Labor Computerarchitektur 1 Lab Computer Architecture 1	S		P	1	4	J	BE	1	1		U	

POG für Studiengangleiter

Studiengang: KTB Stand WS 2008/09 20.09.2008 Kommunikationstechnik BACHELOR

pvers	vert	pnr	pfsem	pdtxt	part	pdauer	ppflich	pminsem	phoesem	angebote	pform	Credits	psws	nachname	partngb	Gruppe
1		3001	3	Mathematik 3 Mathematics 3	L	90	P	3	10	J	KL	5	5	Sigg	C	A
1		3001	3	Mathematik 3 Mathematics 3	L	90	P	3	10	J	KL	5	5	Glatz	C	B
1		3011	3	Informatik 3 Computer Science 3	L	90	P	3	10	J	KL	3	3	Schoop	C	A
1		3011	3	Informatik 3 Computer Science 3	L	90	P	3	10	J	KL	3	3	Rößler	C	B
1		3021	3	Labor Informatik 3 Lab Computer Science 3	S		P	3	10	J	BE	2	2	Schoop	U	A
1		3021	3	Labor Informatik 3 Lab Computer Science 3	S		P	3	10	J	BE	2	2	Rößler	U	B
1		3031	3	Computerarchitektur 2 Computer Architectur 2	L	90	P	3	10	J	KL	4	4	Lindermeir	C	A
1		3031	3	Computerarchitektur 2 Computer Architectur 2	L	90	P	3	10	J	KL	4	4	Keller	C	B
1		3041	3	Labor Computerarchitektur 2 Labor Computer Architecture 2	S		P	3	10	J	BE	1	1	Lindermeir	U	A
1		3041	3	Labor Computerarchitektur 2 Labor Computer Architecture 2	S		P	3	10	J	BE	1	1	Keller	U	B
1		3051	3	Signale und Systeme Signals and Systems	L	90	P	3	10	J	KL	4	4	Doster	C	A
1		3051	3	Signale und Systeme Signals and Systems	L	90	P	3	10	J	KL	4	4	Doster	C	B
1		3061	3	Labor Signale und Systeme Lab Signals and Systems	S		P	3	10	J	BE	1	1	Doster	U	A
1		3061	3	Labor Signale und Systeme Lab Signals and Systems	S		P	3	10	J	BE	1	1	Doster	U	B
1		3071	3	Betriebssyteme Operating Systems	L	90	P	3	10	J	KL	4	4	Väterlein	C	A
1		3071	3	Betriebssyteme Operating Systems	L	90	P	3	10	J	KL	4	4	Weber	C	B
1		3081	3	Labor Betriebssyteme Lab Operating Systems	S		P	3	10	J	BE	1	1	Väterlein	U	A
1		3081	3	Labor Betriebssyteme Lab Operating Systems	S		P	3	10	J	BE	1	1	Weber	U	B

POG für Studiengangleiter

Studiengang: KTB Stand WS 2008/09 20.09.2008 Kommunikationstechnik BACHELOR

pvers	vert	pnr	pfsem	pdtxt	part	pdauer	ppflich	pminsem	phoesem	angebote	pform	Credits	psws	nachname	partngb	Gruppe
1		3091	3	Datenbanken 1 Database Systems 1	L	90	P	3	10	J	KL	4	4	Nonnast	C	A
1		3091	3	Datenbanken 1 Database Systems 1	L	90	P	3	10	J	KL	4	4	Schoop	C	B
1		3101	3	Labor Datenbanken 1 Lab Database Systems 1	S		P	3	10	J	BE	1	1	Nonnast	U	A
1		3101	3	Labor Datenbanken 1 Lab Database Systems 1	S		P	3	10	J	BE	1	1	Schoop	U	B

POG für Studiengangleiter

Studiengang: KTB Stand WS 2008/09 20.09.2008 Kommunikationstechnik BACHELOR

pvers	vert	pnr	pfsem	pdtxt	part	pdauer	ppflich	pminsem	phoesem	angebote	pform	Credits	psws	nachname	partngb	Gruppe
1		4001	4	Betriebswirtschaft Business Economics	L	90	P	3	10	J	KL	4	4	LB Feil	U	A
1		4001	4	Betriebswirtschaft Business Economics	L	90	P	3	10	J	KL	4	4	LB Feil	U	B
1		4011	4	Projektmanagement Project Management	S		P	3	10	J	BE	1	1	LB Stumpp	C	A
1		4011	4	Projektmanagement Project Management	S		P	3	10	J	BE	1	1	LB Stumpp	C	B
1		4021	4	Computerarchitektur 3 Computer Architekture 3	L	90	P	3	10	J	KL	4	4	Friedrich	U	A
1		4021	4	Computerarchitektur 3 Computer Architekture 3	L	90	P	3	10	J	KL	4	4	Friedrich	U	B
1		4031	4	Labor Computerarchitektur 3 Lab Computer Architecture	S		P	3	10	J	BE	1	1	Friedrich	C	A
1		4031	4	Labor Computerarchitektur 3 Lab Computer Architecture	S		P	3	10	J	BE	1	1	Friedrich	C	B
1		4041	4	Systemtechnik 1 System Design 1	L	90	P	3	10	J	KL	4	4	Zimmermann	U	A
1		4041	4	Systemtechnik 1 System Design 1	L	90	P	3	10	J	KL	4	4	Zimmermann	U	B
1		4051	4	Labor Systemtechnik 1 Lab System Design 1	S		P	3	10	J	BE	1	1	Zimmermann	C	A
1		4051	4	Labor Systemtechnik 1 Lab System Design 1	S		P	3	10	J	BE	1	1	Zimmermann	C	B
1		4061	4	Echtzeitsysteme Real Time Systems	L	90	P	3	10	J	KL	4	4	Friedrich	U	A
1		4061	4	Echtzeitsysteme Real Time Systems	L	90	P	3	10	J	KL	4	4	Friedrich	U	B
1		4071	4	Labor Echtzeitsysteme Lab Real Time Systems	S		P	3	10	J	BE	1	1	Friedrich	C	A
1		4071	4	Labor Echtzeitsysteme Lab Real Time Systems	S		P	3	10	J	BE	1	1	Friedrich	C	B
1		4081	4	Rechnernetze 1 Computer Networks 1	L	90	P	3	10	J	KL	4	4	Wiese	U	A
1		4081	4	Rechnernetze 1 Computer Networks 1	L	90	P	3	10	J	KL	4	4	Wiese	U	B
1		4091	4	Labor Rechnernetze 1 Lab Computer Networks 1	S		P	3	10	J	BE	1	1	Wiese	C	A
1		4091	4	Labor Rechnernetze 1 Lab Computer Networks 1	S		P	3	10	J	BE	1	1	Wiese	C	B

POG für Studiengangleiter

Studiengang: KTB Stand WS 2008/09 20.09.2008 Kommunikationstechnik BACHELOR

pvers	vert	pnr	pfsem	pdtxt	part	pdauer	ppflich	pminsem	phoesem	angebote	pform	Credits	psws	nachname	partngb	Gruppe
1		4101	4	Objektorientierte Systeme 1 Object Oriented Systems 1	L	90	P	3	10	J	KL	3	3	Dausmann	U	A
1		4101	4	Objektorientierte Systeme 1 Object Oriented Systems 1	L	90	P	3	10	J	KL	3	3	Dausmann	U	B
1		4111	4	Labor Objektorientierte Systeme 1 Lab Object Oriented Systems 1	S		P	3	10	J	BE	2	2	Dausmann	C	A
1		4111	4	Labor Objektorientierte Systeme 1 Lab Object Oriented Systems 1	S		P	3	10	J	BE	2	2	Dausmann	C	B

POG für Studiengangleiter

Studiengang: KTB Stand WS 2008/09 20.09.2008 Kommunikationstechnik BACHELOR

pvers	vert	pnr	pfsem	pdtxt	part	pdauer	ppflich	pminsem	phoesem	angebote	pform	Credits	psws	nachname	partngb	Gruppe
1		5011	5	Ingenieurmethodiken 2 Methodologies for Engineers 2	L		P	5	10	J	TE	4	3	Gündner	C	
1		5021	5	Betriebliche Praxis Company Internship	S		P	5	10	J	BE	26	0	Doster	U	

POG für Studiengangleiter

Studiengang: KTB Stand WS 2008/09 20.09.2008 Kommunikationstechnik BACHELOR

pvers	vert	pnr	pfsem	pdtxt	part	pdauer	ppflich	pminsem	phoesem	angebote	pform	Credits	psws	nachname	partngb	Gruppe
1		6011	6	Studienarbeit Application Project	L		P	6	10	N	RE	5	5	it professoren	C	
1		6021	6	Labor Digitale Signalverarbeitung Lab Digital Signal Processing	S		P	3	10	N	BE	1	1	Höfer	U	
1		6031	6	Digitale Signalverarbeitung Digital Signal Processing	L	90	P	3	10	N	KL	4	4	Höfer	C	
1		6041	6	Labor Rechnernetze 2 Lab Computer Networks 2	S		P	3	10	N	BE	1	1	Schoop	U	
1		6051	6	Rechnernetze 2 Computer Networks 2	L	90	P	3	10	N	KL	4	4	Schoop	C	
1		6061	6	Labor Übertragunsmedien Lab Transmission Media	S		P	3	10	N	BE	1	1	Buck	U	
1		6071	6	Übertragunsmedien Transmission Media	L	90	P	3	10	N	KL	4	4	Buck	C	
1		6081	6	Labor Funknetze Lab Wireless Networks	S		P	3	10	N	BE	1	1	Melcher	U	
1		6091	6	Funknetze Wireless Networks	L	90	P	3	10	N	KL	4	4	Melcher	C	
1		6101	6	Labor Festnetze Lab Cable Networks	S		P	3	10	N	BE	1	1	Melcher	U	
1		6111	6	Festnetze Cable Networks	L	90	P	3	10	N	KL	4	4	Melcher	C	

POG für Studiengangleiter

Studiengang: KTB Stand WS 2008/09 20.09.2008 Kommunikationstechnik BACHELOR

pvers	vert	pnr	pfsem	pdtxt	part	pdauer	ppflich	pminsem	phoesem	angebote	pform	Credits	psws	nachname	partngb	Gruppe
1		7000	7	Bachelorarbeit und Kolloquium Bachelor Thesis and Defense	P		P	6	10	J	BA	15	15	it professoren	C	
1		7011	7	Wissenschaftliche Vertiefung Scientific Research	L		P	6	10	J	ML	9	9	it professoren	C	

POG für Studiengangleiter

Studiengang: KTB Stand WS 2008/09 20.09.2008 Kommunikationstechnik BACHELOR

pvers	vert	pnr	pfsem	pdtxt	part	pdauer	ppflich	pminsem	phoesem	angebote	pform	Credits	psws	nachname	partngb	Gruppe
				WAHLFÄCHER KTB												
1		8811	6	Algorithmen Algorithms	L		W	4	99	J	ML	2	2	Koch	C	
1		8821	6	Fundamentals of Optoelectronics Fundamentals of Optoelectronics	L		W	4	99	N	ML	2	2		C	
1		8831	6	IT-Sicherheit IT-Security	L		W	4	99	J	ML	2	2	Schoop	C	
1		8841	6	Fernsehtechnik Digital Video Broadcasting	L		W	4	99	N	ML	2	2		C	
1		8851	6	Technischer Vertrieb Sales and Marketing	L		W	4	99	N	ML	2	2		C	
1		8861	6	Programmieren in C# Introduction to C#	L		W	4	99	N	ML	2	2		C	
1		8871	6	Introduction to eCommerce Introduction to eCommerce	L		W	4	99	N	ML	2	2		C	
1		8881	6	Systemarchitekturen mit .NET Software Systems Architectures with .NET	L		W	4	99	J	ML	2	2	LB Erath	C	
				neue WAHLFÄCHER KTB												
1		neu	6	Vertrags- und Internet-Recht Contract and Media Law	L		W	4	99	J	ML	2	2	LB Schließ	C	
1		neu	6	Methoden der künstlichen Intelligenz Artificial Intelligence	L		W	4	99	J	ML	2	2	Weber	C	
1		neu	6	Einführung in CAD Introduction to CAD	L		W	4	99	J	ML	2	2	Koch	C	
1		neu	6	Kfz-Systeme Automotive Systems Engineering	L		W	4	99	J	ML	2	2	LB Marchthaler	C	
1		neu	6	Qualitätstechniken Quality Engineering	L		W	4	99	J	ML	2	2	LB Grübel	C	
1		neu	6	Optische Nachrichtenübertragung Optical Communications Engineering	L		W	4	99	J	ML	2	2	LB Khakzar	C	
1		neu	6	Optische Nachrichtenübertragung 2 Optical Communications Engineering 2	L		W	4	99	J	ML	2	2	LB Khakzar	C	

POG für Studiengangleiter

Studiengang: KTB Stand WS 2008/09 20.09.2008 Kommunikationstechnik BACHELOR

pvers	vert	pnr	pfsem	pdtxt	part	pdauer	ppflich	pminsem	phoesem	angebote	pform	Credits	psws	nachname	partngb	Gruppe
				neue Zusatzfächer TIB												
1		neu	6	Blockseminar XML Summer School Course XML	S		Z	4	99	J	TE			Dausmann	U	
1		neu	6	Blockseminar C# Summer School Course C#	S		Z	4	99	J	TE			Melcher	U	

1. Deputatsnachlässe für Funktionsträger nach § 6 (4) LVVO (Anzahl der gewährten Semesterwochenstunden (DWS)) SS 2010

Funktion	Fakultäten										
	AN	BW	FZ	G	GS	IT	MB	ME	SAGP	VU	WI	Σ
Prorektor							12	12	12			**36**
Dekan, Prodekan, Studiengangleiter1	21	22	28,5	28,5	24	28	27,5	33	43	16	16	**287,5**
Studiengangleiter Aufbaustudiengang	2											**2**
Σ	**23**	**22**	**28,5**	**28,5**	**24**	**28**	**39,5**	**45**	**55**	**16**	**16**	**325,5**

Basis ist die Stellenzahl laut Struktur- und Entwicklungsplan der Hochschule Esslingen

GS: 4 Dekan, 6+6+6 Kohlert, Schindler, Rösler, 2 Ehlers

Der Fakultät G sind die Studierenden der Ingenieurpädagogik zugeordnet. Der Maximalrahmen von 28 DWS nach LVVO wird nicht ausgeschöpft, da die Studierendenzahl klein ist.

2. Zusätzliche Deputatsnachlässe nach § 9 und § 10 LVVO (Anzahl der gewährten DWS)

Funktion	Fakultäten										
	AN	BW	FZ	G	GS	IT	MB	ME	SAGP	VU	WI	Σ
Forschungsprojekt	0	0	0	0	0	0	2	0	8	0	0	**10**
Sonstiges	0	9	4	4,5	0	0	0	8	10	0	0	**35,5**
Σ	**0**	**9**	**4**	**4,5**	**0**	**0**	**2**	**8**	**18**	**0**	**0**	**45,5**

Forschungsprojekte:
MB: Czarnetzki (2)
SAGP: Elsbernd (4), Möller (4)

Sonstiges:
FZ: Schreier (4) - finanziert durch CDHAW
ME: Minuth (8) - finanziert durch CDHAW

BW: Mathis (9) K
G: Hiesgen (4,5) K
SAGP: Simon-Hohm (10) K

3. Deputatsnachlässe nach § 8 (1) der LVVO (7 % des Gesamtumfangs der Lehrverpflichtungen) **SS 2010**
(Anzahl der gewährten DWS)

Ist-Wert:	(Stichtag 01.10.2009)		DWS	
	230,5 Professorenstellen	ergeben	4149	(inkl. HS 2012, Studiengebühren und Planstellen)
	1 Lektorstelle		16	
			4165	
abzüglich Deputatsnachlässe Funktionsträger			325,5	
			3839,5	
davon 7 % ergibt	**268,765** gerundet	269		

Funktion	Fakultäten											
	AN	**BW**	**FZ**	**G**	**GS**	**IT**	**MB**	**ME**	**SAGP**	**VU**	**WI**	**Σ**
Laborleitung	7	12	15	11	0	16	15	17	12	7	11	123
Praktikantenamt	2	4	5	0	0	5	5	6	6	2	4	39
Studieneignungstests												
Auslandsbeauftragte	1	2	2	1	0	2	2	2	3	1	2	18
Gesamt pauschal	**10**	**18**	**22**	**12**	**0**	**23**	**22**	**25**	**21**	**10**	**17**	**180**
zentral												
Gleichstellung	0	0	0	5	0	0	0	0	0	0	0	5
Schulwerbung/Schülerprojekt	1	0	0	1	0	1	3	3	0	0	1	10
Forschung	6	0	0	0	0	5	0	7	0	0	0	18
Leitung RZ	0	0	0	0	0	4	0	0	0	0	0	4
Neuentwicklungen RZ	0	0	0	0	0	2	0	0	0	0	0	2
Leitung ZMF	0	0	0	0	0	0	2	0	0	0	0	2
Leitung IAF	0	0	0	0	0	0	2	2	2	0	0	6
Leitung IBZ	0	0	2	1	0	0	0	0	0	2	0	5
Didaktik + DZ	0	0	0	0	0	0	0	3	0	0	0	3
Leitung IFS	0	2	0	0	0	0	0	0	0	0	0	2
Ethik	0	0	0	0	0	0	0	0	2	0	0	2
Nachhaltigkeit	0	0	0	0	0	1	0	0	0	0	0	1
Stundenplan	0	0	0	6	0	0	0	0	0	0	0	6
Prüfungsamt	0	0	0	6	0	0	0	0	0	0	0	6
Stipendien	0	0	0	0	0	0	1	0	0	0	0	1
Datenschutz	0	0	0	0	0	0	0	0	0	0	2	2
QM	0	0	0	0	0	0	0	0	0	0	0	0
Keep/INATP	0	0	1	0	0	0	0	0	0	0	0	1
FASE-Labor	0	0	1	0	0	0	0	0	0	0	0	1
Entrepreneurship	0	0	0	0	1	0	0	0	0	0	0	1
Campus Online	0	0	0	0	0	0	0	0	0	0	0	0
Behindertenbeauftragter	0	0	0	0	0	0	0	0	1	0	0	1
Ökomanagement	0	0	0	0	0	0	0	0	0	3	0	3
Nachlass Fakultäten	**7**	**2**	**4**	**19**	**1**	**13**	**8**	**15**	**5**	**5**	**3**	**82**

Gesamt	**17**	**20**	**26**	**31**	**1**	**36**	**30**	**40**	**26**	**15**	**20**	**262**

Deputatsermäßigungen für die Leitung von Fakultäten

Grundlage
Lehrverpflichtungsverordnung vom 4. August 2003, hier § 6a Abs. 4

Freistellungspauschale

Fakultät ohne Studiengänge	⇒	≤ 8	Lehrveranstaltungsstunden
Fakultät mit ≤ 11 Professorenstellen	⇒	≤ 12	Lehrveranstaltungsstunden
Fakultät mit 12 ...15 Professorenstellen	⇒	≤ 16	Lehrveranstaltungsstunden
Fakultät mit 16 Professorenstellen	⇒	≤ 20	Lehrveranstaltungsstunden,

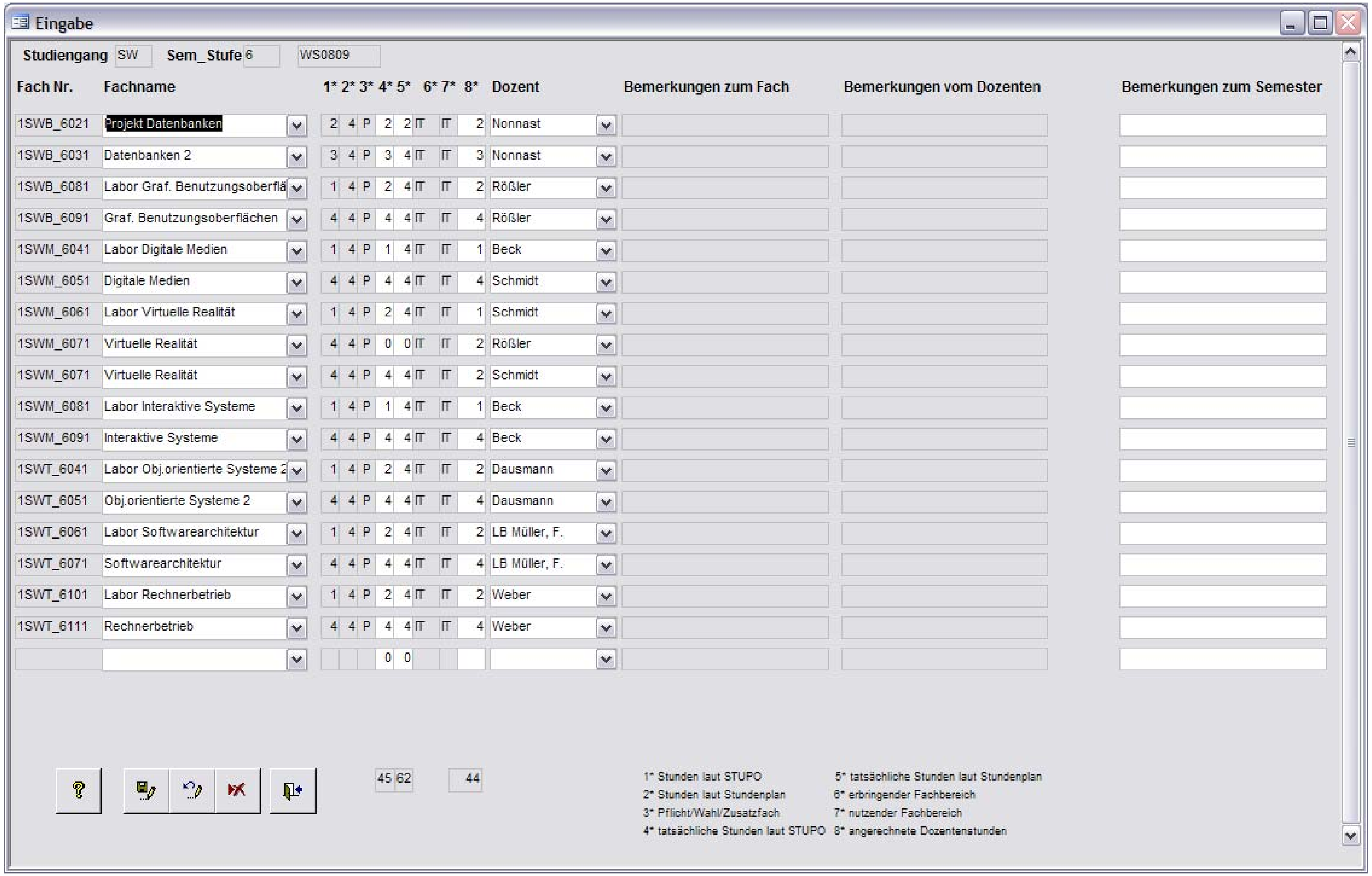
und zusätzlich
1 weitere Lehrveranstaltungsstunde
je weiterer Professorenstelle

Iststand (Stellen laut Angabe Personalabteilung zum 01.10.2009

Fakultät	Professorenstellen	Pauschale (max.)
AN	17	21
BW	18	22
FZ	24,5	28,5
G	24,5	28,5
GS	Sonderregelung (6)	nicht anwendbar
IT	24	28
MB	23,5	27,5
ME	29	33
SAGP	39	43
VU	12	16
WI	13	16

Die Zuteilung der Deputate ist in der Semesterplanung auszuweisen.

Planen



Eingabe

Studiengang SW Sem_Stufe 6 WS0809

Fach Nr.	Fachname	1*	2*	3*	4*	5*	6*	7*	8*	Dozent	Bemerkungen zum Fach	Bemerkungen vom Dozenten	Bemerkungen zum Semester
1SWB_6021	Projekt Datenbanken	2	4	P	2	2	IT	IT	2	Nonnast			
1SWB_6031	Datenbanken 2	3	4	P	3	4	IT	IT	3	Nonnast			
1SWB_6081	Labor Graf. Benutzungsoberflä	1	4	P	2	4	IT	IT	2	Rößler			
1SWB_6091	Graf. Benutzungsoberflächen	4	4	P	4	4	IT	IT	4	Rößler			
1SWM_6041	Labor Digitale Medien	1	4	P	1	4	IT	IT	1	Beck			
1SWM_6051	Digitale Medien	4	4	P	4	4	IT	IT	4	Schmidt			
1SWM_6061	Labor Virtuelle Realität	1	4	P	2	4	IT	IT	1	Schmidt			
1SWM_6071	Virtuelle Realität	4	4	P	0	0	IT	IT	2	Rößler			
1SWM_6071	Virtuelle Realität	4	4	P	4	4	IT	IT	2	Schmidt			
1SWM_6081	Labor Interaktive Systeme	1	4	P	1	4	IT	IT	1	Beck			
1SWM_6091	Interaktive Systeme	4	4	P	4	4	IT	IT	4	Beck			
1SWT_6041	Labor Obj.orientierte Systeme 2	1	4	P	2	4	IT	IT	2	Dausmann			
1SWT_6051	Obj.orientierte Systeme 2	4	4	P	4	4	IT	IT	4	Dausmann			
1SWT_6061	Labor Softwarearchitektur	1	4	P	2	4	IT	IT	2	LB Müller, F.			
1SWT_6071	Softwarearchitektur	4	4	P	4	4	IT	IT	4	LB Müller, F.			
1SWT_6101	Labor Rechnerbetrieb	1	4	P	2	4	IT	IT	2	Weber			
1SWT_6111	Rechnerbetrieb	4	4	P	4	4	IT	IT	4	Weber			
					0	0							

45 62 44

1* Stunden laut STUPO
2* Stunden laut Stundenplan
3* Pflicht/Wahl/Zusatzfach
4* tatsächliche Stunden laut STUPO
5* tatsächliche Stunden laut Stundenplan
6* erbringender Fachbereich
7* nutzender Fachbereich
8* angerechnete Dozentenstunden

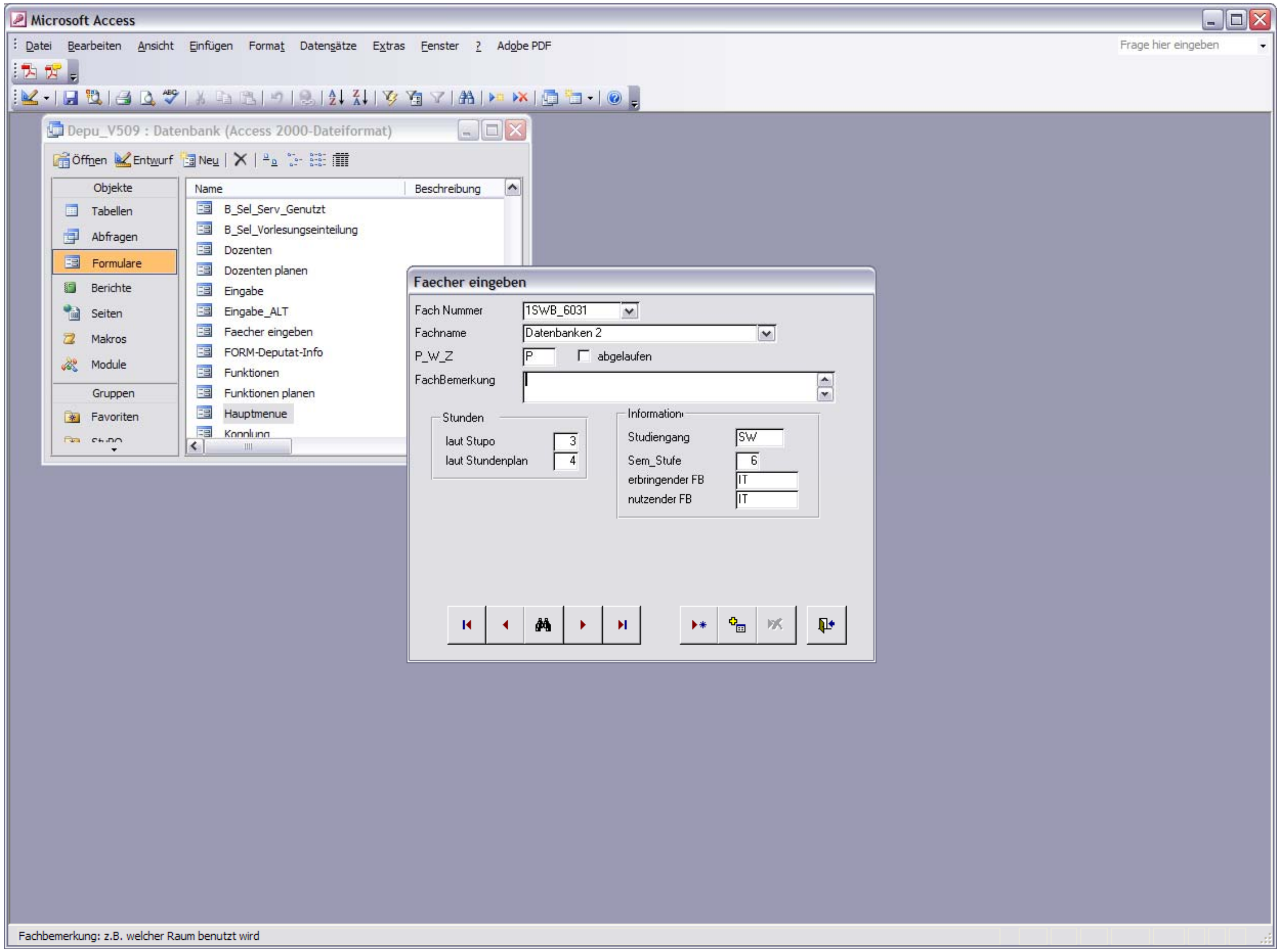
Deputatsnachlässe eingeben

Funktionen

für das Semester: WS0809

Dozent	Funktionsname	Nachlass
Beck	Gleichstellungsbeauftragte	2
Buck	Laborleiter HF	1
Dausmann	Gymnasien	2
Doster	Praktikantenamtsleiter	5
Friedrich	Laborleiter PDV	1
Goll	Laborleiter SuSWT	0
Gündner	Qualitätssicherung	4
Höfer	Laborleiter NT	1
Kappen	Laborleiter Embedded Systems	1
Kappen	Studiengangleiter TI	5
Kull	Laborleiter Mikrosystemtechnik	1
Lindermeir	Laborleiter CAD	1
Malz	Laborleiter Messtechnik	1
Malz	Nachhaltigkeitsbeauftragter	2
Melcher	Studiengangleiter KT	5
Nonnast	Dekan	9
Nonnast	Laborleiter Datenbanken	1
Schmidt	Laborleiter MMVR	1
Schmidt	Prodekan	5
Schoop	Ausland	1
Väterlein	Forschung	5
Väterlein	Studiengangleiter SW	5
Warendorf	Ausland	2
Weber	Laborleiter Betriebssysteme	1
Wiese	Leiter Rechenzentrum	4
Zieher	Fortbildung	18
Zieher	Laborleiter KT	0
Zimmermann	Laborleiter Informationstechnik	1

Stammdatenpflege: Fächer



Service anmelden