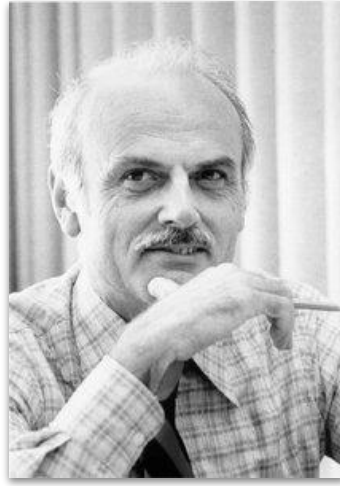


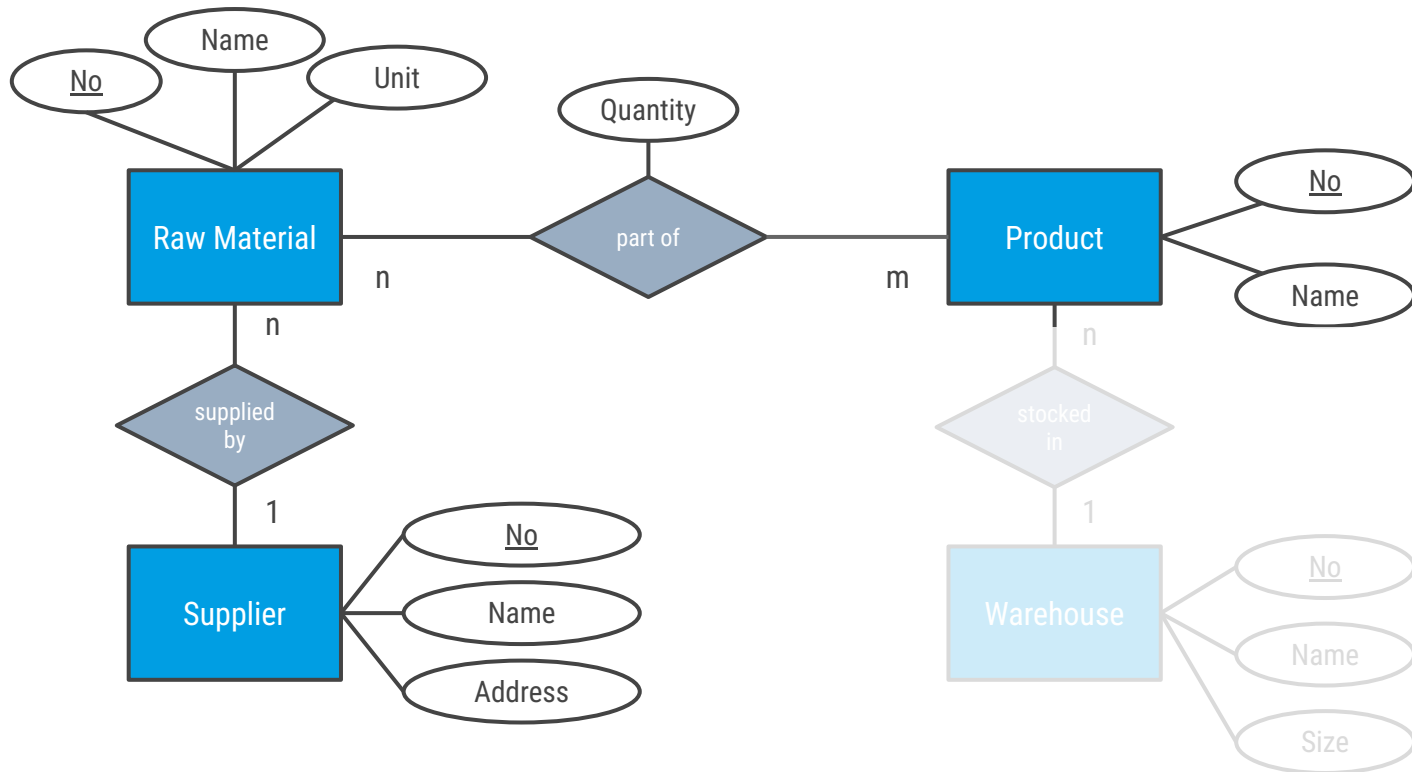


INFORMATION MANAGEMENT

RELATIONAL DATABASES



Codd, E. F., A Relational Model of Data for Large Shared Data Banks, Communications of the ACM, Band 13, Ausgabe 6, Juni 1970, S. 377 – 387, [WEBLINK](#).



Raw Material			
<u>RNo</u>	Name	Unit	SNo
100	Flour	Kilogram	04
201	Olive Oil	Liters	05

Raw Material To Product		
<u>PNo</u>	<u>RNo</u>	Quantity
1	100	150
1	201	0.02

Supplier		
<u>SNo</u>	Name	City
04	Organic Farmer Mayer	Osnabrück
05	Barilla Specialities	Parma

Product	
<u>PNo</u>	Name
1	Pizza Margherita
2	Pizza Funghi

Raw Material			
<u>RNo</u>	Name	Unit	SNo
100	Flour	Kilogram	01
201	Olive Oil	Liters	05

Raw Material To Product		
<u>PNo</u>	<u>RNo</u>	Quantity
1	201	0.02

Supplier		
<u>SNo</u>	Name	City
04	Organic Farmer Mayer	Osnabrück
05	Barilla Specialities	Parma

Product	
<u>PNo</u>	Name
1	Pizza Margherita
2	Pizza Funghi

A Relational Database stores information for one object type in a **relation or table**. Tables can have connections among each other.

Relation or
Table

Primary Key

Foreign Key

Raw Material			
<u>RNo</u>	Name	Unit	SNo
100	Flour	Kilogram	04
201	Olive Oil	Liters	05

Attribute or Column

In the relational model, there are two special types of attributes:



Primary Key

Every table (relation) has exactly one primary key. The value for this key must be unique for all rows in the table. A primary can be a combination of multiple columns.



Foreign Key

Foreign keys connect tables. The value of a foreign key in one table corresponds to the value of the primary key in the referenced table.

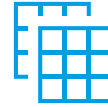
The primary goal of the relational model is consistency



The first relational databases were developed to manage information in operational processes.

Here, it was important that:

- ... the data is correct
- ... transactions are handled safely
- ... data can be changed efficiently



These goals are achieved mainly through **normalization** of the the data model.

3 goals of normalization

Reduce complexity

- ✓ Group information that belongs together
- ✓ Reality-compliant modeling

Eliminate redundancy

- ✓ Store information only once
- ✓ Enable reuse

Efficient data operations

- ✓ Changes affect only one location
- ✓ No dependencies on deleting data

Reality check: data modeling in practice

Wrong understanding

- ⚡ “No redundancy? Disk space doesn’t cost much!”
- ⚡ “The user is responsible for data quality!”
- ⚡ “We validate data only in the frontend on entry.”

Short-term thinking

- ⚡ IT systems are developed under time pressure
- ⚡ “Make it work, make it nice!”
- ⚡ “This is just a temporary solution anyway.”

Competing interests

- ⚡ The user pays, so his wishes (features) come first
- ⚡ The frontend developer also models the data

The first three normal forms (1NF, 2NF, 3NF)



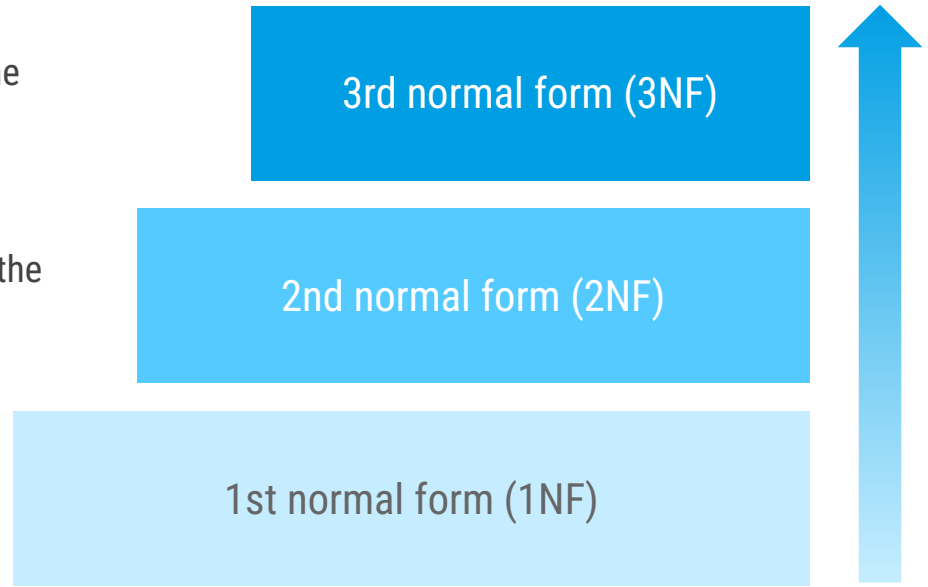
No attribute is transitively dependent on the primary key



All attributes are functional dependent on the whole primary key of the table



Every attribute (column) contains atomic values



Example for bad data modeling

Production Orders									
<u>OrderNo</u>	Date	CNo	FirstName	LastName	<u>PNo</u>	PName	Qty	WNo	WName
101	04.05.2015	42	Wim	Mayer	20	Stool	5	1010	Finished Goods
101	04.05.2015	42	Wim	Mayer	40	Leg	20	1012	Intermediate Goods
101	04.05.2015	42	Wim	Mayer	45	Seat	5	1012	Intermediate Goods
102	05.05.2015	51	Katrin	Scholz	20	Stool	2	1010	Finished Goods
102	05.05.2015	51	Katrin	Scholz	40	Leg	8	1012	Intermediate Goods
102	05.05.2015	51	Katrin	Scholz	45	Seat	2	1012	Intermediate Goods

Example for bad data modeling: 2NF

Orders				
<u>OrderNo</u>	Date	CNo	FirstName	LastName
101	04.05.2015	42	Wim	Mayer
102	05.05.2015	51	Katrin	Scholz

Products			
<u>PNo</u>	PName	WNo	WName
20	Stool	1010	Finished Goods
40	Leg	1010	Intermediate Goods
45	Seat	1012	Intermediate Goods

Order Positions		
<u>OrderNo</u>	<u>PNo</u>	Qty
101	20	5
101	40	20
101	45	5
102	20	2
102	40	8
102	45	2

Example for bad data modeling: 3NF

