Relationale Datenbankenmodelle *Grundlagen*

- ist das am weitesten verbreiteten Datenmodell, welches in der Datenbankentwicklung als Standard genutzt wird
- vier Elemente machen diese aus :
    - Tabellen
    - Attribute
    - relationale algebra
- Was ist eine Entität?
    - Stellt ein Objekt eines Themenkreis da Mit Objekten mit gleichen Merkmalen
    - *zb. Firma,Student,Kurs,Dozent*
    - Entitätsmenge: *Alle Datensätze einer Entität* repräsentiert alle Datensätze, die zu einer Entiät gehören
- **Relation(Tabelle)**
    - umfasst eine Entität inklusive der dazugehörigen Entitätsmenge.
    - eine Komplette Relation besteht aus tupeln eine Entität *Tupel -> rep. Alle Merkmalswerte einer Entiät einer Entitätsmenge*
    - Alle Tupel einer Entität bilden die Entitätsmenge
- Attribut
    - Ein Attribut beschreibt die Entität

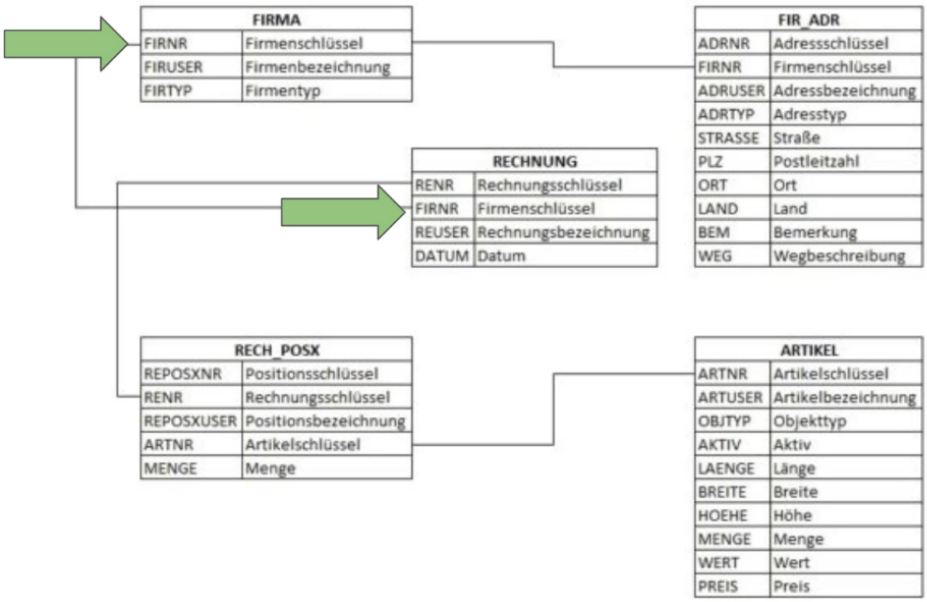
> Entität/Tupel --> Konkreter Datensatz Entitätsmenge --> Alle Datensätze Attribute --> Spalten/Felder Beziehungen --> Verknüpfungen zwischen Tabellen

Eigenschaften des Relationalen Datenbankmodells

- Jeder Datensatz besteht aus einer Tupel von eigenschaften
- --> Spalten der Tabelle
- Realtions schema:
    - Anzahl, Typ der Attribute
    - Verknüpfungen *Beziehungen* über Primärschlüssel herstellen
    - zb --> `id INT(11) PRIMARY KEY` *Primary key wird verwendet um Tabellen miteinander zu verbinden*

Beispiel verbundene Tabellen:

Beispiel für ein Relationales Datenbankmodell



> Dieses Beispiel braucht korrekte Normalisierung und deren Normalformen um korrekt erstellt werden zu können

Normalisierung

> ziel ist eine Redundanzfrei Datenspeicherung --> Daten sind nur EINMAL irgendwo gespeichert

- Normalisierung soll außerdem auch Anomalien entfernen, im Normalisierungprozess gibt es fünf Noramlformen
  - In der Datenbankentwicklung ist die *Dritte Normalform* oft ausreichend *perfekte balance aus Redundance performance*
  - *Eine redundanzenfreie datenbank ist eine Datenbank ohne doppelte Daten auskommt*
  - jedoch kann Redundanz manchmal aus performancegründen Sinn machen
  - --> Redundanzen können mittels der *Normalisierung* entfernt werden

Anomalien

- Einfüge
  - bei fehlerhaften design können manchmal Daten gar nicht übernommen werden, wenn zb der Primärschlüssen keinen Wert erhalten hat
- Änderung *Update*
  - Dinge werden nicht automatisch mitgeändert *Inkonsistente Werte*
- Löschen
  - Eine Datei ist gelöscht. Ist aber noch in einem anderen Part der Datenbank vorhanden

Normalisierung und Abhängigkeit

- Funktionale Abhängigkeit : zu jedem x genau ein y
- Voll funktionale Abhängigkeit alle teile sind funktional abhängig

- Transitive Abhängigkeit: y von x und z von y somit z von x *wird mit 3.Normalform erreicht*

Normalformen

- Nullte Normalform
  - Sammeln von unstrukturierten information --> alle Datenelemte der realen Welt ist in der Tabelle zusammengefasst
  - zb. unnormalisierte Rechnungsinfromationen *Informationen sind unnormalisiert*
- Erste Normalform (1NF)
  - alle Infromationen liegen in der Tabelle *atomar* vor
  - alle informationen haben eine eigen Tabellenspalte
  - zb PLZ und ORT liegen nicht in der Gleichen spalte vor
  - Name / Vorname Spalten
  - *Man kann die werte nicht weiter aufteilen* --> Atomatisiert
- Zweite Normalform (2NF)
  - PRÜFT ob eine vollständige funktionale tabelle vorliegt oder nur *eine* abhängigkeit besteht
  - wid oft schon passiv erreicht bei Erstellung eines ER-Diagramms
  - Logische Aufspaltung von Komplexen Sachverhalten, Geschäftsprozesse in Relation bringen
  - jedes nichtschlüsselattribut muss von jedem SchlüsselKandidaten voll Funktional abhängig sein
  - Beispiel:
    - Nachname ist nicht eindeutig daher wird jedem Kunden eine Kuden Nummer Zugeordnet

R.-Nr.	Datum	Name	Vorname	Straße	Hnr.	PLZ	Ort	Artikel	Anzahl	Preis	Währung
187	01.01.2012	Mustermann	Max	Musterstr.	1	12345	Musterort	Bleistift	5	1,00	Euro

Rechnung

R.-Nr.	Datum	Knr.
187	01.01.2012	007

Kunde

Knr.	Name	Vorname	Straße	Hnr.	PLZ	Ort
007	Mustermann	Max	Musterstr.	1	12345	Musterort

Rechnungsposition

R.-P.-Nr.	R.-Nr.	Art.-Nr.	Anzahl
1	187	69	5

Artikel

Art.-Nr.	Artikel	Preis
69	Bleistift	1,00

    -
    - > Beispiel für 2. Normalform
- Dritte Normalform(3NF)
  - Wenn es sich in 2NF befindet und kein Atrribut Transativ ist
  - zb plz in eine Eigene Tabelle abspalten und sie under der PLZ zuordnen --> indirekte Abhängigkeit

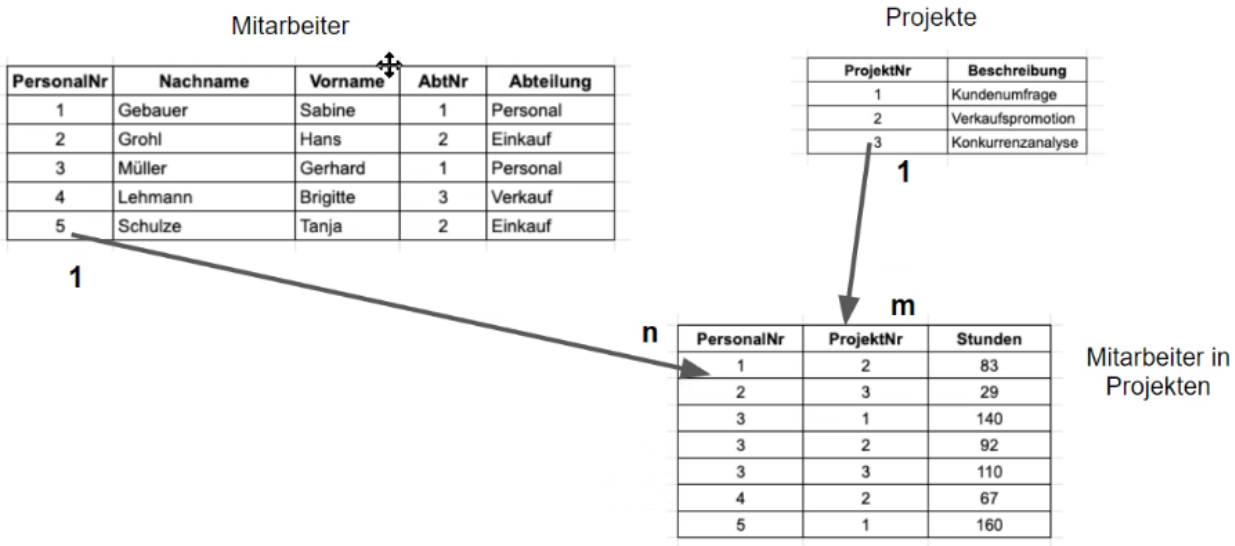
Beispiel:

PersonalNr	Name	AbtNr	Abteilung	ProjektNr	Projektbeschreibung	Stunden
1	Sabine Gebauer	1	Personal	2	Verkaufspromotion	83
2	Hans Grohl	2	Einkauf	3	Konkurrenzanalyse	29
3	Gerhard Müller	1	Personal	1,2,3	Kundenumfrage, Verkaufspromotion, Konkurrenzanalyse	140, 92, 110
4	Brigitte Lehmann	3	Verkauf	2	Verkaufspromotion	67
5	Tanja Schulze	2	Einkauf	1	Kundenumfrage	160

> Die Beispieltabelle in der 0NF keine Normalisierung *Zusammentragen der Daten* Hier gibt es viel redundanz zb. VerkaufsPromotion

PersonalNr	Name	Vorname	AbtNr	Abteilung	ProjektNr	Projektbeschreibung	Stunden
1	Gebauer	Sabine	1	Personal	2	Verkaufspromotion	83
2	Grohl	Hans	2	Einkauf	3	Konkurrenzanalyse	29
3	Müller	Gerhard	1	Personal	1	Kundenumfrage	140
3	Müller	Gerhard	1	Personal	2	Verkaufspromotion	92
3	Müller	Gerhard	1	Personal	3	Konkurrenzanalyse	110
4	Lehmann	Brigitte	3	Verkauf	2	Verkaufspromotion	67
5	Schulze	Tanja	2	Einkauf	1	Kundenumfrage	160

> Schlüsselkandidaten herausfinden, Atomatisieren, Name vorname trennen. *Elementare Daten*
>
> zb Wurde Müller aufgeteilt damit es mehr sortiert ist der punkt wurde Atomatisiert
>
> Problem ist aber das es jetzt keinen Eindeutigen Primärschlüssel mehr gibt



PersonalNr	Nachname	Vorname	AbtNr	Abteilung
1	Gebauer	Sabine	1	Personal
2	Grohl	Hans	2	Einkauf
3	Müller	Gerhard	1	Personal
4	Lehmann	Brigitte	3	Verkauf
5	Schulze	Tanja	2	Einkauf

ProjektNr	Beschreibung
1	Kundenumfrage
2	Verkaufspromotion
3	Konkurrenzanalyse

PersonalNr	ProjektNr	Stunden
1	2	83
2	3	29
3	1	140
3	2	92
3	3	110
4	2	67
5	1	160

Lösung: Separate Relationen

> Für die umwandlung in 2NF Personal, Abteilung und Project trennen
>
> ProjektNr ist nicht abhängig vom primärschlüssel daher sollte aufgetrennt werden *seperate realtion* n zu m relation

> zb kann ein neues projekt hinzugefügt werden indem man einfach einen neuen eintrag in Projekte macht oder zb namensänderung

Mitarbeiter

PersonalNr	Nachname	Vorname	AbtNr
1	Gebauer	Sabine	1
2	Grohl	Hans	2
3	Müller	Gerhard	1
4	Lehmann	Brigitte	3
5	Schulze	Tanja	2

n

1

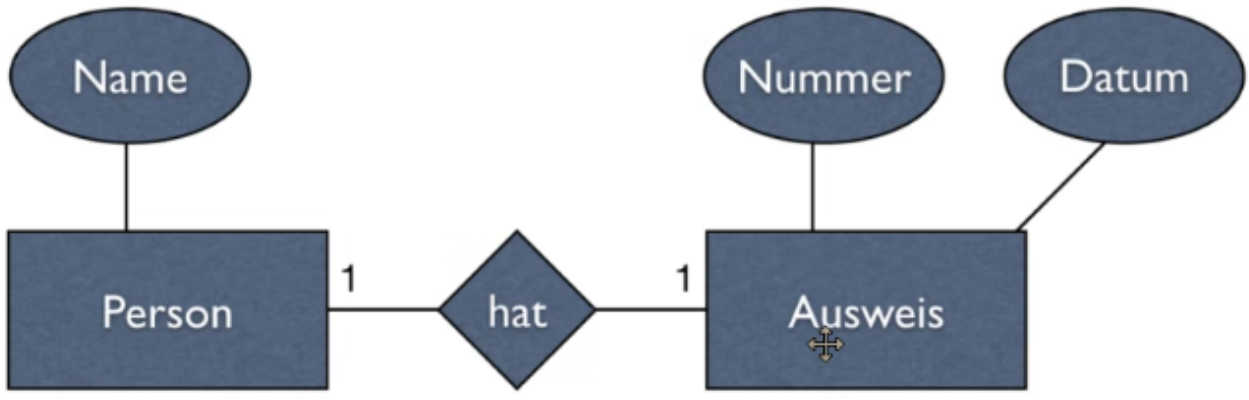
Abteilungen

AbtNr	Abteilung
1	Personal
2	Einkauf
3	Verkauf

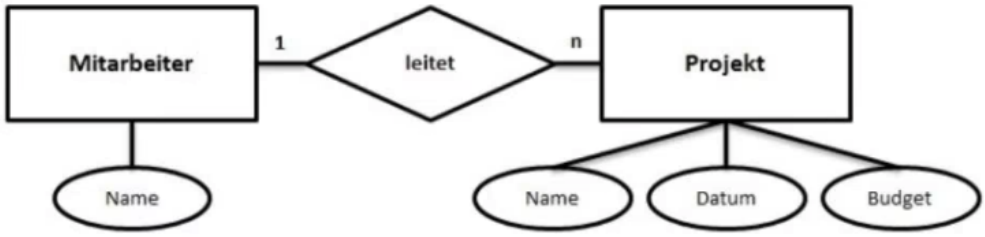
> bei 3NF könnte man die abteilung auch nochmal abspalten

Enitiy-Relationship-Modell *ER*

> Gebildet aus :
>
> - Entitäten *Individuell identifizierbares model*
> - Beziehungen *Verknüpfung/zusammenhang zwischen zwei oder mehreren Entitäten*
> - Atributte *Eine Eigenschaft die im Kontext zu einer Entität steht* Ähnlich wie bei OOP
> - Kardinalität In welcher relation zwei Entitäten stehen zb 1:n oder 1:1 ein mitarbeiter kan 1:n projekte leiten kann mehre muss aber auch keines. 1:1 währe zb ein projekt mit einem Startdatum
>   - 1:1 Benutzt man nur wenn man zb eine tabelle spaltet

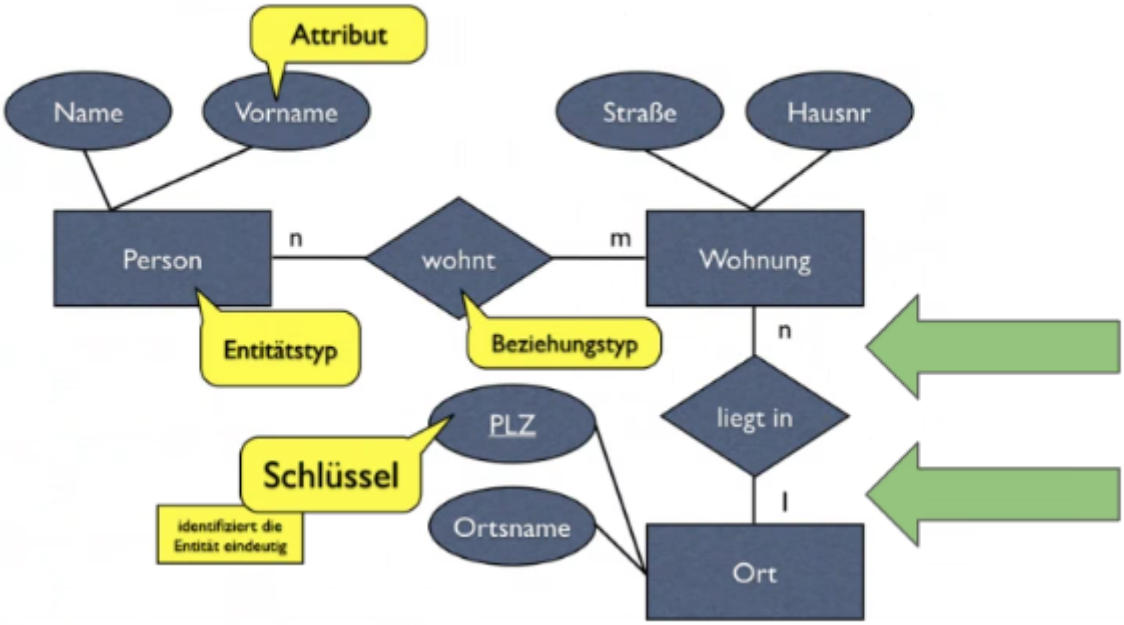


Beispiel:



Ein Mitarbeiter hat einen Namen.
Ein Projekt hat einen Namen, ein Datum und ein Budget.
Ein Mitarbeiter kann mehrere Projekte leiten, aber nur ein Projekt kann von genau einem Mitarbeiter geleitet werden.
Diese o.g. Notation nennt man **Chen-Notation**, diese beinhaltet auch die **Kardinalität**.

Genaueres: Chen-Notation



> Beispiel einer 1:n Beziehung

Video: Entity Relationsjhip Modellierung