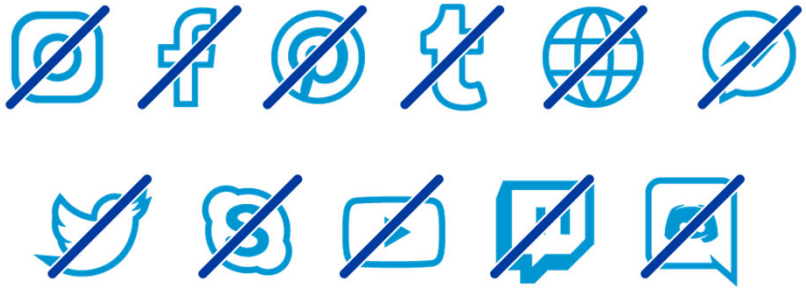
DATABASES

Source: https://en.itpedia.nl/2017/11/26/wat-is-een-database/

Prof. Dr. Ulrike Herster
Hamburg University of Applied Sciences

HAW HAMBURG

COPYRIGHT



The publication and sharing of
slides, images and sound recordings of this
course is not permitted

ORGANISATION
CONTACT

- Prof. Dr. Ulrike Herster
  - Mail: ulrike.herster@haw-hamburg.de
  - Room: 10.02, BT7
  - Office hours by arrangement

Scan Me

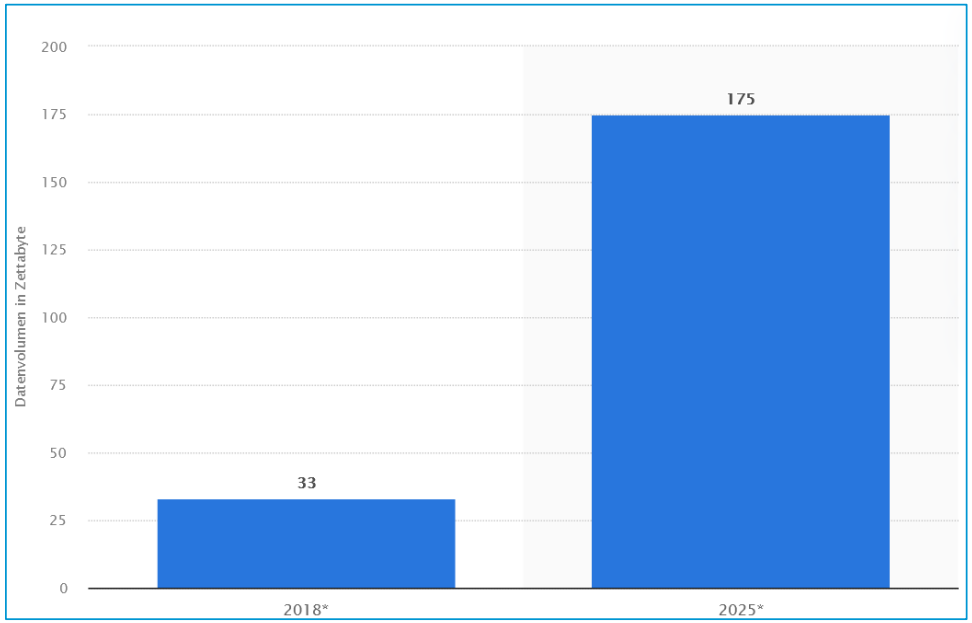
Source: Photo by Melinda Gimpel on Unsplash

DATABASES - WHY?

Source: https://www.linkedin.com/posts/oxana-zeitler_what-happens-in-an-internet-minute-2023-activity-7062646772900212736-05me?trk=public_profile_like_view

DATABASES - WHY?

- Amount of digital data generated annually worldwide in 2018 and 2025 in Zettabytes:



- 1 Zettabyte = 1.000.000.000.000.000.000.000 Bytes = 1.000.000.000.000 GB

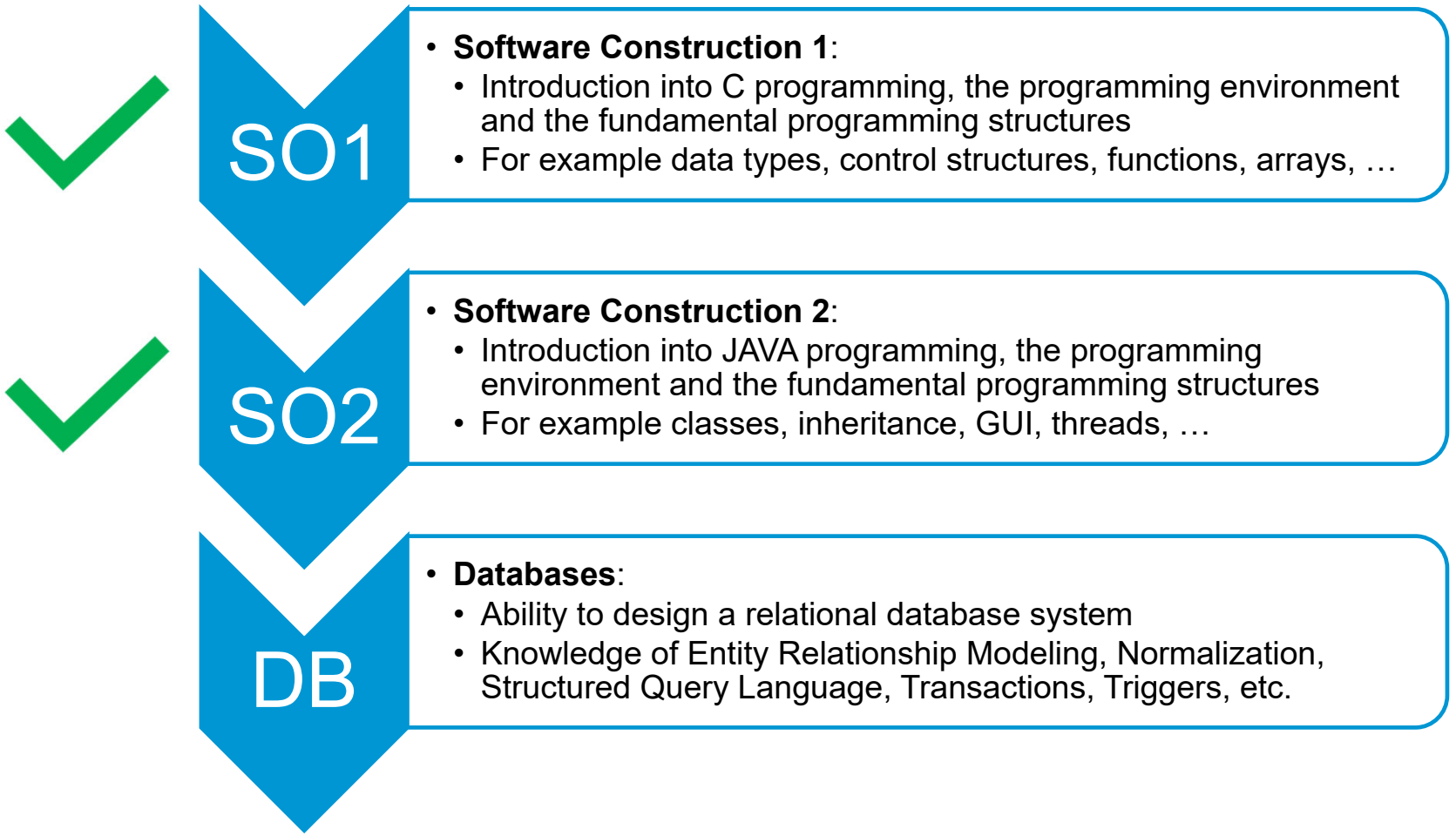
Source: https://de.statista.com/statistik/daten/studie/267974/umfrage/prognose-zum-weltweit-generierten-datenvolumen/ [4]

DATABASES - WHY?

- Crucial for a successful study
- Databases are one of the central parts of a company's software: thus, they are critical for every company and every public authority
- The actionability of every company depends on the availability, the completeness and the correctness of the data
- Increasing importance and use of AI and Machine Learning
- Excellent job opportunities, e.g., Data Engineer, Data scientist, Data analyst, BI Developer, etc.]
- *Because it's fun* ☺

Source: www.akamai.com

EMBEDDING IN YOUR STUDY



ORGANIZATION
LECTURES AND LABORATORY

Lectures for giving theory and practice

- Fundamentals of databases
- Designing database models
- Defining data structures
- Building databases
- Implementing transactions
- Applying views
- Handling security, integrity & triggers

Laboratory for applying concepts and deepening knowledge

ORGANIZATION
LECTURES AND LABORATORY: TIMETABLE

- 11 Lectures
  → Lectures for giving theory and practice
- Laboratories
  → Laboratory for applying concepts and deepening knowledge
- For actual dates see schedule on HAW-website

CW	Mondays 08:10 – 11:25 a.m.	Tuesdays 08:10 – 11:25 a.m.	Thursdays 12:10 – 15:40 p.m.
15			DB
16	DB		DB
17	DB		DB
18	DBL/All Groups		DB
19	DBL/01		
20	DBL/02		DB
21		DBL/03	
22	DBL/01		DB
23	DBL/02		DB
24	DBL/03		
25	DBL/01		DB
26	DBL/02		DB
27	DBL/03		

ORGANIZATION
LECTURES AND LABORATORY

Scan Me

- Lecture material at moodle.haw-hamburg.de
- Enrolment key: db_2024

- The slides will be provided as script, no additional text script
- Therefore, your notes are essential!

ORGANIZATION
LABORATORY

Source: Photo by Mimi Thian on Unsplash

- Assignments should be worked on in teams
  - Work in fixed **teams of two**
    Division of the teams of two in lab on 29.04.2024
  - Working together means discussing things, explaining each other, helping each other out
  - Every team member must be ability to explain the **solution of each assignment**

- For PVL (precondition for examination):
  - **Presence in all laboratories is obligatory!**
    In case of illness: Send a sick note and make up the Lab on another date
  - All assignments for the labs must be successfully solved
  - Each student must present **at least two** assignments on blackboard

ORGANIZATION
LABORATORY

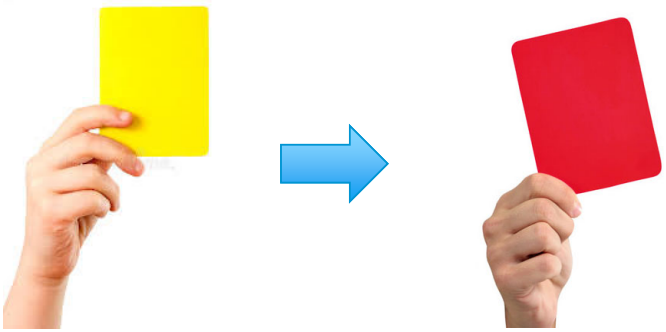
- Assignments are published before laboratory
- Each laboratory consists of two parts:
  1. Upfront assignments
     - Submitting the solutions Friday before lab date
       E.g., solution of lab on 06.05.2024 must be submitted until 03.05.2024 11:59 p.m. via moodle
       - Only one submission per team of two
       - No re-submission after a laboratory
       - Submit only **PDF-files**
  2. Live assignments
     - They can be solved in advance or during the Lab
     - Discussion during the laboratory

Source: Photo by Mimi Thian on Unsplash

ORGANIZATION LABORATORY

- For each laboratory date
  - Punctual participation
  - **Each** team member must be able to explain the solution to **all upfront assignments**
- In case of one violation of the rules yellow card
- In case of second violation exclusion from exercise!!!
- **Participation of all laboratory dates is obligatory, unexcused absence leads to immediate exclusion from the laboratory**

Source: Photo by Mimi Thian on Unsplash



ORGANIZATION
LABORATORY – 1. LAB ON 29.04.2024

- Joint lab with all three lab groups
- **Attendance is mandatory!**
- Division of the teams of two to work on the lab assignments
- **Bring your own device with a working mySQL database!**
- You do not have to submit solutions in moodle or implement the assignments in advance

ORGANIZATION LABORATORY

- Contacts
  - Julian Moldenhauer → Lab group 01 or group 2
    julian.moldenhauer@haw-hamburg.de
  - Furkan Yildirim → Lab group 01 or group 2
    furkan.yildirim@haw-hamburg.de
  - Ulrike Herster → Lecture and Lab group 03
    Ulrike.Herster@haw-hamburg.de

Source: Photo by Mimi Thian on Unsplash

ORGANIZATION
USING A DBMS

Source: Foto von Burst auf Unsplash

- Unfortunately, computers with a running DBMS will not be available in the PC pools in this semester (due to the cyber attack) ☹
- **Each student must install a DBMS on their home computer**
  (mySQL preferred, MariaDB, PostgresSQL, Oracle,...)
- Each lab team must bring a laptop with a running DBMS
  → So, make sure in time that at least one student of each lab team has a laptop with a DBMS!
- If this is not the case, please contact us directly before the first lab
- Install a DBMS promptly and try out the examples from the lecture

ORGANIZATION
USING A DBMS

- You can download the mySQL Workbench using the following link: https://dev.mysql.com/downloads/workbench/

Scan Me

- You can find some tutorials in moodle

Installing mySQL

It is expected in the fourth semester that you are able to install the appropriate software yourself. If you have problems, you can watch online tutorials. Here are some videos which might be helpful too.

Note: The videos do not refer to the latest version. Exchange with other students. You can help each other via the Moodle room in the forum.

ORGANIZATION
PASSING THE LECTURE

- Requirements (PVL)
  - → Successful attendance to each laboratory
  - → Presentation of at least two assignments in the laboratories

- Written exam at the end of semester
  - → Content of lecture and laboratory

ORGANIZATION
QUESTIONS

- If you have any questions during the lecture,
  feel free to ask

- If you have any questions afterwards
  or regarding lab assignments,
  please feel free to use the **forum** in our moodle room
  → This has a significant learning benefit for both the questioner and the answerer!

ORGANIZATION
RECOMMENDED READING

- **R. Elmasri, S.B. Navathe:**
  Fundamentals of Database Systems
- Books from C.J. Date:
  - An Introduction to Database Systems
  - SQL and Relational Theory: How to write accurate code
- German Books
  - Heuer, Saake:
    Datenbanken Konzepte und Sprachen
  - H. Faeskorn-Woyke et al.:
    Datenbanksysteme
  - Ralf Adams:
    SQL: Der Grundkurs für Ausbilung und Praxis
- …

ORGANIZATION
NOTES

- Allocate time for learning, create a schedule
- Try out the examples discussed in the lecture
- Create your laboratory yourself
- Keep your eye on the ball! The end of the semester comes faster than you think
- Take an active part, ask questions

Learning databases only in a face-to-face class is like taking a swimming course on dry land!

ORGANIZATION
LEARNING OUTCOMES

Who

- The students

What

- develop relational database systems for simple applications on their own

With what

- by
  - describing application areas and background of database systems
  - illustrating the architectures of relational database systems
  - modeling database schema for applications with Entity Relationship diagrams
  - applying normalization on a database schema
  - realizing the modeled database schema within a relational database system
  - applying SQL (structured query language) for extraction and manipulation of data within the database

For what

- for guaranteeing the actionability of every company which depends on the availability, the completeness and the correctness of the data

ORGANIZATION
LEARNING OUTCOMES

What not

- The students are not trained to be a database expert
- The students are not trained to be a database administrator
- The lecture concentrates on relational databases and only gives a short overview of other types of databases
- The students do not learn every detail of SQL

→ There are many dialects (historical, vendor)

- The students do not learn implementational details of DBMS

ORGANIZATION
JOURNEY IN THE PAST SEMESTERS

Source: Photo by Matt Duncan on Unsplash

ORGANIZATION
OUR JOURNEY IN THIS SEMESTER

- Integrity, Trigger & Security
- Database Applications
- Transactions
- Subqueries & Views
- More Features
- Notations & Guidelines
- Constraints
- Relationships
- Simple Entities and Attributes
- Basics

Source: Foto von Justin Kauffman auf Unsplash

ORGANIZATION
WHO ARE YOU?

- Do you already have experience with databases?
- Who attended the lecture Algorithms & Data Structures with me last winter semester?
- Is one of you here in the second trial?
- Who just wants the 5 points and pass with minimal effort?
- What is your expectation of this lecture?

ORGANIZATION
OUR JOURNEY IN THIS SEMESTER

- Integrity, Trigger & Security
- Database Applications
- Transactions
- Subqueries & Views
- More Features
- Notations & Guidelines
- Constraints
- Relationships
- Simple Entities and Attributes
- **Basics**

Source: Foto von Justin Kauffman auf Unsplash

BASICS
DEFINITION

- What is a database?

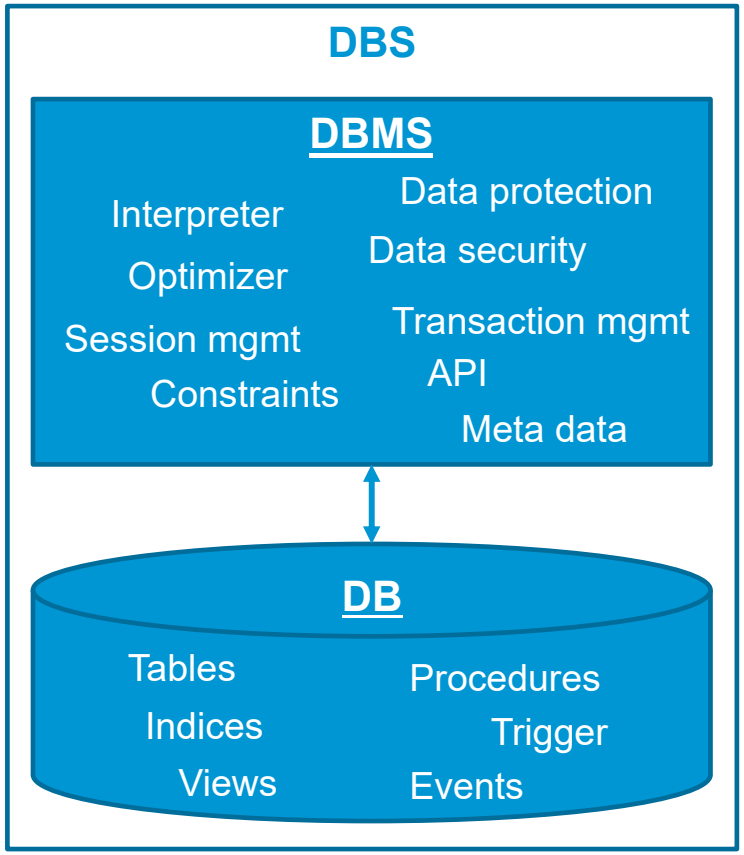
Source: https://www.youtube.com/watch?v=t8jgX1f8kc4

BASICS
DEFINITION

- A "**database**" refers to a set of related data and the way it is organized
- Access to this data is usually provided by a "**database management system**" (DBMS) consisting of an integrated set of computer software that allows users to interact with one or more databases and provides access to all the data contained in the database (although restrictions may exist that limit access to particular data)
- The DBMS provides various functions that allow entry, storage and retrieval of large quantities of information and provides ways to manage how that information is organized

Source: www.wikipedia.org

BASICS
DEFINITION



DB

→ Manage data logically and physically

DBMS

→ Offers tools for managing, editing, and evaluating data

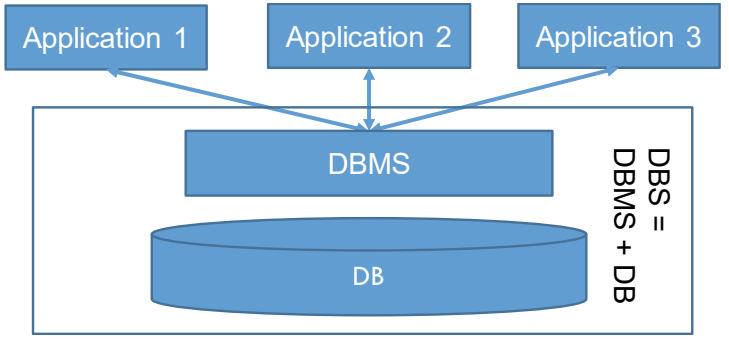
Source: Adams „SQL“

BASICS
DEFINITION

A **DBMS** is a software system that enables the use of a database approach.

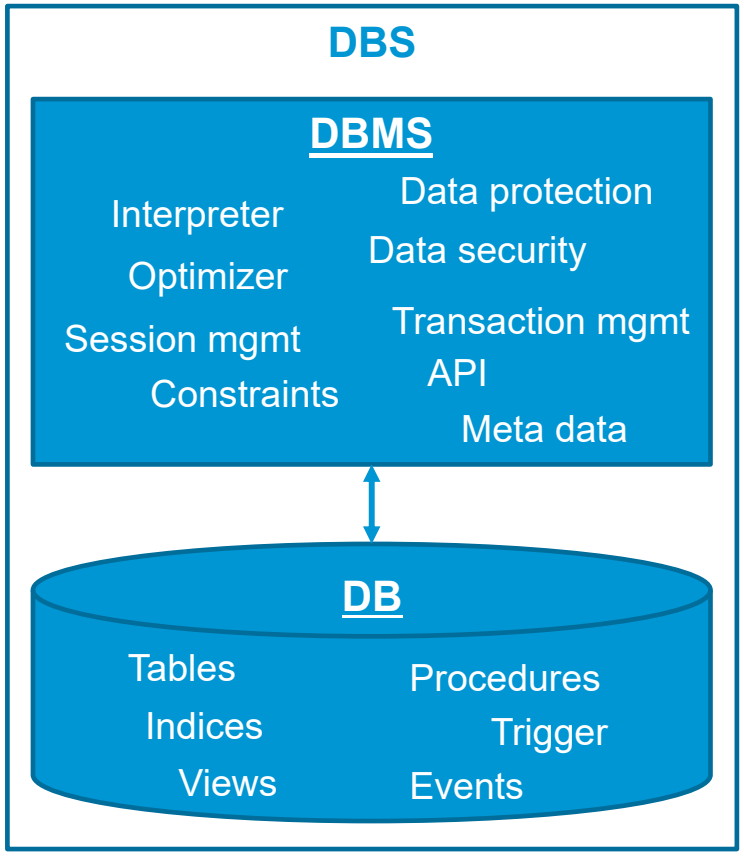
The primary purpose of a DBMS is to provide a systematic method of creating, updating, storing, and retrieving the data stored in a database.

It enables end users and application programmers to share data, and it enables data to be shared among multiple applications rather than propagated and stored in new files for every new application.



Source: Mullins, C. S. 2002. Database Administration:
The Complete Guide to Practices and Procedures.
New York: Addison-Wesley

BASICS
DEFINITION



The essential functions of a DBMS:

- Storage, overwriting and deletion of data
- Management of metadata
- Mechanisms for data security
- Mechanisms for data protection
- Mechanisms for data integrity
- Enabling multi-user operation by transactions
- Optimization of queries
- Enabling triggers and stored procedures
- Providing key figures on DBMS technology and operation

Source: Adams „SQL“
https://de.wikipedia.org/wiki/Datenbank

BASICS
DATABASE EXAMPLES

Source: www.unitop-welt.de

BASICS
DATABASE EXAMPLES

- Customer Relationship Management
  E.g., Amazon has more than 310 million customers worldwide
- Controlling and Accounting
  E.g., Amazon closed the 2023 financial year with sales of around USD 574.79 billion
- Merchandise Management System
  E.g., Amazon has a range of more than 250 million products
- Enterprise Resource Planning
- Content Management Systems
- …

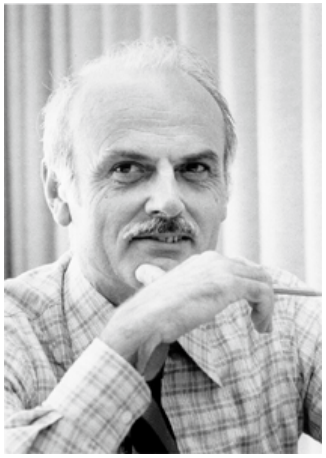
Quelle: https://www.edesk.com/de/blog/amazon-statistik/
https://de.statista.com/statistik/daten/studie/75292/umfrage/nettoumsatz-von-amazoncom-seit-2004/
https://business.amazon.de/de/funktionen-entdecken/einkauf-vereinfachen/produktauswahl

BASICS
HISTORY

- In 1960`s, files were used for storing data
  - Misconception as files are designed for a specific application
  - Thus, a considerable part of daily business was to copy, mix, and restructure the files

- In 1970`s Edgar F. Codd developed the relational databases
  - He worked at IBM
  - He developed the first relational database system “System R”
    - Oracle exploited the results of System R and led SQL to commercial success
    - IBM follows with SQL/DS and DB2

    → today relational databases are the most widespread



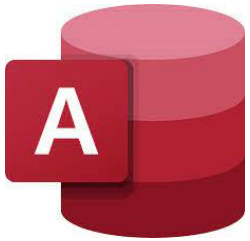
Source: www.wikipedia.org

BASICS
WHY EVEN USE DBS?

- If not by using a DBMS, how would you store your data?
- Maybe here?

Text Files

MS Excel



MS Access

Etc...

Others

- What are the disadvantages?

BASICS
WHY EVEN USE DBS?

- Disadvantages

Text Files

- Structuring of data
- Data types
- Amount of data
- Data Validation
- No Security
- Poor performance & Querying
- Backup of data
- Maintenance
- etc.

MS Excel

- Disadvantages of text files also apply partially to Excel
- Cumbersome to share

MS Access

- Performance with larger data sets
- Data Management
- Single-User/Small team use
- Concurrency & control features

BASICS
DATABASE VS. SPREADSHEET

Source: https://www.youtube.com/watch?v=djEZeF4KTaM

BASICS
DATABASE VS. MS EXCEL

Problems with MS Excel

- It's easy to make **accidental changes** to data
- It's **hard to replicate** an old analysis on new data
- It's **slow** with large sets of data
- It's **cumbersome to share** giant spreadsheets over email

Source: https://news.codecademy.com

BASICS
SQL DATABASE VS. MS EXCEL: WHAT ARE THEY BEST USED FOR?

Database:

- Larger datasets: depending on the software and database, this can be very very large
- Organization /Structure: SQL tables are stricter about consistent data types and restricts users if they try to enter the wrong type
- Collaborative work
- Prepping data for further analysis in another software
- Consistent reports or calculations: you can save and share queries
- More secure, as changes are always traceable and auditable

MS Excel:

- Smaller data sets: under 1 million rows, even north of 100,000 it will likely slow down your computer
- Manually entering data
- More flexible structure: any cell can be of any data type, regardless of what column it's in
- Outputting graphs and visualizations
- Built-in spell check and other useful functions
- Working independently on a project

BASICS
DATABASE VS. MS ACCESS

Aspect	DB	MS Excel	MS Access	Note
Initial training	-	++	+	Initial training is necessary E.g., separation of the presentation and editing of data from processing and storage
Large data sets	++	-	+	Access has performance problems with several thousand entries
Access by multiple users	++	+	+	Multiple users can access the database simultaniously
Database Design	++	-	+	The direct display of tables leads to denormalized tables
Platform independence	++	-	-	Access limited to Windows
Application development	-	++	++	A SQL database can never stand alone
Integration with MS Office	-	++	++	

Source: https://launix.de/

BASICS
DATABASE VS. MS ACCESS: TECHNICAL COMPARISON

Aspect	mySQL	MS Access
Database Size	16 terabytes	2 GB
Simultaneous users	32.767 users	255 users
Number of objects	2.147483.647 objects per database	32.768 objects per database

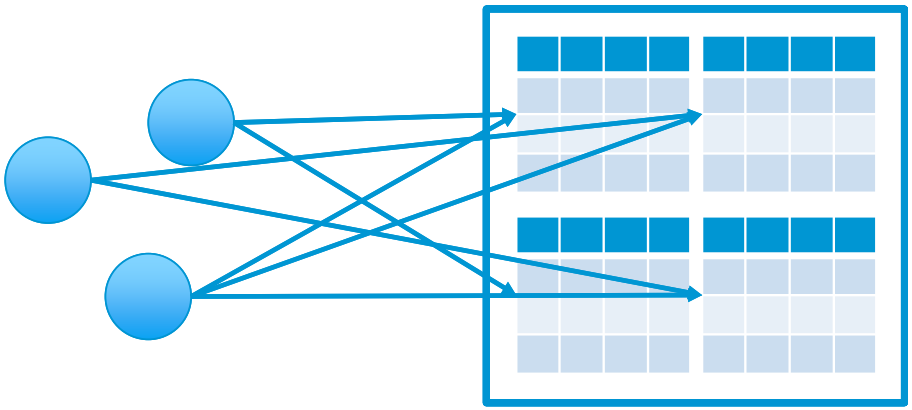
Source: https://database.guide/microsoft-access-vs-sql-server/

BASICS
DBMS

- Different DB-Models
  - Relational model
  - Hierachical model
  - Network model
  - Object relational model
  - Object oriented model
  - XML-based model

BASICS
RDBMS VS. ODBMS

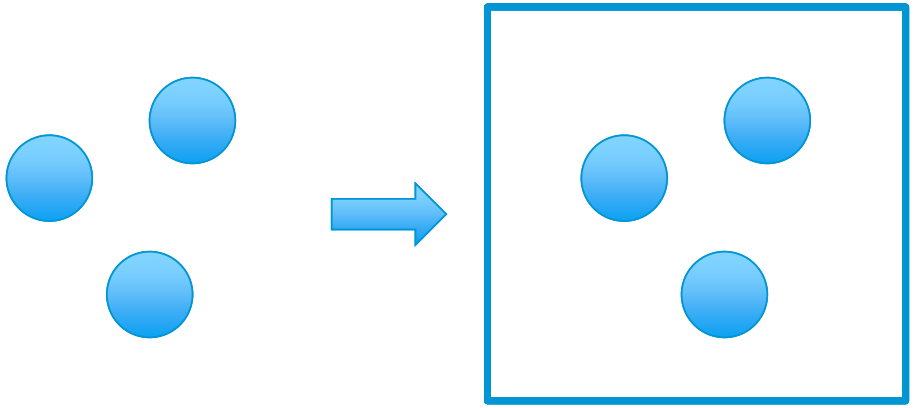
- RDBMS



Some objects

RDBMS with tables

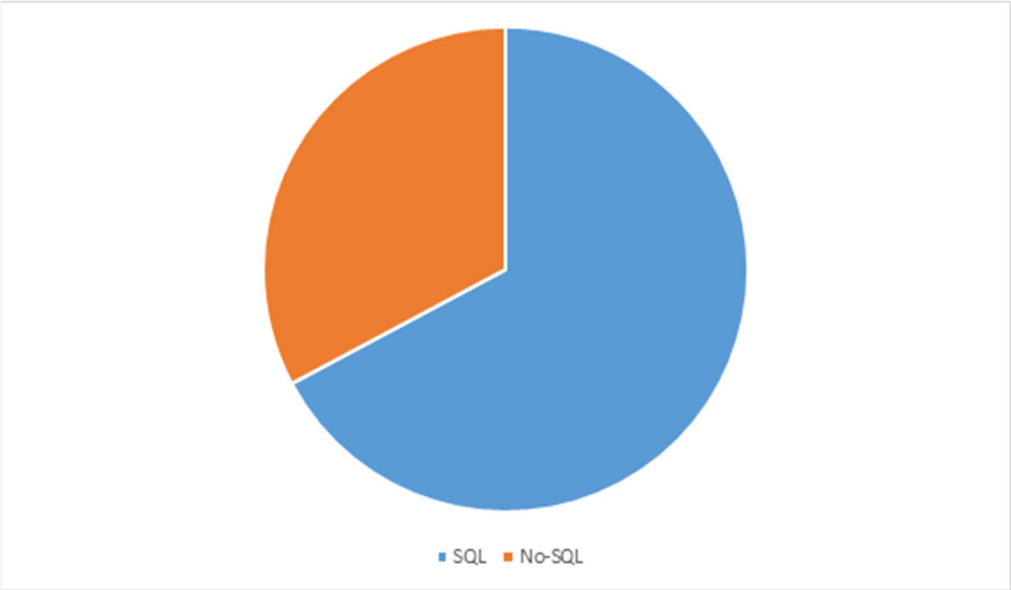
- ODBMS



Some objects

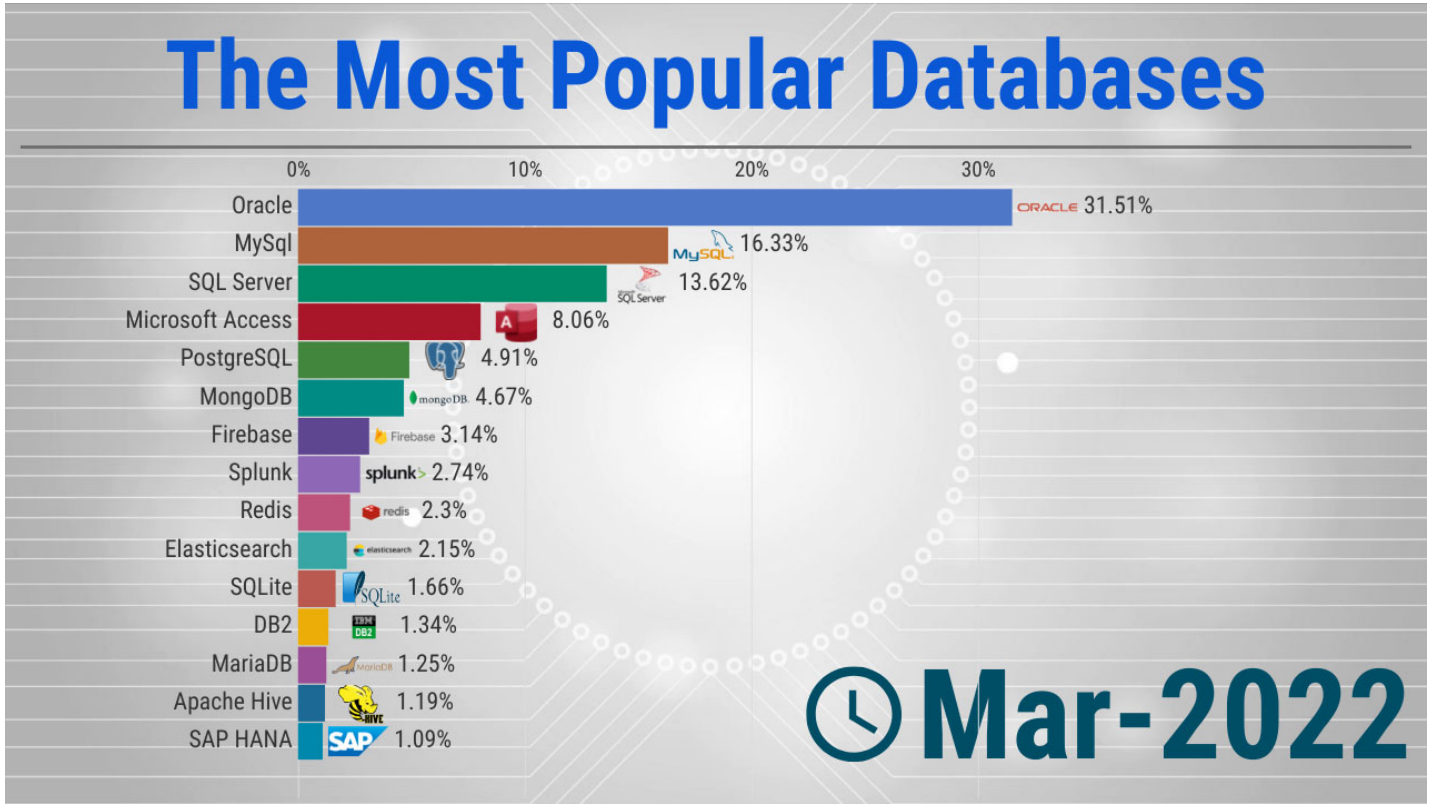
ODBMS with objects

BASICS
RELATIONAL DATABASE MARKET 2020



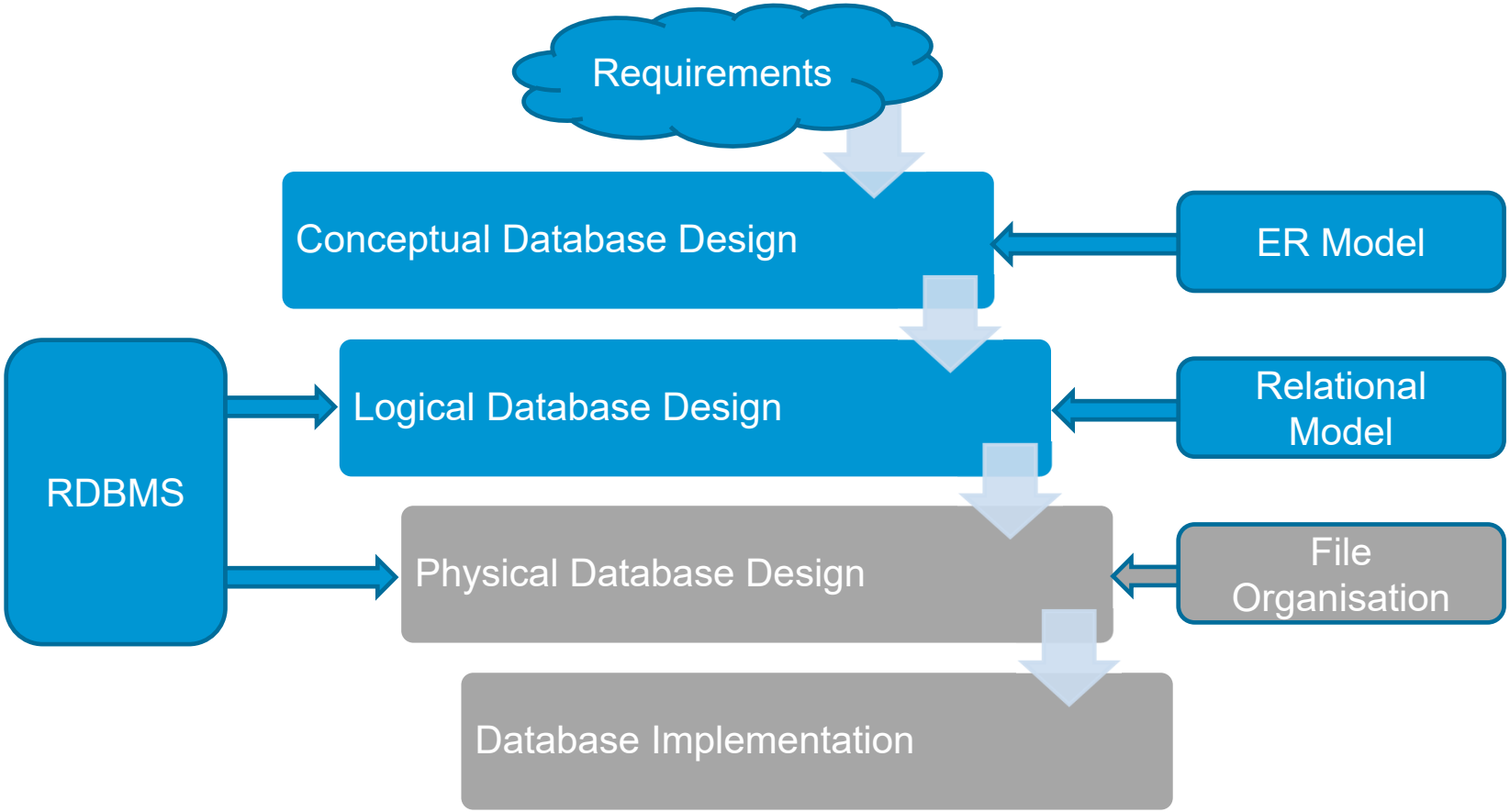
Source: https://www.industryarc.com/Report/19213/relational-database-market.html

BASICS
MOST POPULAR DATABASES 2022

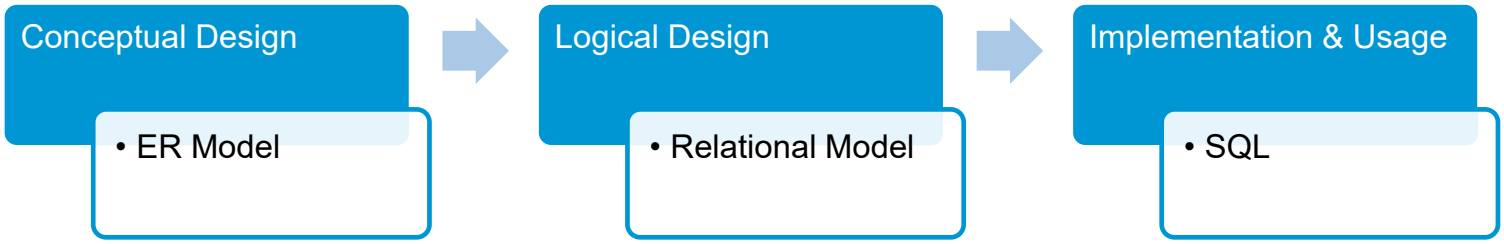


Source: https://statisticsanddata.org/data/most-popular-databases-2006-2022/

BASICS
DATABASE DESIGN



BASICS
DATABASE DESIGN



BASICS
EXAMPLE: CONTACT LIST

Source: Photo by Thom Holmes on Unsplash

What things exist in the real world?
What properties do they have?
How do they relate to each other?



Steven
12.05.1968

Maria
8.05.1972

Jane
30.07.1944

John
10.06.1996

Ann
12.04.1998

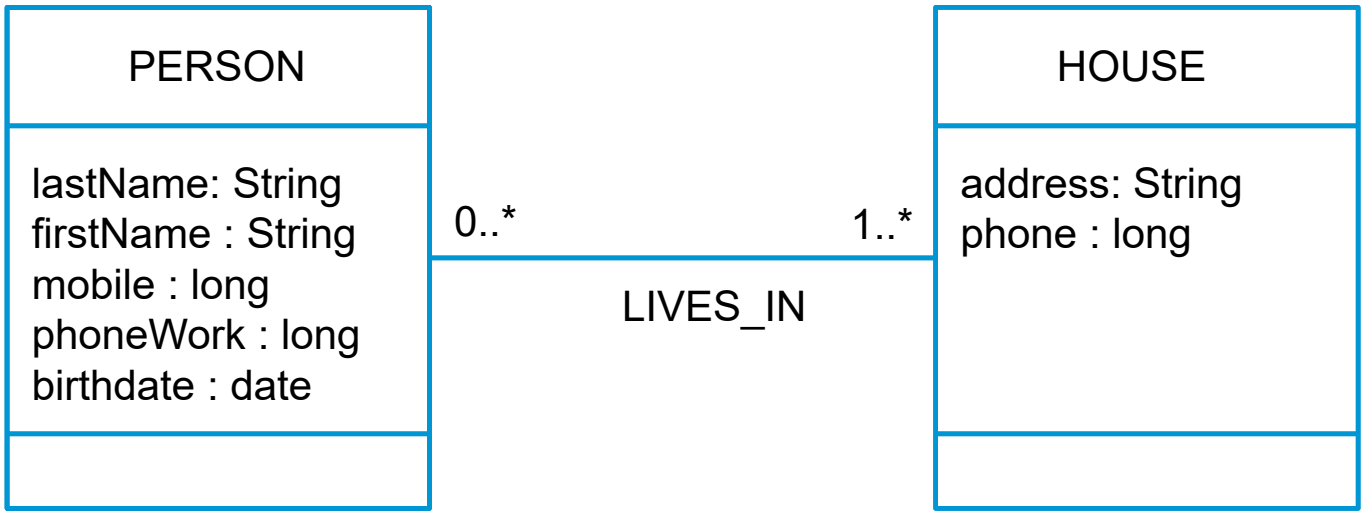
221 Baker Street, 1NW London
0044 20 7946 0000

112 Baker Street, 1NW London
0044 20 7946 1000

55 Oxford Street, W1 London
0044 20 7946 3000

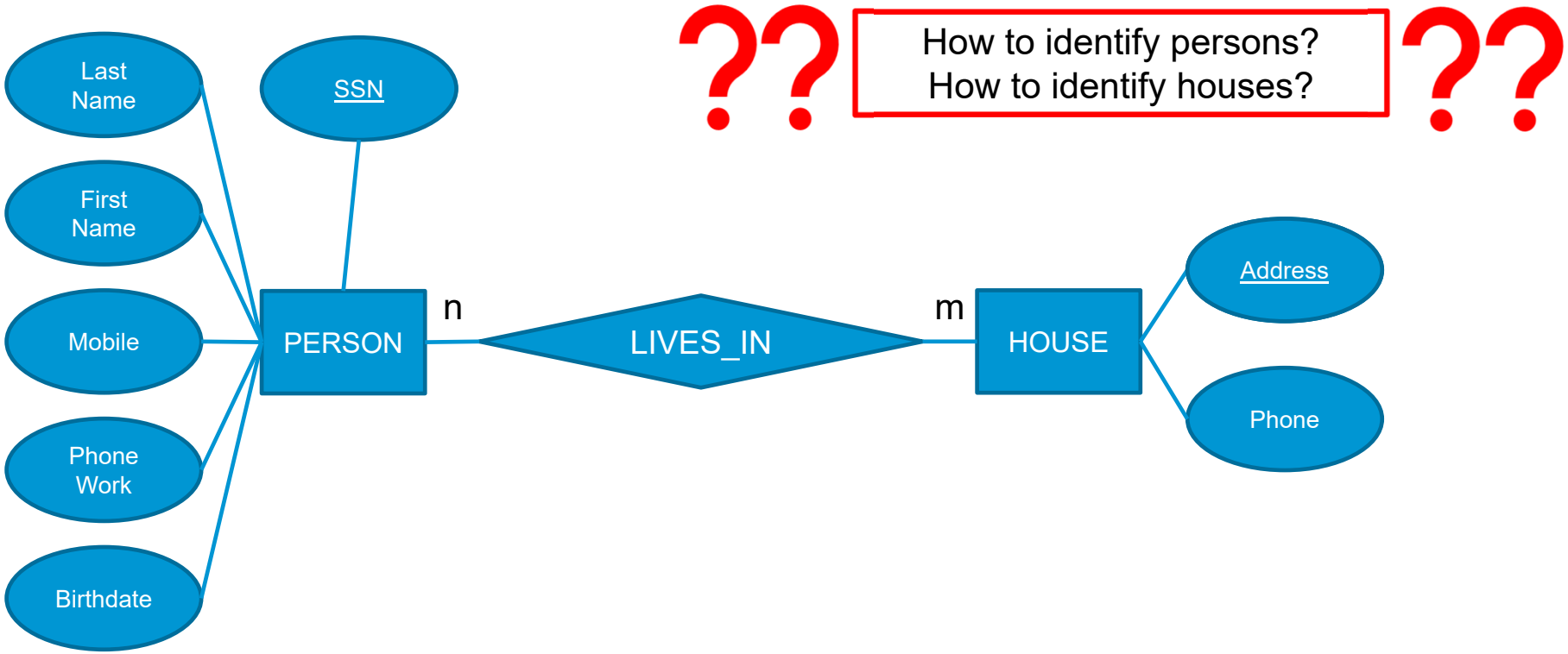
BASICS
EXAMPLE: CONTACT LIST

1. Option 1: Conceptual design with a Class Diagram



BASICS
EXAMPLE: CONTACT LIST

1. Option 2: Conceptual design with ER Model

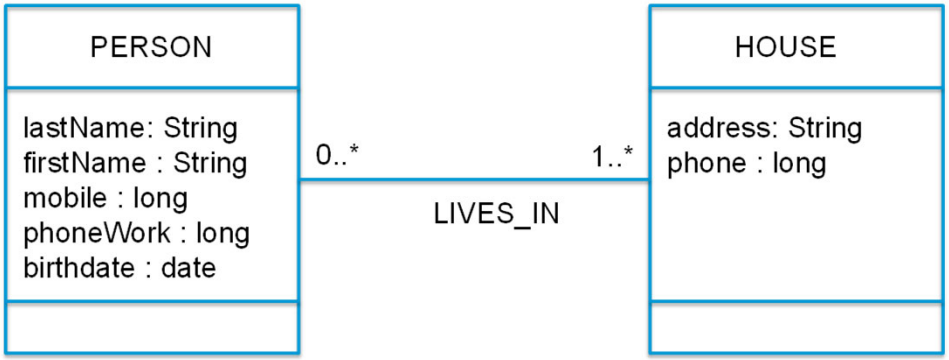


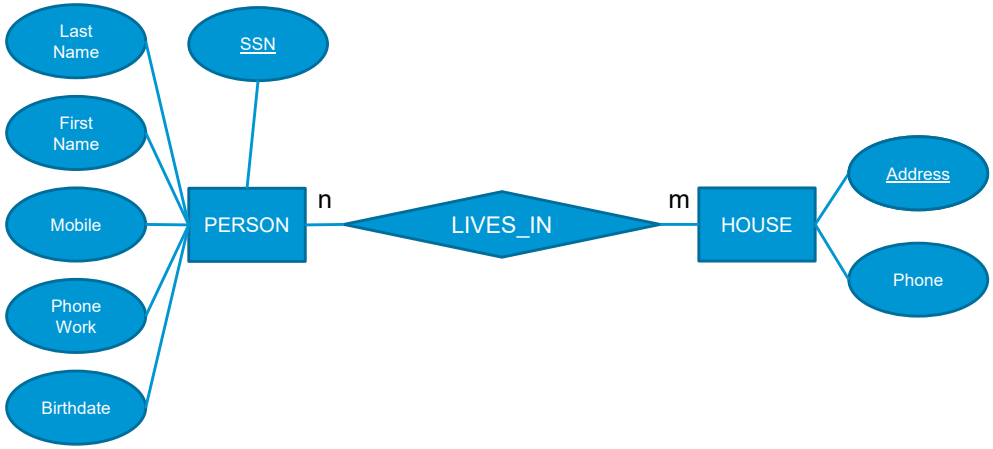
BASICS
EXAMPLE: CONTACT LIST

- 1. Conceptual design: Class diagram vs. ERM

You can easily translate a class diagram into an ERM

Some differences between ERM and Class Diagrams
- There are no methods in an ERM
- ERM offers multivalued attributes
- Identifying keys are marked in ERM
- …





BASICS
EXAMPLE: CONTACT LIST

2. Logical design with Relational Data Model

PERSON	SSN	FirstName	LastName	Mobile	PhoneWork	Birthdate

LIVES_IN	SSN (FK)	Address (FK)

HOUSE	Address	Phone

BASICS
EXAMPLE: CONTACT LIST

PERSON

SSN	Last Name	First Name	Mobile	Phone work	Birthdate
123456789	Miller	Jane	0044 7701 123456	0044 20 7946 0001	11.04.1979
234567891	Miller	Steven	0044 7701 123457	0044 20 7946 0002	13.05.1977
345678912	Miller	Maria			21.07.2015
…	…	…	…	…	…

HOUSE

Address	Phone
221 Baker Street, 1NW London	0044 20 7946 0000
112 Baker Street, 1NW London	0044 20 7946 1000
55 Oxford Street, W1 London	0044 20 7946 3000
…	…

LIVES_IN

SSN	Address
123456789	221 Baker Street, 1NW London
234567891	221 Baker Street, 1NW London
345678912	221 Baker Street, 1NW London
…	…

BASICS
EXAMPLE: CONTACT LIST

3. Implementing in database

```
CREATE TABLE Person(
      SSN           CHAR(9)       NOT NULL ,
      FirstName     VARCHAR(15)   NOT NULL ,
      LastName      VARCHAR(15)   NOT NULL ,
      Mobile        VARCHAR(30) ,
      PhoneWork     VARCHAR(30) ,
      Birthdate     DATE ,
      PRIMARY KEY ( SSN )
);
```

BASICS
EXAMPLE: CONTACT LIST

4. Using the database

```
SELECT FirstName, LastName
FROM   Person
WHERE  LastName = 'Miller' ;
```

```
SELECT FirstName, LastName
FROM   Person P, lives_in L, House H
WHERE  Phone = '0044 20  7946 0000' AND
       H.Address = L.Address AND
       L.SSN = P.SSN ;
```

BASICS
EXAMPLE: CONTACT LIST

PERSON

SSN	Last Name	First Name	Mobile	Phone work	Birthdate
123456789	Miller	Jane	0044 7701 123456	0044 20 7946 0001	11.04.1979
234567891	Miller	Steven	0044 7701 123457	0044 20 7946 0002	13.05.1977
345678912	Miller	Maria			21.07.2015
…	…	…	…	…	…

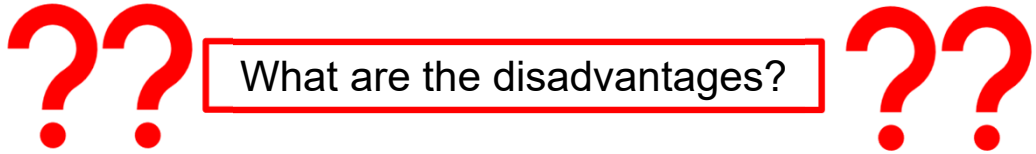
HOUSE

Address	Phone
221 Baker Street, 1NW London	0044 20 7946 0000
112 Baker Street, 1NW London	0044 20 7946 1000
55 Oxford Street, W1 London	0044 20 7946 3000
…	…

LIVES_IN

SSN	Address
123456789	221 Baker Street, 1NW London
234567891	221 Baker Street, 1NW London
345678912	221 Baker Street, 1NW London
…	…

BASICS
EXAMPLE: CONTACT LIST



Last Name	First Name	Address	Phone	Mobile	Phone work	Birthdate
Miller	Jane	221 Baker Street, 1NW London	0044 20 7946 0000	0044 7701 123456	0044 20 7946 0001	11.04.1979
Miller	Steven	221 Baker Street, 1NW London	0044 20 7946 0000	0044 7701 123457	0044 20 7946 0002	13.05.1977
Miller	Maria	221 Baker Street, 1NW London	0044 20 7946 0000			21.07.2015
Miller	Josef	221 Baker Street, 1NW London	0044 20 7946 0000	0044 7701 123458		10.11.2010
Miller	Judy	221 Baker Street, 1NW London	0044 20 7946 0000			01.12.2013
Smith	John	112 Baker Street, 1NW London	0044 20 7946 1000	0044 7701 456789		06.06.1987
Smith	Jane	112 Baker Street, 1NW London	0044 20 7946 1000	0044 7701 456780	0044 20 7946 1003	05.03.1989
Smith	Hannah	112 Baker Street, 1NW London	0044 20 7946 1000			10.01.2012
Miller	Ann	55 Oxford Street, W1 London	0044 20 7946 3000	0044 7701 678912	0044 20 7946 3001	18.06.1984
Miller	Paul	55 Oxford Street, W1 London	0044 20 7946 3000			
Miller	Steven	55 Oxford Street, W1 London	0044 20 7946 3000			
…	…	…	…	…	…	…

BASICS
EXAMPLE: AD

- Let‘s create a model for lecture Algorithms and Data Structures

Bachelor Information Engineering	
Algorithms and Data Structures	
Abbreviation	AD / ADL
Module responsibility	Prof. Dr. Dierks
Duration / Semester/ Regular cycle	One semester / 3rd semester / winter and summer semester
Credits (CP) / Semester hours per week (SHW)	6 LP 3 + 1 SHW
Workload	Attendance: 72 h Self-study: 108 h
Type of module	Mandatory module
Prerequisites	Programming experience recommended
Language	English
Learning outcomes	• The students understand that choosing data structures and algorithms affects the efficiency of their programs • The students know state-of-the-art solutions for typical problems and they are able to apply them in their own programs to improve the quality thereof. • The students know theoretical limits of sorting and searching and they are able to apply this knowledge to analyze the complexity of new programming problems • The students are able to synthesize efficient programs by applying the taught algorithms and data structures • The students learn that it makes sense to apply state-of-the-art algorithms to produce competitive software that is scalable
Learning content	**Lecture:** • Introduction with elementary algorithms and complexity estimations, complexity • Abstract data types and their implementation • Sorting, Divide-and-Conquer, Pivot, Mergesort, Priority Queue • Search algorithms • Finite-state automata • Introduction to graph- and optimization algorithms **Laboratory:** • Empiric detection of complexity depending on problem size by counting the number of steps • Dynamic behavior of sorting algorithms • Tree traversals, search algorithms
Usability of the module	Degree program relevance
Requirements for the recognition of credits (Study and exam requirements)	Regular examination type for module testing: Lecture: Successful passing in written exam (K)(PL) Laboratory: Successful participation in exercises (LA)(PVL)
Courses	AD (Lecture) ADL (Laboratory / Exercises)

BASICS
DATA VS. INFORMATION

- Data:

John Meyer	123456
Judy Fisher	234567
William Smith	345678

- Information:
  We define information as data that have been processed in such a way that the knowledge of the person who uses the data is increased

Students

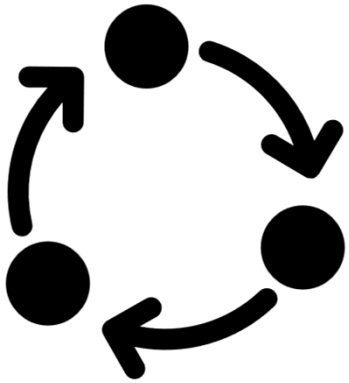
Course: Databases **Term:** Summer 2021

Student	Matr_no	Term
John Meyer	123456	2
Judy Fisher	234567	4
William Smith	345678	3

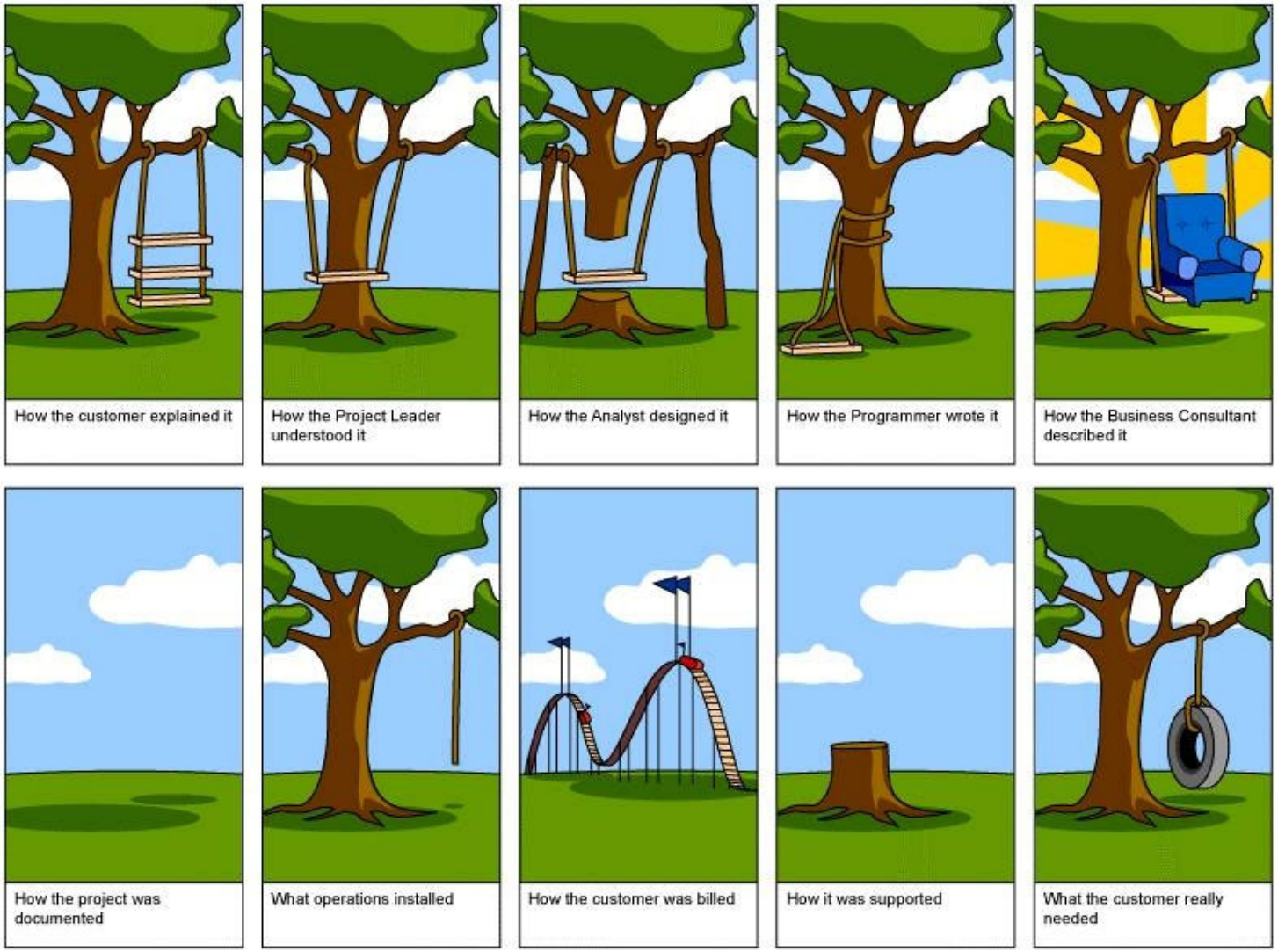
Source: Hoffer, Ramesh, Topi: Modern Database Management

BASICS
DEVELOPMENT CYCLE FOR DB APPLICATIONS

1. Design database by logic
   → What must be in the database?
2. Design database for system
   → How should the data be saved?
3. Develop database applications
   → How is the data processed?
4. Fill database
   →How to insert the data?
5. Maintain database
   → the database runs and runs and runs…



BASICS
DEVELOPMENT CYCLE FOR DB APPLICATIONS



Source: http://www.jhouseconsulting.com/jhouseconsulting/wp-content/uploads/2008/10/howthecustomerexplainedit.jpg

BASICS
THE ANSI-SPARC ARCHITECTURE

- The ANSI-SPARC Architecture is an abstract design standard for a Database Management System (DBMS), first proposed in 1975
- ANSI = American National Standards Institute
  SPARC = Standards Planning And Requirements Committee
- The ANSI-SPARC model however never became a formal standard
- No mainstream DBMS systems are fully based on it (they tend not to exhibit full physical independence or to prevent direct user access to the conceptual level), but the idea of logical data independence is widely adopted

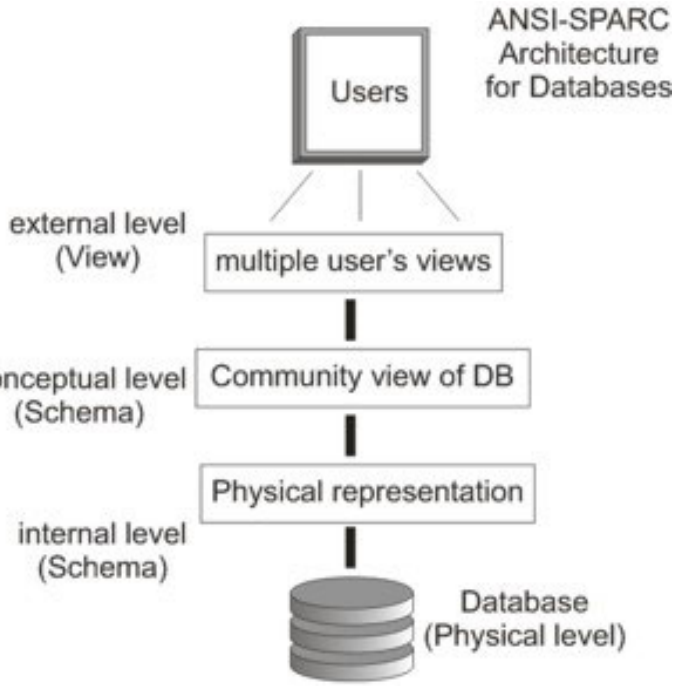
Source: www.wikipedia.org,
https://www.youtube.com/watch?v=6SH5FLccHi0

BASICS
THE ANSI-SPARC ARCHITECTURE

External Schema

Conceptual Schema

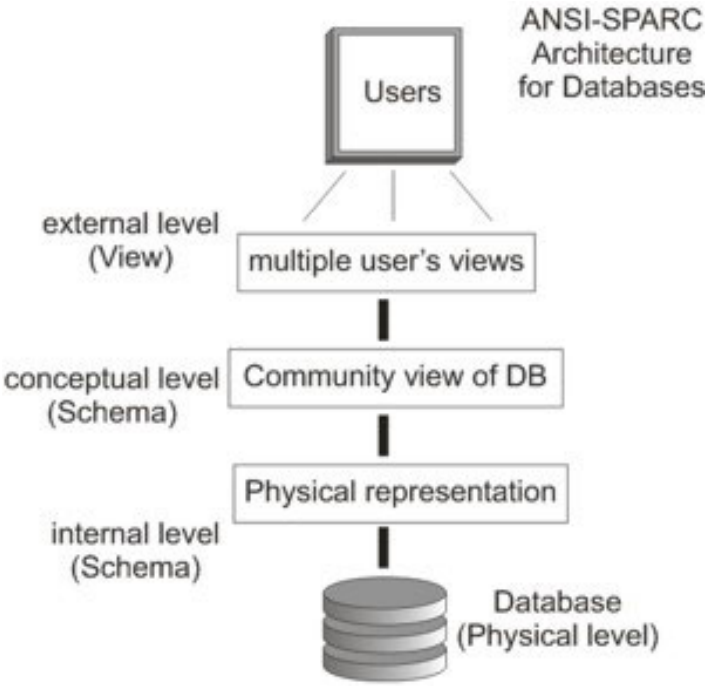
Internal Schema



Source: www.wikipedia.org

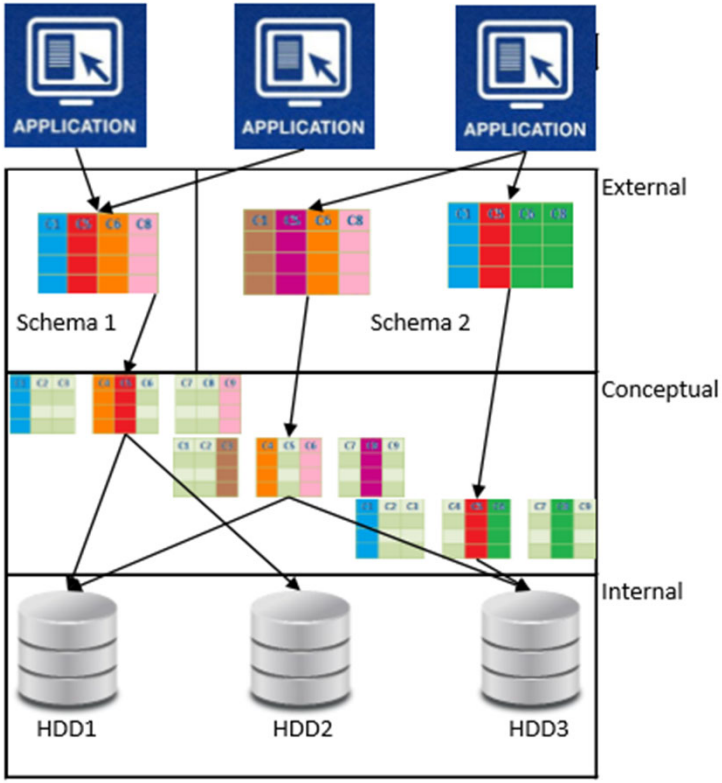
BASICS
THE ANSI-SPARC ARCHITECTURE

- The **external schemas** describe the different external views of the data and there may be many external schemas for a given database
- The **conceptual schema** describes all the data items and relationships between them, together with integrity constraints
  → There is only one conceptual schema per database
- The **internal schema** at the lowest level contains definitions of the stored records, the methods of representation, the data fields, and indexes
  → There is only one internal schema per database



Source: www.wikipedia.org

BASICS
THE ANSI-SPARC ARCHITECTURE



Source: https://stackoverflow.com/questions/9771884/ansi-sparc-practical-explanation

BASICS
DATA INDEPENDENCE

- Database Systems (and its data) must be accessible for a long time
  - Example: Insurance company (decades)
  - Often longer than the lifetime of applications, operating systems, hardware

- Need for data independence
  - Avoid tight coupling of applications, data, and operating environment
  - Physical data independence
  - Logical data independence

BASICS
DATA INDEPENDENCE - PHYSICAL DATA INDEPENDENCE

- **Definition**: Ability to modify the physical storage structures or locations of data without affecting the logical view of the data
- **Examples**:
  - If you change the underlying hardware, file systems, storage devices, or even relocate the database to a different server, applications interacting with the database should remain unaffected
  - Means, the physical implementation details (e.g., storage format, indexing methods) are hidden from the application layer.
- **Benefits**:
  - Ease of adapting to changing technology or infrastructure without rewriting application code
  - Allows for performance optimization without altering the application's logical data access methods

BASICS
DATA INDEPENDENCE - PHYSICAL DATA INDEPENDENCE

- The logical view on DB remains the same, even if method for storage changes

- Example: A database system has been set up to manage all students at the HAW Hamburg. This is based on an Oracle-DBMS and stores the data on a hard disk in the data center. Even in ten years the data will still be interesting / relevant although the hard drive is no longer available, or the disk is no longer located in the computer center.

- Applications are independent of the method of physical storage
  - Will there be magnetic hard drives in 10 years?

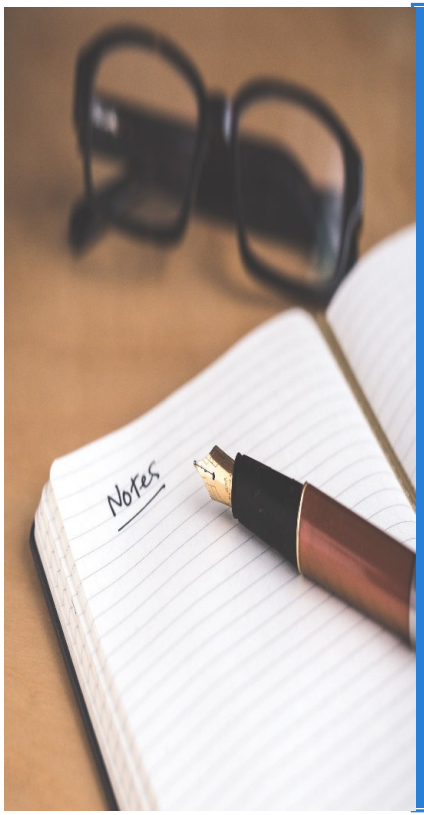
BASICS
DATA INDEPENDENCE - LOGICAL DATA INDEPENDENCE

- **Definition**: Ability to change the logical schema (table structures, relationships, and attributes) without impacting the applications that use the database
- **Examples**:
  - Suppose you want to add a new column to a table or modify the relationships between tables in your database.
    With logical data independence, you can make these changes without requiring modifications to the application code that interacts with the database.
    The application continues to work with the data as before, with the logical changes abstracted from it
- **Benefits**:
  - Provides flexibility in adapting the database schema to evolving business requirements without causing disruption to the applications using the database
  - Simplifies the process of making changes to the data model & maintains data consistency

BASICS
DATA INDEPENDENCE - LOGICAL DATA INDEPENDENCE

- The logical view of the DB remains the same, even if the application interface changes
- Each application only knows the relevant part of the data
- Data Independence: Applications are independent of changes to other parts of the data
- Example: For the administration of the students of HAW there is still the DBS, which stores the data in an Oracle-DBMS on a hard disk in the computer center. In 10 years, some applications will have to process additional attributes (e.g., 'part-time student'); This must not affect the functionality of the old programs; However, no data redundancy must be created, and everything must be stored in one DB!

BASICS
DATA INDEPENDENCE



If we have data independence, applications are robust against changes in the DB or the operational environment.

Source: Photo by David Travis on Unsplash

HOMEWORK

Each student must install a DBMS on
their home computer
(mySQL preferred, MariaDB, PostgresSQL, Oracle,...)!!!

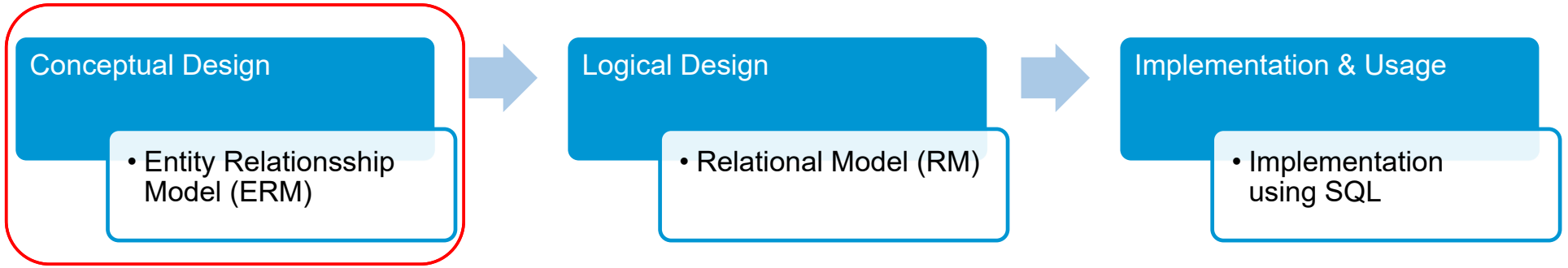
Source: Foto von K8 auf Unsplash

ORGANIZATION
OUR JOURNEY IN THIS SEMESTER

- Integrity, Trigger & Security
- Database Applications
- Transactions
- Subqueries & Views
- More Features
- Notations & Guidelines
- Constraints
- Relationships
- **Simple Entities and Attributes**
- Basics

Source: Foto von Justin Kauffman auf Unsplash

SIMPLE ENTITIES AND ATTRIBUTES
DATABASE DESIGN



SIMPLE ENTITIES AND ATTRIBUTES
ERM: ENTITY-RELATIONSHIP MODEL

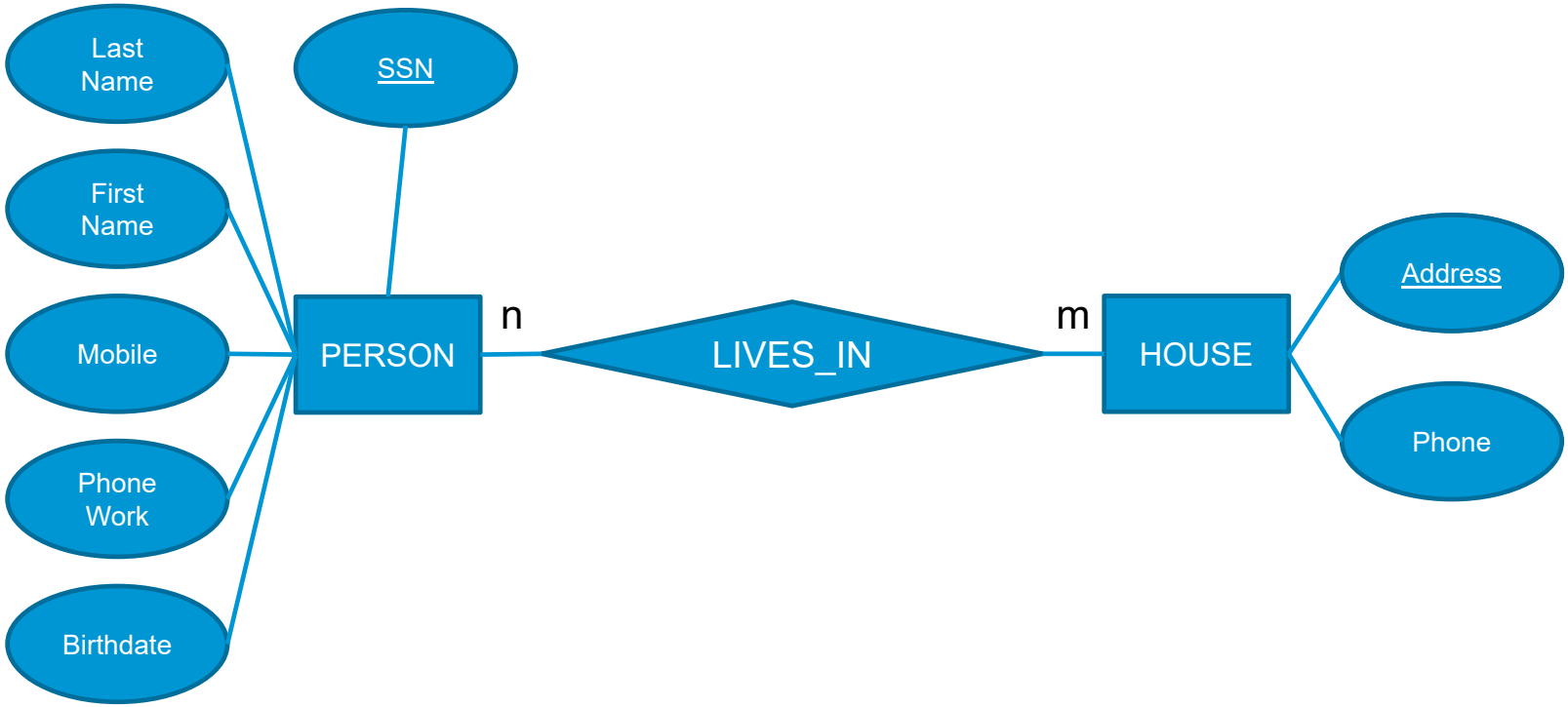
- Entity-Relationship-Model for logical draft of the database
- The focus is on the business requirements
- This language is not implemented in any DBMS

SIMPLE ENTITIES AND ATTRIBUTES
ERM: ENTITY-RELATIONSHIP MODEL

- Introduced by Peter Chen in 1976
- An ERM describes interrelated things of interest in a specific domain of knowledge
- A basic ERM is composed of entity types (which classify the things of interest) and specifies relationships that can exist between entities (instances of those entity types)
- Elements:
  - Entity: A distinguishable thing existing (physically or conceptually) in the real world.
  - Relationship: Between entities
  - Attribute: Property of an entity or relationship

SIMPLE ENTITIES AND ATTRIBUTES
ERM: ENTITY-RELATIONSHIP MODEL - EXAMPLE CONTACT LIST

1. Conceptual design with ER Model

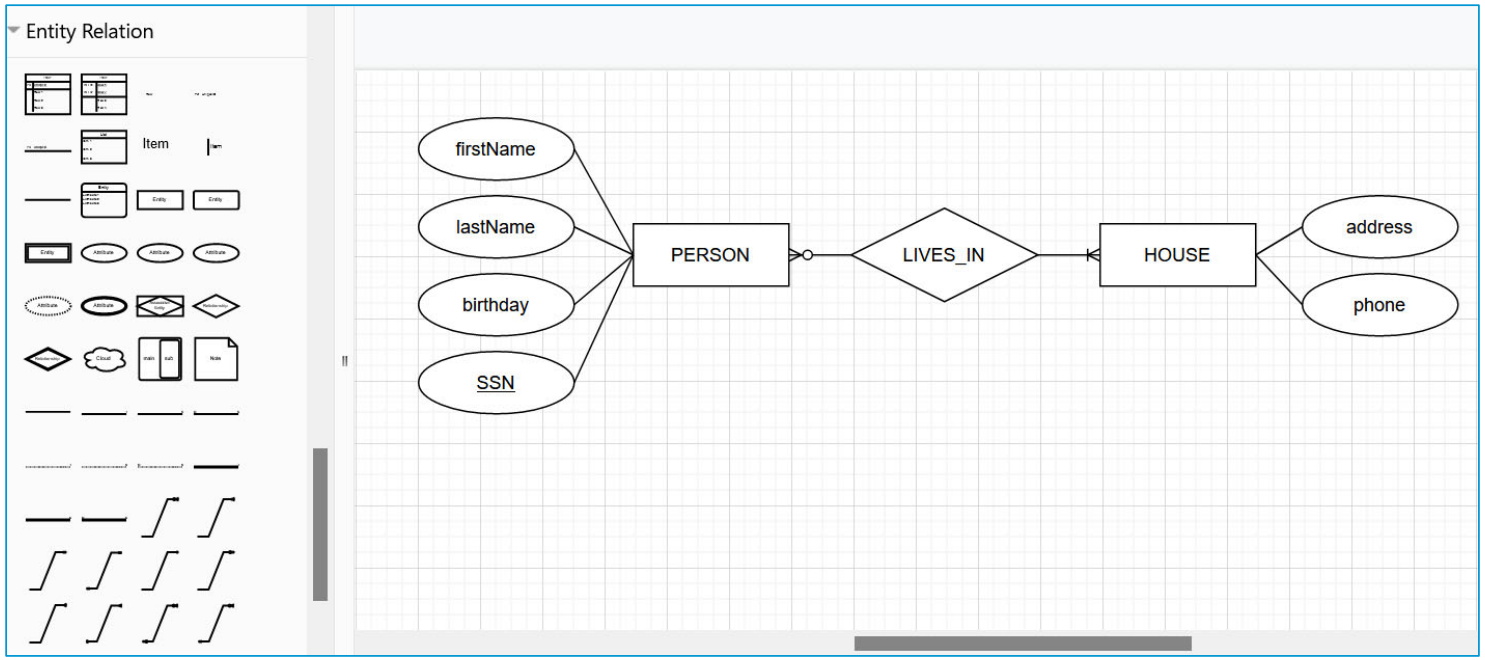


SIMPLE ENTITIES AND ATTRIBUTES
ERM: ENTITY-RELATIONSHIP MODEL

- Entity Type
  - Represented as rectangle in ERM
  - Singular noun
- Attribut Type
  - Represented as ovals
  - Noun
- Relationship Type
  - Represented as Diamond in ERM
  - Always between entities
  - Verb
  - Has cardinalities

SIMPLE ENTITIES AND ATTRIBUTES
ERM: ENTITY-RELATIONSHIP MODEL - TOOLS

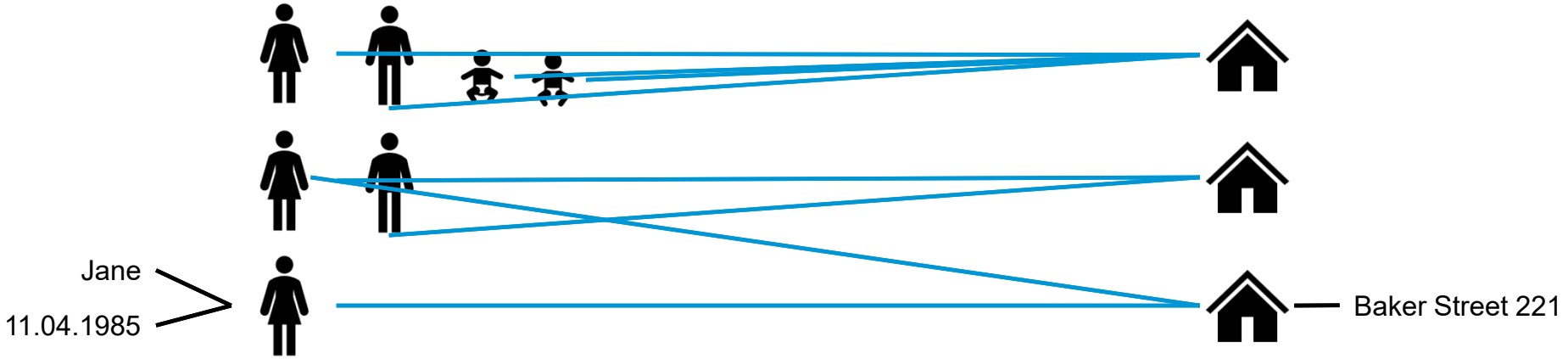
- Any drawing tool, paper or whiteboard
- Examples for draing tools
  - Draw.io
  - Lucidchart
  - Creately
  - GitMinds
  - Gliffey
  - …



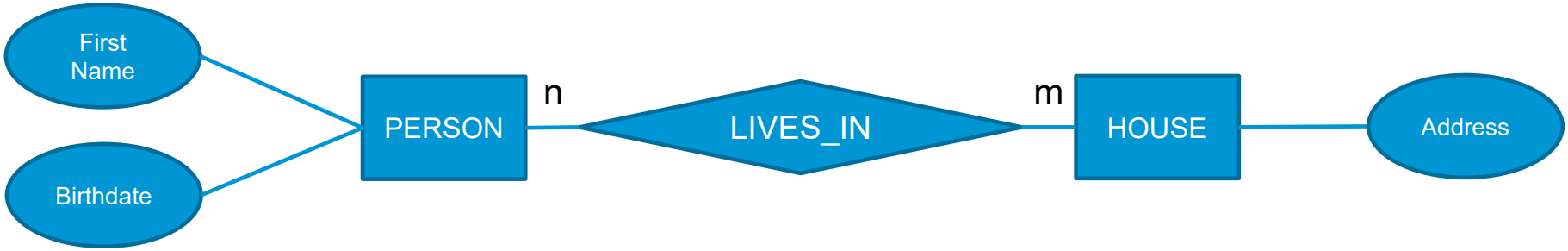
Example with Draw.io

SIMPLE ENTITIES AND ATTRIBUTES
ERM: ENTITY-RELATIONSHIP MODEL - EXAMPLE

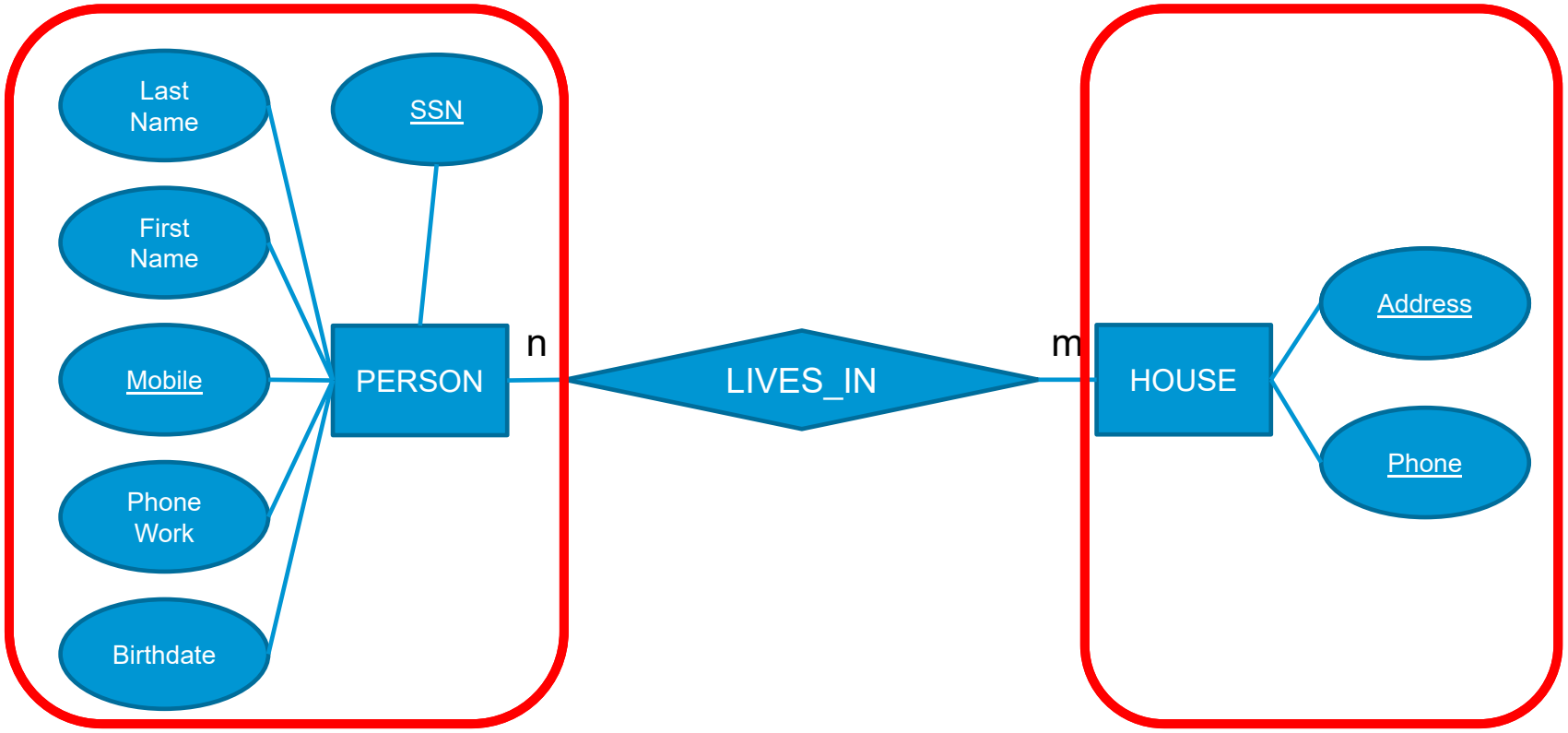
Some entities (in the Real World)



Entity types (abstraction of the real world)



SIMPLE ENTITIES AND ATTRIBUTES
ERM: ENTITY-RELATIONSHIP MODEL - EXAMPLE



SIMPLE ENTITIES AND ATTRIBUTES
ERM: ENTITY-RELATIONSHIP MODEL

- Entity: A distinguishable thing existing (physically or conceptually) in the real world

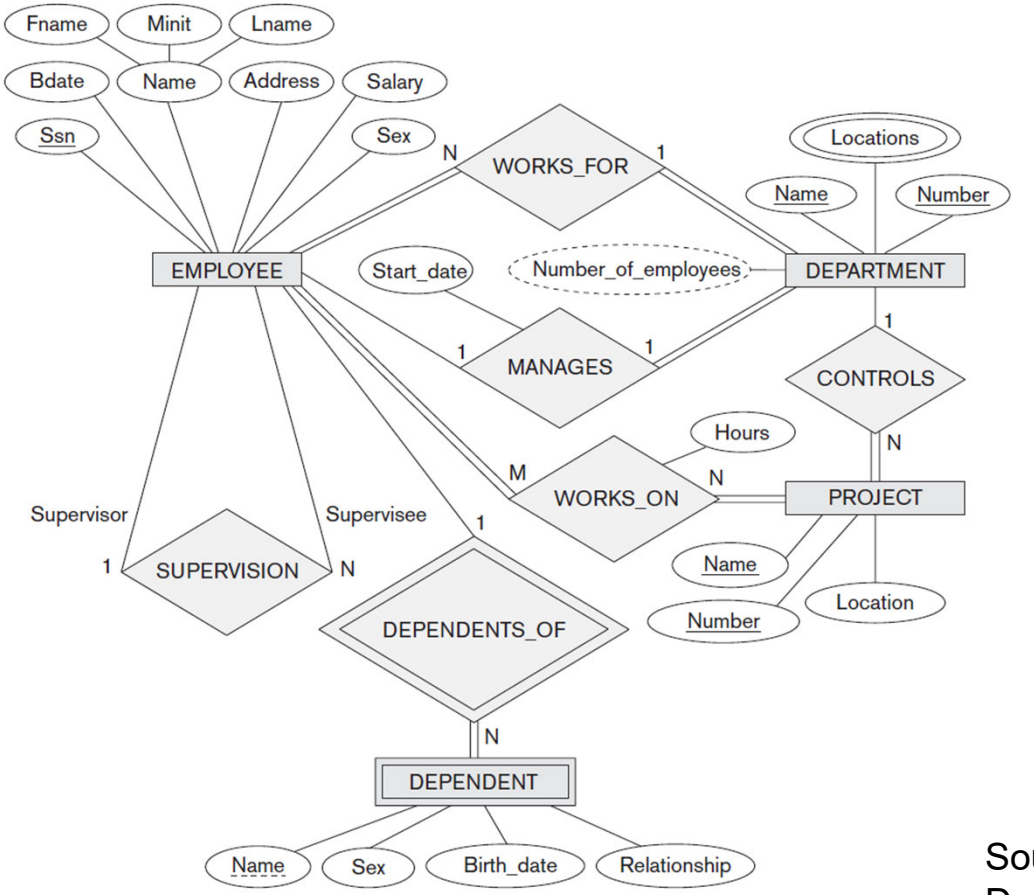
- Abstraction of entities → Entity Type
  (Compare to: Objects → Class)
- Several entities → Entity Set
- Abstraction of relationships → Relationship Type

SIMPLE ENTITIES AND ATTRIBUTES
ERM: CONTINUOUS EXAMPLE - COMPANY

- The company is organized into departments
- Each department has a unique name, a unique number, a manager (employee) with start date, and several locations
- A department controls a number of projects, each with unique name, unique number, single location
- We store each employee's name, ssn, address, salary, sex, birthdate
- An employee is assigned to one department, but may work on several projects, also from other departments
- We keep track of the hours per week per project
- We also keep track of the supervisor
- We want to keep track of each employee's dependents for insurance purposes, namely first name, sex, birth date, and relationship to employee.

Source: Elmasri, Fundamentals of Database Systems, Page 204 ff

SIMPLE ENTITIES AND ATTRIBUTES
ERM: CONTINUOUS EXAMPLE - COMPANY



Source: Elmasri, Fundamentals of Database Systems, Page 204 ff

SIMPLE ENTITIES AND ATTRIBUTES
ERM: SIMPLE ENTITIES

- Entity Type is a basic object in an ERM
- Represents a thing in the real world with an indepedent existence, e.g. person, car, job, company, university course
- Has attributes, e.g. name, address, age
- A particular entity will have a value for each of its attributes

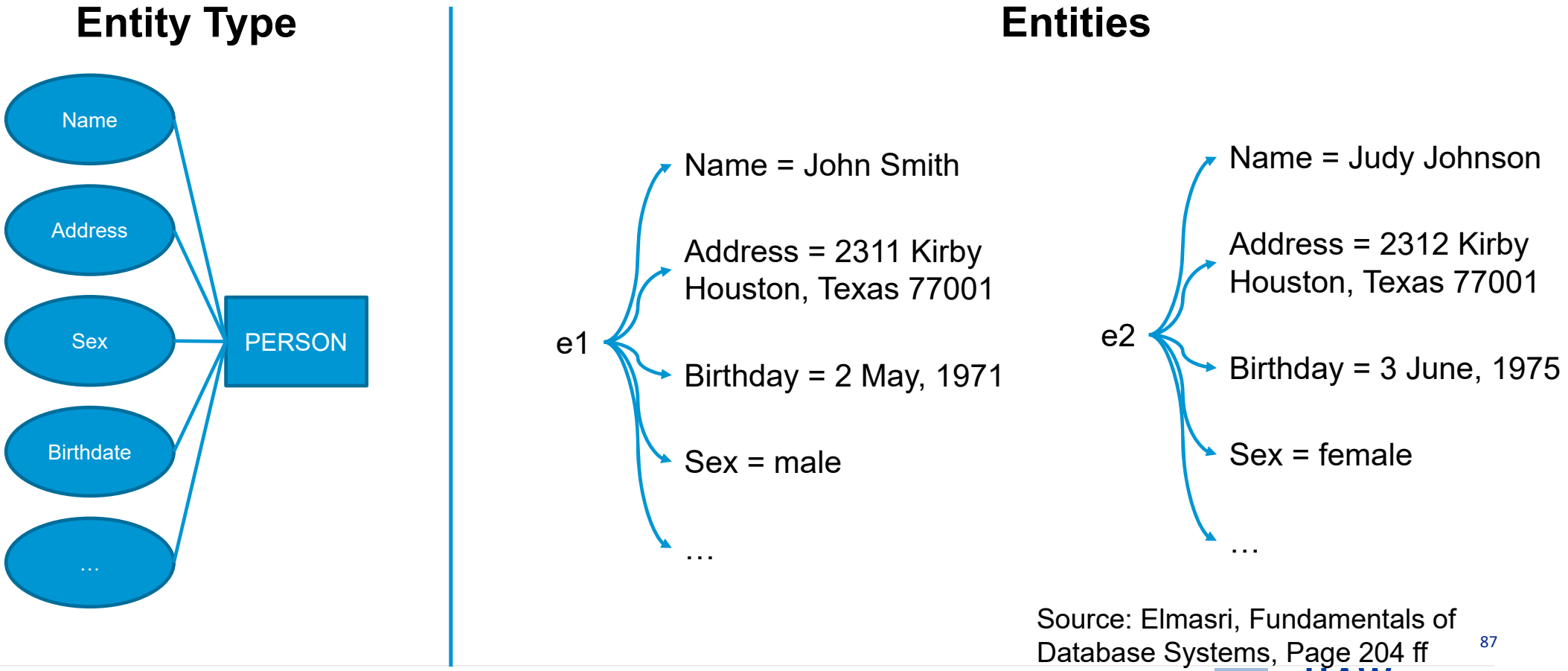
Source: Elmasri, Fundamentals of Database Systems, Page 204 ff

SIMPLE ENTITIES AND ATTRIBUTES
ERM: SIMPLE ENTITIES

- An **entity type** defines a collection (or set) of entities that have the same attributes
- Each entity type is described by its name and attributes
- The collection of all entities of a particular entity type is called an **entity set**

Source: Elmasri, Fundamentals of Database Systems, Page 204 ff

SIMPLE ENTITIES AND ATTRIBUTES
ERM: SIMPLE ENTITIES



Source: Elmasri, Fundamentals of Database Systems, Page 204 ff

SIMPLE ENTITIES AND ATTRIBUTES
ERM: SIMPLE ENTITIES

- Identification of actual objects
  → Could be real world objects, persons, forms
- Categories
  - Actual objects
  - Persons, Roles, Organizations
  - Actions
  - Interfaces
  - General Information
- Good name for Entity Type?
- Not an Entity Type, if
  - No attributes nor relationships
  - Contains same attributes as another entity type (not distinguishable)

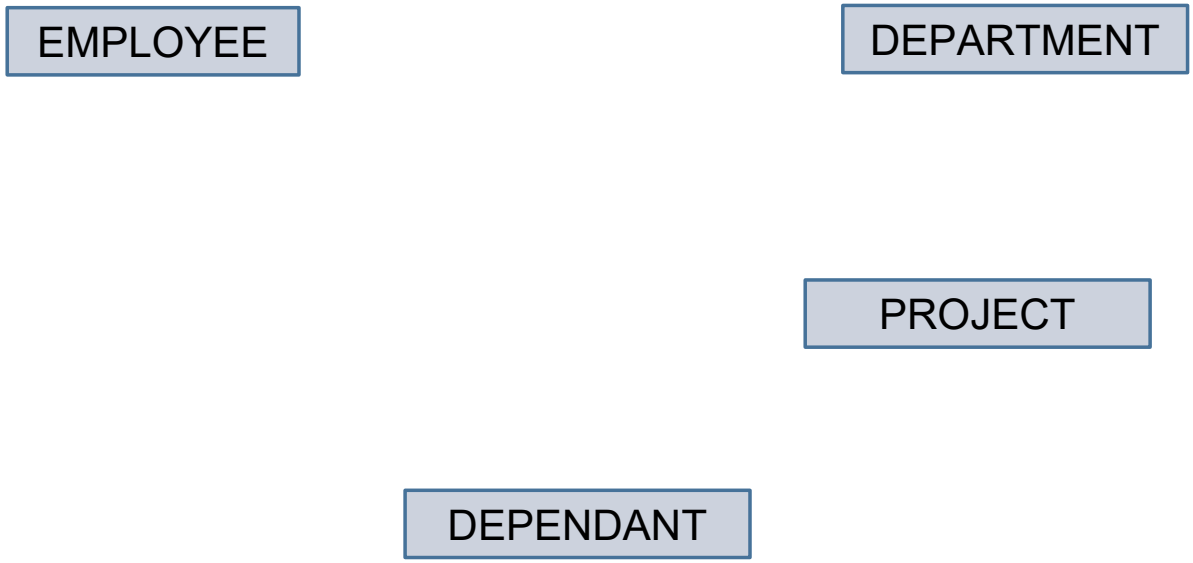
SIMPLE ENTITIES AND ATTRIBUTES
ERM: SIMPLE ENTITIES - CONTINUOUS EXAMPLE - COMPANY

What are the entity types?

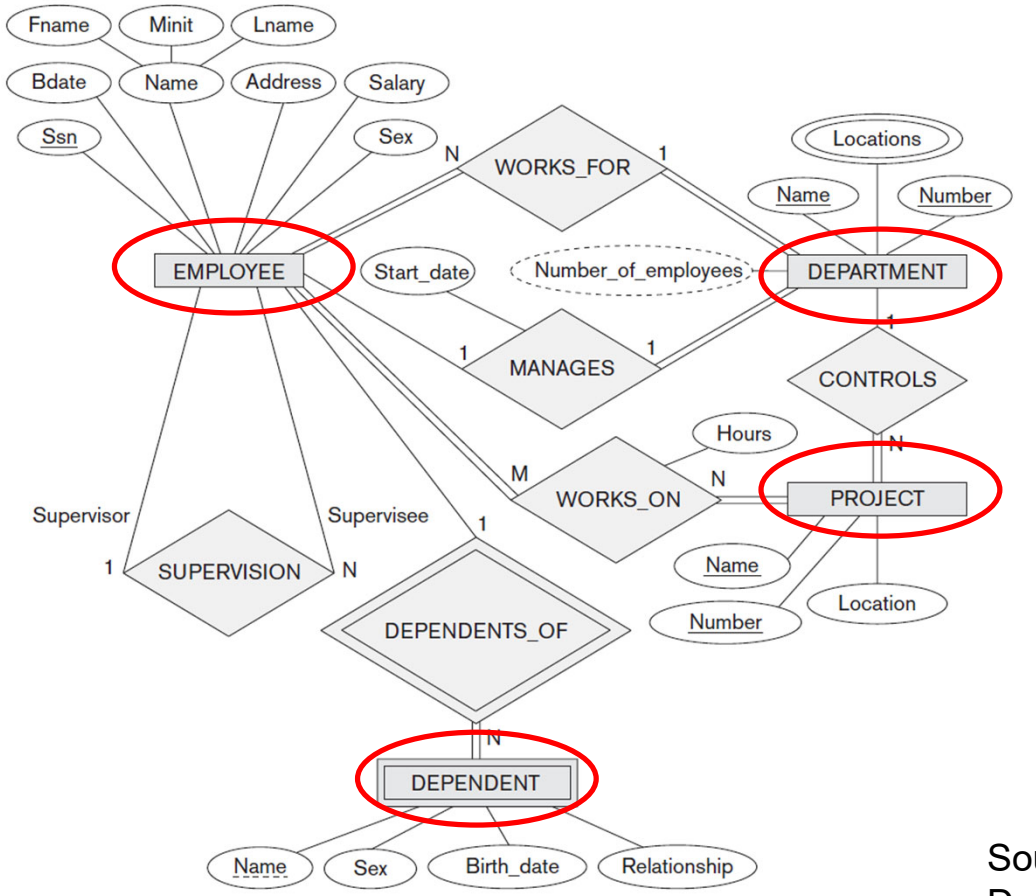
- The company is organized into departments
- Each department has a unique name, a unique number, a manager (employee) with start date, and several locations
- A department controls a number of projects, each with unique name, unique number, single location
- We store each employee's name, ssn, address, salary, sex, birthdate
- An employee is assigned to one department, but may work on several projects, also from other departments
- We keep track of the hours per week per project
- We also keep track of the supervisor
- We want to keep track of each employee's dependents for insurance purposes, namely first name, sex, birth date, and relationship to employee.

Source: Elmasri, Fundamentals of Database Systems, Page 204 ff

SIMPLE ENTITIES AND ATTRIBUTES
ERM: SIMPLE ENTITIES



SIMPLE ENTITIES AND ATTRIBUTES
ERM: SIMPLE ENTITIES



Source: Elmasri, Fundamentals of Database Systems, Page 204 ff

SIMPLE ENTITIES AND ATTRIBUTES
ERM: ATTRIBUTES

- Is the attribute relevant for the problem?
- An attribute must belong to an Entity Type (or a Relationship Type)
- Key Attributes (identifying attributes)?
- Good name
  → Unique name within Entity Type, not necessarily in model

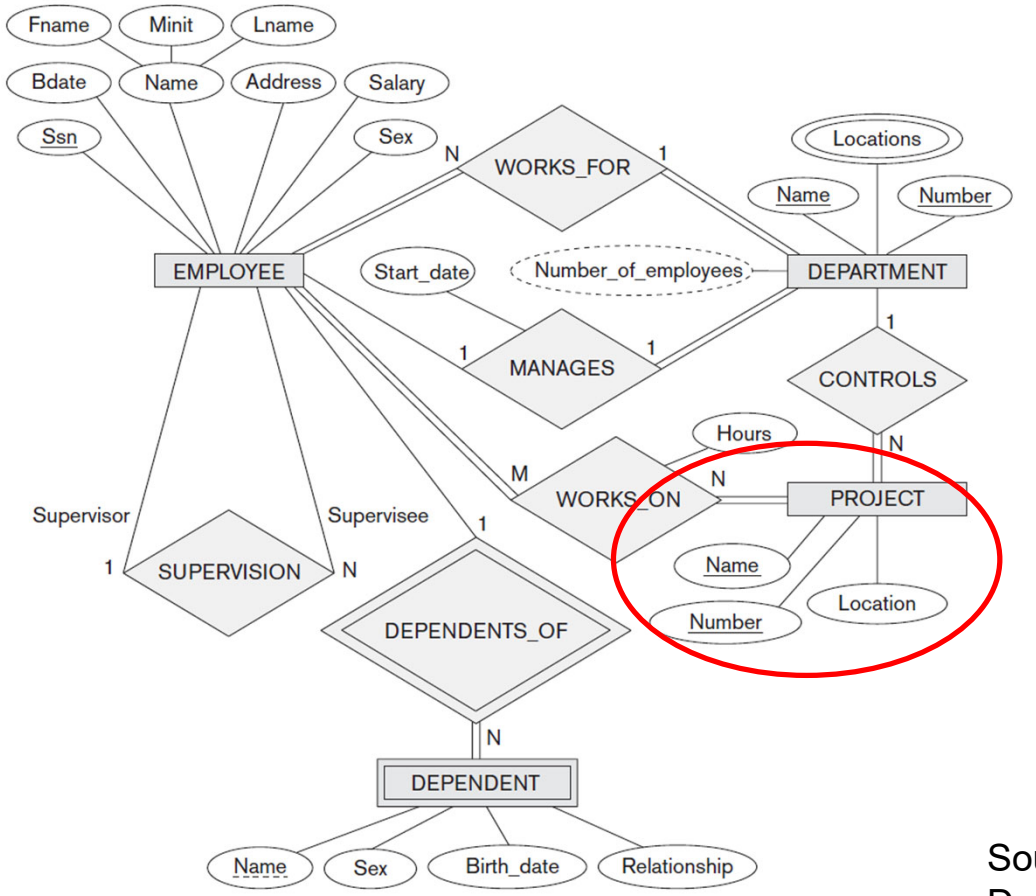
SIMPLE ENTITIES AND ATTRIBUTES
ERM: ATTRIBUTES

What could be the attributes of entity type PROJECT?
What could be the identifying (key) attributes?

- …
- A department controls a number of projects, each with unique name, unique number, single location
- …

Source: Elmasri, Fundamentals of Database Systems, Page 204 ff

SIMPLE ENTITIES AND ATTRIBUTES
ERM: ATTRIBUTES



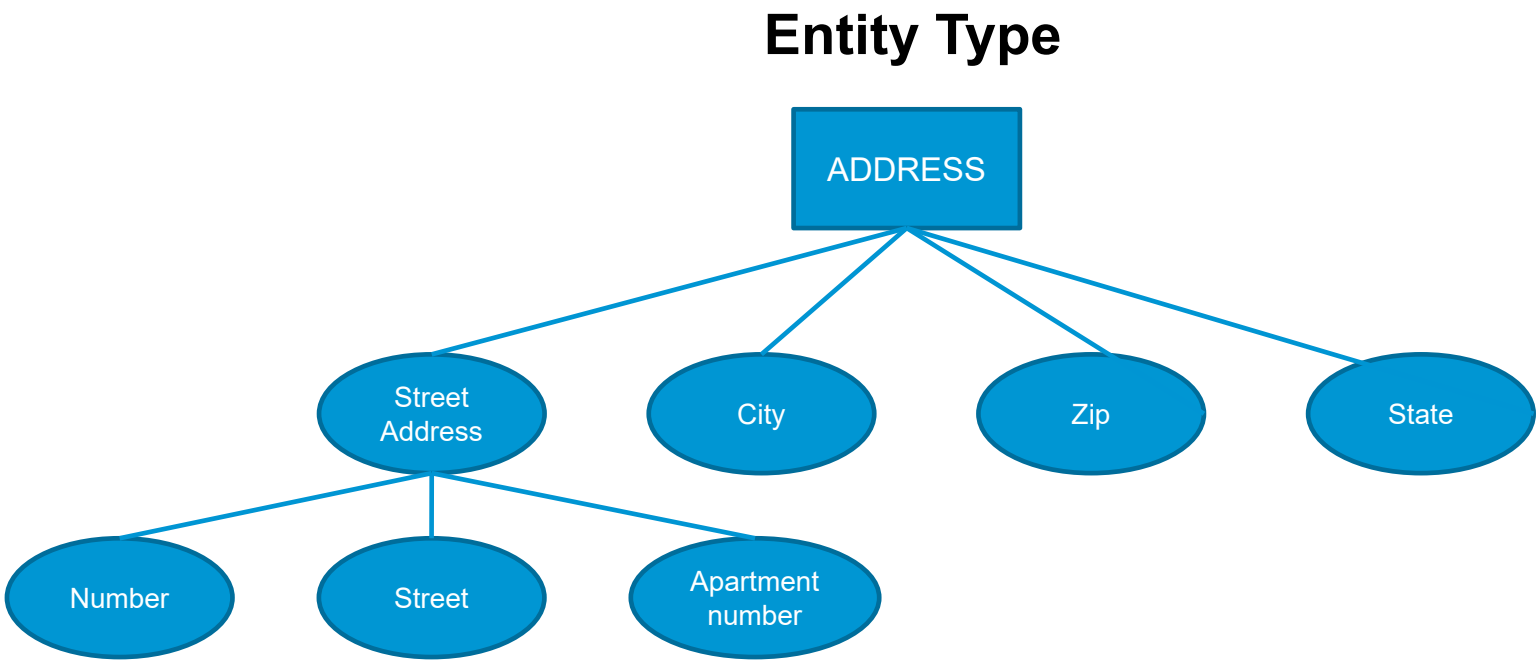
Source: Elmasri, Fundamentals of Database Systems, Page 204 ff

SIMPLE ENTITIES AND ATTRIBUTES
ERM: ATTRIBUTES – COMPOSITE ATTRIBUTES

- Composite vs. Simple (atomic) attributes
  - Attributes which are not divisible are called *simple* or *atomic attributes*
  - Composite attributes can form a hierarchy
  - Composite attributes are useful to model situations in which a user sometimes refers to the composite attribute as a unit but at other times refers specifically to its components
  - If the composite attribute is referenced only as a whole, there is no need to subdivide it into component attributes
  - Composite attributes are attached to their component attributes by straight lines

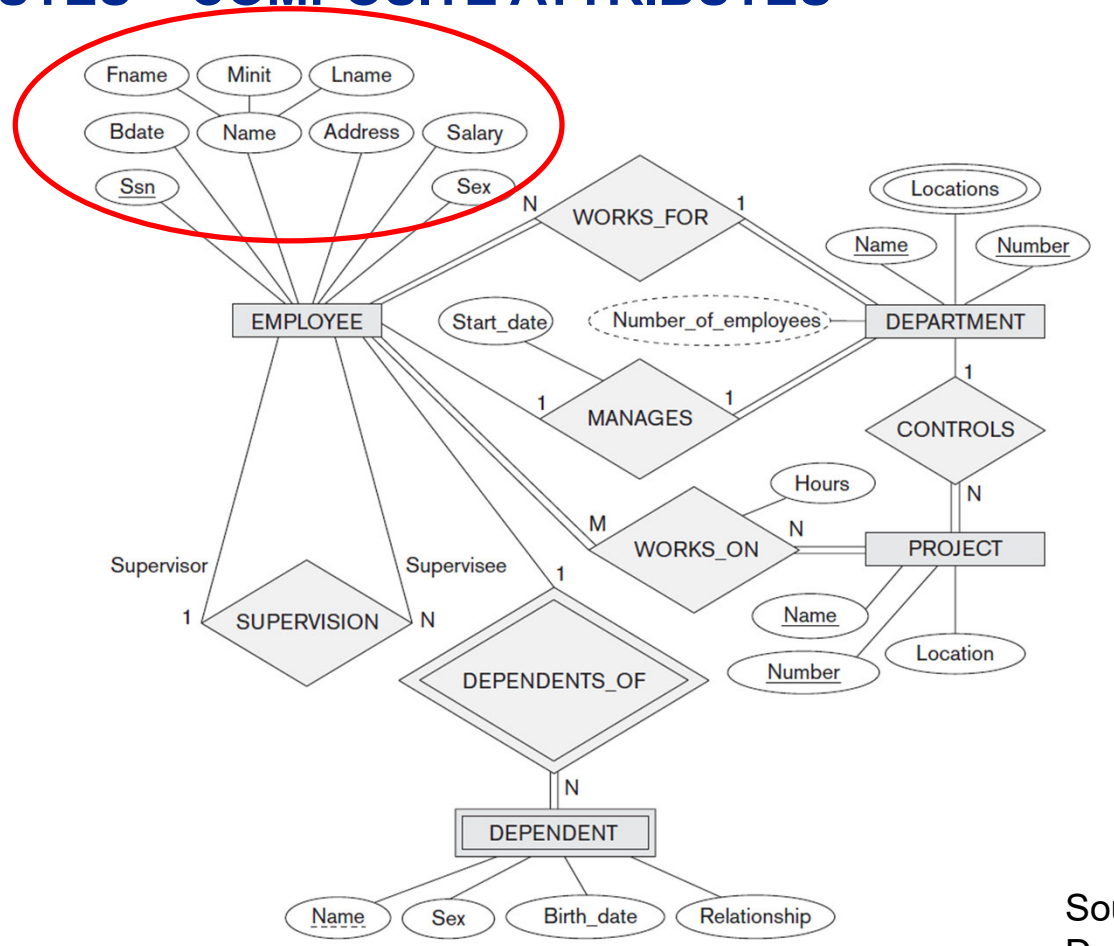
Source: Elmasri, Fundamentals of Database Systems, Page 204 ff

SIMPLE ENTITIES AND ATTRIBUTES
ERM: ATTRIBUTES – COMPOSITE ATTRIBUTES



Source: Elmasri, Fundamentals of Database Systems, Page 204 ff

SIMPLE ENTITIES AND ATTRIBUTES
ERM: ATTRIBUTES – COMPOSITE ATTRIBUTES



Source: Elmasri, Fundamentals of Database Systems, Page 204 ff

SIMPLE ENTITIES AND ATTRIBUTES
ERM: ATTRIBUTES – KEY ATTRIBUTES (IDENTIFYING ATTRIBUTES)

- How can we identify an actual entity within an entity set?
- Attributes must be used
  → Key Attributes (also called identifying attributes)
- Sometimes several attributes together form a key attribute (identifying attribute), meaning that the combination of the attribute values must be distinct for each entity
  - If a set of attributes possesses this property, the proper way to represent this in the ER model that is to define a composite attribute and designate it as a key attribute of the entity type
  - Notice that such a composite key attributes must be minimal; that is, all component attributes must be included in the composite attribute to have the uniqueness property
- Key attributes are underlined
- If two attributes are underlined separately, then each is an identifying attribute on its own

Source: Elmasri, Fundamentals of Database Systems, Page 204 ff

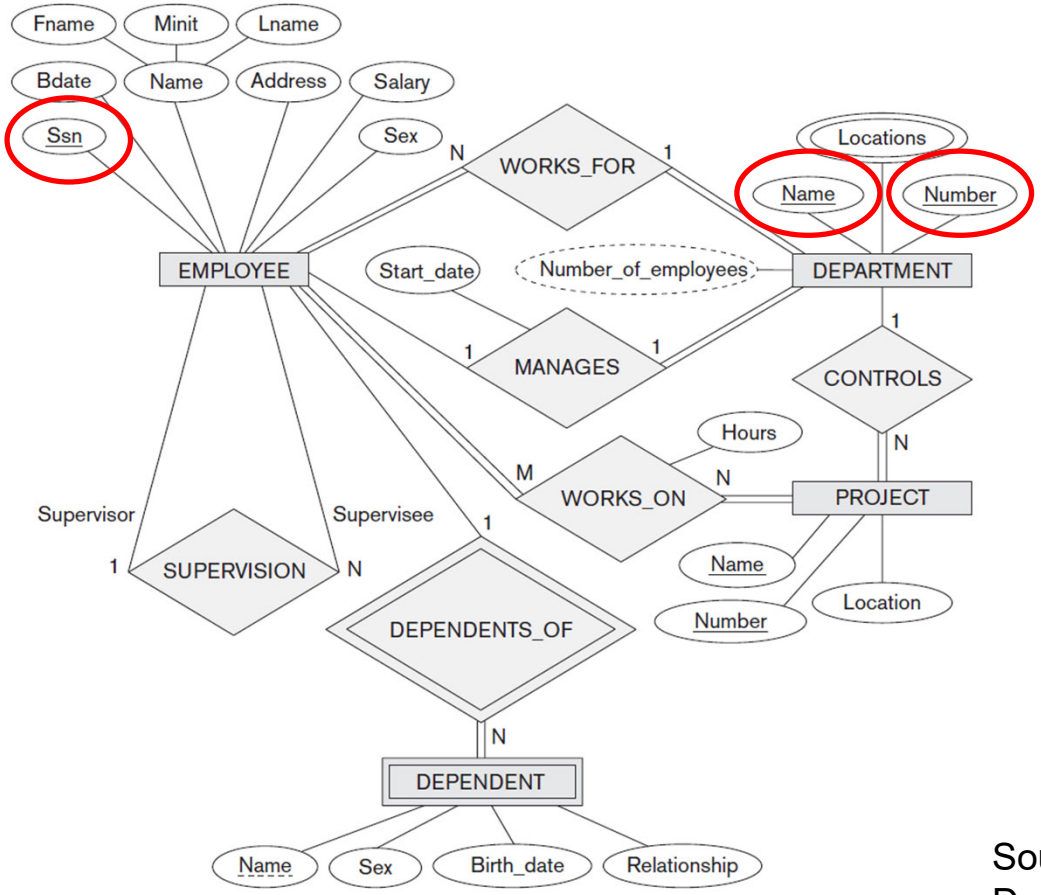
SIMPLE ENTITIES AND ATTRIBUTES
ERM: ATTRIBUTES – KEY ATTRIBUTES (IDENTIFYING ATTRIBUTES)

What are key attributes for entity type EMPLOYEE and DEPARTMENT?

- The company is organized into departments
- Each department has a unique name, a unique number, a manager (employee) with start date, and several locations
- A department controls a number of projects, each with unique name, unique number, single location
- We store each employee's name, ssn, address, salary, sex, birthdate
- An employee is assigned to one department, but may work on several projects, also from other departments
- We keep track of the hours per week per project
- We also keep track of the supervisor
- We want to keep track of each employee's dependents for insurance purposes, namely first name, sex, birth date, and relationship to employee.

Source: Elmasri, Fundamentals of Database Systems, Page 204 ff

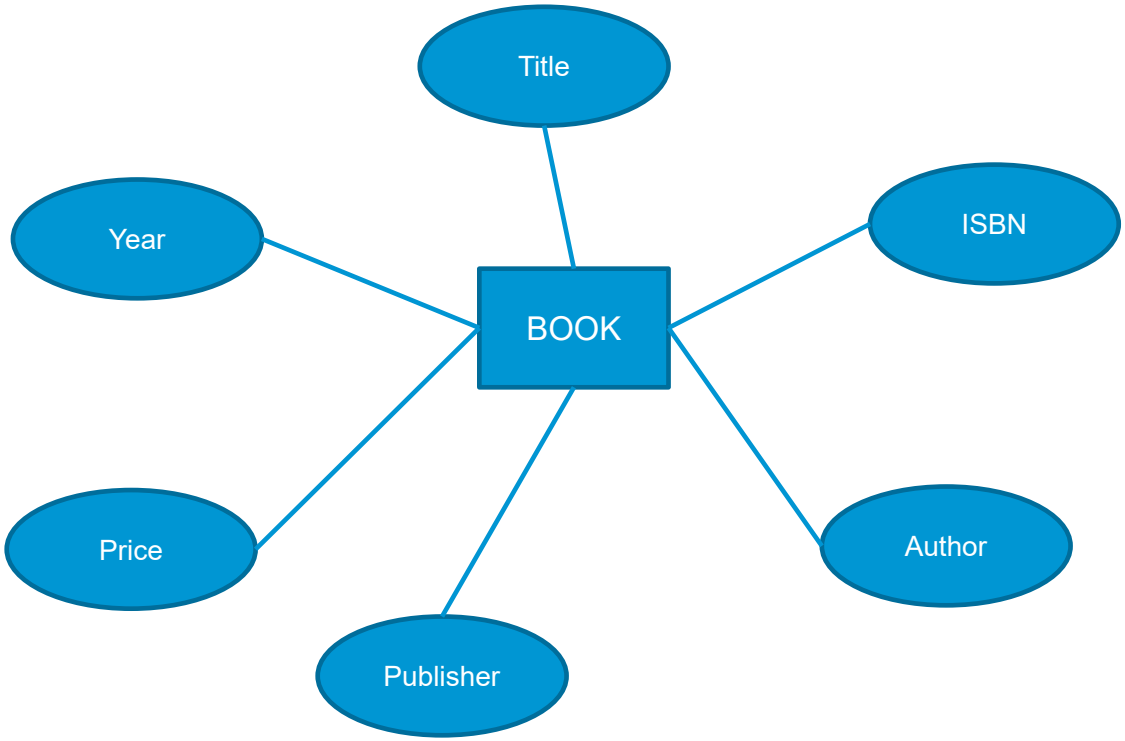
SIMPLE ENTITIES AND ATTRIBUTES
ERM: ATTRIBUTES – KEY ATTRIBUTES (IDENTIFYING ATTRIBUTES)



Source: Elmasri, Fundamentals of Database Systems, Page 204 ff

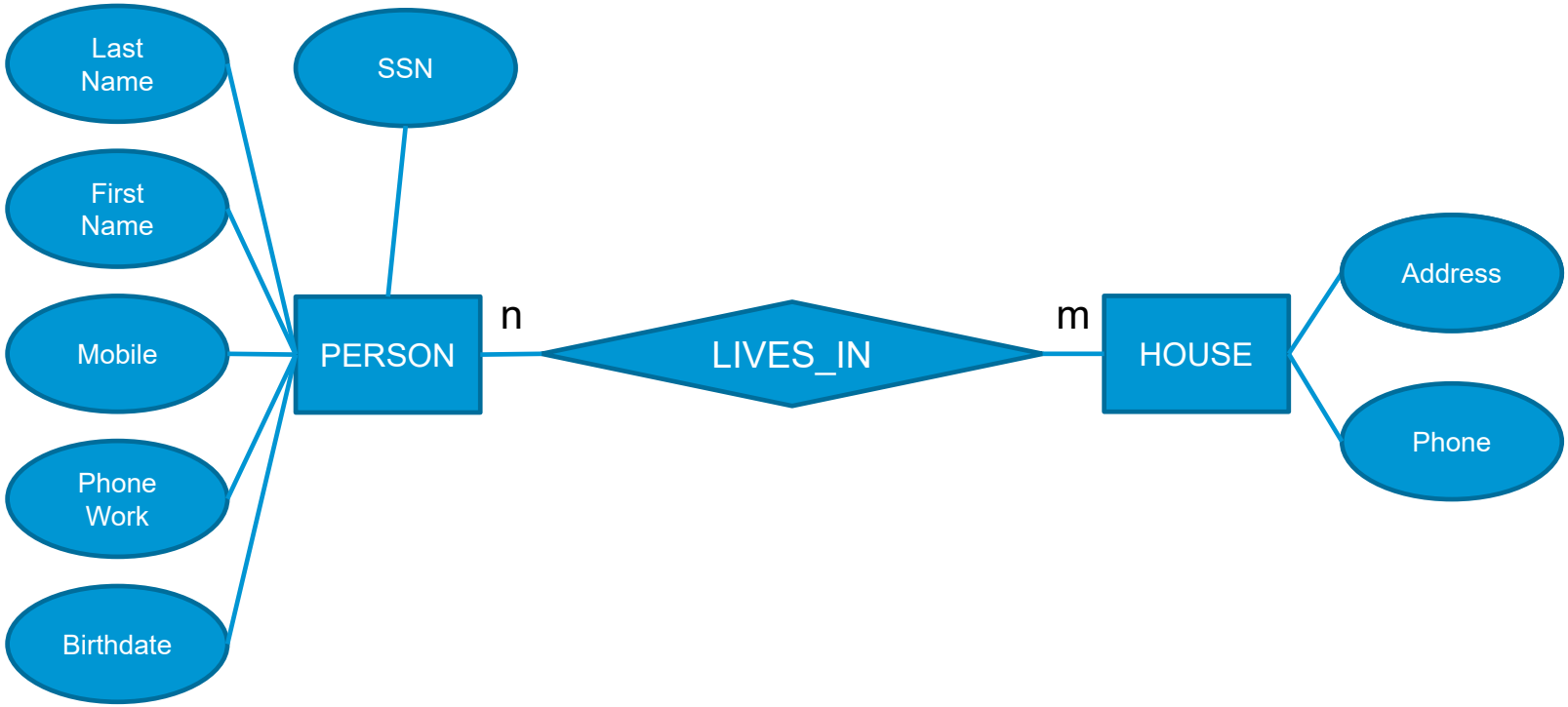
SIMPLE ENTITIES AND ATTRIBUTES
ERM: ATTRIBUTES – KEY ATTRIBUTES (IDENTIFYING ATTRIBUTES)

What are key attributes for entity type BOOK?



SIMPLE ENTITIES AND ATTRIBUTES
ERM: ATTRIBUTES – KEY ATTRIBUTES (IDENTIFYING ATTRIBUTES)

What are key attributes for entity type PERSON and HOUSE?



SIMPLE ENTITIES AND ATTRIBUTES
RM: DATABASE DESIGN