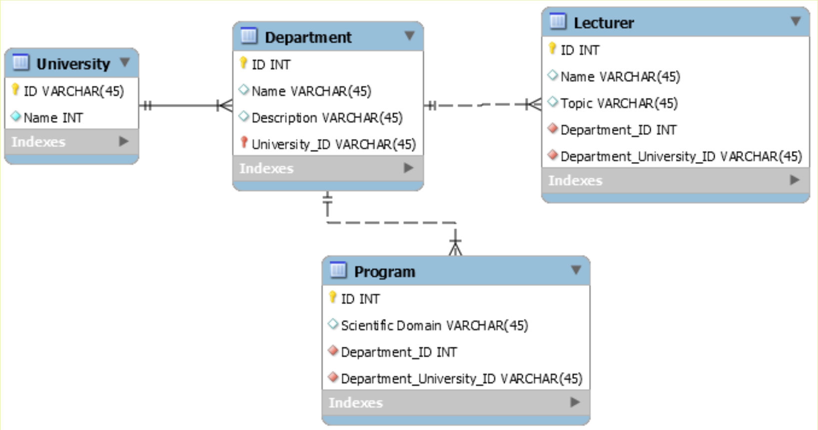
LE 3 – IBIS – Database Systems

Additional Concepts (engl)



Prof. Dr. Markus Grüne, FB03, Wirtschaftsinformatik

Learning Objectives

Learn the difference between structure, semi-structured, and unstructured data.

Know how to describe Business Data Objects in a Data Dictionary.

Understand ER diagrams and ER modeling and learn how to apply it.

Understand the relational model and (on the whiteboard) see the design of relations from the ER model.

Types of data from the „structural“ point of view

Structured Data

- data that has a fixed arrangement, such as a fixed order, defined attributes, or fixed data types.

This structure is found in relational database tables as well as in data with similarly structured file formats (e.g., CSV).

The term “data mining” is used when looking at the analysis of structured data and the resulting knowledge extraction.

Types of data from the „structural“ point of view

Semi-structured Data

- data that have a certain structure, but do not fit into a relational or object-oriented database schema.

Websites with their XML or HTML files are among others. an example of semi-structured data.

One example for analysis of semi-structured data is Web Mining.

Types of data from the „structural“ point of view

Unstructured Data

Unstructured data either

- does not have a predefined data model or
- can not be mapped in a relational database table.

Examples of unstructured data are e.g. Text documents, PDF files, videos, pictures or social media content.

Other forms

Flow data → data such as video streams

Data Dictionary

During the requirements phase (last week!), your primary focus is not on the actual data in the database or the technical design required to implement the business data objects within the database.

Instead, your focus should be on **how the business stakeholders group fields into business data objects**.

One way for the proper definition of requirements for structural data objects is to define a **data dictionary** (not to be confused with the data dictionary in a database system).

Beatty, Joy; Chen, Anthony (2015)

Elements of a Data Dictionary

Business data objects

- are representations of the real-world objects that business users encounter while performing their jobs.
- Examples: credit application, a purchase order, a product.

Fields

- are the characteristics or attributes that describe or define a business data object.
- Example: an order might have an ID, products, shipping address, billing address, payment information, order date, and estimated ship date.

Properties

- A field has properties that specifically define the field and business rules that govern the field.

Beatty, Joy; Chen, Anthony (2015)

Sample Data Dictionary Elements

Property	Description	Example	Notes	Usage
Properties That Define the Business Data Objects and Fields				
ID	A unique identifier for the field. Use a numbering convention that is consistent with the requirements ID numbering	DD001		Necessary
Business Data Object	The name of the business data object that the field is part of.	Customer	The Last Name field is part of the Customer business data object.	Necessary
Field Name	The name the business uses to refer to the field.	Last Name		Necessary
Description	Defines the field. Provides any relevant information beyond the name.	Last Name is a family name or surname of the customer. If the customer only has one name, use the Last Name field.	This is a possible description of the Last Name field.	Optional
Alternate Names	Other names this field is known as. Ideally, you only have one name for each field. However, when you are merging systems or creating a system that is used by multiple groups, a field might have two different, well-established names. If a common name is not clearly understood by all, use this property. This also happens when the names are not synonyms in everyday language but have specific usage within the company. These names can be included in the description if you want to exclude this property.	Family Name	An alternate name for the Last Name field could be Family Name.	Optional
Associated Business Data Object	When a field is another business data object, use this reference and do not repeat the object's information in this row.	Name	Any other business data object. In this case, there might be a Name business data object that has the field's first name, middle name, and last name.	Optional
Data Field	The name under which the data is stored in the system's data store.	LName		Optional
Unique Values?	Whether or not the value for the field has to be unique. This is used if the field is a unique identifier that can be used to differentiate between business data objects of the same type.	No	Multiple customers could have the same last name. This property would be Yes for a field such as social security number.	Optional
Data Type	The type of data used to populate the field. It is best to create and use a set of standard types defined outside an individual Data Dictionary across all of your objects. Also, include formatting information such as patterns for phone numbers or number of decimal digits for real numbers.	Alpha	Basic standard types: Alpha, Numeric, Alphanumeric, or Boolean. More elaborate types: Integer, Real Number, Percent, ZIP/Postal Code, or Phone Number. Alternatively, include formatting information: 3-digit code number, 9-digit number (999.999.9999), or 5 digits plus optional 4 digits (99999- 8888).	Recommended
Length	The maximum number of digits or characters of the field.	50		Recommended

The Excel file is based on Beatty and Chen (2015). You can find it on Moodle.

Entity Relationship Diagrams (ERD)

The **ER model** is the underlying *modeling language*.

It is a graphical representation of the **logical structure of a Database**.

With help of the ER model, the **business analyst** models entities which exist in a system and the relationships between those entities.

The **ER diagram** is the result of modeling with the ER model.

It provides a preview of how all your tables should connect and which what fields each table will own. ER diagrams are the output of visual database modeling.

Entity relationship diagrams

- display the relationships of entity set stored in a database.
- help to explain the logical structure of databases.

Entity Relationship Diagrams

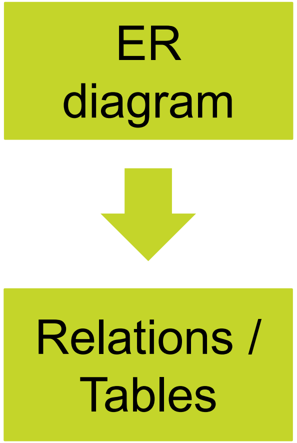
ER diagrams are translatable into relational tables which allows you to build databases quickly (more on this later).

ER diagrams can be used by database designers as a blueprint for implementing data in specific software applications.

Many database management systems distinguish between ER diagrams for logical and physical modeling.

Goals of database diagrams:

- Database diagrams can be used for visualization of data requirements.
- Function as a means of communication between different stakeholders, e.g., database designers, functional analysts, business, testing, etc.



Components of ERDs

The ER model is based on three basic concepts:

- Entities
- Attributes
- Relationships

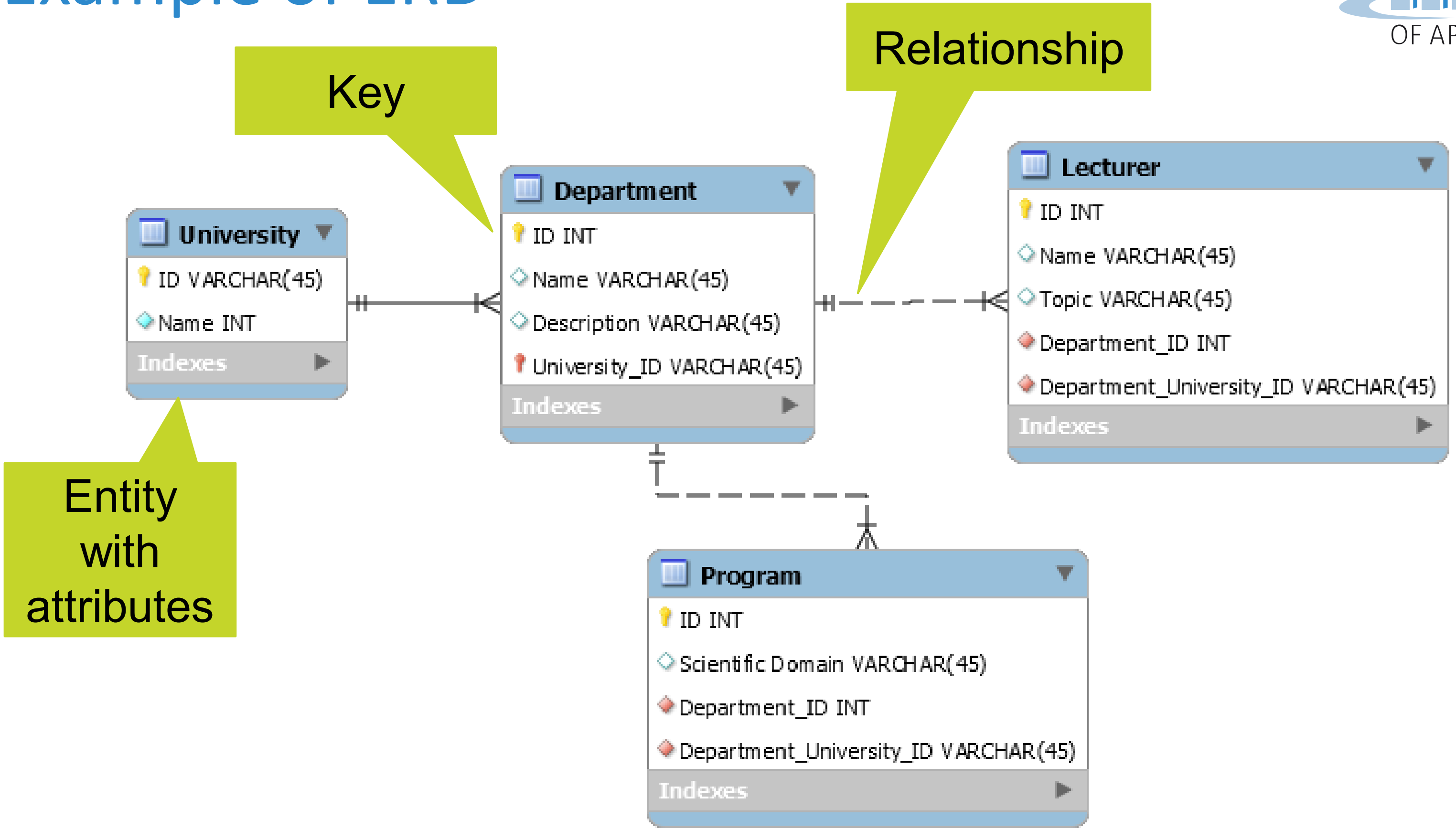
For example, in a University database, we might have **entities** for Students, Courses, and Lecturers. Students entity can have **attributes** like No., Name, and DeptID.

They might have **relationships** with Courses and Lecturers.

University and Department are two examples of entities (more precise: entity types).

Each entity has attributes. For University, the attributes are: ID and Name.

Example of ERD



ERD example – explanation

Previous slide:

A university may have some departments.

All these departments employ various lecturers and offer several programs.

A lecturer from the specific department takes each course, and each lecturer teaches various groups of students (omitted).

In this course, we will use the **Crow Foot notation** instead of the more simplified Chen Notation which is often used in text books.

The Crow Foot notation is the standard notation in most Database Modeling Tools.

A look at relationships

A relationship is an association among two or more entities.

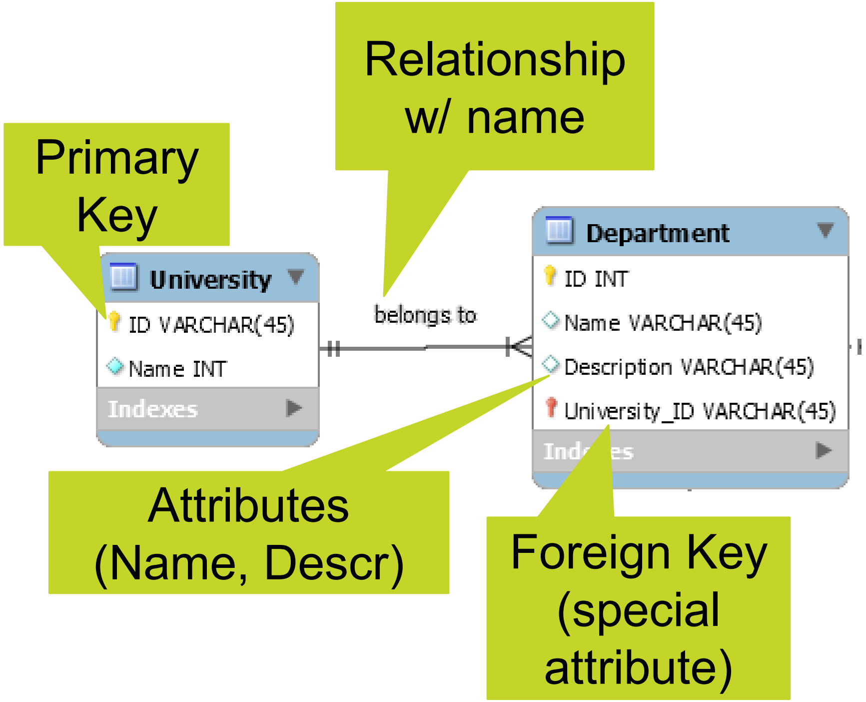
On the right: relationships between University and Department is depicted.

The entities “take part” in the respective relationship.

Relationships can have a name.

A relationship is defined by so-called foreign keys in the database (an attribute that points to the other table).

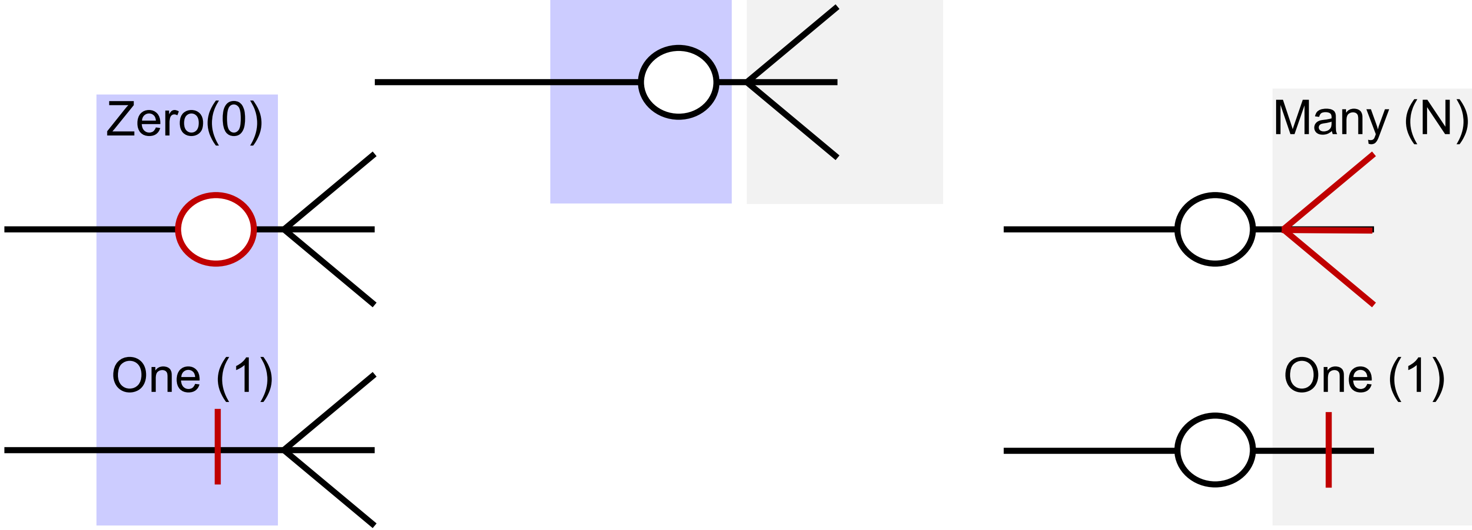
“University_ID” in Department points to the elements in the main table “University”.



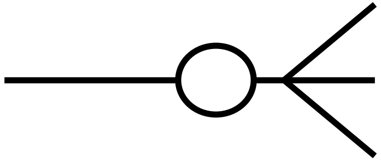
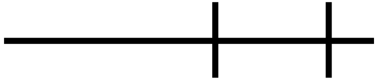
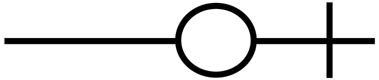
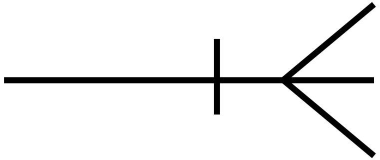
Crow Foot Notation for Relationships Cardinality and Modality

Modality minimum number an instance in one entity can be associated with an instance in the related entity.

Cardinality maximum number an instance in one entity can be associated with instances in the related entity.



Modalities, Cardinalities in Crow Foot

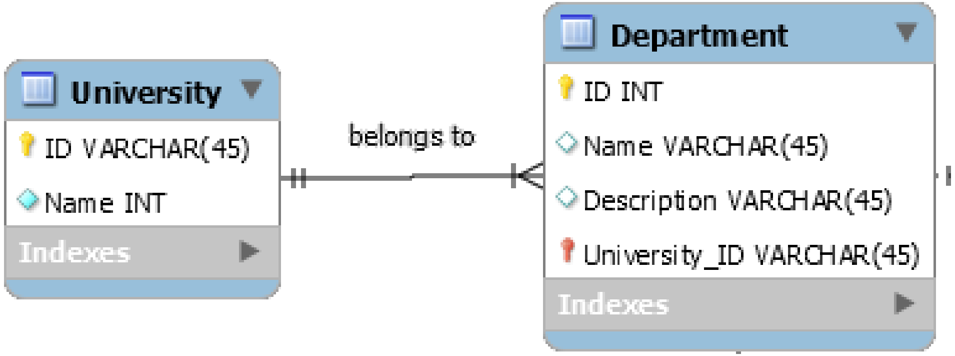
Symbol	Description	Symbol	Description
	Zero or more → N side		One and only one → 1 side
	Zero or one → 1 side		One or more → N side

Relationships interpreted

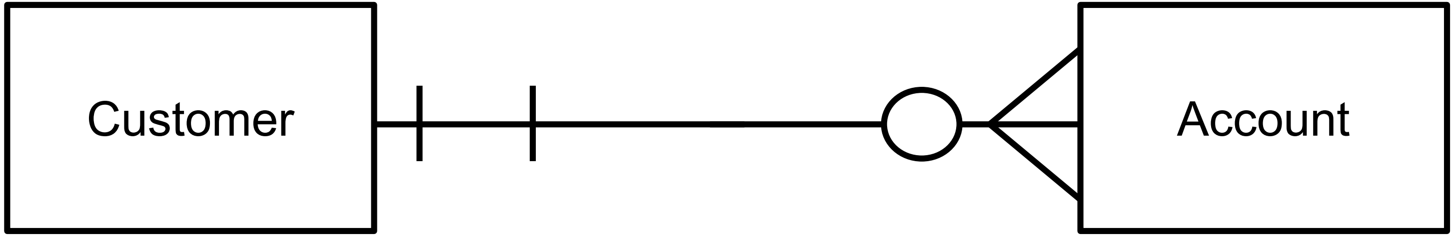
Reading the diagram:

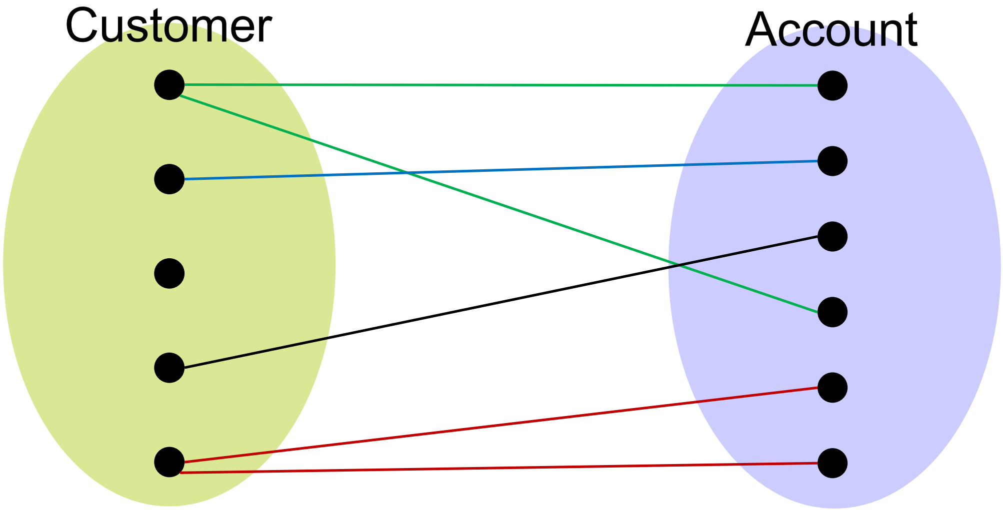
From the right: One Department belongs to **one and only one** University.

From the left: One University has **one or more** Departments.



Interpretation of relationships





Each customer has zero or more accounts

Each account has exactly one customer

Steps for creating an ERD

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Entity identification – Read the specification. All nouns are candidates for entities.

Relationship identification – Identify the relationships. They are expressed by verbs.

Cardinality identification – Look for numbers and stuff like „multiple courses“, „at least one“

Identify Attributes

Create diagram – Put all the information together

This is sometimes hard. As in step one, attributes are nouns in the text, but they relate in some way to the entities. E.g., a teacher has a date of birth.

Some remarks

The ER diagrams features described in this slide deck can be extended.

You will learn more features as you work on a specific use case.

If you are interested, have a look at

- Identifying relationships
- Is-a-relationships (aggregations)
- Weak entities

All the information presented here will help you also in understanding DWH dimensional modeling later on.

The Relational model

We are now looking at the inner design of a database. Until now, we only looked at diagrams for documenting concepts often originating from Business Requirements Documents.

A **relational database** is a collection of **relations** which can be represented as **tables**.

- All entities from the ER diagrams and (almost) all **relationships can be transferred into tables** / relations.
- **Columns** of the tables **define the attributes** of the relation.
- Lines / rows / tuples / individual entities define the **entries of the table** / relation.

A relation in a database

Every relation has attributes and should have a key.

The attributes have a data type or domain.

Question:
In how far is this different from spreadsheets?

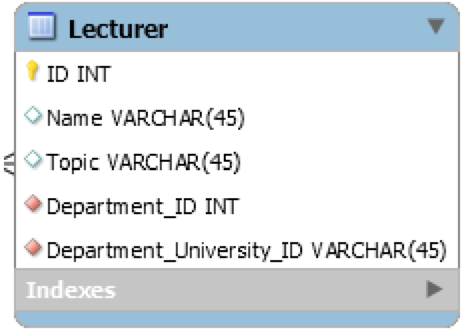


Table Name: Lecturer

Column Name	Datatype	PK	NN	UQ
ID	INT	☑	☑	☐
Name	VARCHAR(45)	☐	☐	☐
Topic	VARCHAR(45)	☐	☐	☐
Department_ID	INT	☐	☑	☐
Department_University_ID	VARCHAR(45)	☐	☑	☐
		☐	☐	☐

Relational modeling

Designing a good data model is not easy.

It is important to keep only information in one table that forms an entity, i.e., which belongs to a concept.

So, in a table for „Cars“ you would not store the persons who drove the car in the past. But you would store, e.g., the information on its horse power.

There are a number of rules for designing „goog“ table structures. These are called „normalization“.

In the following slides, you will see bad design and rules how to fix this. The bad design leads to so-called „anomalies“.

Anomalies – Insertion Anomaly

ID	Name	DOB	Place of Residence	DeptNo	DeptName	DeptMgr
1	Theo Rhetic	12-03-1986	Wetzlar	1	Logistics	Schmitt
2	Sophia Hagia	23-06-1976	Istanbul	1	Logistics	Schmitt
3	Paris Dijon	11-11-1995	Moskow	2	Service	Werner
4	Yevgenij Syrtchuk	23-02-1987	Patna	1	Logistics	Schmidt
				3	Development	Wolf
5	He Hu Must Notbenamed	03-09-1977	Death Valley	3	Development	Wolf

Anomalies – Update Anomaly

ID	Name	DOB	Place of Residence	DeptNo	DeptName	DeptMgr
1	Theo Rhetic	12-03-1986	Wetzlar	1	Logistics	Schmitt
2	Sophia Hagia	23-06-1976	Istanbul	1	Logistics	~~Schmitt~~ Kumar
3	Paris Dijon	11-11-1995	Moskow	2	Service	Werner

Anomalies – Deletion Anomaly

ID	Name	DOB	Place of Residence	DeptNo	DeptName	DeptMgr
1	Theo Rhetic	12-03-1986	Wetzlar	1	Logistics	Kumar
2	Sophia Hagia	23-06-1976	Istanbul	1	Logistics	Kumar
~~3~~	~~Paris Dijon~~	~~11-11-1995~~	~~Moskow~~	~~2~~	~~Service~~	~~Werner~~

Anomalies – Insertion Anomaly

ID	Name	DOB	Place of Residence	Degrees	DeptNo	DeptName	DeptMgr
1	Theo Rhetic	12-03-1986	Wetzlar	B.A., M.Sc.	1	Logistics	Kumar
2	Sophia Hagia	23-06-1976	Istanbul	B.A.	1	Logistics	Kumar
3	Paris Dijon	11-11-1995	Moskow	B.Sc.	2	Service	Werner
4	Yevgenij Syrtchuk	23-02-1987	Patna	B.Sc., PMP, B.A.	1	Logistics	Kumar

First Normal Form

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

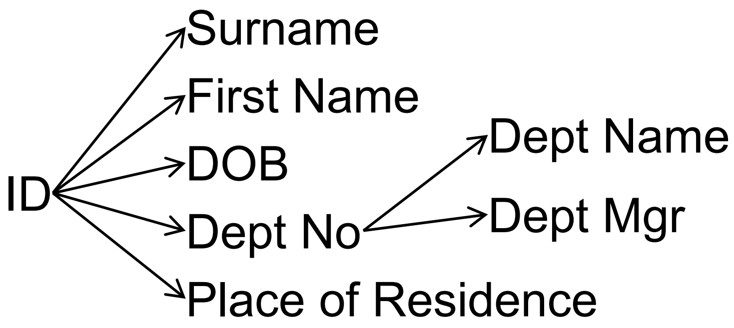
ID	Surname	First Name	DOB	Place of Residence	Dept No	DeptName	DeptMgr
1	Rhetic	Theo	12-03-1986	Wetzlar	1	Logistics	Kumar
2	Hagia	Sophia	23-06-1976	Istanbul	1	Logistics	Kumar
3	Dijon	Paris	11-11-1995	Moskow			
4	Syrtchuk	Yevgenij	23-02-1987	Patna			

ID	PersID	Degree	Year
1	1	B.A.	2006
2	1	M.Sc.	2008
3	2	B.A.	2007
4	3	B.Sc.	2010
…	…	…	…

An in-depth explanation on Normal Forms will follow later.

First Normal Form

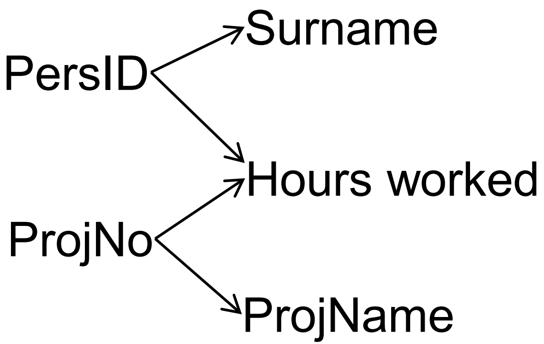
ID	Surname	First Name	DOB	Place of Residence	Dept No	DeptName	DeptMgr
1	Rhetic	Theo	12-03-1986	Wetzlar	1	Logistics	Kumar
2	Hagia	Sophia	23-06-1976	Istanbul	1	Logistics	Kumar
3	Dijon	Paris	11-11-1995	Moskow	2	Service	Werner
4	Syrtchuk	Yevgenij	23-02-1987	Patna	1	Logistics	Kumar



Second Normal Form

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

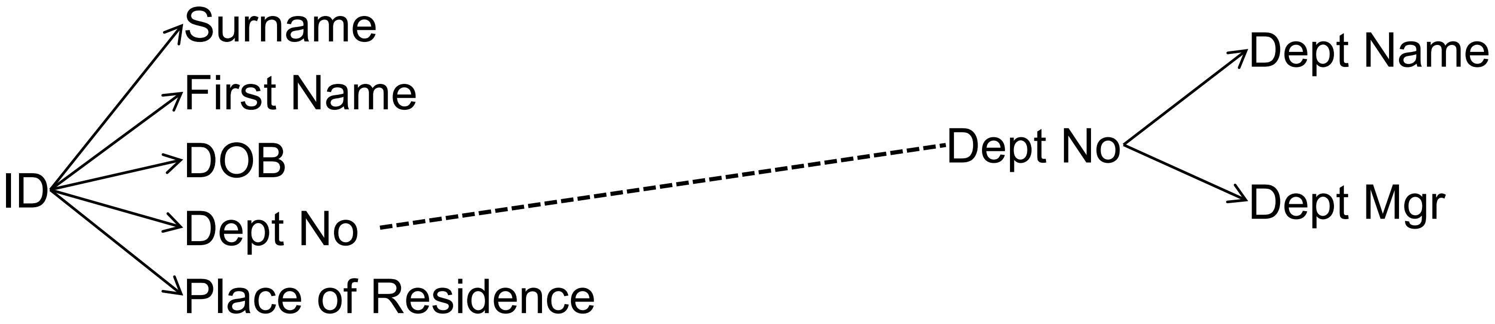
PersID	Surname	ProjNo	ProjName	Hours worked
1	Rhetic	10	Intranet	30
2	Hagia	10	Intranet	45
3	Dijon	21	Extranet	90
3	Dijon	10	Intranet	5



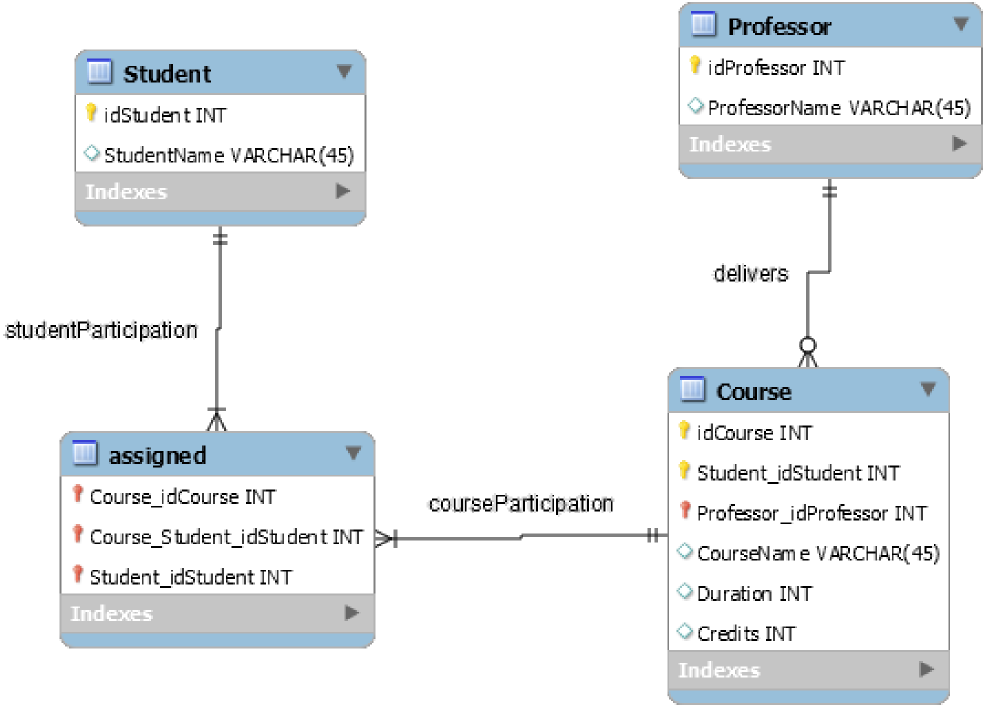
Third Normal Form

ID	Surname	First Name	DOB	Place of Residence	Dept No
1	Rhetic	Theo	12-03-1986	Wetzlar	1
2	Hagia	Sophia	23-06-1976	Istanbul	1
3	Dijon	Paris	11-11-1995	Moskow	2
4	Syrtchuk	Yevgenij	23-02-1987	Patna	1

Dept No	DeptName	DeptMgr
1	Logistics	Kumar
2	Service	Werner



A final look at the database



Key Takeaways

In this chapter we looked at structured data from different angles.

First, we learned the ER model and how to apply it for conceptual modeling, i.e., designing models that we can show our business.

Second, we took a look at how to derive tables from our ER models. And we learned how to create „good“ designs.

The contents of this chapter are a first look only. But they provide enough details for understanding DWH concepts in upcoming chapters.

Literature

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Internet sources:

Published sources: