

Senior dev: Haha that's weird it's almost like the database is gone

The intern:



Datenbanken



Crazy wa?

Intern after deleting the production database



WHEN EVERYBODY IS HAVING A RELATIONSHIP
BUT YOU ONLY HAVE RELATIONAL DATABASES



Wat machen wir??

1. Theorie

1. Was ist ne Datenbank undso
2. Probleme, Anomalien usw.
3. Normalisierung

2. ERM

1. Syntax
2. Relationale Form mit Regeln

3. SQL

1. Befehle
2. Vllt. Aufgaben ka was ich vorhabe noch zu machen



THEORIE

Wofür den ganzen Scheiß???

Datenorganisation / Daten strukturiert speichern

1. Eine Datenbank ist ein verteiltes, integriertes Computersystem, das Nutzdaten und Metadaten enthält. Nutzdaten sind die Daten, die Benutzer in der Datenbank anlegen und aus denen Informationen gewonnen werden. Metadaten werden oft auch als Daten über Daten bezeichnet und helfen, die Nutzdaten der Datenbank zu strukturieren.
2. Eine Datenbank ist eine geordnete, selbstbeschreibende Sammlung von Daten, die miteinander in Beziehung stehen.



THEORIE

Wofür den ganzen Scheiß???

Datenorganisation / Daten strukturiert speichern

Eine Datenbank ist ein System zur strukturierten Speicherung von Nutzdaten mithilfe von Metadaten, wobei die Daten miteinander verknüpft sein können.



THEORIE

Wieso nicht einfach Dateien Nutzen häää??

Weil das Dumm ist

Emil Schmidt; 0231-1020449 ;Kaiserstrasse 5, Musterhausen; 12345; Helena Meier; ...

Hans Weiher 3, Musterhausen;12345;Ingo Fuchs;

Johanna Schulze;0410-1241221;Alte Poststr. 5, Musterhausen;12345;He1ena Meier; ...

Markus Schulte; 04514-123414 ;Goethestr. 7, Musterburg;12354;Ingo Fuchs;

Hans Weiher 3, Musterhausen;12345;He1ena Meier;

Johanna Schulze;0410-1241221;Alte Poststr. 5, Musterhausen;12345;He1ena

Hans Müller;0221-2415932;Am Weiher 3, Musterhausen;12345;Ingo

Fuchs; ...

THEORIE

Emil Schmidt; 0231—1020449 ;Kaiserstrasse 5, Musterhausen; 12345; Helena Meier; ...

Hans Weiher 3, Musterhausen;12345;Ingo Fuchs;

Johanna Schulze;0410-1241221;Alte Poststr. 5, Musterhausen;12345;He1ena Meier; ...

Markus Schulte; 04514-123414 ;Goethestr. 7, Musterburg;12354;Ingo Fuchs;

Hans Weiher 3, Musterhausen;12345;He1ena Meier;

Johanna Schulze;0410-1241221;Alte Poststr. 5, Musterhausen;12345;He1ena

Hans Müller;0221-2415932;Am Weiher 3, Musterhausen;12345;Ingo Fuchs; ...

- Langsamer Zugriff
- Schwierige Datenaktualisierung
- Hohe Redundanz
- Inkonsistenzen
- Begrenzte Flexibilität

THEORIE – Datenbanksystem

Datenbank

- Da wo die Daten wirklich leben
- In Form von Tabellen



DBMS

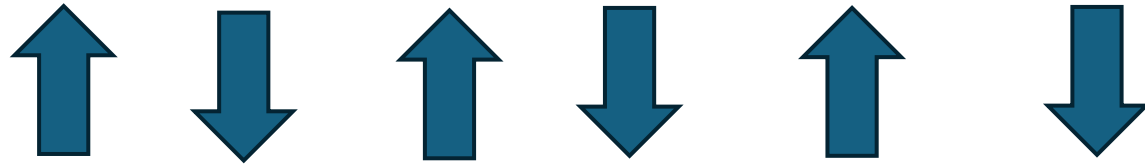
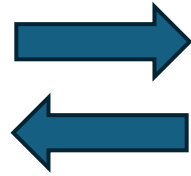
- Software zur Verwaltung der Datenbank
- MySQL, SQLite, Microsoft Access, ...
- DB-Struktur, Datenmanipulation, Datenabfrage, Datenkontrolle, Datenübertragung

THEORIE – Datenbanksystem

~~BDSM~~ DBMS



Wer auch immer



Datenbank



Tabellen mit Daten

THEORIE

Was erwarten wir von einer Datenbank / Welche Ziele soll es erfüllen?

- **Daten sind Unabhängig**

- Daten sind unabhängig von Programmen gespeichert.
- Benutzer arbeiten nur mit der Datenstruktur, nicht mit der technischen Umsetzung
- Das Datenbankverwaltungssystem verwaltet die physische Speicherung und Gerätezugriffe

- **Benutzerfreundlichkeit**

- SQL + Benutzeroberfläche

- **Mehrfachzugriff**

- Mehrere Leute können gleichzeitig zugreifen (Hinzufügen, Löschen, Ändern, ...)

THEORIE

- **Flexibilität**

- Alle Daten wie man will verknüpfen und zugreifen

- **Effizienz**

- **Datenschutz**

- Schutz vor unbefugtem Zugriff

- **Datensicherheit**

- Vor Programm/Hardware-Ausfällen schützen

- **Datenintegrität**

- Daten vollständig, korrekt und widerspruchsfrei (referentielle Integrität & Konsistenz)

- **Redundanzfreiheit**

- Jedes Element nur einmal speichern

THEORIE – Relationale Datenbank

- Eine relationale Datenbank speichert Daten in Tabellen, die über Schlüssel miteinander verbunden sind und mithilfe von SQL abgefragt werden können.



Schlüssel –
eindeutiges Attribut

Relation – Tabelle

Attribut – Name der Spalte

PK	Attr1	Attr2
1	abc	123
2
3
34	Donald	54

Fremdschlüssel

PK	Attr1	FK#
1	...	3

Datensatz – Zeile

Beziehung



THEORIE – Relationale Datenbank

Vorteile

- Daten sind unabhängig von der physischen Speicherung
- Änderungen an der Datenstruktur beeinflussen Programme kaum
- Redundanzen werden reduziert
- flexible Abfragen möglich (z. B. mit SQL)
- Transaktionen sorgen für sichere Verarbeitung (bei Fehlern kann ein Rollback alles rückgängig machen)

Nachteile

- höherer Bedarf an Speicher und Rechenleistung
- einfache Nutzung kann zu schlechtem Datenbankdesign führen



THEORIE – 3 Schichten Architektur



Externe Ebene

Beschreibt die benutzerspezifische Sicht, also welche Daten relevant für den Nutzer sind
Welche Daten sieht welche Benutzergruppe?

Konzeptuelle Ebene

Beschreibt die Gesamtstruktur der Daten und die Beziehungen zwischen denen
Was sind die Daten und wie hängen sie logisch zusammen?

Interne Ebene

Beschreibt die physische Umsetzung der Konzeptionellen Ebene
Wie werden die Daten physisch gespeichert?

THEORIE – Anomalien usw.

Begriff	Bedeutung	Beispiel
Redundanz	Mehrfache Speicherung derselben Daten	Die Postleitzahl „12344“ steht in mehreren Tabellen.
Inkonsistenz	Widersprüchliche oder fehlerhafte Daten	In einer Tabelle steht „Berlin“, in einer anderen „Hamburg“ für denselben Schüler.
Referentielle Integrität	Beziehungen zwischen Tabellen bleiben konsistent	Ein Kurs darf nur Schüler enthalten, die in der Schülertabelle existieren. Fremdschlüssel regelt die Verknüpfung.

THEORIE – Anomalien

Nr	Fahrer	FührerscheinNr	Modul	Bezeichnung	Termin	Schulungsanbieter
1	Hans Kaiser 0162/433789	G0208812341	1	Eco-Training	30.04.15	Logistik Service
			2	Kontrollgeräte	28.05.19	Deutschherrnufer 1
			5	Ladungssicherung	20.08.19	60594 Frankfurt
			1	Eco-Training	12.02.20	
2	Jupp Kurz 0178/128875	77123001221	3	Sicherheit im Fokus	05.03.19	Fahrschule Weiß Karl-Herd-Weg 56 63075 Offenbach
			2	Kontrollgeräte	28.05.19	Logistik Service Deutschherrnufer 17 60594 Frankfurt
3	Kurt Fenner 0162/688799	M0023441234	1	Eco-Training	25.06.19	Cockpit Schulung Westanlage 165 44227 Darmstadt

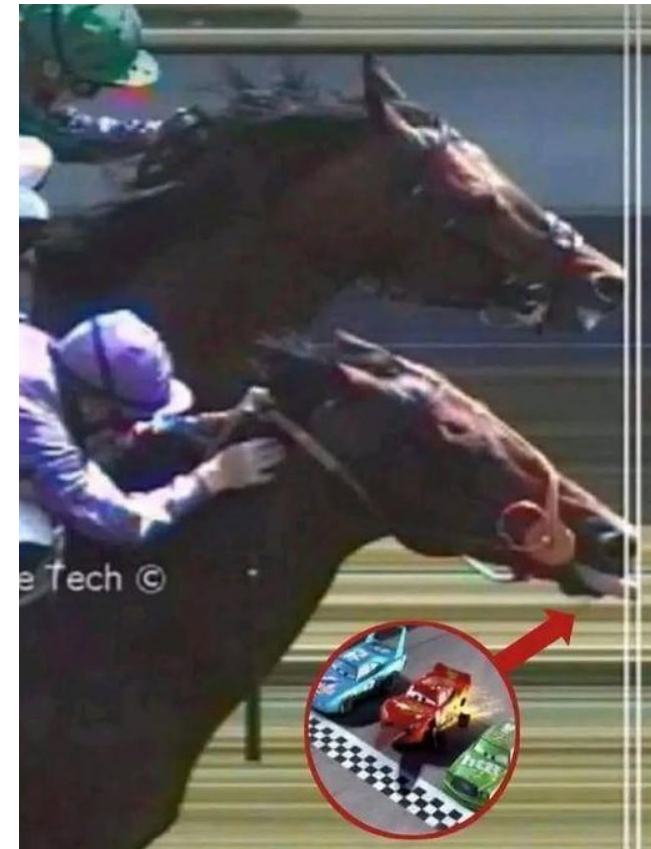
THEORIE – Anomalien

- Einfüge(Insert)-Anomalie
 - Einfügen von Daten die nicht vollständig sind. Bspw.: Primärschlüssel fehlt
- Änderungs(Update)-Anomalie
 - Beim Ändern von Datensätze muss zwangsläufig andere Datensätze verändert werden.
- Lösch>Delete)-Anomalie
 - Beim Löschen eines Datensatzes werden andere zusammenhängende Datensätze parallel mit gelöscht.

THEORIE – Referentielle Integrität wahren

Was passiert wenn ein Fremdschlüssel sich ändert oder gelöscht wird?

- **NO ACTION**
 - Es passiert nichts; der Schlüssel bleibt
- **CASCADE**
 - Die Änderung wird weitergegeben
- **RESTRICT**
 - Änderung wird verhindert, auch Primärtabelle
- **SET NULL**
 - Setzt den Verweis auf NULL
- **SET DEFAULT**
 - Setzt den Verweis auf einen definierten Wert



THEORIE – **NORMA**lisierung

Friend: "Does your cat bite?"
Me: "Worse. He judges you."



Was das und Warum?

ReNr	ReDatum	KuNr	KuName	KuOrt	ArtNr	ArtName	ArtPreis
1	13.08.1999	4	Meier	Hamburg	4, 2, 6	Prozessor, Speicher, Monitor	350, 120, 900
2	13.08.1999	1	Schulze	Berlin	6	Monitor	900
3	14.08.1999	5	Fischer	Hamburg	1, 4	Gehäuse, Prozessor	150, 350

Ziel: Dateninkonsistenzen und Anomalien vermeiden.

Normalisierung strukturiert Daten so, dass Redundanzen kontrollierbar und Inkonsistenzen vermeidbar werden

THEORIE – **NORMA**lisierung 1. Normalform

<u>ReNr</u>	<u>ReDatum</u>	<u>KuNr</u>	<u>KuName</u>	<u>KuOrt</u>	<u>ArtNr</u>	<u>ArtName</u>	<u>ArtPreis</u>
1	13.08.1999	4	Meier	Hamburg	4	Prozessor	350
1	13.08.1999	4	Meier	Hamburg	2	Speicher	120
1	13.08.1999	4	Meier	Hamburg	6	Monitor	900
2	13.08.1999	1	Schulze	Berlin	6	Monitor	900
3	14.08.1999	5	Fischer	Hamburg	1	Gehäuse	150
3	14.08.1999	5	Fischer	Hamburg	4	Prozessor	350

- Jedes Attribut in einer Tabelle ist Atomar
- Alle Schlüsselattribute vorhanden
- Keine Wiederholgruppe





THEORIE – **NORMA**lisierung 2. Normalform

Rechnungen

ReNr	ReDatum	KuNr#
1	13.08.1999	4
2	13.08.1999	1
3	14.08.1999	5

Kunden

KuNr	KuName	KuOrt
4	Meier	Hamburg
1	Schulze	Berlin
5	Fischer	Hamburg

Artikel

ArtNr	ArtName	ArtPreis
4	Prozessor	350
2	Speicher	120
6	Monitor	900
1	Gehäuse	150

Rechnungsposition

ReNr#	ArtNr#
1	4
1	2
1	6
2	6
3	1
3	4

- befindet sich in der ersten Normalform
- keine teilweisen Abhängigkeiten. Jedes Attribut **vollständig** vom gesamten Primärschlüssel abhängt

Boobs?!?!?!?!?



THEORIE – **NORMA**lisierung 3. Normalform

Rechnungen

ReNr	ReDatum	KuNr#
1	13.08.1999	4
2	13.08.1999	1
3	14.08.1999	5

Kunden

KuNr	KuName	KuOrt
4	Meier	Hamburg
1	Schulze	Berlin
5	Fischer	Hamburg

Artikel

ArtNr	ArtName	ArtPreis
4	Prozessor	350
2	Speicher	120
6	Monitor	900
1	Gehäuse	150

Rechnungsposition

ReNr#	ArtNr#
1	4
1	2
1	6
2	6
3	1
3	4

- befindet sich in der zweiten Normalform
- Keine **transitiven Abhängigkeiten** existieren (ein Attribut hängt nicht indirekt über ein anderes Nicht-Schlüsselattribut vom Primärschlüssel ab).

THEORIE – **NORMA**lisierung 3. Normalform



Keine **transitiven Abhängigkeiten** existieren (ein Attribut hängt nicht indirekt über ein anderes Nicht-Schlüsselattribut vom Primärschlüssel ab).

<u>MITARBEITERID</u>	NAME	ABTEILUNG	ABTEILUNGSLEITER
1	Anna	Verkauf	Müller
2	Ben	Verkauf	Müller
3	Carla	IT	Schmidt

COCK?!?!?!?!?



THEORIE – **NORMA**lisierung 3. Normalform

Keine **transitiven Abhängigkeiten** existieren (ein Attribut hängt nicht indirekt über ein anderes Nicht-Schlüsselattribut vom Primärschlüssel ab).

Abteilung

<u>ABTEILUNGID</u>	ABTEILUNG	ABTEILUNGSLEITER
1	Verkauf	Müller
2	IT	Schmidt

Mitarbeiter

<u>MITARBEITERID</u>	NAME	<u>ABTEILUNGID#</u>
1	Anna	1
2	Ben	1
3	Carla	2



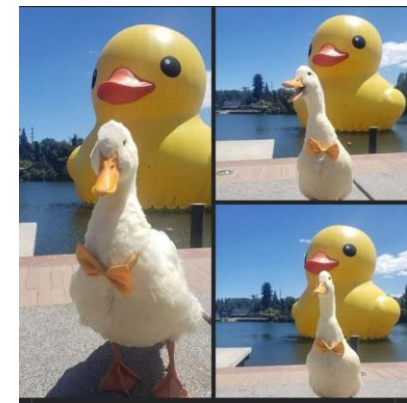
THEORIE – **NORMA**®lisierung

- **Vorteile**

- Reduziert **Fehlerquellen** (z. B. Tippfehler).
- Änderungen an wiederholten Daten (z. B. Übersetzung von „Herr“/„Frau“) sind zentral möglich.
- Datenmodell wird **stabiler und konsistenter**.

- **Nachteil**

- Abfragen erfordern **Zusammenführung mehrerer Tabellen** → erhöht **Komplexität** und kann **Performance** verringern.
- Mehraufwand bei Programmierung und Datenzugriff.



THEORIE – **NORMA**lisierung

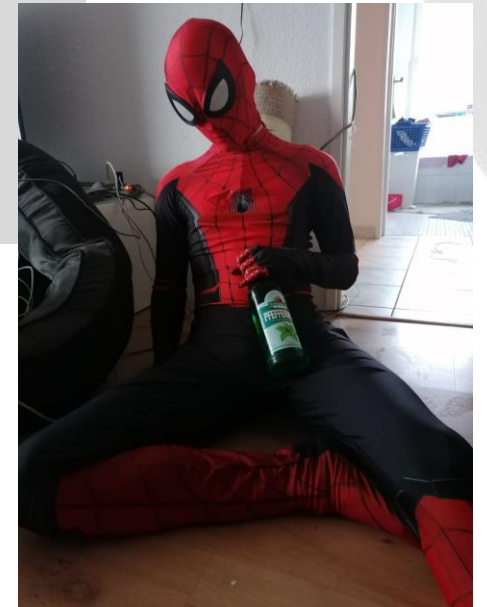
Normalisierung strukturiert Daten so, dass Redundanzen kontrollierbar und Inkonsistenzen vermeidbar werden, auf Kosten von Komplexität und manchmal Performance.



Intermission

(Das bedeutet Pause, für die die kein Englisch können)

- Fenster auf, einmal tief durchatmen, so tun als wärst du im Wellness-Retreat.
- Ein Glas Wasser trinken und dich fühlen wie ein Hydration-Profi.
- 10 Kniebeugen oder 15 Sekunden Plank (oder zumindest drüber nachdenken).
- Kurz aus dem Fenster starren und so tun, als würdest du über dein Leben reflektieren.
- Einen Snack holen und behaupten, das sei „Energie-Management“.
- Schultern kreisen, Nacken dehnen, Rücken einmal strecken.
- Eine Mini-To-Do-Liste für die nächsten 30 Minuten schreiben.
- Einen schlechten Witz googeln und minimal schmunzeln.
- Schreibtisch minimal aufräumen, damit es produktiv aussieht.
- Einmal bewusst langsam durch den Raum gehen – Dramaturgie inklusive.
- Eine Pflanze anschauen und sie motivieren, weiter Photosynthese zu betreiben. 🌱



ERM



<u>LeserID</u>	Name	Alter
1	Donald	14
2	Liam	86
3	Thom	26

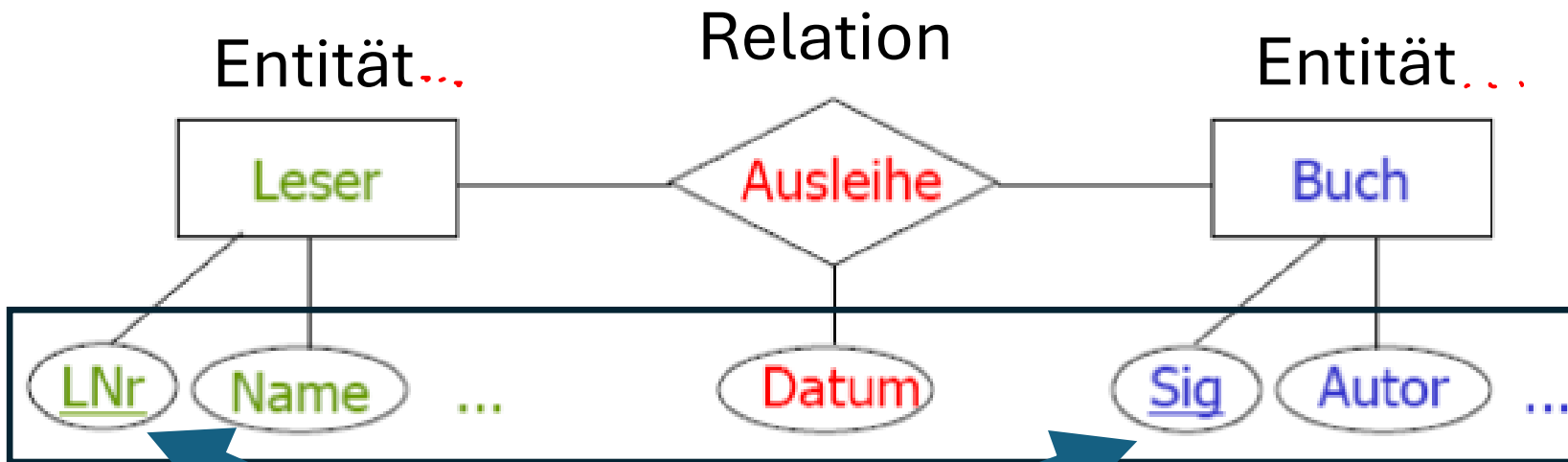
Entität

Entität

Entität



ERM



Primär Schlüsselattribute

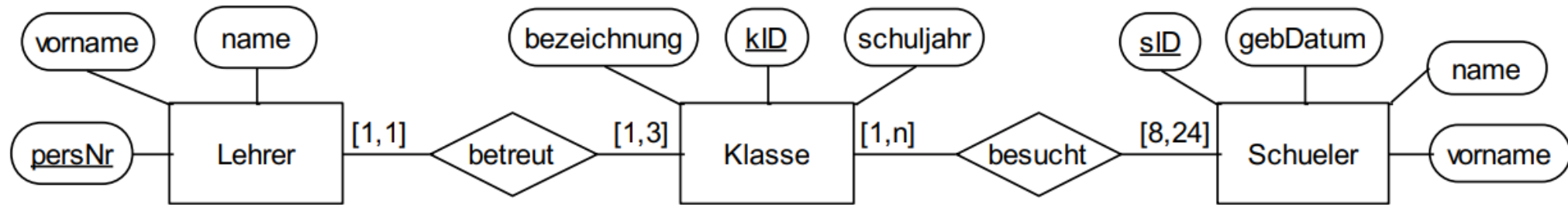
Fremdschlüssel werden im Diagramm **nicht dargestellt.**
Sie werden durch die Relation impliziert



ERM

[min, max]

Kardinalitäten

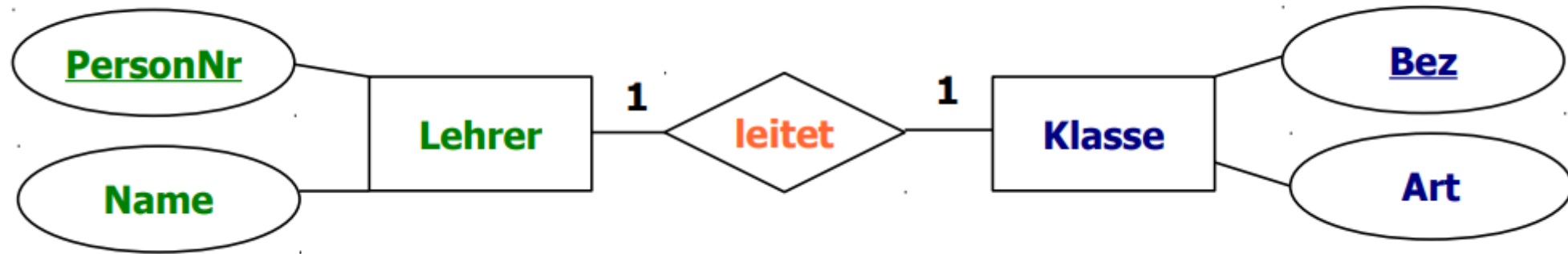


- 1:1 Beziehung
- 1:n Beziehung
- n:m Beziehung
- c:1 Beziehung



ERM

1:1 Beziehung



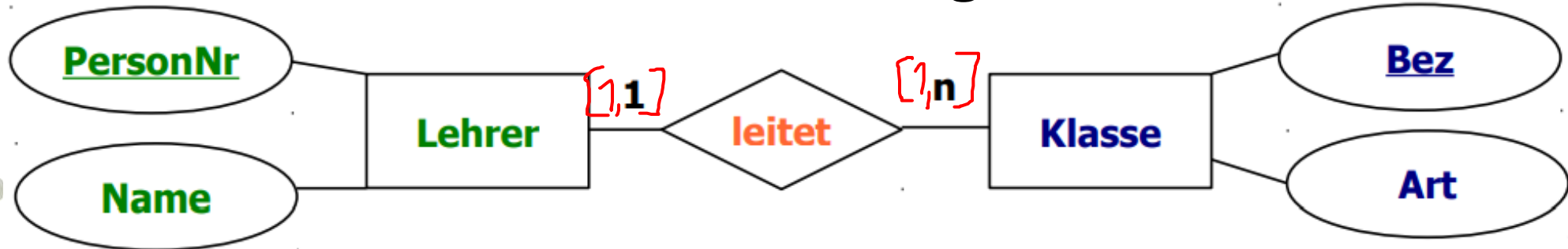
Die Informationen werden in einer Tabelle zusammengefasst.

Tabelle *Lehrer*

<u>PersonNr</u>	Name	Bez	Art
12345	Meier	12BGD0	Berufliches Gymnasium
14446	Müller	12FOI1	Fachoberschule
...			

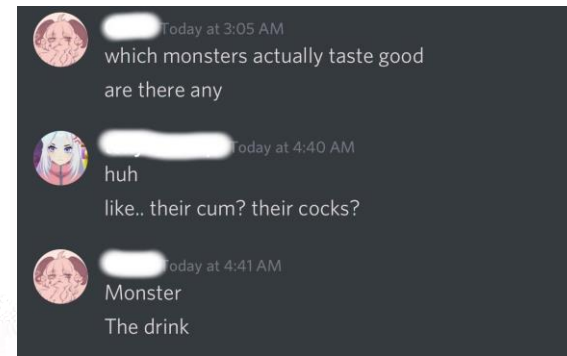
ERM

1:n Beziehung



2 Tabellen sind notwendig:

- Tabelle Lehrer
- Tabelle Klasse, wobei diese den Primärschlüssel von Lehrer als Fremdschlüssel speichern



ERM

1:n Beziehung

Tabelle *Lehrer*

<u>PersonNr</u>	Name	Vorname	Wohnort
12345	Meier	Fritz	Rauischholzhausen
14446	Müller	Walter	Wettenberg
...			

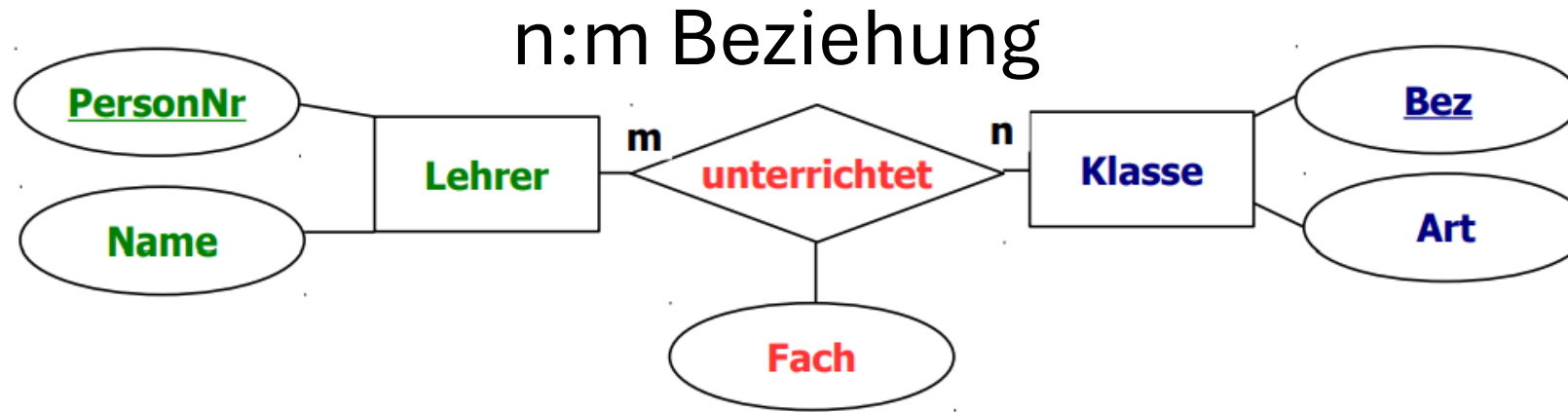
Tabelle *Klasse*

<u>Bez</u>	PersonNr#	Art	...
12BDD0	12345	Berufliches Gymnasium	
12FO11	14446	Fachoberschule	
13BGD0	12345	Berufliches Gymnasium	

PersonNr wird zum Fremdschlüssel in Klasse



ERM



3 Tabellen sind notwendig:

- Tabelle Lehrer
- Tabelle Klasse
- Neue Tabelle Unterricht mit den Primärschlüsseln von Lehrer und Klasse

ERM

n:m Beziehung

Tabelle *Lehrer*

<u>PersonNr</u>	Name	Vorname	Wohnort
12345	Meier	Fritz	Gießen
14446	Müller	Walter	Marburg
...			

Tabelle *Klasse*

<u>Bez</u>	Art
12BGD0	Berufliches Gymnasium
13BGD0	Berufliches Gymnasium
12FOI1	Fachoberschule

<u>U_ID</u>	<u>PersonNr#</u>	<u>Bez#</u>	Fach
1	12345	12BGD0	Englisch
2	14446	12FOI1	Mathematik
3	12345	13BGD0	Deutsch

Tabelle *Unterricht*

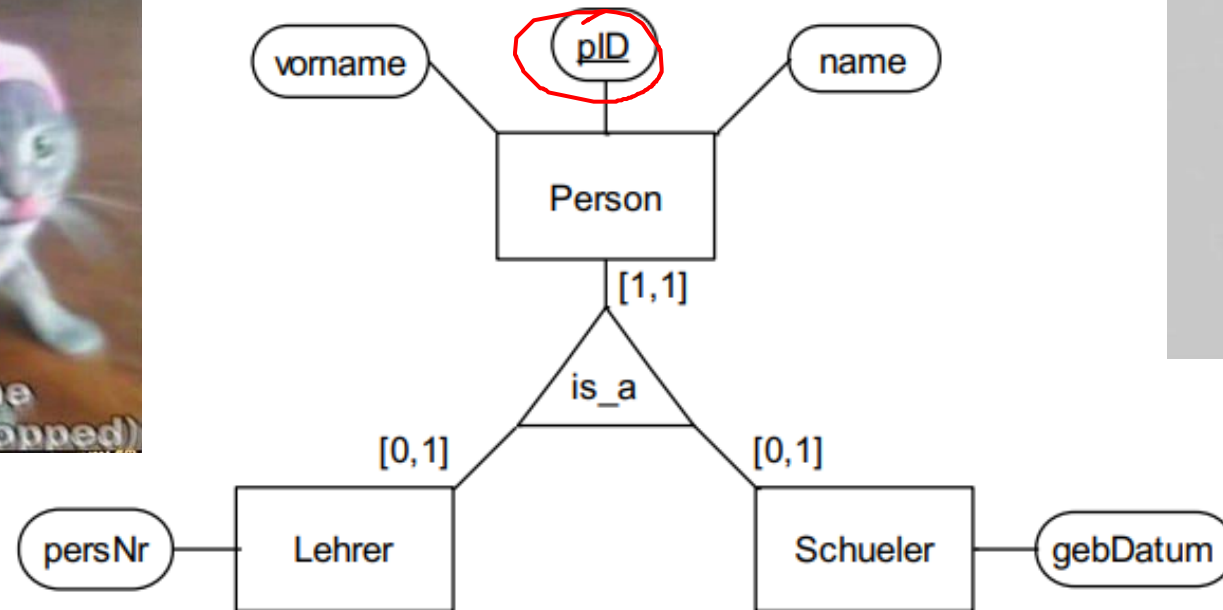
Fremdschlüsseln können zusammen ein Primärschlüssel bilden, müssen aber nicht



ERM



1:c Beziehung



Kowalski please give me an analysis



ERM

1:c Beziehung

Person

<u>pld</u>	vorname	name
20	Tahm	Kench
21	Hoarah	Loux
673	Donald	Duck
674	Sebastian	Schweinsteiger

Lehrer

persNr	<u>pld#</u>
1	20
2	21

Schüler

gebDatum	<u>pld#</u>
<i>10.3.2000</i>	673
<i>2.10.2001</i>	674

How the 10/0 Tahm Kench looks at your adc



ERM – Transformation

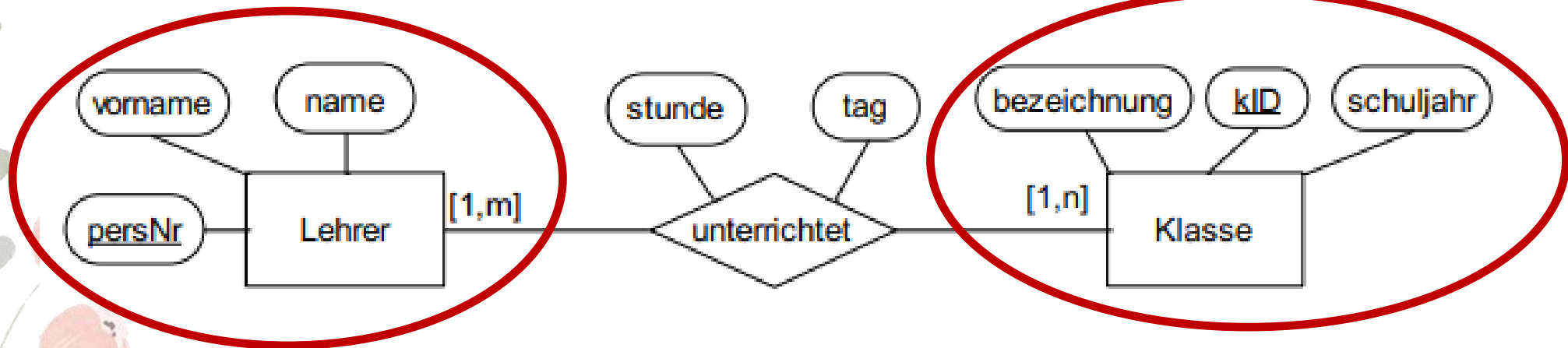


Lehrer(persNr, name, vorname)
Klasse(kID, bezeichnung, schuljahr, klassenlehrer#)
Schueler(sID, name, vorname, gebDatum)
Schueler_Klasse(sID#, kID#)



ERM – Transformation

REGEL 1: Für jede Entitätstyp, eine Relation erstellen mit Attributen und Primärschlüssel

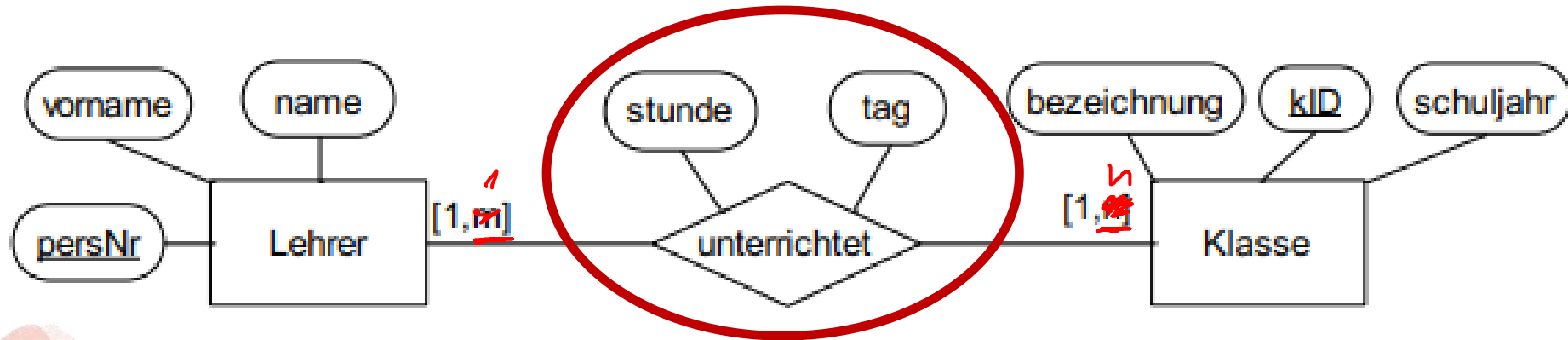


Lehrer(persNr, name, vorname)
Klasse(kID, bezeichnung, schuljahr)

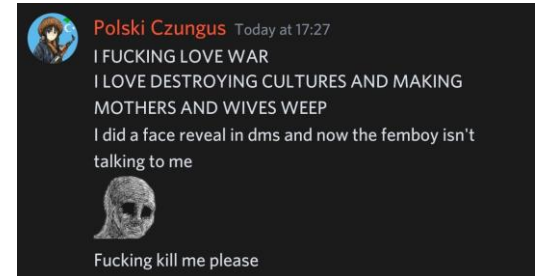


ERM – Transformation

REGEL 2: Für eine Beziehung mit eigenen Attributen wird eine gesonderte Tabelle angelegt.

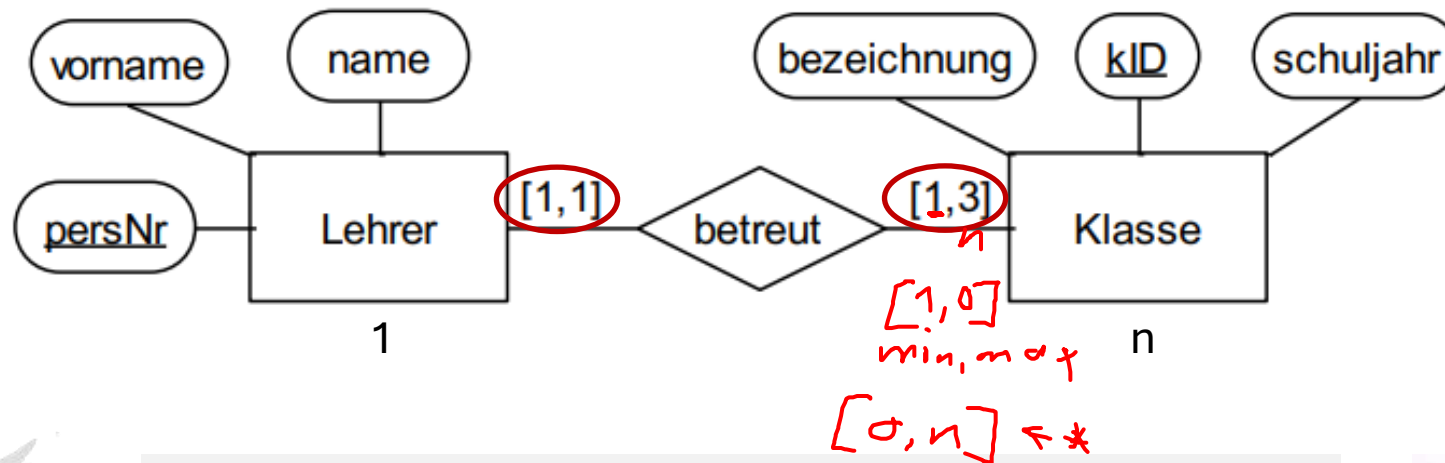


```
Lehrer(persNr, name, vorname)
Klasse(kID, bezeichnung, schuljahr)
Unterricht(stunde, tag, persNr#, kID#)
```



ERM – Transformation

REGEL 3: Bei einer 1:n Beziehungen, wird der Primärschlüssel des 1-Entitätstyps Fremdschlüssel im n-Entitätstyps

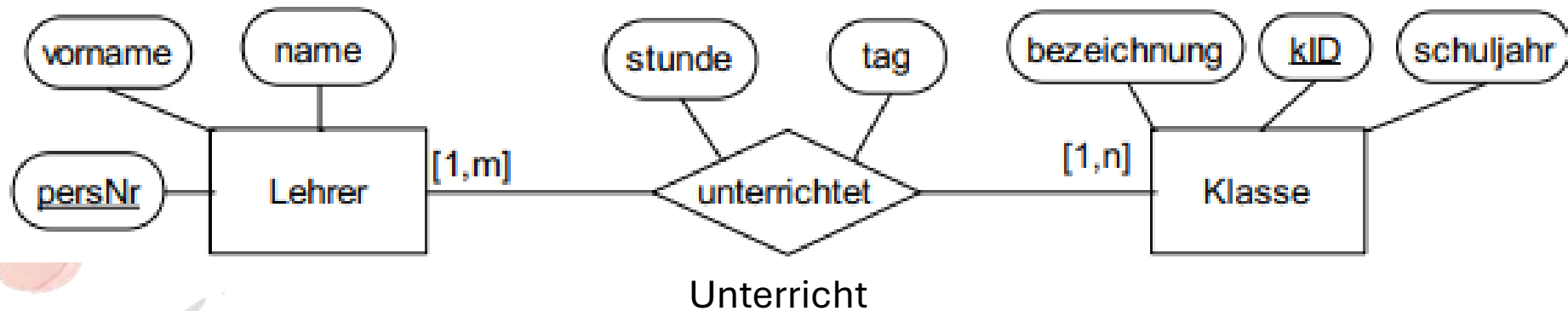


Lehrer(persNr, name, vorname)
Klasse(kID, bezeichnung, schuljahr, klassenlehrer#)



ERM – Transformation

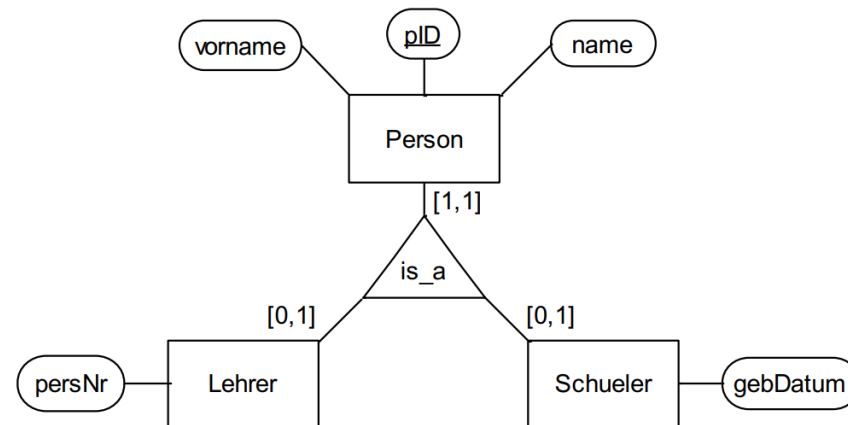
REGEL 4: Bei n:m Beziehung wird eine zusätzliche Relation erstellt, mit den Primärschlüsseln der beteiligten Relationen und (wenn abgebildet) zusätzlichen Attributen



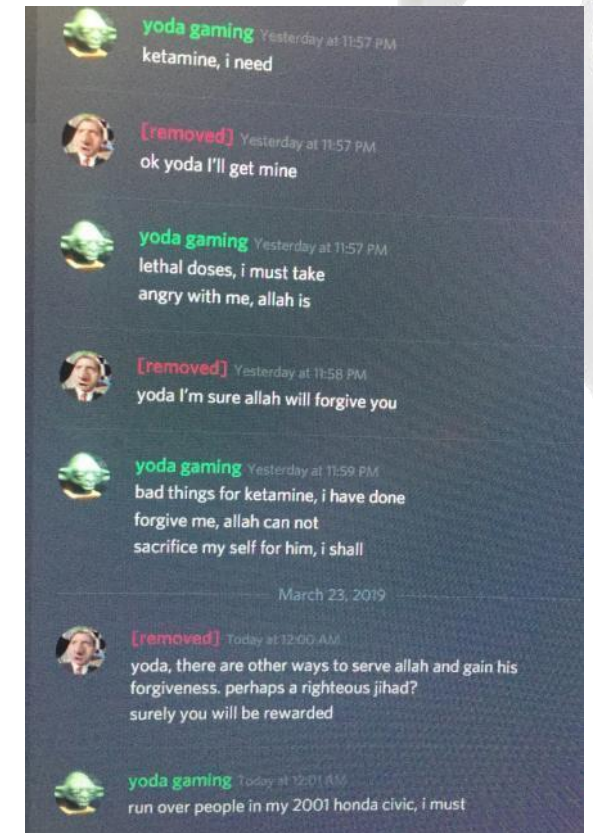
Lehrer(persNr, name, vorname)
Klasse(kID, bezeichnung, schuljahr)
Unterricht(stunde, tag, persNr#, kID#)

ERM – Transformation

REGEL 5: Bei c:1 Beziehungen spricht man von Generalisierung bzw. Spezialisierung. Der Primärschlüssel der Obermenge wird als Fremdschlüsseln der Teilmenge hinzugefügt

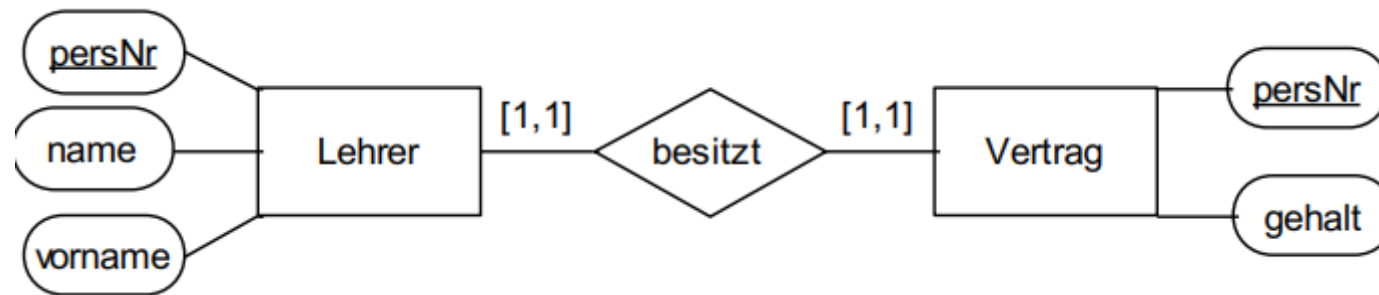


Person(pID, name, vorname)
Lehrer(pID#, persNr)
Schueler(pID#, gebDatum)



ERM – Transformation

REGEL 6: Bei 1:1 Beziehungen dient der Primärschlüssel einer der beteiligten Relation als Fremdschlüssel der andern. Alternativ können beide Entitätstypen in eine Relation zusammengefasst werden



Lehrer(persNr, name, vorname,
vertragNr#)
Vertrag(persNr, gehalt)

Lehrer(persNr, name, vorname)
Vertrag(persNr, gehalt,
lehrerNr#)

Lehrer(persNr, name, vorname,
gehalt)

Intermission (Es ist Zeit für ne Pause meine Kerle)

- Einmal aufstehen, Arme über den Kopf strecken, so tun als wärst du extrem sportlich.
- Kühlschrank aufmachen, reinschauen, wieder zumachen. Klassisches Ritual.
- 30 Sekunden lang bewusst langsam atmen.
- Eine Mini-Nachricht an dich selbst schreiben: „Ich regel das.“
- Ein Glas Wasser holen und dabei würdevoll gehen.
- Kurz aus dem Fenster schauen und so tun, als wärst du Hauptfigur in einem Indie-Film.
- Zwei Minuten lang alles ignorieren und einfach nichts tun. 😊



SQL

SELECT [DISTINCT] select_ausdruck, ...
[FROM Tabellen]
[WHERE where-Bedingung]
[GROUP BY {positive_ganzzahl | spalten_name} [ASC | DESC], ...]
[HAVING where-Bedingung]
[ORDER BY {positive_ganzzahl | spalten_name} [ASC | DESC], ...]
[LIMIT [offset,] zeilen]



SQL

- SELECT: gib diese Spalten aus
- FROM: aus Tabelle
- WHERE: wobei Bedingung ... gilt



```
SELECT NACHNAME, GEBURTSJAHR  
FROM AUTOR  
WHERE NACHNAME ='Lagerlöf';
```

SQL

Select * → id, name, Ort, datum

Select name → name

id	name	Ort	datum
----	------	-----	-------

```
SELECT DISTINCT *  
FROM AUTOR, BÜCHER  
WHERE NACHNAME = 'Lagerlöf' AND Veröffentlichung > 1990;
```

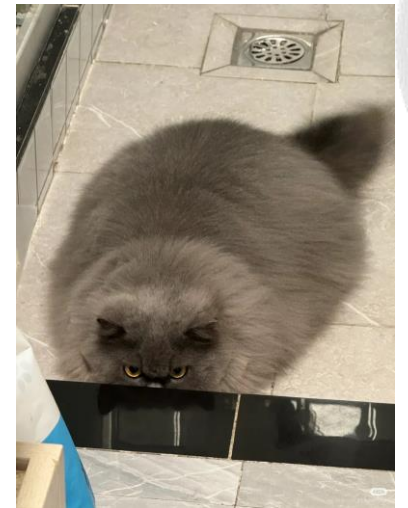
- SELECT * : gib alle Attribute aus
- DISTINCT: Doppelte Daten erscheinen nur einmal
- FROM mehrere Tabellen gleichzeitig ansprechen
- WHERE mit AND verknüpfbar
- Verschiedenste Operatoren (=, <, >, <=, ~~>=~~, +, -, ...)
- ...



SQL

```
SELECT DISTINCT *  
FROM AUTOR, BÜCHER  
WHERE NACHNAME ='Lagerlöf' AND Veröffentlichung > 1990;
```

- SELECT * : gib alle Attribute aus
- DISTINCT: Doppelte Daten erscheinen nur einmal
- FROM mehrere Tabellen gleichzeitig ansprechen
- WHERE mit AND verknüpfbar
- Verschiedenste Operatoren (=, <, >, <=, +, -, ...)
- ...



SQL

```
SELECT DISTINCT *, Preis / 2 AS ,Rabatt'  
FROM AUTOR, BÜCHER  
WHERE NACHNAME ='Lagerlöf' AND Veröffentlichung > 1990;
```



- SELECT * : gib alle Attribute aus
- DISTINCT: Doppelte Daten erscheinen nur einmal
- FROM mehrere Tabellen gleichzeitig ansprechen
- WHERE mit AND, OR verknüpfbar
- Verschiedenste Operatoren (=, <, >, <=, +, -,...)
 - Auch bei SELECT möglich (Preis / 2)
- AS ... : Benennt Spalte bei Ausgabe um
- ... (Schaut euch Lernblatt an)

SQL

Aggregatfunktionen SELECT...

COUNT(*): Anzahl von Reihen

SUM(...): Die Summe einer Spalte

AVG(...): Der Durchschnittswert einer Spalte

MAX(...): Der höchste Wert in einer Spalte

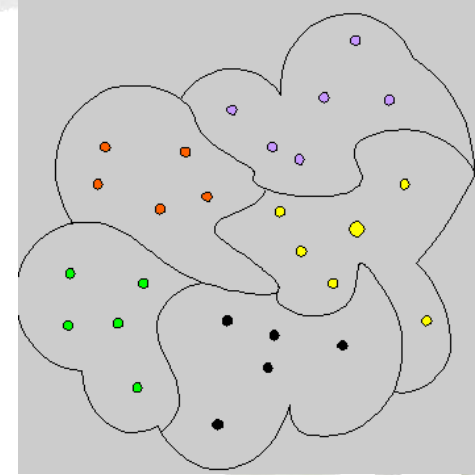
MIN(...): Der niedrigste Wert in einer Spalte

Gut Kombinierbar mit GROUP

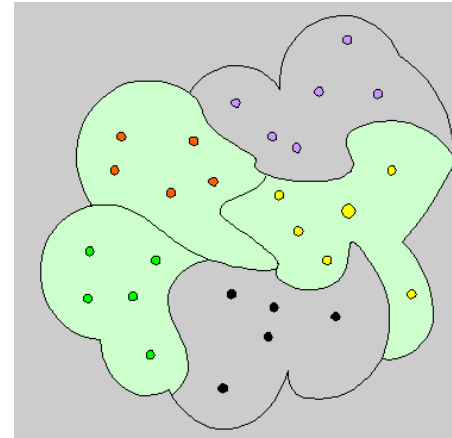


SQL

region
Afrika
Antarktis
Arktik-Region
Asien
Europa
Mittelamerika
Mittlerer Osten
Nordamerika
Osteuropa
Ozeanien
Südamerika
Südostasien
Welt

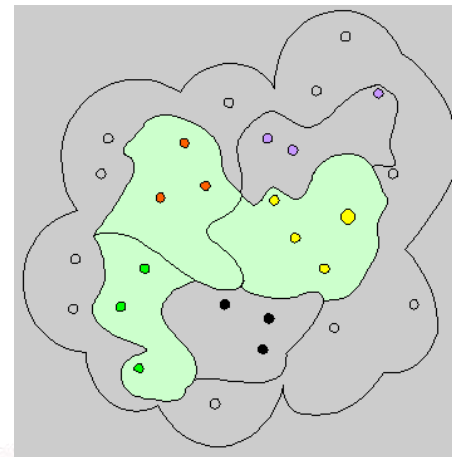


```
select Region, sum(Einwohner)
from cia
group by Region
having sum(Einwohner) > 1E08
```



Region	sum(Einwohner)
Afrika	123.337.822,00
Asien	2.819.135.450,00
Nordamerika	375.912.439,00
Südamerika	172.860.370,00
Südostasien	224.784.210,00

```
select Region, sum(Einwohner)
from cia
where Einwohner > 1E08
group by Region
having sum(Einwohner) > 250000000
```

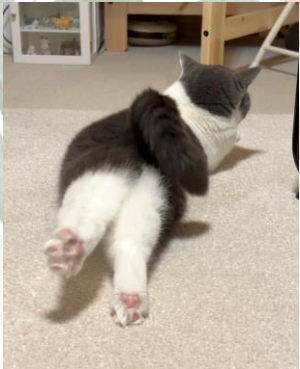


Region	sum(Einwohner)
Asien	2.819.135.450,00
Nordamerika	375.912.439,00

SQL - Unterabfragen

```
SELECT Buch.Titel AS Ungelesen  
FROM Buch  
WHERE NOT IN ( SELECT BuchNr FROM Ausleihe WHERE  
Buch.BuchNr = Ausleihe.BuchNr)
```

```
SELECT Buch.Titel AS Ungelesen  
FROM Buch  
WHERE NOT IN (1234, 5123, 213)
```



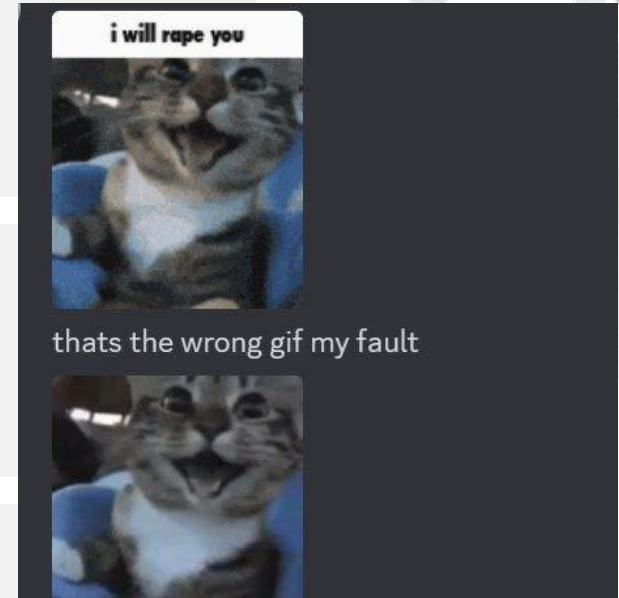
SQL

JOIN – Tabellen verbinden/verknüpfen

```
SELECT *  
FROM Buch, Autor  
WHERE Autor.AutorId = Buch.AutorId
```

```
SELECT *  
FROM Buch  
JOIN Autor ON Autor.AutorId = Buch.AutorId
```

```
SELECT *  
FROM Buch  
JOIN Autor USING(AutorId)
```



SQL – Was brauchen wir?

i forgot



Ende

