Erhvervsakademi og
Professionshøjskole

Databases

Software Development
OPBSW20FD1

Bernadette Besenhofer
berb@ucl.dk

Communication

Talk to me :)

Email me at: berb@ucl.dk

Messages on *itslearning* might be answered with delay

Check the plan on itslearning

Check the message board on itslearning (before lessons) !

Today

01

Agenda

- Intro
- Relational Data Model
- Database Design
- Entity-Relationship Diagrams

Book Cp. 1

Intro - Databases and DBMS

DATA - Working with Data

Problems

1. Size
2. Ease of updating
3. Accuracy
4. Security
5. Redundancy
6. Importance

Solutions

1. Scalable
2. Accessible
3. Accurate
4. Secure
5. Consistent
6. Permanent

What is a ...

Database

A database is a collection of related information that is organized so that it can be easily accessed, managed and updated.

Database Management System (DBMS)

A database management system (DBMS) is a computerized system that enables users to create and maintain a database.

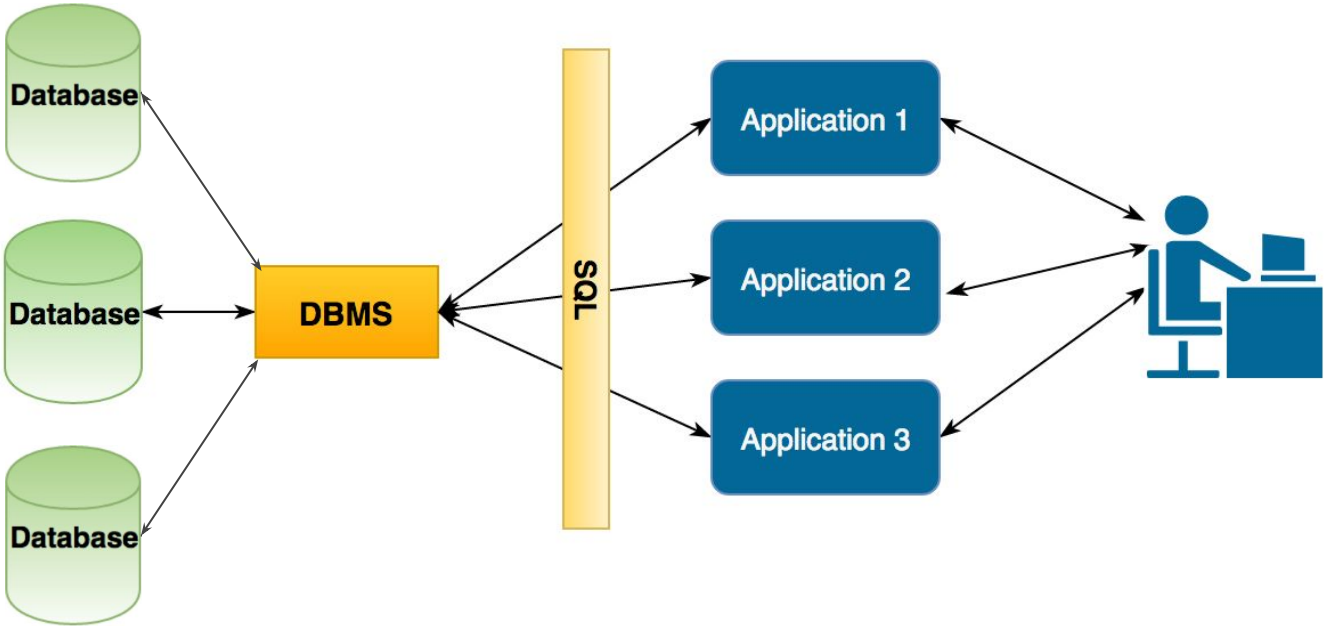
For example MySQL, PostgreSQL, MS SQL Server, Oracle, ...

MySQL ORACLE DATABASE Microsoft SQL Server PostgreSQL MariaDB

Database Management System

DBMS provide functionality for:

- Data definition
- Update
- Retrieval
- Administration



When and why to use a Database

Advantages

- Can store very large numbers of records efficiently
- It is very quick and easy to find information
- It is easy to add new data and to edit or delete old data
- Data can be searched and sorted easily
- Data can be used in different applications
- More than one person can access the same database at the same time - multi-access
- Security

Disadvantages

- Database systems are complex, difficult, and time-consuming to design
- Substantial hardware and software start-up costs
- Damage to database affects all applications using it
- Extensive conversion costs in moving from a file-based system to a database system
- Initial training required for all programmers and users

History of Databases

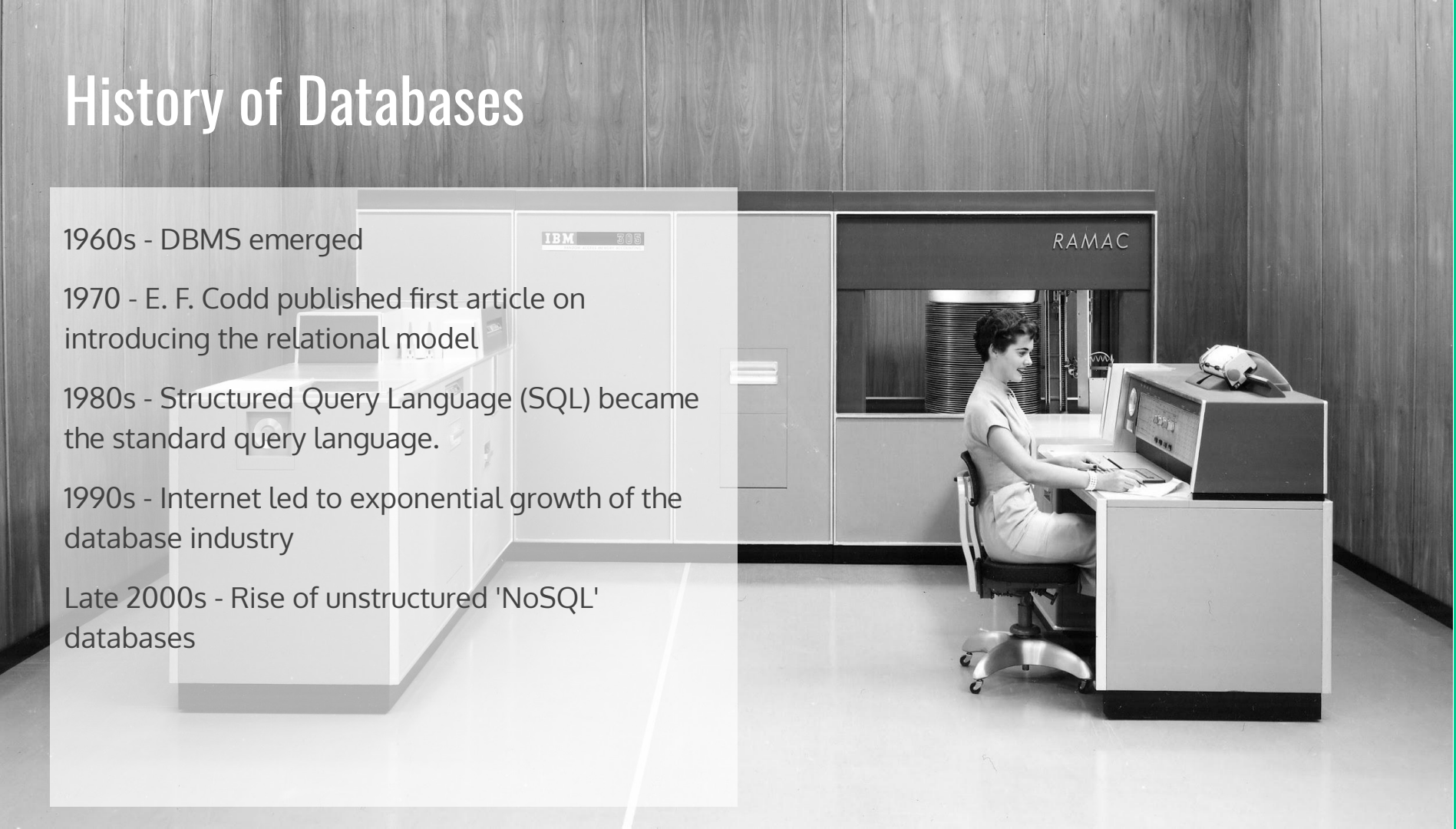
1960s - DBMS emerged

1970 - E. F. Codd published first article on introducing the relational model

1980s - Structured Query Language (SQL) became the standard query language.

1990s - Internet led to exponential growth of the database industry

Late 2000s - Rise of unstructured 'NoSQL' databases



Database Types

- Hierarchical databases
  (Data is organized into a tree-like structure, similar to file system)
- **Relational databases**
  (Data structured to recognize relations between stored items of information - **most used type**)
- Object-oriented databases
  (Information is represented in the form of objects as used in object-oriented programming)
- **NoSQL databases**

Data Interchange - XML

XML

https://developer.mozilla.org/en-US/docs/XML_Introduction

- Stands for Extensible Markup Language
- Author of the document defines the markup elements
- Markup language used to create documents of hierarchical structure
- Used for sharing structured information between systems, apps, people, ...
- Optional use of schema/validation

```
<students>
  <student>
    <name>John</name> <age>23</age> <city>Agra</city>
  </student>
  <student>
    <name>Steve</name> <age>28</age> <city>Delhi</city>
  </student>
  <student>
    <name>Peter</name> <age>32</age> <city>Chennai</city>
  </student>
  <student>
    <name>Chaitanya</name> <age>28</age> <city>Bangalore</city>
  </student>
</students>
```

Data Interchange - JSON

JSON
http://json.org/

- Stands for JavaScript Object Notation
- Lightweight data-interchange format
- "Self-describing" and easy to understand
- Language independent

```
{"students":[
   {"name":"John", "age":"23", "city":"Agra"},
   {"name":"Steve", "age":"28", "city":"Delhi"},
   {"name":"Peter", "age":"32", "city":"Chennai"},
   {"name":"Chaitanya", "age":"28", "city":"Bangalore"}
]}
```

DBMS Popularity Ranking

359 systems in ranking, August 2020

Rank			DBMS	Database Model	Score		
Aug 2020	Jul 2020	Aug 2019			Aug 2020	Jul 2020	Aug 2019
1.	1.	1.	Oracle	Relational, Multi-model	1355.16	+14.90	+15.68
2.	2.	2.	MySQL	Relational, Multi-model	1261.57	-6.93	+7.89
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	1075.87	+16.15	-17.30
4.	4.	4.	PostgreSQL	Relational, Multi-model	536.77	+9.76	+55.43
5.	5.	5.	MongoDB	Document, Multi-model	443.56	+0.08	+38.99
6.	6.	6.	IBM Db2	Relational, Multi-model	162.45	-0.72	-10.50
7.	↑ 8.	↑ 8.	Redis	Key-value, Multi-model	152.87	+2.83	+8.79
8.	↓ 7.	↓ 7.	Elasticsearch	Search engine, Multi-model	152.32	+0.73	+3.23
9.	9.	↑ 11.	SQLite	Relational	126.82	-0.64	+4.10
10.	↑ 11.	↓ 9.	Microsoft Access	Relational	119.86	+3.32	-15.47

http://db-engines.com/en/ranking

Relational Model

Relational Data Model

Relational Model was proposed by E.F. Codd to model data in the form of relations or tables.

Relational data model is the primary data model, which is used widely around the world for data storage and processing.

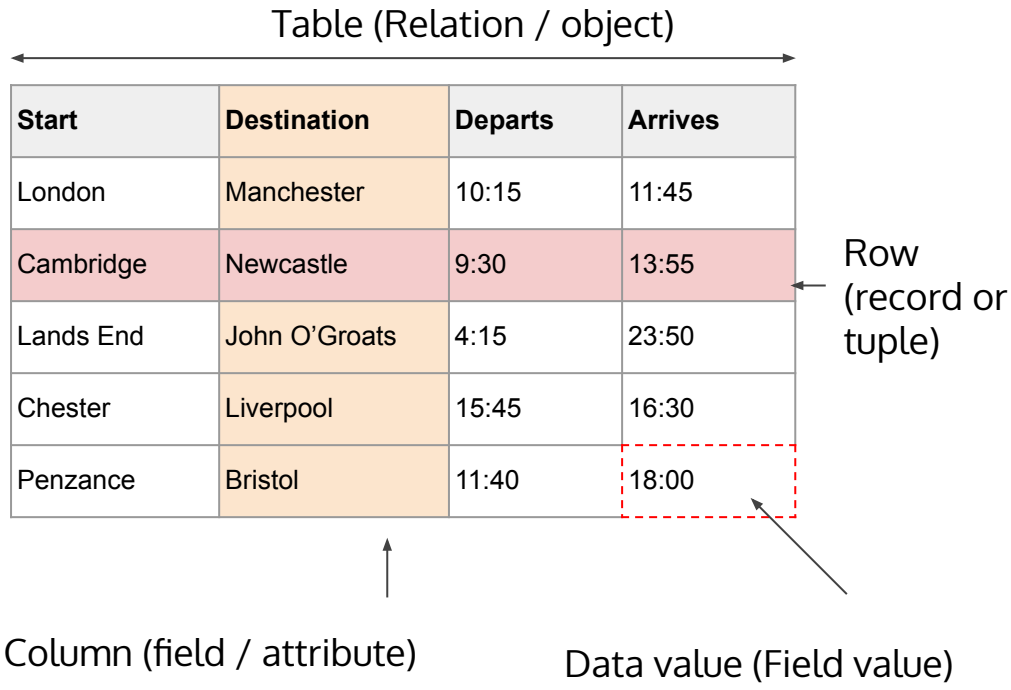
Relational Database organizes data in tables (or *relation*).

A database **schema** is a description of a database.

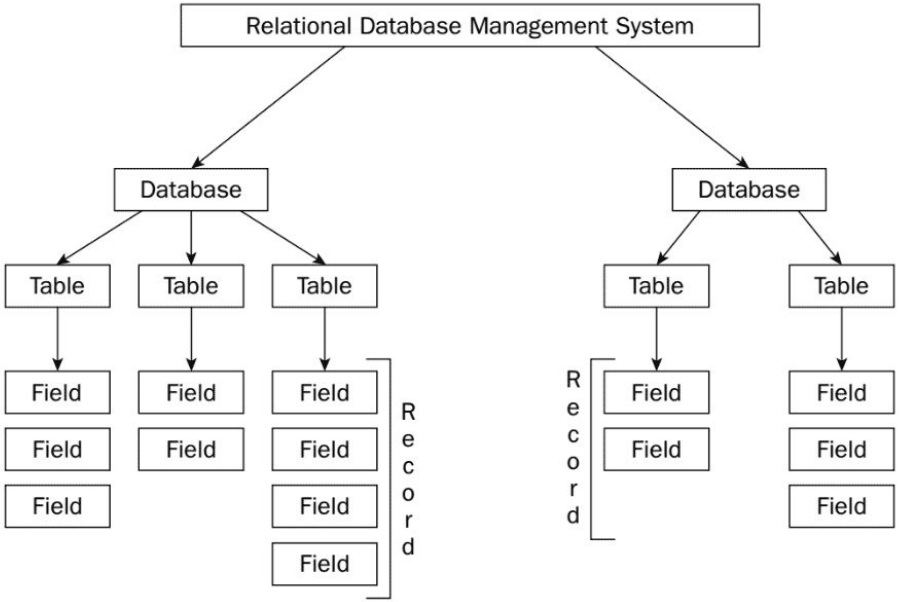
A **table** is made up of rows and columns.

A **row** is also called a record (or *tuple*).

A **column** is also called a field (or *attribute*).



Relational Database Structure



Relationships are defined between tables, to indicate how data is connect.

The types of relationships include:

1. one-to-many
2. many-to-many
3. one-to-one

SQL (Structured Query Language) was developed to work with relational databases.

Example World DB

Table *Country*

Code	Name	Continent	Region	SurfaceAr
ABW	Aruba ...	North America	Caribbean	193.00
AFG	Afghanistan ...	Asia	Southern and Centra...	652090.0
AGO	Angola ...	Africa	Central Africa	1246700.
AIA	Anguilla ...	North America	Caribbean	96.00
ALB	Albania ...	Europe	Southern Europe ...	28748.00

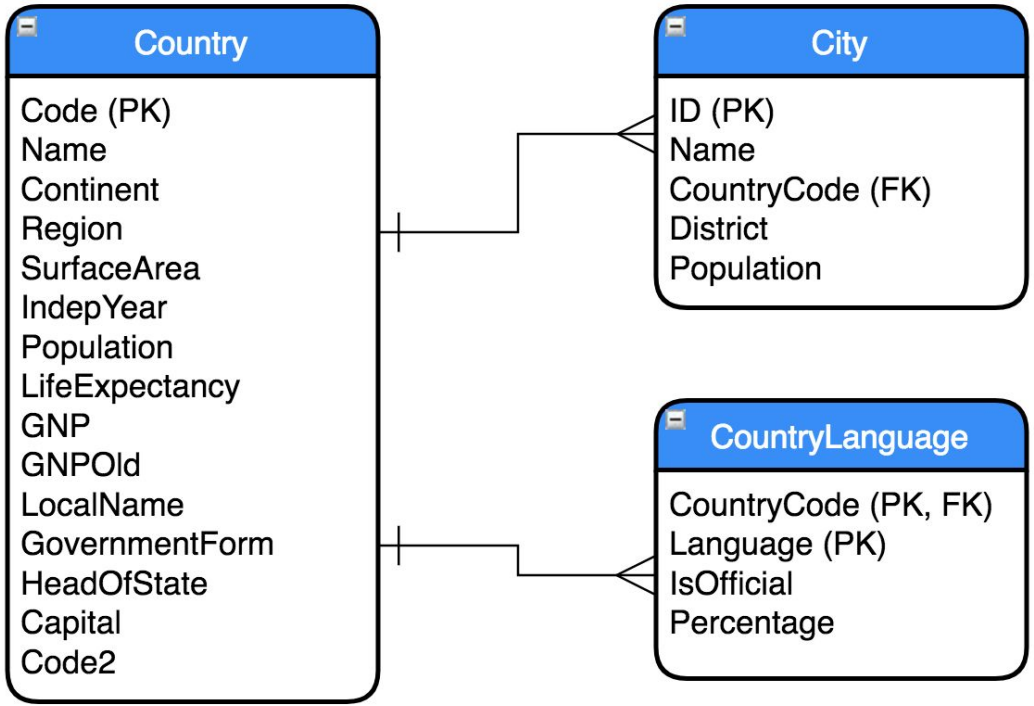
Table *City*

ID	Name	CountryCode	District	Pop
1	Kabul	AFG	Kabol	178
2	Qandahar ...	AFG	Qandahar	237
3	Herat	AFG	Herat	186
4	Mazar-e-Sharif ...	AFG	Balkh	127
5	Amsterdam ...	NLD	Noord-Holland ...	731

Table *CountryLanguage*

CountryCode	Language	IsOfficial	Percentage
ABW	Dutch ...	T	5.3
ABW	English ...	F	9.5
ABW	Papiamento ...	F	76.7
ABW	Spanish ...	F	7.4
AFG	Balochi ...	F	0.9
AFG	Dari	T	32.1

World DB - ER Diagram



Primary Key

- Uniquely identify each record of a table
- Must have unique value
- Can not contain NULL values
- ONE primary key per table
- A field or combination of fields
- Can be natural key (occurs in data) or surrogate key (generated by system)

Code	Name	Continent	Region
DJI	Djibouti	Africa	Eastern Afri
DMA	Dominica	North America	Caribbean
DNK	Denmark	Europe	Nordic Cour
DOM	Dominican Republic	North America	Caribbean
DZA	Algeria	Africa	Northern Afr
ECU	Ecuador	South America	South Amer
EGY	Egypt	Africa	Northern Afr
ERI	Eritrea	Africa	Eastern Afri
ESH	Western Sahara	Africa	Northern Afr

Foreign Key

- Establishes and enforces a link between data in two tables
- Refers to the primary key in another table

The Primary Key from the One-Table becomes Foreign Key in the Many-Table!

Table **Country**

Code	Name	Continent	Region
DJI	Djibouti	Africa	Eastern Afri
DMA	Dominica	North America	Caribbean
DNK	Denmark	Europe	Nordic Cour
DOM	Dominican Republic	North America	Caribbean
DZA	Algeria	Africa	Northern Afr
ECU	Ecuador	South America	South Amer
EGY	Egypt	Africa	Northern Afr
ERI	Eritrea	Africa	Eastern Afri
ESH	Western Sahara	Africa	Northern Afr

Table **City**

ID	Name	CountryCode	District
3315	København	DNK	Københa
3316	Århus	DNK	Århus
3317	Odense	DNK	Fyn
3318	Aalborg	DNK	Nordjylla
3319	Frederiksberg	DNK	Frederik

Designing a Database
Data Modelling

Good Database Design

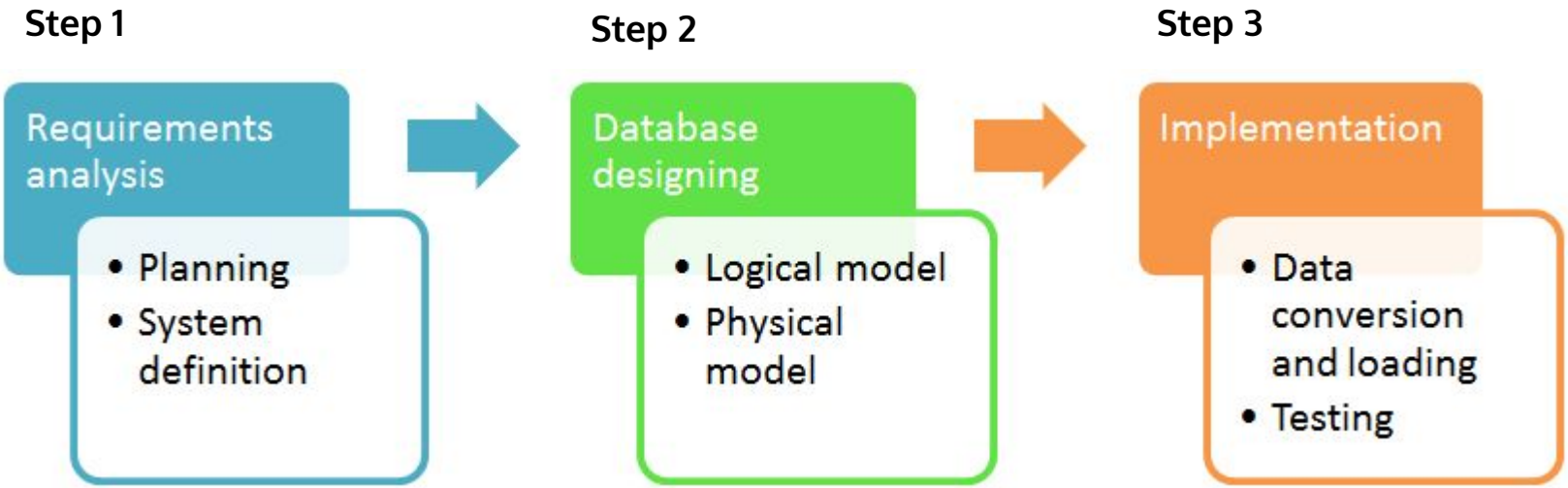
What?

- Eliminate data redundancy
- Ensure consistent data
- Save only data that is needed

Why?

- Simplify data extraction
- Faster execution
- Reduces changes in later development cycle

Database Design Process



http://www.guru99.com/database-design.html

Requirements Analysis (Step 1)

Obtain and analyse your data needs

- Analyze the organization
- Define any problems, possibilities or constraints
- Define the objectives
- Agree on the scope

RESULT: Concisely written set of users' requirements

Data Modelling (Step 2)

A **data model**—a collection of concepts that can be used to describe the structure of a database

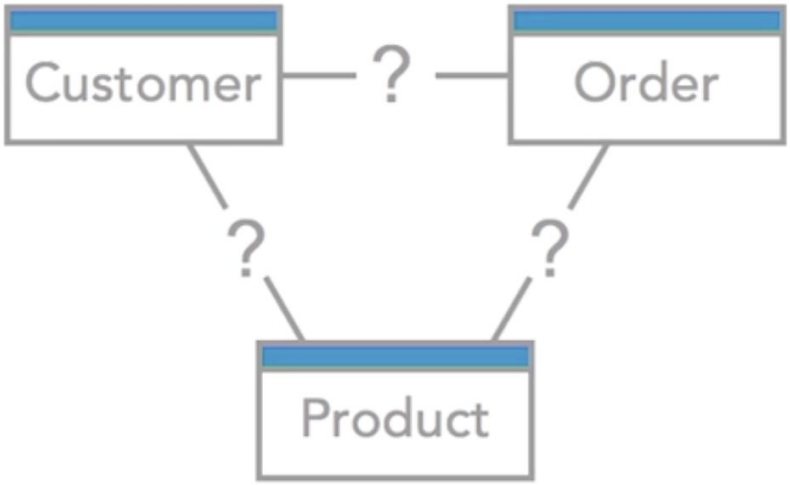
Define your (Relational) Data Model

1. entities
2. attributes and
3. relationships

Entities are basically people, places, or things you want to keep information about.
E.g library system may have the *book*, *library* and *borrower* entities.

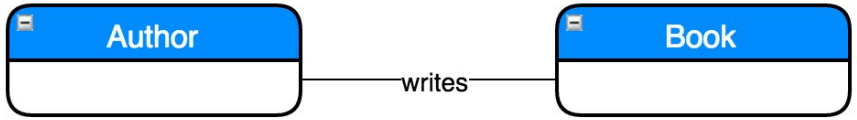
Attributes describe details of the entity
E.g. Book entity has a title, page count, isbn attributes

Relationship describes how to entities are related to each other
E.g. A book is published by a publisher

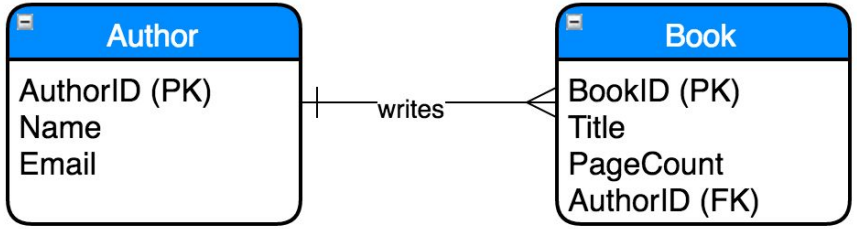

Customer
?
Order
?
?
Product

Three Types (Steps) of Data Modelling

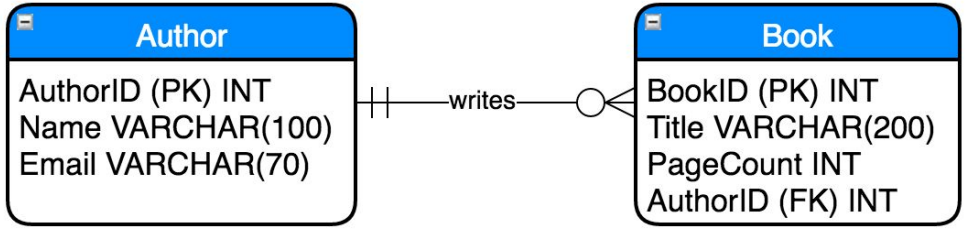
Conceptual data model



Logical data model



Physical data model

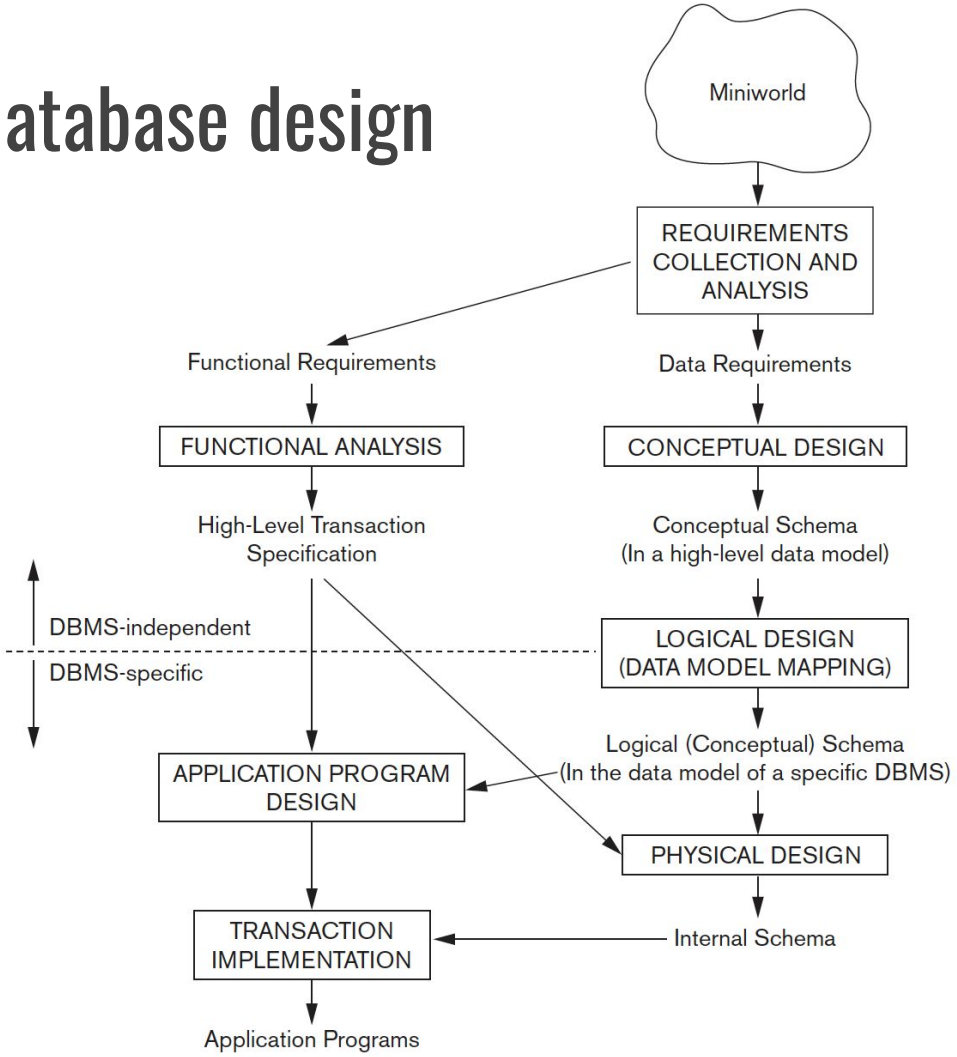


Three Types of Data Modelling

- **Conceptual data model**
  overview of what should be included - (High-level)
- **Logical data model** – more detailed than the conceptual data model, contains attributes, but doesn't contain data types
- **Physical data model** – the most detailed type of models. Its exact makeup depends on which database management system is used (Low-level)

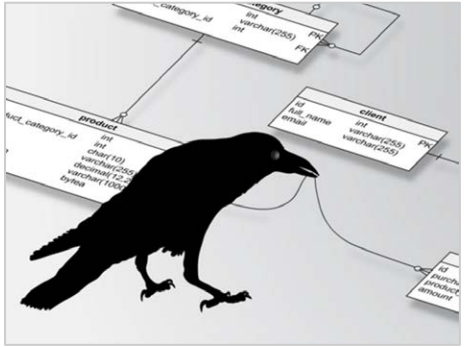
Feature	Conceptual	Logical	Physical
Entity names	X	X	
Entity relationships	X	X	
Attributes		X	
Primary keys		X	X
Foreign keys		X	X
Table names			X
Column names			X
Column data types			X

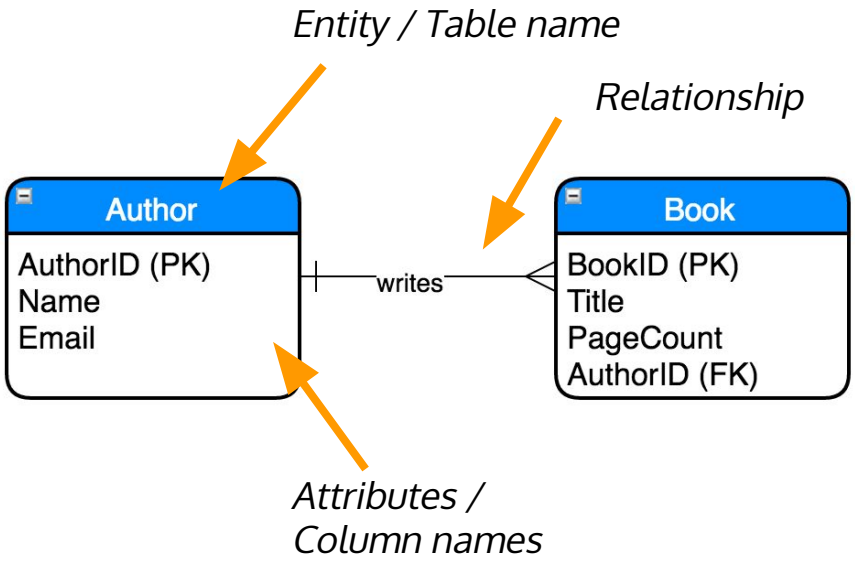
Main phases of database design

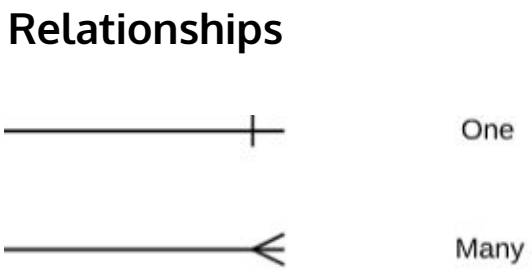


Source: Book [EL] Page 92

ER Diagram (using Crow's Foot Notation)







https://www.vertabelo.com/blog/technical-articles/crow-s-foot-notation
https://www.vertabelo.com/blog/documentation/crow-s-foot-notation-in-vertabelo

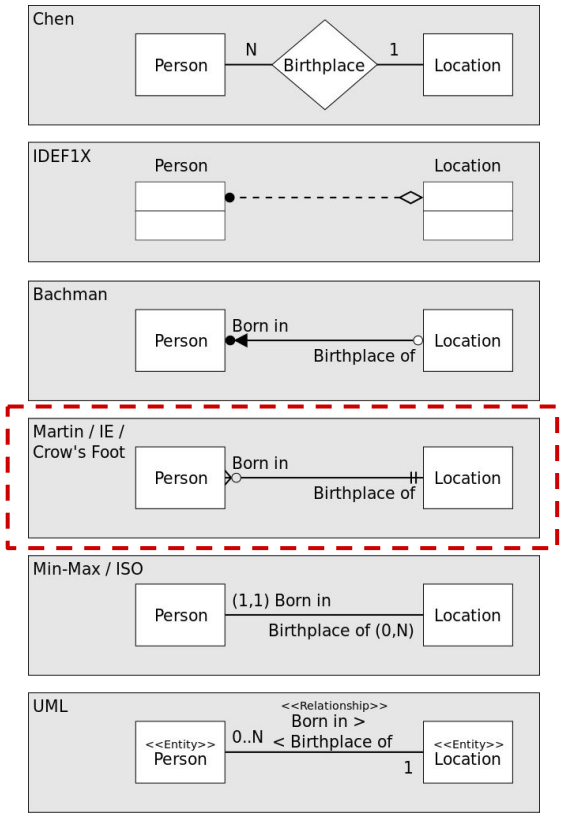
Entity Relationship Diagram (ERD or ER Diagram)

An Entity Relationship Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system.

They are used to model and design relational databases.

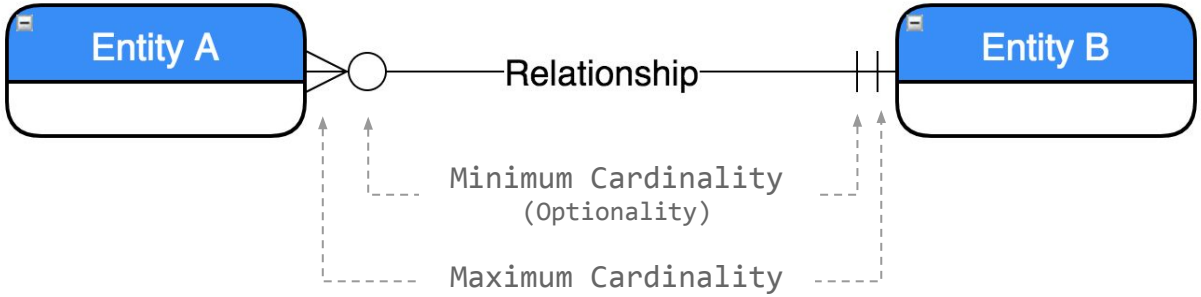
- **Entity** is thing or object in the real world *(e.g. instructor, student)*
- ***Attributes*** *are used to describe entities (e.g. name, id)*
- **Relationship** is an association between entities *(e.g. instructor advises students)*

Different notations to display ERD



Relationships

Cardinality indicates the **number** of times an entity can be connected to another (=participation).

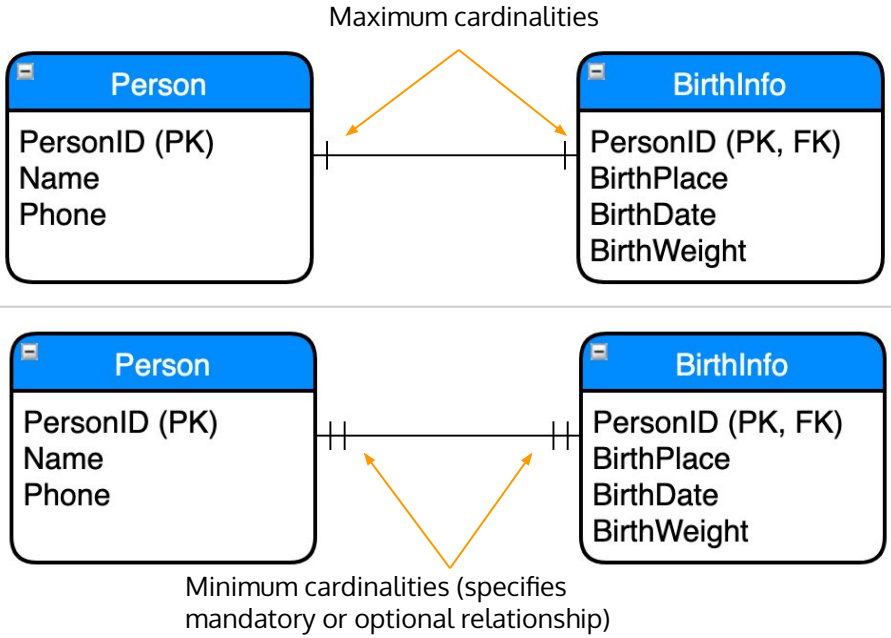


|	One
⋔	Many
○	Zero (=Optional)

Is the relationship mandatory or optional?
(i.e., "Can entity A exist without entity B?"; Must have / May have)

One-to-One Relationship (1:1)

One-to-one relationship between person and birth info

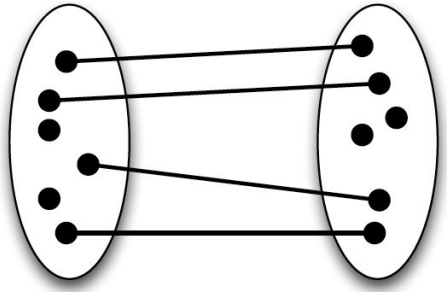


- A person has **one** BirthInfo entry
- Specific BirthInfo record belongs to **one** person

- A person **MUST** have BirthInfo entry
- Specific BirthInfo record **MUST** belong to one person

Relationship between Person and BirthInfo is **mandatory.**
Relationship between BirthInfo and Person is **mandatory**.

One-to-One relationship



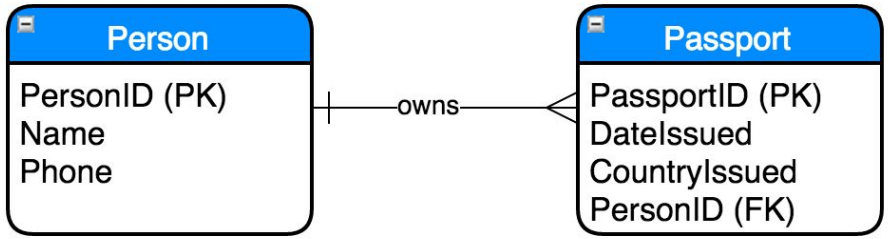
Examples 1:1

Person - LoginDetails

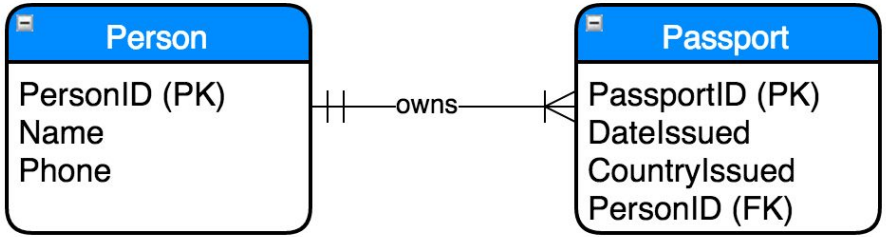
Customer - ContactInfo

One-to-Many Relationship (1:N)

One-to-many relationship between person and passport

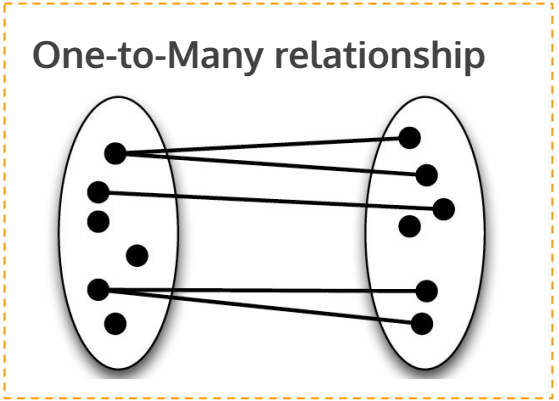


- A person can own **many** passports
- A passport is owned by one specific person



- A person MUST own **at least** one passport (one or more)
- A passport MUST be owned by a person (Passport is always associated with specific person)

Relationship between Person and Passport is **mandatory.**
Relationship between Passport and Person is **mandatory**.



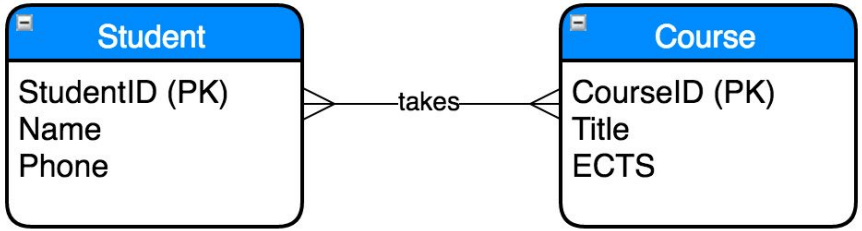
Examples 1:N

Customer - Order

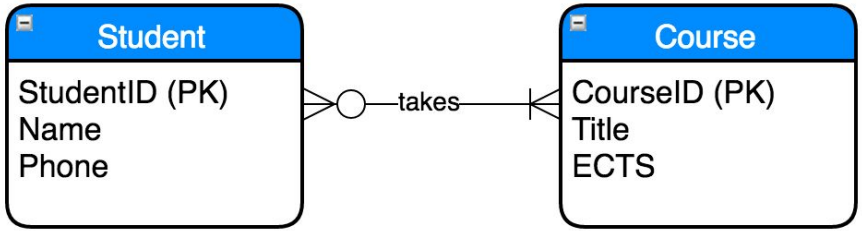
Company - Department

Many-to-Many Relationship (M:N)

Many-to-many relationship between an instructor and a student



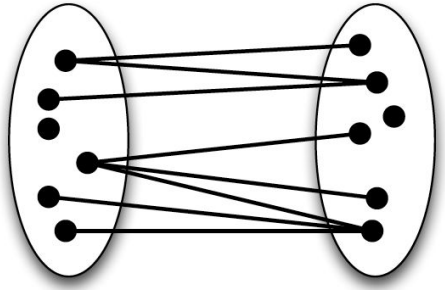
- A student takes **many** course
- A course is taken by **many** students



- A student MUST take **at least** one course (one or more)
- A course MAY be taken by a student (zero to many)

N:M relationship only displayed in logical data model, DBMS can't store this relationship and need to be broken up with a junction/binding table

Many-to-Many relationship

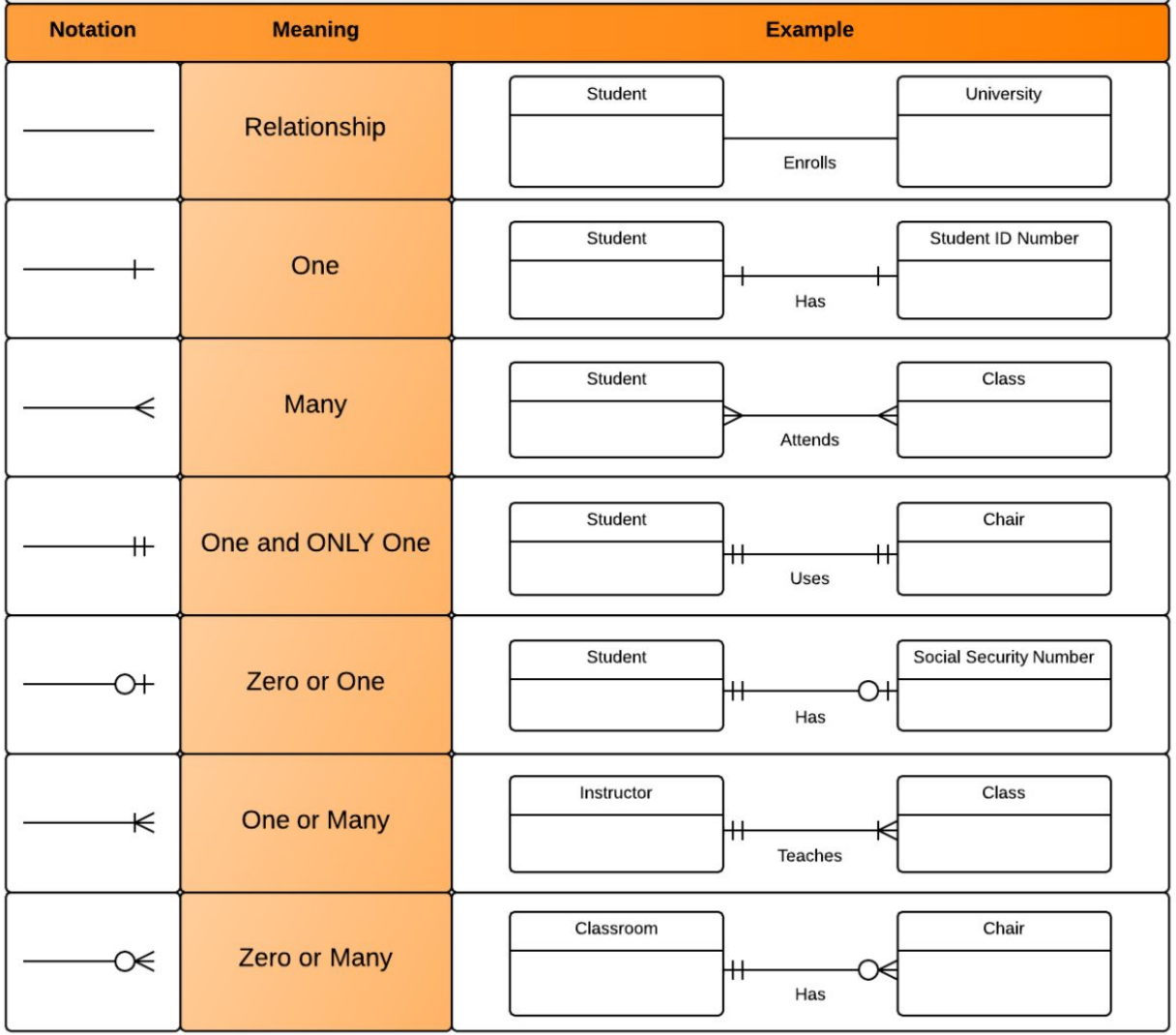


Examples M:N

Order - Product

Employee - Project

Examples

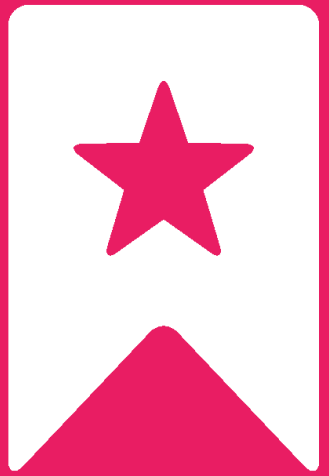

Notation
Meaning
Example
Relationship
Student
Enrolls
University
One
Student
Has
Student ID Number
Many
Student
Attends
Class
One and ONLY One
Student
Uses
Chair
Zero or One
Student
Has
Social Security Number
One or Many
Instructor
Teaches
Class
Zero or Many
Classroom
Has
Chair

ER Diagram Modelling Rules

Relationship	ER Diagram
One-to-one	**PK** from one of the tables becomes **FK** in the other table. Doesn't matter which table holds the FK.
One-to-many	**PK** from the one-table becomes **FK** in the many-table.
Many-to-many	**N:M relationship only displayed in logical data model.** In physical data model, relationship needs to be broken into two 1:N relationships. **Create new table** (= junction/binding table)

schema = table
PK = primary key
FK = foreign key

The Primary Key from the One-Table becomes Foreign Key in the Many-Table!



Data modelling - basic design rules

A well-designed table is one that:

- seeks to minimize redundant data (same info kept in two different tables)
- represents a single subject
- has a primary key (a field or set of fields whose values will uniquely identify each record in the table)
- does not contain multi-part fields (e.g., "UCL, Seebladsgade, Odense C")
- does not contain multi-valued fields (e.g., an Author field shouldn't hold values of the form "Jensen, Holst, Andersen")
- does not contain repeated groups (e.g. using Author1, Author2, Author3)
- does not contain fields that rely on other fields (e.g. store DOB not Age)

Source: https://www.e-education.psu.edu/spatialdb/l2_p4.html

Task 1.1: Data Modelling

Use e.g. https://www.draw.io/ for drawing the diagram

Draw the ER-Diagrams (logical) for the following:

1. A person can have multiple emails addresses and multiple phone numbers
2. A book can be written by multiple authors but only published by one publishing company
3. A blog post is written by one user and can have multiple comments
4. A charity project on a fundraising website can have many donations by many donors.

Task 1.2: Pet Hospital

HEALTH HISTORY REPORT

PET ID	PET NAME	PET TYPE	PET AGE	OWNER	VISIT DATE	PROCEDURE
246	ROVER	DOG	12	WILL JENSEN	13 JAN 2015	RABIES VACCINATION
					27 MAR 2015	EXAMINE and TREAT WOUND
					02 APR 2015	HEART WORM TEST
298	SPOT	DOG	2	TERRY KIM	21 JAN 2015	TETANUS VACCINATION
					10 MAR 2015	HEART WORM TEST
341	MORRIS	CAT	4	WILL JENSEN	23 JAN 2014	RABIES VACCINATION
					13 JAN 2015	RABIES VACCINATION
519	TWEEDY	BIRD	2	TERRY KIM	30 APR 2015	ANNUAL CHECK UP
					30 APR 2015	EYE WASH

Task 1.2: Pet Hospital

1. Analyse (write down requirements)
   e.g. A pet has one owner, an owner can have many pets, ...
2. Create a ER Diagram (logical data model)

Task 1.3: SQL queries

Try to solve the following queries.
Do this task on your own.
It will show me the class level.

Handin queries as .sql file (plain text file)

1. -- Show all films that have a length of more than 180 minutes
2. -- Show actors that have a first name of 'Bob' or 'Fred'
3. -- Show all film titles that contain the word 'mad'
4. -- Count how many 'PG' rated films there are
5. -- Show films with the highest rental rate. Display 10 only and the highest rates first.
6. -- How many days has it been since the film with id 1 has been updated
7. -- Show how many films there are per rating
8. -- Show customer email, street and postal code
9. -- How much do I have to pay, if I rent all films in the category 'Horror'
10. -- Write an SQL query yourself (as complex or simple as you like :)

Homework with Handin

Task 1.2: ER-Diagram Pet Hospital

Task 1.3: SQL Queries (sakilaDB)

You need:

- MS SQL server running locally
  (see info next slide)
- Add 'sakila' demo database
  (Download and installation info on itslearning)

*> **Handin** on itslearning*

Counselling today
12:15 - 14:00

I will be available in this room.

Email me, if you want counselling during this time via Zoom.

Prep for next class

Read the following guides:

CREATE DATABASE
https://www.sqlservertutorial.net/sql-server-basics/sql-server-create-database/

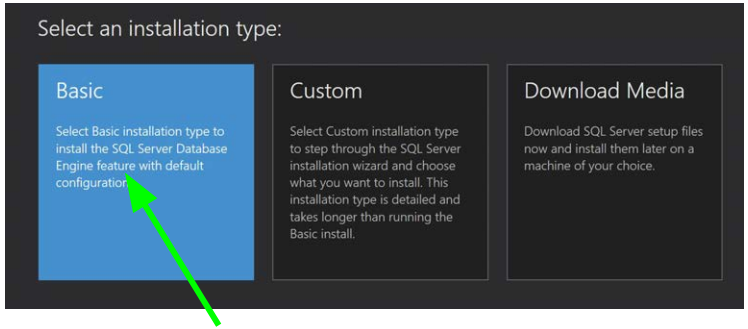
CREATE TABLE
https://www.sqlservertutorial.net/sql-server-basics/sql-server-create-table/

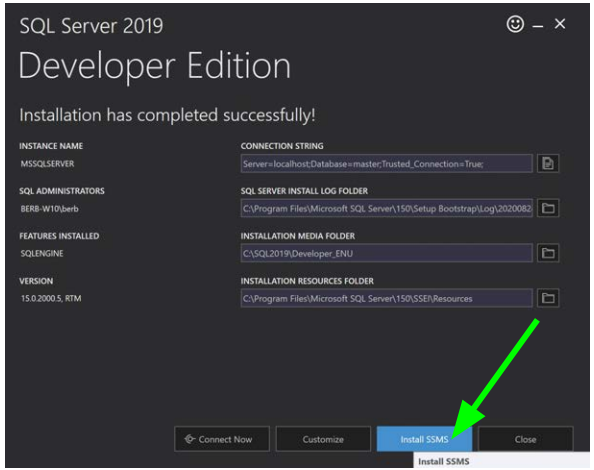
INSERT INTO
https://www.sqlservertutorial.net/sql-server-basics/sql-server-insert/

Data Types
https://www.sqlservertutorial.net/sql-server-basics/sql-server-data-types/

Installing MS SQL Server

1. Install **MS SQL Server 2019 (Developer version)**
   https://www.microsoft.com/en-us/sql-server/sql-server-downloads
2. Install **SQL Server Management Studio (SSMS)**
   Download SQL Server Management Studio or use button on dialog box after Install 1
3. Open SSMS and connect to your SQL Server





>> Tutorial on how to install ...
https://www.linkedin.com/learning/microsoft-sql-server-2019-essential-training/install-sql-server-developer-edition