CODERS.BAY

DATABASE

16.6.2020 Basics of databases

Relational databases

e.g. MySQL, Oracle, Microsoft SQL

Other databases:

- Document-based databases
- hierarchical database
- nosql-databases (e.g. Graph Ql) → new stuff

⏩ **We are learning MySQL**.
When googling for help you need to check if the answer refers to the language MySQL.

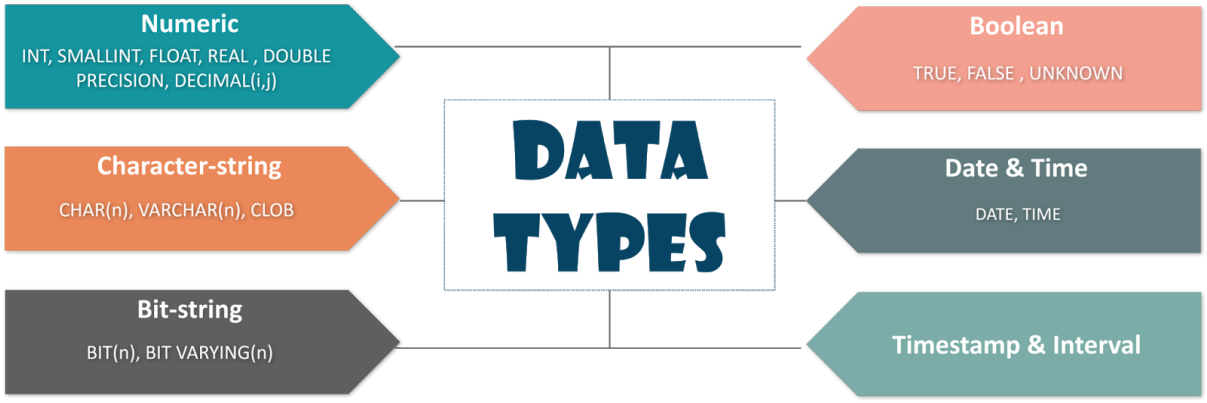
Data modelling

Sketch of a use case for a database. Has to be made for all databases and differs later in the implementation (which DB and language is used), when we setup the database.

1. Conceptual schema
2. Logic schema (readable by machines)
3. Implementation

⏩ For DB we are using the **ER-model with the CHEN-notation.**
UML is not used for DB, more for programming use cases.

Data Types



1. Numerics (Zahlen)

INT - for whole numbers, no decimals
Used for e.g. IDs, to calculate

Numeric data Types have 2 options.
signed: allows negative numbers
unsigned: does not allow negative numbers

TINYINT: max. 255

SMALLINT: max. 65535
MEDIUMINT: ...
INT
BIGINT

DECIMAL - if you need perfect numbers

DECIMAL	**999**	**,**	**99**
naming	precision		scale

other names for decimal:
DEC
NUMERIC
FIXED

FLOAT 4 bytes (about 7 digits)
DOUBLE 8 bytes (about 15 digits)

Data types that can store a lot of data and decimal numbers, but they do not maintain precision. Better use DOUBLE, because it's more precise.

If precision does not have to be perfect, minor differences are ok at the end of a decimal.

2. Binary (Binärdaten)

consists of 0 and 1, nothing else
BINARY
VARBINARY

3. Strings (Zeichenketten)

CHAR - fixed length, for strings
e.g. CHAR (10) → The size parameter specifies the column length in characters.
Which character set? → unicode → encoded with UTF8
Default for text data type text: anything

What values are allowed? The default number is 1.
0 → null
255 → max. number for 8 bit number

Note:
If you do not need the foll space then you waste space.

If you export data with spaces SQL strips those spaces away.
→ PAD_CHAR_TO_FULL_LENGTH

VARCHAR - variable length

A VARIABLE length string (can contain letters, numbers, and special characters).
Attention: every string has different bytes, the encoding has also different bytes (e.g. chinese has 3 bytes)
Max. 65535 bytes (if a character takes 1 byte, the it´s 65535 characters)
Row-limit: 65535

TEXT:
Use depends on the size of the text.
No row limit like char because the text data is stored elsewhere in MySql.
If you need more text in your rows then use text data.
Default for text data type text: NULL

“CLOB”: character large object

TEXT: max. 65535 bytes
TINYTEXT: max. 255 bytes
MEDIUMTEXT: 16 mio. bytes
LONGTEXT: 4 billion. bytes

ENUM
list of values
Max. 3000 options
allows you to pick one of many options
everything is stored as a number

1 very good
2 ok
3 bad → #3 is stored as an empty string, no data is stored, just the number (save space)
0 is invalid

ENUM(“very good”, “ok”, “bad”)
→ entering #2 = “ok”
The string has to match the string.

Using enum can be done with database design. It might be easier.

SET
Similar to ENUM
allows you to pick many of many options - a set of data
Max. 64 options possible

Note: Do not use SET for M:N relations, solve this with database design (intermediate table)

BLOB - For BLOBs (Binary Large OBjects). Holds up to 65,535 bytes of data
TINYBLOB - For BLOBs (Binary Large OBjects). Max length: 255 bytes
MEDIUMBLOB - For BLOBs (Binary Large OBjects). Holds up to 16,777,215 bytes of data
LONGBLOB - For BLOBs (Binary Large OBjects). Holds up to 4,294,967,295 bytes of data

4. Booleans (Boolesche Werte)

true or false
If a boolean is compared to “null” or “unknown”, then the result is always “unknown”.

5. Datetimes (Datum-/ Zeit-Werte)

DATE
TIME
DATETIME
TIMESTAMP

DATE	TIME (6)	DATETIME
2020-01-22	22:54:30.123456 Seconds up to 6 decimals!	Date and time in one!

Date conventions: YY MM TT yy mm tt dd hh ss,... attention to detail!

Check documentation: MySQL 8.0 Reference Manual :: 11.2 Date and Time Data Types

Einheit	Kürzel
year	yyyy, yy
quarter	qq, q
month	mm, m
dayofyear	dy, y
day	dd, d
week	wk, ww
weekday	dw
hour	hh
minute	mi, n
second	ss, s
millisecond	ms

6. Intervals (Intervalle)

Similar to datetime, difference between 2 datetimes

7. XML

Format for plattform neutral exchange of data

SQL Data Types for MySQL, SQL Server, and MS Access

https://www.youtube.com/watch?v=UGu9unCW4PA&list=PL_c9BZzLwBRKn20DFbNeLAAbw4ZMTIZPH

How to use and write date formats:

Student (Matrnr: INTEGER, Name: VARCHAR, Semester: TINYINT)

→ Der Student (Objekt) hat die Attribute Matrnr, Name und Semester. Es gibt immer ein Schlüsselattribut.
→ Objekte stehen in Beziehung zueinander. Beziehungen können auch Attribute haben.

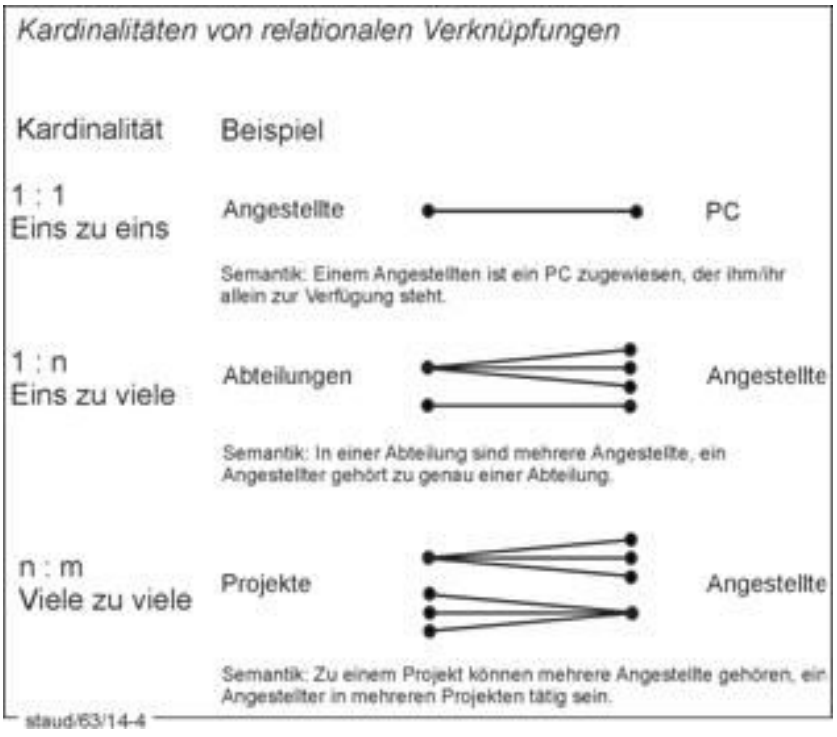
Relations

Different kinds of relations

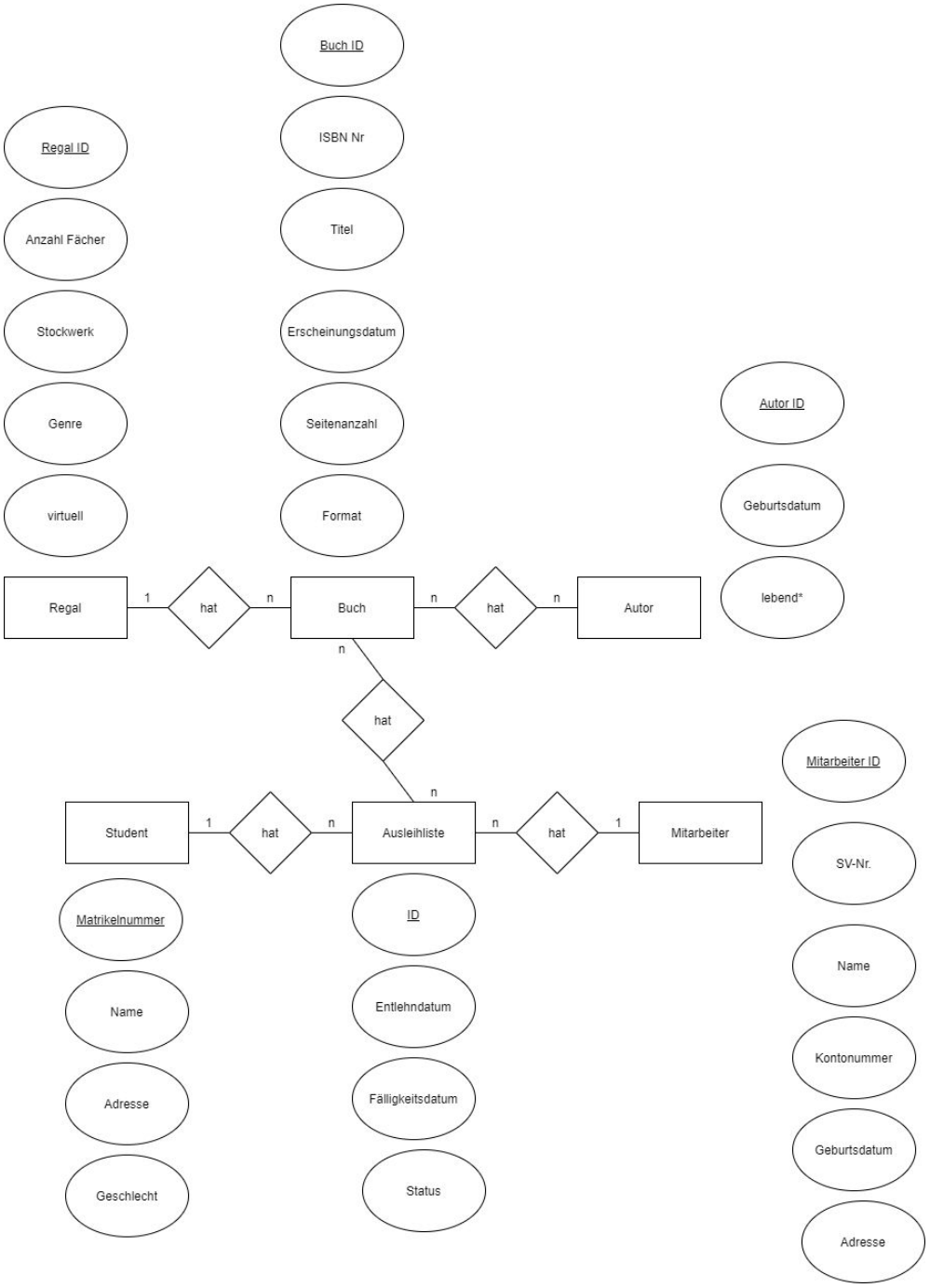
1. binär
2. unär
3. ternär

Describing relations with cardinalities

1. 1:1
2. 1:n
3. n:n



First draft of ER-diagram - a library:



MIN, MAX-Notation

Bei der (min,max)-Notation wird für jeden an einer Beziehung beteiligten Entitätstyp ein geordnetes Paar mit einem **Minimal- und einem Maximalwert** angegeben. Diese Werte geben

an, an wie vielen Beziehungsausprägungen die Entitätausprägungen mindestens teilnehmen müssen und an wie vielen sie höchstens teilnehmen dürfen.

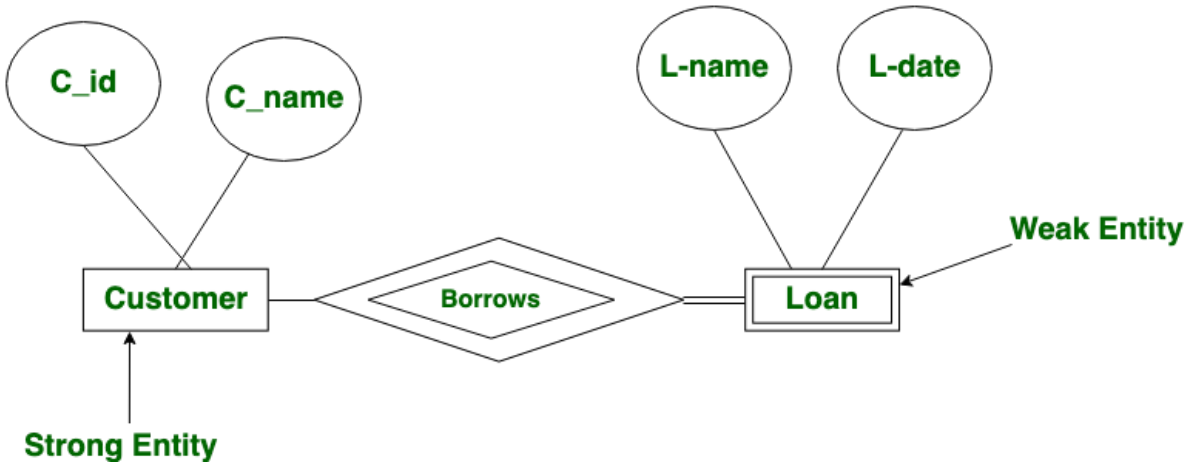
Spieler-Entität (1,1) → da jeder Spieler zu genau einer Mannschaft gehören soll

Mannschafts-Entität (11,11) → da es für jede Mannschaft genau 11 Spieler geben soll, die zu ihr gehören

Other possible options:
0,0
0,1
0,*
1,*

Starke und schwache Entitäten

A **weak entity** is **dependent on a strong entity** to ensure the its existence. Unlike a strong entity, a weak entity does not have any primary key. It instead has a partial discriminator key.



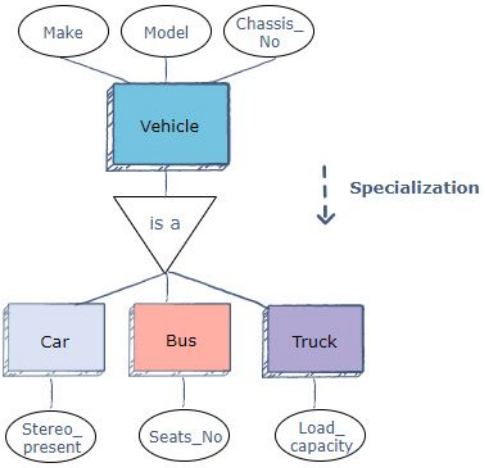
E.g. A bank loan does not exist without a customer. A Customer can have a loan.

A weak entity is represented by a double rectangle.
The relation between one strong and one weak entity is represented by a double diamond.
https://www.geeksforgeeks.org/weak-entity-set-in-er-diagrams/

Generalisierung und Spezialisierung

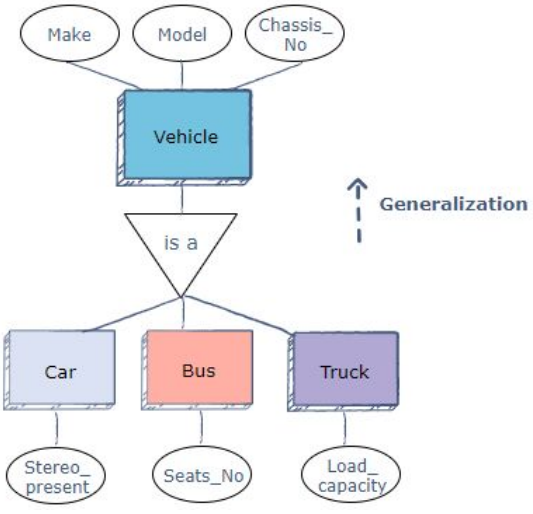
Specialisation

Top-down approach where a higher-level entity is divided into multiple specialised lower-level entities.



Generalisation

Bottom-up approach: lower-level entities are combined to a higher-level entifies.



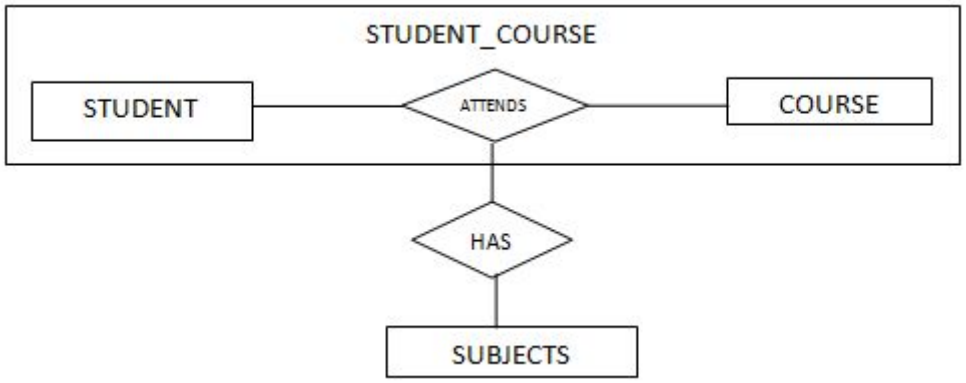
NOTE: In both cases Make, Model, and Chassis_No are shared attributes among the entities Car, Bus, Truck. These entities have their own attributes as well.

is a-Beziehung

Type A is a subtype of type B when A's specification implies B's specification.

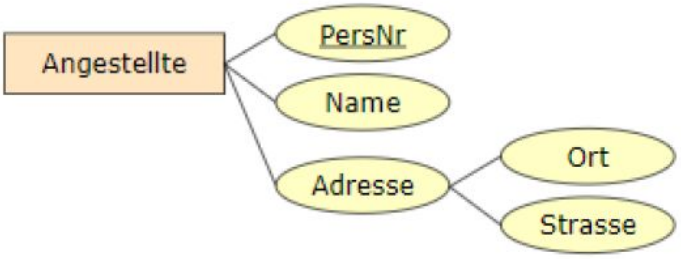
Aggregation

Relations with corresponding entities are aggregated.



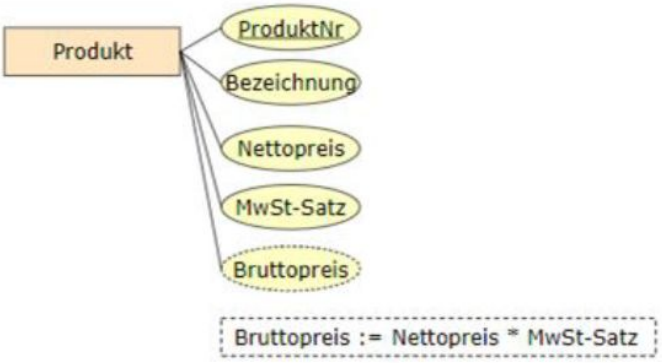
Strukturierte Attribute

Groups of attributes that share similar attributes.



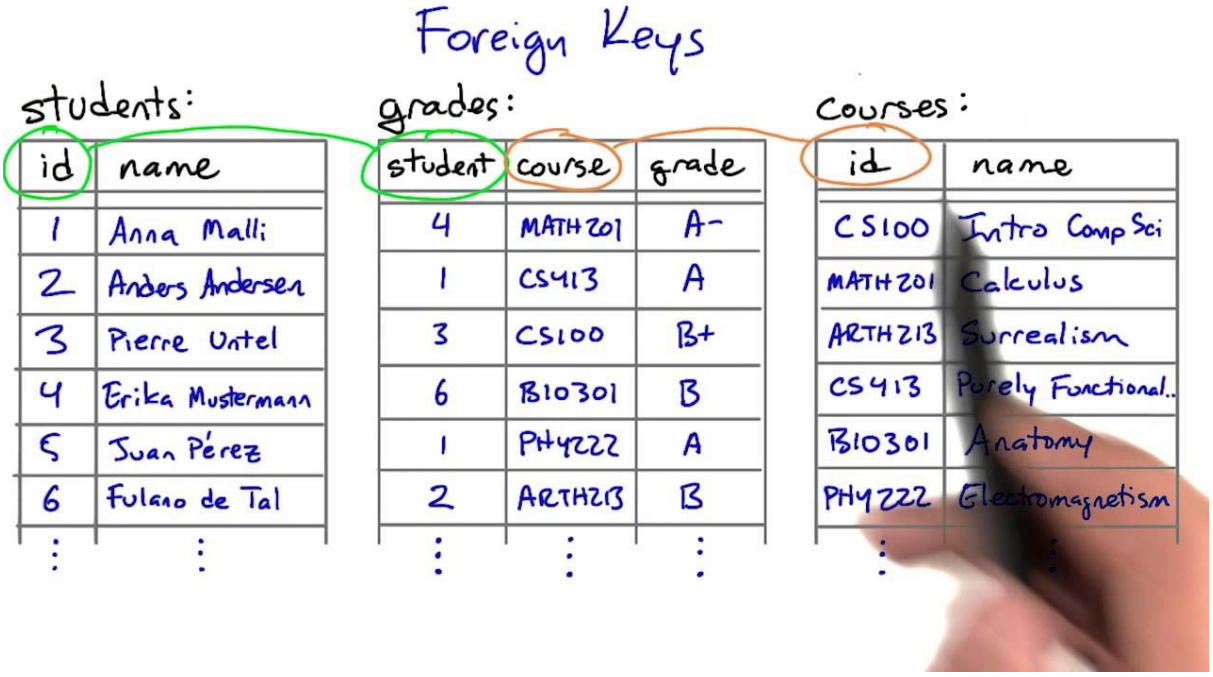
Abgeleitete Attribute (berechnet sich aus anderen Attributen)

“Calculated attributes” who´s values can´t be stored in an attribute, but calculated via a DB query.



Fremdschlüssel

Primary keys from other entities used as referential key in another entity.



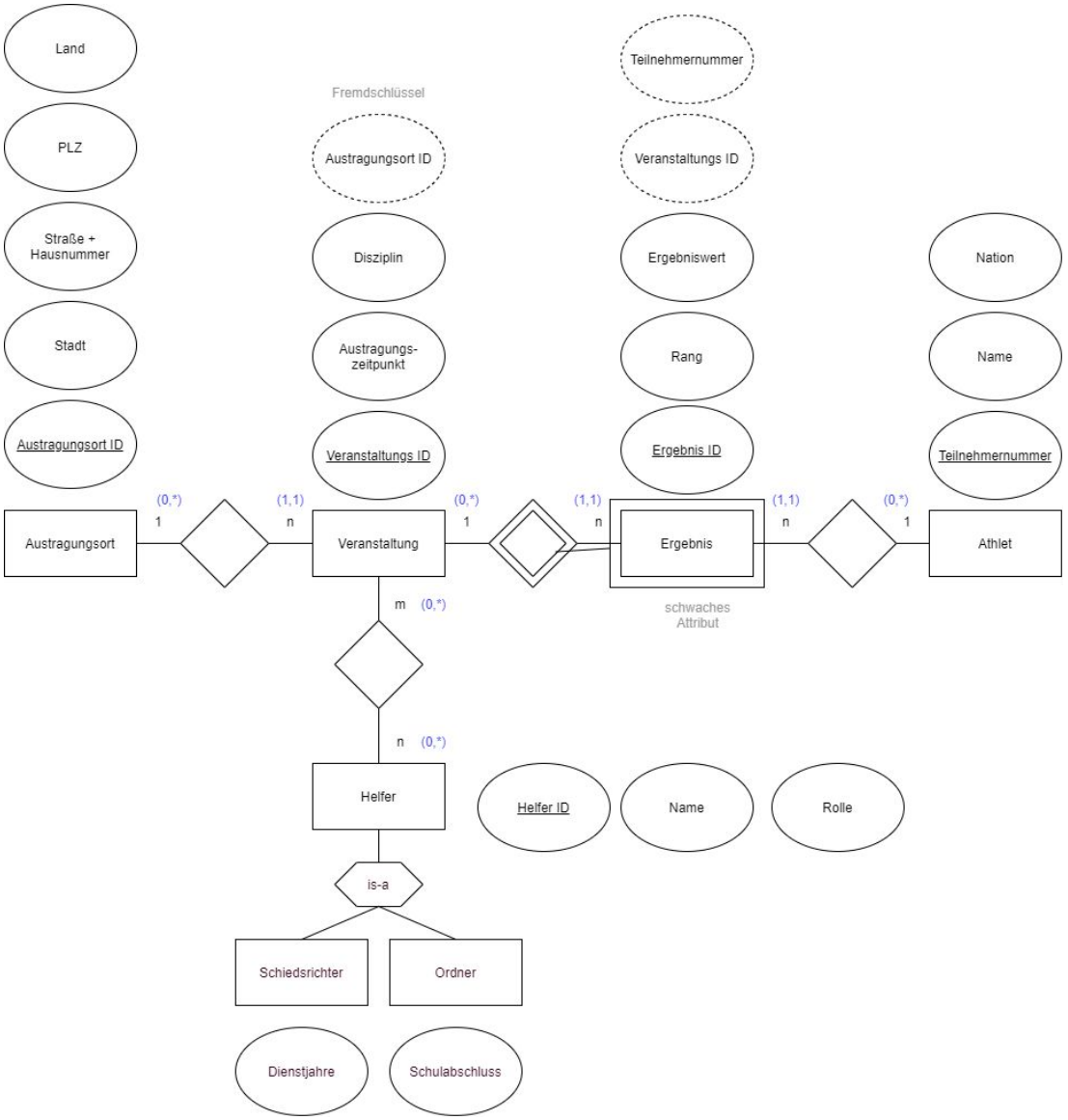
Kreuztabellen bei n:m Beziehungen mit zusammengesetzten Primärschlüssel

Artificial primary key that is a combination of the primary keys of two entities in a n:m relation.

Vorgehensweise:

1. ER-Diagramm
2. Relation in Tabellen darstellen
3. Textuelle Notation

Exercise:



Austragungsort: {[Austragungsort ID:integer (PK), Stadt:string, Straße+Hausnummer:string, PLZ:integer, Land:string]}

Veranstaltung: {[Veranstaltungs ID:integer (PK), Austragungszeitpunkt:datetime, Disziplin:string, Austragungsort ID:integer (FK)]}

Ergebnis: {[Ergebnis ID:integer (PK), Rank:integer, Ergebniswert:string, Veranstaltungs ID:integer (FK), Teilnehmernummer:integer (FK)]}

Athlet: {[Teilnehmernummer:integer (PK), Name:string, Nation:string]}

Helfer: {[Helfer ID: integer (PK), Name:string, Rolle:string]}

Helfer_Ergebnis_Liste: {[Helfer ID:integer (PK, FK), Veranstaltung ID:integer (FK, PK)]}