

# Digital Skills für Ingenieur\*innen

4. Vorlesungstag

*„Datenbanken“*

## ER-Modell → Relationen

- Nach der Entwicklung des **ER-Modells**, wird das Modell in **datenverarbeitungsnahe Beschreibungen** für Datenbanken umgesetzt.
- Im Falle von **relationalen Datenbanksystemen**, werden die Daten in sogenannten **Relationen** oder **Tabellen** gespeichert.
- Jede Relation besteht aus einer bestimmten **Anzahl von Spalten oder Attributen** sowie einer **Menge von Zeilen oder Tupeln** (auch Datensätze genannt).

## Grundlagen DBS

- Diese Art der Modellierung wird relationales Datenbankmodell genannt und geht auf die Arbeiten von Edgar F. Codd zurück in den 1970ern zurück.
- Grundlage für relationale DBS ist die **relationale Algebra (RA)**
- Die relationale Algebra ist eine **Abfragesprache für relationale Datenbanken**. Sie besteht aus **Relationen, Operatoren und Integritätsbedingungen**, die zu Integritätsarten gehören. **Ein oder zwei Relationen werden durch die Operatoren wieder auf eine Relation abgebildet** (die RA ist abgeschlossen gegenüber ihren Operationen).

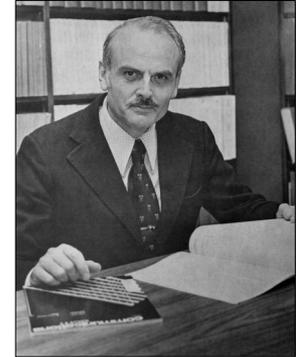


Abbildung 1:  
Edgar F. Codd.jpg

## Beispiele für Operationen aus der RA

- **Auswahl (Selection)**

Beschreibung: Auswahl von Datensätzen, die eine bestimmte Bedingung erfüllen

- **Projektion (Projection)**

Beschreibung: Auswahl bestimmter Attribute aus einer Tabelle

- **Vereinigung (Union)**

Beschreibung: Kombination von Tupeln aus zwei Tabellen (ohne Duplikate)

...

## Begrifflichkeiten einer Tabelle/Relation

<i>Tabellenname/Name der Relation</i>			
<u>Schlüsselattribut <math>A_1</math></u>	Attribut $A_2$	...	Attribut $A_n$
Attributwert $a_{1,1}$	Attributwert $a_{1,2}$	...	Attributwert $a_{1,n}$
Attributwert $a_{2,1}$	Attributwert $a_{2,2}$	...	Attributwert $a_{2,n}$
...	...	...	...
Attributwert $a_{m,1}$	Attributwert $a_{m,2}$	...	Attributwert $a_{m,n}$

## Begrifflichkeiten einer Tabelle/Relation

Relationenschema

<i>Tabellenname/Name der Relation</i>			
<u>Schlüsselattribut <math>A_1</math></u>	Attribut $A_2$	...	Attribut $A_n$
Attributwert $a_{1,1}$	Attributwert $a_{1,2}$	...	Attributwert $a_{1,n}$
Attributwert $a_{2,1}$	Attributwert $a_{2,2}$	...	Attributwert $a_{2,n}$
...	...	...	...
Attributwert $a_{m,1}$	Attributwert $a_{m,2}$	...	Attributwert $a_{m,n}$

## Begrifflichkeiten einer Tabelle/Relation

Datensatz

<i>Tabellenname/Name der Relation</i>			
<u>Schlüsselattribut <math>A_1</math></u>	Attribut $A_2$	...	Attribut $A_n$
Attributwert $a_{1,1}$	Attributwert $a_{1,2}$	...	Attributwert $a_{1,n}$
Attributwert $a_{2,1}$	Attributwert $a_{2,2}$	...	Attributwert $a_{2,n}$
...	...	...	...
Attributwert $a_{m,1}$	Attributwert $a_{m,2}$	...	Attributwert $a_{m,n}$

## Beispiel einer Tabelle/Relation

<i>Mitarbeiter</i>				
<u>ID</u>	Name	Vorname	PLZ	Ort
1002	Müller	Michael	66538	Neunkirchen
1005	Meyer	Sabine	66538	Neunkirchen
6475	Schmidt	Peter	66113	Saarbrücken
1324	Kaiser	Manfred	66113	Saarbrücken

## Normalisierung von Datenbanken

- Unter Normalisierung eines relationalen Datenbankmodells versteht man die **Aufteilung von Attributen in mehrere Relationen** (Tabellen) **mithilfe der Normalisierungsregeln und deren Normalformen**, sodass eine Form entsteht, die **keine vermeidbaren Redundanzen** mehr enthält.
- Ziel der Normalisierung ist eine **redundanzfreie Datenspeicherung**.
- Werden sämtliche Daten **nur ein einziges Mal erfasst und gespeichert**, dann sind sie **redundanzfrei**.
- Um die Forderung Datenkonsistenz und Redundanzfreiheit zu erfüllen, müssen Datenbanken einer **Normalisierung** unterzogen werden.

## Probleme in nicht normalisierten Tabellen

- Zweck der Normalisierung ist **Redundanzen** (mehrfaches Festhalten des gleichen Sachverhalts) **zu verringern**.
- Redundanzen in Daten haben den Nachteil, dass **bei der Aktualisierung von Daten in der Datenbank Anomalien entstehen können** (Änderungen werden nur an einer Stelle getätigt und müssten eigentlich an mehreren Stellen erfolgen).
- **Redundanzen kosten Speicher**.

## Probleme bei „Lernangebotsübersicht“

- Das **Einfügen** eines neuen Kurses „Datenmanagement“ ist nicht möglich, da der Kurs durch eine\*n Schüler\*in belegt werden muss, um ihn zu erfassen.
- Durch das **Löschen** des Schülers Nummer 5 Sebastian Jürgens geht der Kurs Nummer 4 Informatik auch verloren.
- Heirat des Lehrers/der Lehrerin Müller: aus dem Namen Müller wird Müller-Schmidt. Diese **Namensänderung** muss mehrfach in der Tabelle passieren. Ansonsten besteht die Gefahr der Dateninkonsistenz.
- Das **Ändern**, **Einfügen** und **Löschen** von Daten kann zu sogenannten **Anomalien** führen.

## Prozess der Normalisierung – 1. Normalform

- Eine Relation befindet sich in der ersten Normalform, falls alle Attribute **atomare Werte** aufweisen.
  - × Beispiel für nicht-atomares Attribut: Adresse, welche aus PLZ, Ort, Straße und Hausnummer sich zusammensetzt.
- Zur Bildung der ersten Normalform müssen die nicht atomaren Attributwerte durch **Einfügen zusätzlicher Zeilen, Spalten** oder **neuer Relationen** aufgeteilt werden.
- Relation Lernangebotsübersicht kann durch **Hinzufügen von Zeilen** (oder durch Auslagern der nichtatomaren Attribute mit dem vorhandenen Schlüssel) in eine neue Relation in die erste Normalform überführt werden.

## 1. Normalform bei „*Lernangebotsübersicht*“

- Relation Lernangebotsübersicht kann durch **Hinzufügen von Zeilen** (oder durch Auslagern der nichtatomaren Attribute mit dem vorhandenen Schlüssel) in eine neue Relation in die erste Normalform überführt werden.
- Es kann beobachtet werden, dass durch diesen Vorgang die **Redundanz zunimmt**.
- Der bisherige **Primärschlüssel verliert** seine **Eindeutigkeit**.

## Prozess der Normalisierung – 2. Normalform

A	B	C
1	1	3
1	1	3
1	2	4

- Eine Relation befindet sich in der zweiten Normalform, falls
  1. die **Relation in der ersten Normalform** ist und
  2. **jedes Nichtschlüsselattribut vom Primärschlüssel voll funktional abhängig ist.**

Anmerkung zu (voll) funktionaler Abhängigkeit:

$A, B \rightarrow C$ , A und B bestimmen eindeutig C (funktionale Abhängigkeit)

Aus der funktionalen Abhängigkeit  $A, B \rightarrow C$  lässt sich die volle funktionale Abhängigkeit  $B \rightarrow C$  gewinnen.

- Regel zum Prüfen der zweiten Bedingung:  
Wenn Nichtschlüsselattribute **von einem Teil des Schlüssel eindeutig** identifiziert werden, dann liegt **keine 2. Normalform vor!**

## 2. Normalform bei „Lernangebotsübersicht“

- In der Relation Lernangebotsübersicht nehmen wir die **Attributkombination S.-Nr. und L.-Nr. als der neue Primärschlüssel**.
- Aus dem **Teilschlüssel L.-Nr. lässt sich bereits eindeutig** auf das Attribut "Beschreibung" schließen.
- Aus **S.-Nr. lässt sich eindeutig** auf die Attribute "Name", "Vorname", "Klasse" und "Klassenlehrer" schließen.
- Somit sind zwei neue Relationsschemen Lernangebot und Schüler zu erzeugen.

## Prozess der Normalisierung – 3. Normalform

- Eine Relation befindet sich in der dritten Normalform, falls
  1. die **Relation in der zweiten Normalform** ist und
  2. **zwischen den Nichtschlüsselattributen keine Abhängigkeiten existieren, d. h. aus keinem Nichtschlüsselattribut folgt ein anderes Nichtschlüsselattribut** (Fachbegriff: keine transitiven Abhängigkeiten).
- Regel zum Prüfen der zweiten Bedingung:  
Wenn aus einem **Nichtschlüsselattribut ein anderes Nichtschlüsselattribut folgt**, dann liegt **keine 3. Normalform** vor!

## 3. Normalform bei „Lernangebotsübersicht“

- In den Relation Lernangebotsübersicht und Lernangebot gibt es **nur ein Nichtschlüsselattribut**. Daher liegen die Tabellen in der 3. Normalform vor.
- In der Relation Schüler folgt aus dem **Nichtschlüsselattribut "Klasse"** jedoch der **"Klassenlehrer"**.
- Somit ist eine **neue Relation Klassenübersicht** mit den Attributen **"Klasse"** (**Primärschlüssel**) und **"Klassenlehrer"** zu erzeugen.

## Wichtige Erkenntnisse (nach Aufteilung in 3NF)

- Die Tabellen stehen nach Aufteilung in Relationen in einer **Primärschlüssel/Fremdschlüssel-Beziehung**.
- Ein **Primärschlüssel** ist eine Spalte oder eine Spaltengruppe in einer Tabelle, deren Werte eine Zeile in der Tabelle **eindeutig kennzeichnen**.
- Ein **Fremdschlüssel** ist eine Spalte oder eine Spaltengruppe in einer Tabelle, deren Werte den Werten des Primärschlüssels in einer anderen Tabelle entsprechen.
- Mit Hilfe dieser Beziehung wird die **Referentielle Integrität** der Datensätze gewahrt.

## Referentielle Integrität

- Die **Referentielle Integrität** besteht grundsätzlich **aus zwei Teilen**:
  - 1) Ein **neuer Datensatz mit einem Fremdschlüssel kann nur dann in einer Tabelle eingefügt werden**, wenn in der referenzierten Tabelle ein **Datensatz mit entsprechendem Wert im Primärschlüssel existiert**.
  - 2) Eine **Datensatzlöschung oder Änderung des Schlüssels in einem Primär-Datensatz ist nur möglich**, wenn zu diesem Datensatz **keine abhängigen Datensätze in Beziehung stehen**.

## Eigenschaften von Transaktionen in DBMS (ACID)

- Als Transaktion wird in der Informatik eine Folge von Programmschritten bezeichnet, die als eine logische Einheit betrachtet werden, weil sie den Datenbestand nach fehlerfreier und vollständiger Ausführung in einem konsistenten Zustand hinterlassen.
- **Atomicity (Atomarität)**: Transaktion ist eine Folge von Datenbank-Operationen, die **ganz oder gar** nicht ausgeführt wird. Global erst für gültig erklärt und in Kraft gesetzt, wenn **ALLE** Datenbankoperationen erfolgreich und vollständig abgeschlossen werden können.
- **Consistency (Konsistenzerhaltung)**: Nach Beendigung einer Transaktion ist die Datenbank in einem **konsistenten Zustand** (wenn sie es davor auch war). **Überprüfung aller Integritätsbedingungen** vor Abschluss der Transaktion.

## Eigenschaften von DBMS (ACID)

- Isolation (Abgrenzung): Bei einem Datenzugriff einer Transaktion, werden die **notwendigen Daten für andere Transaktionen gesperrt**. Nebenläufige Transaktionen können sich so nicht gegenseitig beeinflussen.
- **Durability** (Dauerhaftigkeit): Daten müssen nach dem erfolgreichen Abschluss einer Transaktion **dauerhaft in der Datenbank gespeichert sein**. Dauerhafte Speicherung der Daten muss auch nach einem Systemfehler garantiert sein (z.B. mittels Transaktionslogs).