



Foto: Thomas Josek

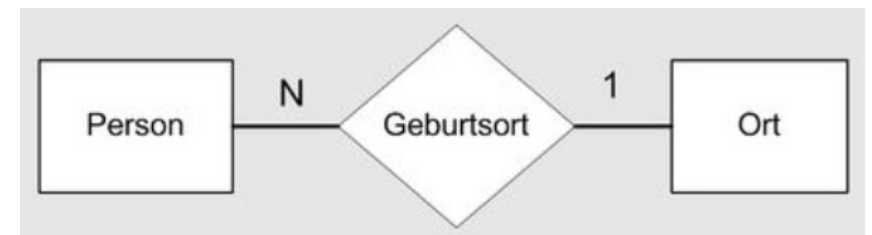
Datenmodellierung

Woche 4: Relationale Datenbanken

Institut für Digital Humanities, Historisch-Kulturwissenschaftliche Informationsverarbeitung | Prof. Dr. Øyvind Eide

Relationales Modell (Codd 1970), Entity-Relationship-Modell (Chen 1976)

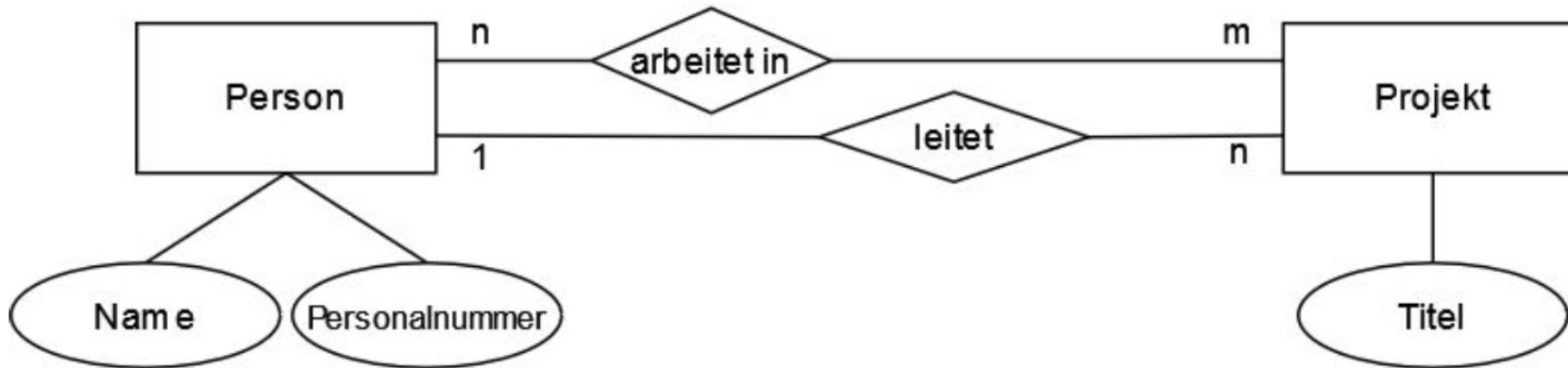
- Beschreibt für einen relevanten Ausschnitt der Realität die Struktur der Daten und ihrer Beziehungen zueinander
- **Entität**: individuelles, eindeutig identifizierbares Objekt oder Sachverhalt, z.B. der Person Øyvind Eide
- **Entitätstyp/Beziehungstyp** (Tabelle): Entitäten/Beziehungen können zusammengefasst werden, z.B. Personen, Geburtsorte
- **Attribut** (Eigenschaft, Spalte): Entitäten haben Attribute, z.B. haben Personen das Attribut Vorname
- **Relation** (zweidimensionale Tabelle): Relationen geben die Beziehungen zwischen Entitäten wieder
- **Tupel** (Zeile): Instanz, ein Datensatz
- Entity-Relationship-Diagramm zur Darstellung des konzeptuellen Entwurfs
- Kardinalität (1:1, 1:n, n:m)



Übung: Anforderungserhebung und Konzeption einer relationalen Datenbank

- Wählen Sie ein Beispiel aus (weitere Details können Sie sich ggf. ausdenken):
 - Datenbank über Personen und ihre Haustiere
 - Datenbank von Bestellungen in einem Onlineshop
 - Datenbank über Filme und ihre Schauspieler*innen und Regisseur*innen
 - Datenbank der Gebäude der Universität zu Köln
- Überlegen Sie sich:
 1. Welche Entitäten sind dabei relevant?
 2. Welche Beziehungen bestehen zwischen den Entitäten?
 3. Welche Attribute haben die Entitäten?
 4. Welche Kardinalitäten haben die Beziehungen?
- Machen Sie sich Notizen und/oder erstellen Sie Diagramme

Übung: Anforderungserhebung und Konzeption einer relationalen Datenbank



- Überlegen Sie sich:
 1. Welche Entitäten sind dabei relevant?
 2. Welche Beziehungen bestehen zwischen den Entitäten?
 3. Welche Attribute haben die Entitäten?
 4. Welche Kardinalitäten haben die Beziehungen?
- Machen Sie sich Notizen und/oder erstellen Sie Diagramme

Logisches Datenbankschema

Nachdem man die Anforderungen für die Datenbank geklärt und ein konzeptuelles Modell der Datenbank erstellt hat (wie in der Übung), kann die Datenbank jetzt logisch modelliert werden:

- Für jeden Entitätstyp wird eine Tabelle angelegt
 - Name der Tabelle: Entitätstyp
 - Spalten der Tabelle: Attribute
 - Evtl. Aggregation/Generalisierung (z.B. 'Gemälde' und 'Skulpturen' zusammenfassen zu Entitätstyp 'Werke')
- **Schlüssel** zur eindeutigen Identifizierung jedes Datensatzes
- **Normalisierung** zur Vermeidung von Redundanz
 - Normalformen
- Integritätsbedingungen
- Datentypen

Normalisierung

- Wird in der Phase des Entwurfs durchgeführt
- Änderungen der Tabellenstruktur
 - Überführung komplexer Tabellen in einfache Tabellen durch Aufteilung der Attribute einer Tabelle auf mehrere Tabellen
- Ziel: Vermeidung unerwünschter Nebeneffekte wie Redundanz und Inkonsistenz
- Keine mehrfache Speicherung von Daten
- Konsistenz
 - Keine widersprechenden Daten oder unterschiedlichen Werte für das gleiche Phänomen
- Änderungs-, Einfüge- und Lösch-Anomalien vermeiden
 - Stabile und flexible Datenstrukturen
 - Aktualisierung der Datenbank vereinfachen → Änderungen lediglich an einer Stelle
 - Unnötige Verwendung von Speicherplatz vermeiden

Schlüssel

- **Schlüssel:** Attribut oder Attributkombination (= Spalte(n)), die einen Datensatz (= Zeile) eindeutig identifiziert
 - Kombination von Attributen wird genutzt, wenn ein Attribut zur eindeutigen Identifizierung nicht ausreicht
 - Häufig werden fortlaufende Zahlen als Schlüssel zu einer Tabelle hinzugefügt (IDs)
- **Schlüsselkandidat:** Schlüssel mit minimaler Anzahl Attribute
 - Eine Relation kann mehrere Schlüsselkandidaten haben
- **Schlüsselattribut:** Attribut, das Teil mindestens eines Schlüssels/Schlüsselkandidatens ist
 - Alle anderen Attribute sind Nicht-Schlüsselattribute

Schlüssel

- **Primärschlüssel:** beliebig ausgewählter Schlüsselkandidat, der zur eindeutigen Identifizierung jeder Zeile benutzt wird
 - Jede Tabelle sollte einen Primärschlüssel haben
 - Zusammengesetzter Primärschlüssel: wenn der Primärschlüssel aus mehreren Attributen besteht
- **Fremdschlüssel:** Schlüssel einer Tabelle, der auf den Primärschlüssel einer anderen Tabelle verweist

Schlüssel und Kardinalität

Beispiel: Person 1 : 1 Ausweis

→ Primärschlüssel einer Tabelle muss als Fremdschlüssel in der anderen eingesetzt werden (Richtung egal)

Person			Ausweis		
PersID	Vorname	Nachname	AuswID	Enddatum	<i>PersID</i>
1	Maria	Solano	388	21.07.2025	1

Person				Ausweis	
PersID	Vorname	Nachname	<i>AuswID</i>	AuswID	Enddatum
1	Maria	Solano	388	388	21.07.2025

Schlüssel und Kardinalität

Beispiel: Museum 1 : n Kunstwerk

→ Entitätstyp auf der 'n'-Seite wird um den Fremdschlüssel ergänzt

Museum

MID	Name	Ort
1	Museum Ludwig	Köln

Werk

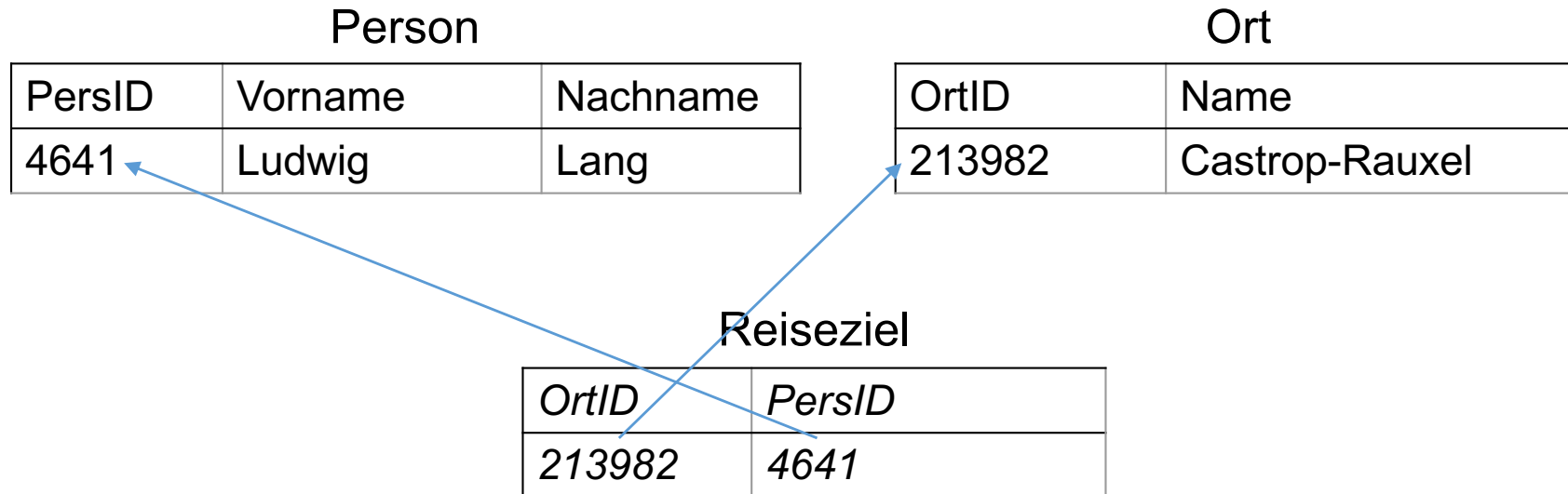
WID	Titel	Jahr	<i>MID</i>
521	Wildschweine	1913	1

Schlüssel und Kardinalität

Beispiel: Person n : m Ort (Reiseziel)

→ n : m Beziehung wird durch eine dritte Tabelle in zwei 1 : n Beziehungen aufgeschlüsselt

→ Primärschlüssel der neuen Tabelle meist zusammengesetzter Schlüssel aus Primärschlüsseln der Ausgangstabellen



Erste Normalform (1NF)

- **Alle Attribute der Tabelle enthalten atomare Werte und die Tabelle ist frei von Wiederholungsgruppen**
- Oder: kein Attribut kann in weitere sinnvolle Teilbereiche aufgespaltet werden
 - **Atomar:** Wertebereiche sind nicht zusammengesetzt
 - **Wiederholungsgruppen:** Attribute enthalten gleiche/gleichartige Informationen
- **Vorgehen für 1NF:**
 1. Nicht-atomare Daten in Spalten aufteilen
 2. Gleichartige Daten in Zeilen aufteilen
 3. Berechnete Werte weglassen
 4. Primärschlüssel festlegen
- **Nutzen:** Abfragen werden ermöglicht/erleichtert

Erste Normalform (1NF)

ISBN	Name	Titel	Standort
978-3-518-36777-3	Max Frisch	Andorra	Regal 20-4, Regal 20-5
978-3-7891-1851-7	Astrid Lindgren	Pippi Langstrumpf	Regal 13-99

ISBN	Name	Titel	Standort1	Standort2
978-3-518-36777-3	Max Frisch	Andorra	Regal 20-4	Regal 20-5
978-3-7891-1851-7	Astrid Lindgren	Pippi Langstrumpf	Regal 13-99	

Erste Normalform (1NF)

<u>ISBN</u>	Vorname	Nachname	Titel	<u>ExemplarID</u>	Standort
978-3-518-36777-3	Max	Frisch	Andorra	1	Regal 20-4
978-3-518-36777-3	Max	Frisch	Andorra	2	Regal 20-5
978-3-7891-1851-7	Astrid	Lindgren	Pippi Langstrumpf	1	Regal 13-99

- Trennung von Name in Vorname und Nachname
 - Erlaubt es, nach dem Nachnamen zu sortieren
- Aufteilung der Standorte auf mehrere Zeilen
 - Es können weitere Standorte hinzugefügt werden ohne die Tabellenstruktur durch Hinzufügen neuer Attribute (Spalten) ändern zu müssen
- Zusammengesetzter Primärschlüssel aus ISBN und ExemplarID

Zweite Normalform (2NF)

1NF + Jedes Nicht-Schlüsselattribut ist von jedem Schlüsselkandidaten vollständig funktional abhängig

- Vollständig funktional abhängig: Jedes Nicht-Schlüssel-Attribut ist von allen Attributen eines zusammengesetzten Schlüsselkandidaten abhängig
- Relationen in 1NF, deren Schlüsselkandidat(en) nicht zusammengesetzt sind, erfüllen automatisch 2NF!

Vorgehen für 2NF:

- Primärschlüsselattribute betrachten:
 - Zu jedem Attribut bestimmen, von welchem Primärschlüsselattribut es abhängig ist
 - Zu jedem zusammengesetzten Primärschlüsselattribut eine eigene Tabelle erstellen mit den Attributen, die von dieser Kombination abhängig sind

Nutzen: Redundante Daten vermeiden, indem eine Tabelle in mehrere jeweils monothematische Tabellen aufgeteilt wird; Vermeiden von Anomalien (Inkonsistenzen)

Zweite Normalform (2NF)

<u>ISBN</u>	Vorname	Nachname	Titel	<u>ExemplarID</u>	Standort
978-3-518-36777-3	Max	Frisch	Andorra	1	Regal 20-4
978-3-518-36777-3	Max	Frisch	Andorra	2	Regal 20-5
978-3-7891-1851-7	Astrid	Lindgren	Pippi Langstrumpf	1	Regal 13-99

<u>ISBN</u>	Vorname	Nachname	Titel
978-3-518-36777-3	Max	Frisch	Andorra
978-3-7891-1851-7	Astrid	Lindgren	Pippi Langstrumpf

<u>ISBN</u>	<u>ExemplarID</u>	Standort
978-3-518-36777-3	1	Regal 20-4
978-3-518-36777-3	2	Regal 20-5
978-3-7891-1851-7	1	Regal 13-99

- Keine Wiederholung von Vor-, Nachname und Titel notwendig → Redundanzvermeidung

Dritte Normalform (3NF)

2NF + Jedes Nicht-Schlüsselattribut ist von keinem Schlüsselkandidaten transitiv abhängig

- Transitive Abhängigkeit: Ist Y von X funktional abhängig und Z von Y, so ist Z von X funktional abhängig
- Keine transitive Abhängigkeit: Keine funktionale Abhängigkeit der Nicht-Schlüsselattribute untereinander
- Alle Nicht-Schlüsselattribute sind voll funktional abhängig vom Primärschlüssel (bzw. von den Schlüsselkandidaten)

Vorgehen für 3NF:

- Nicht-Schlüsselattribute betrachten:
 - Attribute bestimmen, die von Nicht-Schlüsselattributen abhängig sind
 - Für diese Nicht-Schlüsselattribute weitere Tabellen erstellen mit deren abhängigen Attributen

Dritte Normalform (3NF)

Rechnungsnummer	Datum	Vorname	Nachname	PLZ	Ort
1	29.04.2024	Luc	Dupont	20354	Hamburg

Rechnungen

<u>Rechnungsnummer</u>	Datum	KundenID
1	29.04.2024	1

Kund*innen

<u>KundenID</u>	Vorname	Nachname	PLZ
1	Luc	Dupont	20354

PLZ_Ort

<u>PLZ</u>	Ort
20354	Hamburg

Übung: Normalformen

- Arbeiten Sie gern in 2er-3er Gruppen, wenn Sie möchten
- Bringen Sie die folgende Tabelle in die 1., dann die 2. und dann die 3. Normalform

BestNr	Datum	KdNr	KdName	KdAdresse	Artikel	ArtPreis	Betrag
1001	01.03.2021	K01	Schmidt	Kaiserstr. 5, 53113 Bonn	A10, Block A4, 200 A15, Block A5, 100 A50, Kugelschreiber, 30	0,45 0,30 1,50	90,00 30,00 45,00
1002	01.03.2021	K02	Müller	Händelstr. 67, 53115 Bonn	A60, Radiergummi, 100 A50, Kugelschreiber, 20 A80, Büroklammer, 1000	0,05 1,60 0,01	5,00 32,00 10,00
1003	03.03.2021	K01	Schmidt	Kaiserstr. 5, 53113 Bonn	A10, Block A4, 100 A50, Kugelschreiber, 50	0,50 1,40	50,00 70,00

Die Daten stellen die Bestellungen eines Schreibwarengroßhandels dar. Im Beispiel hat der Kunde Schmidt am 01.03.2021 200 Blöcke A4 zum Einzelpreis von 0,45 €, 100 Blöcke A5 zum Preis von 0,30 € und 30 Lineale zum Preis von 1,50 € bestellt. Die Tabelle enthält weitere Daten wie Kunden- und Bestellnummern, die Adressen der Kunden sowie die Gesamtbeträge der einzelnen Artikel einer Bestellung.

Eine Besonderheit ergibt sich daraus, dass der Einzelpreis für Artikel nicht immer gleich ist – am 03.03. hat der Kunde Schmidt weitere 100 Blöcke A4 bestellt, die jetzt 0,50 € pro Stück kosten – d.h. bei großen Bestellungen sinkt der Preis der Artikel.

Schrittweise Lösung und Erklärung:

<https://cpothmann.de/wp-content/uploads/2021/07/Text-Normalisierung.pdf>

Weitere Normalformen

- Die ersten drei Normalformen sind zunächst ausreichend, weitere müssen Sie sich nicht merken
- Die weiteren Normalformen sind:
 - Boyce-Codd-Normalform (BCNF)
 - Vierte Normalform (4NF)
 - Fünfte Normalform (5NF)

Normalisierung: Vor- und Nachteile

- Vorteile: Stabile Datenstrukturen, Vermeidung von Redundanz und Inkonsistenz/Anomalien
- Nachteile: Performance von Abfragen kann aufgrund der Aufteilung in viele einzelne Tabellen leiden
 - Abfragen werden langsamer, denn bei übergreifenden Abfragen müssen alle relevanten Tabellen wieder mittels JOIN-Befehlen verbunden werden
 - Abfragen sind weniger bequem, wenn z.B. keine ableitbaren/berechneten Werte enthalten sind
 - Berichterstellung ist aufwendiger
 - Ausweg (in begründeten Fällen): teilweise Denormalisierung: die Zahl der Tabellen wird für eine bessere Performance verringert, wobei hingegen Redundanzen in Kauf genommen werden

Konsistenz/Integrität in Datenbanken

Integritätsbedingungen:

- **Entitätsintegrität:** ein Primärschlüssel muss eindeutig sein, er darf nicht leer sein (NOT NULL)
- **Referentielle Integrität:** Ein Fremdschlüssel-Wert darf nur auf einen existierende Datensätze verweisen oder andernfalls leer sein
- **Bereichsintegrität:** Attributwerte müssen im festgelegten Wertebereich liegen
- **Logische Konsistenz:** Es können weitere Integritätsbedingungen definiert werden (z.B., das Todesdatum muss nach dem Geburtsdatum liegen)

Datentypen:

- Durch Definition eines bestimmten Datentyps, dem die Werte eines Attributs entsprechen müssen, kann sichergestellt werden, dass nur Werte des passenden Typs eingegeben werden (z.B. nur Datum oder nur Zeichenkette einer bestimmten Länge)

Logisches Datenbankschema

Nachdem man die Anforderungen für die Datenbank geklärt und ein konzeptuelles Modell der Datenbank erstellt hat (wie in der Übung), kann die Datenbank jetzt logisch modelliert werden:

- Für jeden Entitätstyp wird eine Tabelle angelegt
 - Name der Tabelle: Entitätstyp
 - Spalten der Tabelle: Attribute
 - Evtl. Aggregation/Generalisierung (z.B. 'Gemälde' und 'Skulpturen' zusammenfassen zu Entitätstyp 'Werke')
- **Schlüssel** zur eindeutigen Identifizierung jedes Datensatzes
- **Normalisierung** zur Vermeidung von Redundanz
 - Normalformen
- Integritätsbedingungen
- Datentypen

Übung: Logisches Datenbankschema

- Überführen Sie Ihr konzeptuelles Datenbankschema aus der letzten Sitzung in ein logisches Datenbankschema (erstellen Sie Tabellen, ggf. Notizen)
 - Identifizieren Sie die verschiedenen Entitätstypen und Beziehungstypen und erstellen Sie jeweils eine Tabelle (wenden Sie evtl. Generalisierung an)
 - Bringen Sie die Tabellen in die 1NF
 - Bringen Sie die Tabellen in die 2NF
 - Bringen Sie die Tabellen in die 3NF
 - Markieren Sie dabei bzw. definieren Sie neu die nötigen Primärschlüssel und Fremdschlüssel
- Wenn Sie noch Zeit haben:
 - Formulieren Sie weitere Integritätsbedingungen (z.B. NOT NULL, UNIQUE)
 - Notieren Sie die jeweiligen Datentypen der Attribute (z.B. Text, Zahl, Datum, Boolean)

Zum Nachlesen (optional)

- Klinke, Harald. 2017. “Datenbanken.” S. 109–127 in: Fotis Jannidis, Hubertus Kohle, Malte Rehbein (Hg.): Digital Humanities: Eine Einführung.
<https://link.springer.com/book/10.1007/978-3-476-05446-3>
- Andreas Kelz. 1998. “Relationale Datenbanken. Eine Einführung.” <https://www.hdm-stuttgart.de/~riekert/lehre/db-kelz/>
- Christian Pothmann. 2021. “Normalisierung einer Datenbank.” <https://cpothmann.de/wp-content/uploads/2021/07/Text-Normalisierung.pdf>
- Sowie diverse Lehrbücher, Online-Tutorials oder auch Wikipedia (für ein Grundverständnis)

Ausblick auf die nächste Sitzung

Implementierung:

- Der logische Datenbankentwurf kann schließlich implementiert werden
- Datenbanksprache für relationale Datenbanken → SQL (Structured Query Language)
 - Definition von Datenstrukturen
 - Einfügen, Verändern, Löschen
 - Abfragen
 - Rechteverwaltung
- Basis: relationale Algebra (z.B. Selektion, Kreuzprodukt)
- Relationale Datenbankmanagementsysteme → MySQL
 - *Studienleistung 1*

Zusammenfassung relationale Datenbanken:

- Vor- und Nachteile des relationalen Datenbankmodells
- SQL vs. NoSQL

Hausaufgabe zur nächsten Sitzung

- **Bitte bringen Sie wenn mögliche einen Laptop mit!**
- **Bitte beachten Sie außerdem die E-Mail, die ich Ihnen allen senden werde, was Sie vor der Sitzung installieren sollten!**
- Sie benötigen beides für die erste Studienleistungs-Aufgabe, die Sie in der Sitzung anfangen können.
- Außerdem ist es vorteilhaft, wenn Sie die Übungen zuhause fertig bearbeiten, da die Studienleistung darauf aufbaut
 - “Anforderungserhebung und Konzeption einer relationalen Datenbank”
 - “Logisches Datenbankschema”