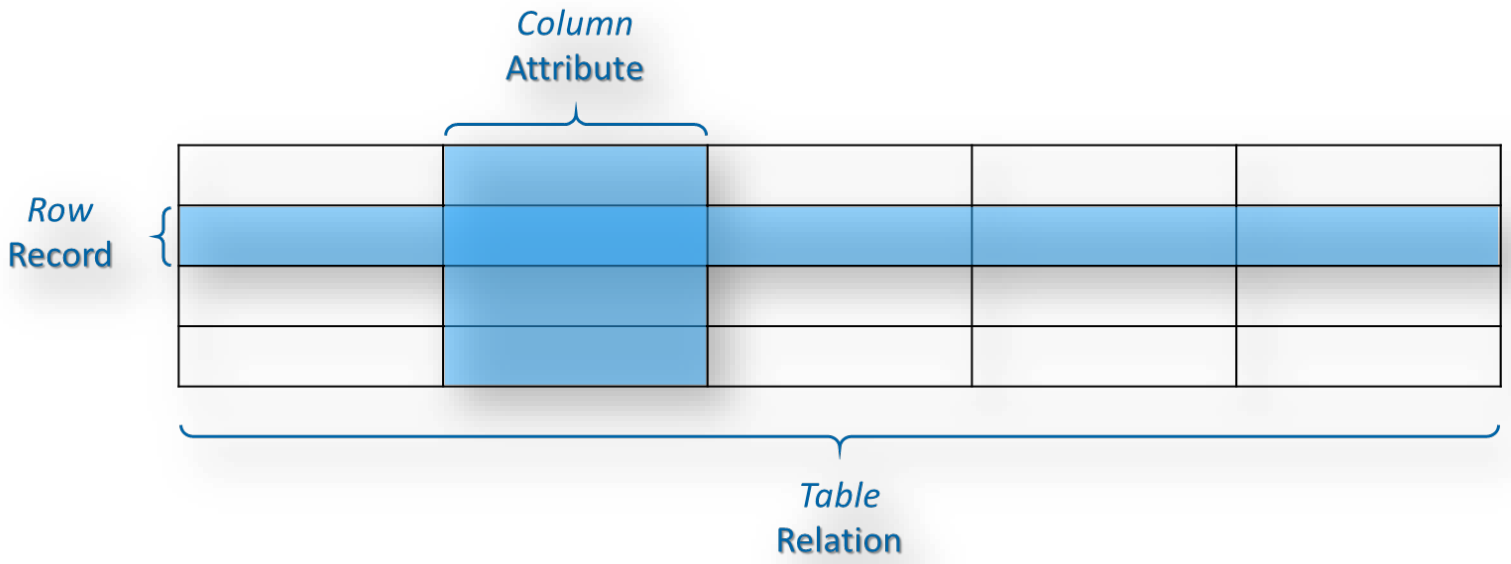
Relational Databases



Relational Databases

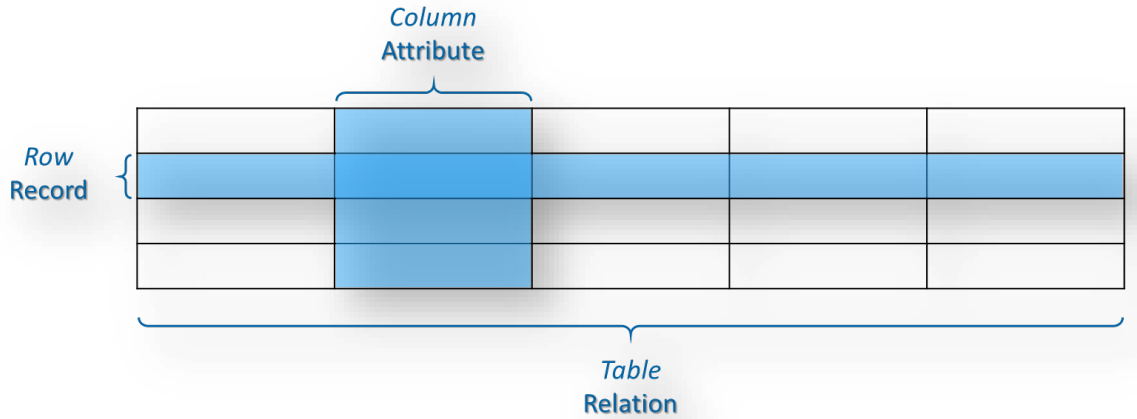
"[Codd's] relational model was at first very controversial; people thought that the model was too simplistic and that it could never give good performance."

Jim Gray

"Database Systems: A Textbook Case of Research Paying Off," in *Computer Science: Reflections on the Field, Reflections from the Field*
2004

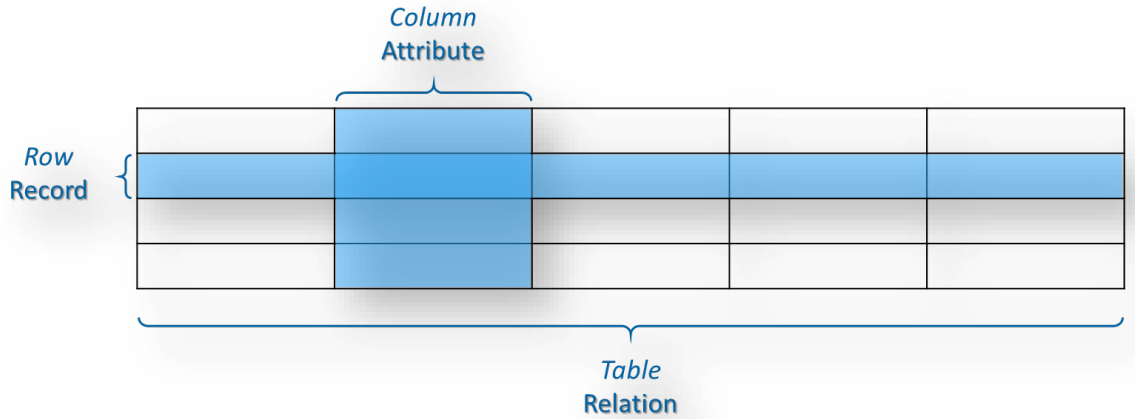
Content

- Relations
- Keys
- Anomalies
- Normalization
- Redundancy
- Referential Integrity



Content

- Relations
- Keys
- Anomalies
- Normalization
- Redundancy
- Referential Integrity



Relations

Why are relational databases called “relational”?

The name comes from the mathematical notion of “relation.” It all started with E. F. Codd who in 1970 (in the article *A Relational Model of Data for Large Shared Data Banks*) proposed something now called **relational algebra** as the mathematical foundation of databases.

In mathematics, a relation is a subset of a Cartesian product. A Cartesian product A × B is the set of all pairs (a,b) where a ∈ A, b ∈ B; a product A × B × C is the set of all triples (a,b,c) where a ∈ A, b ∈ B, c ∈ C; and so on. So, a relation between three sets is a certain set of triples. For example, a relation between the sets {0,1,2}, {0,1,2} and {-2,-1,0,1,2} may be defined like this:

r = { (0,0,0), (0,1,-1), (0,2,-2),
(1,0,1), (1,1,0), (1,2,-1),
(2,0,2), (2,1,1), (2,2,0) }

The relation r above is subtraction in {0,1,2}: (x,y,z) ∈ r if x-y = z.

Relations

Why are relational databases called “relational”?

r = { (0,0,0), (0,1,-1), (0,2,-2),
(1,0,1), (1,1,0), (1,2,-1),
(2,0,2), (2,1,1), (2,2,0) }

subtraction

minuend	subtrahend	difference
0	0	0
0	1	-1
0	2	-2
1	0	1
1	1	0
1	2	-1
2	0	2
2	1	1
2	2	0

Relations

Why are relational databases called “relational”?

A table in a database is a relation – an enumeration of the facts that hold.

A typical example of a table, this one taken from the original Codd’s paper:

supply

supplier	part	project	quantity
1	2	5	17
1	3	5	29
2	3	7	9
2	7	5	4
4	1	1	12

SELECT supplier, part FROM supply
WHERE quantity > 10;

supplier	part
1	2
1	3
4	1

Anything you do with tables in a database yields a relation.
The result of an SQL query is a relation.

Relations

Why are relational databases called “relational”?

A table in a database is a relation – an enumeration of the facts that hold.

A typical example of a table, this one taken from the original Codd’s paper:

supply

supplier	part	project	quantity
1	2	5	17
1	3	5	29
2	3	7	9
2	7	5	4
4	1	1	12

Tables are Relations

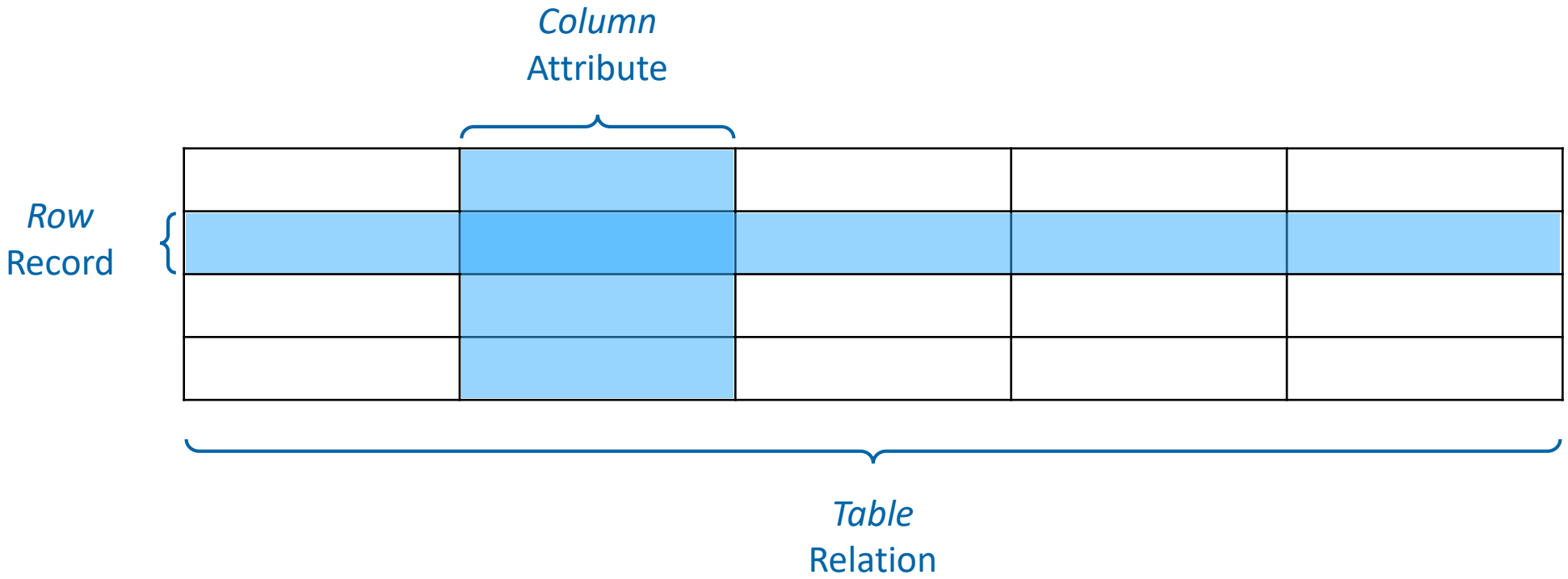
```
SELECT supplier, part FROM supply
WHERE quantity > 10;
```

supplier	part
1	2
1	3
4	1

Anything you do with tables in a database yields a relation.
The result of an SQL query is a relation.

Relations

Tables



Content

- Relations
- Keys
- Anomalies
- Normalization
- Redundancy
- Referential Integrity

supply

ID	supplier	part	project	quantity
101	1	2	5	17
102	1	3	5	29
103	2	3	7	9
104	2	7	5	4
105	4	1	1	12

Keys

How to identify a specific record?

Key

identification number

supply

supplier	part	project	quantity
1	2	5	17
1	3	5	29
2	3	7	9
2	7	5	4
4	1	1	12

supply

ID	supplier	part	project	quantity
101	1	2	5	17
102	1	3	5	29
103	2	3	7	9
104	2	7	5	4
105	4	1	1	12

- must be unique per table
- one key attribute per table is required, several are possible
- database-management systems provide automatic generation

Keys

Primary and Alternative Keys

Out of all *candidate keys*, only one gets selected as primary key (PK), remaining keys are known as alternative or secondary keys.

employee

ID	EmployeeNr	Department	Name	Age
101	MN0345	MN	John Smith	52
102	SE2376	SE	Peter Johnson	29
103	SE8568	SE	Alice Winter	32
104	MN3785	MN	Mary Jones	24
105	MN9448	MN	Peter McAlister	47

primary key *alternative key* *alternative key?*

Keys

Composite Keys

It is possible to create a key by combining attributes.

employee

EmployeeNr	Department	Name	Age	ZIP	City
MN0345	MN	John Smith	52	8008	Zürich
SE2376	SE	Peter Johnson	29	8006	Zürich
SE8568	SE	Alice Winter	32	9000	St.Gallen
MN3785	MN	Mary Jones	24	3002	Bern
MN9448	MN	Peter McAlister	47	3004	Bern

composite key:
(Name + ZIP)

Keys

Relationships

Records in a table can be linked to records in other tables (or within the same table) by adding an attribute for the primary key (PK) of the linked records. Such attributes are called **foreign keys (FK)**.
Such a link is called a **relationship**.

employee

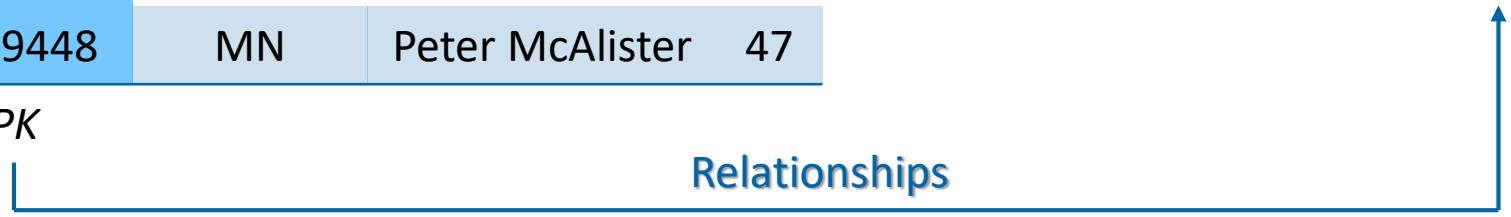
EmployeeNr	Department	Name	Age
MN0345	MN	John Smith	52
SE2376	SE	Peter Johnson	29
SE8568	SE	Alice Winter	32
MN3785	MN	Mary Jones	24
MN9448	MN	Peter McAlister	47

PK

project

ProjectNr	PM	Budget
P870	MN3785	50’000
P348	MN0345	120’000
P101	MN0345	1’000

PK *FK*



Content

- Relations
- Keys
- Anomalies
- Normalization
- Redundancy
- Referential Integrity

ID	EmployeeNr	Department	Name	Age
101	MN0345	MN	John Smith	52
~~102~~	~~SE2376~~	~~SE~~	~~Peter Johnson~~	~~29~~
~~103~~	~~SE8568~~	~~SE~~	~~Alice Winter~~	~~32~~
104	MN3785	MN	Mary Jones	24
105	MN9448	MN	Peter McAlister	47

Anomalies

- **Insert Anomaly**
  Inserting new records might require data that is not available yet (e.g. department). This would result in a NULL entry. Who updates the data later?

- **Update Anomaly**
  If the name of a department changes, a lot of records have to be updated. What if one gets missed?

- **Delete Anomaly**
  If records are deleted, there's a risk that information is deleted for good. What if all employees of the department MN are deleted? The department name is no longer in the database.

ID	EmployeeNr	Department	Name	Age
101	MN0345	MN	John Smith	52
102	SE2376	SE	Peter Johnson	29
103	SE8568	SE	Alice Winter	32
104	MN3785	MN	Mary Jones	24
105	MN9448	MN	Peter McAlister	47

Anomalies

- **Insert Anomaly**
  Inserting new records might require data that is not available yet (e.g. department). This would result in a NULL entry. Who updates the data later?

- **Update Anomaly**
  If the name of a department changes, a lot of records have to be updated. What if one gets missed?

- **Delete Anomaly**
  If records are deleted, there's a risk that information is deleted for good. What if all employees of the department MN are deleted? The department name is no longer in the database.

ID	EmployeeNr	Department	Name	Age
101	MN0345	MN	John Smith	52
102	SE2376	SE	Peter Johnson	29
103	SE8568	SE	Alice Winter	32
104	MN3785	MN	Mary Jones	24
105	MN9448	MN	Peter McAlister	47
106	?	*SE*	?	?

Anomalies

- **Insert Anomaly**
  Inserting new records might require data that is not available yet (e.g. department). This would result in a NULL entry. Who updates the data later?

- **Update Anomaly**
  If the name of a department changes, a lot of records have to be updated. What if one gets missed?

- **Delete Anomaly**
  If records are deleted, there's a risk that information is deleted for good. What if all employees of the department MN are deleted? The department name is no longer in the database.

ID	EmployeeNr	Department	Name	Age
101	MN0345	MN	John Smith	52
102	SE2376	SE	Peter Johnson	29
103	SE8568	SE	Alice Winter	32
104	MN3785	MN	Mary Jones	24
105	MN9448	MN	Peter McAlister	47

Anomalies

- **Insert Anomaly**
  Inserting new records might require data that is not available yet (e.g. department). This would result in a NULL entry. Who updates the data later?

- **Update Anomaly**
  If the name of a department changes, a lot of records have to be updated. What if one gets missed?

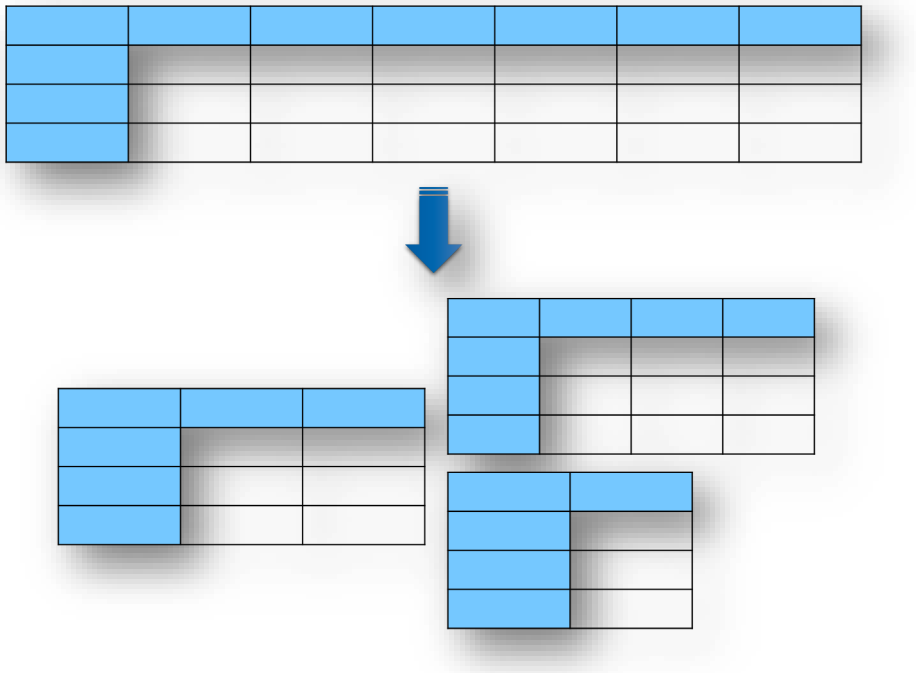
- **Delete Anomaly**
  If records are deleted, there's a risk that information is deleted for good. What if all employees of the department MN are deleted? The department name is no longer in the database.

ID	EmployeeNr	Department	Name	Age
101	MN0345	MN	John Smith	52
~~102~~	~~SE2376~~	~~SE~~	~~Peter Johnson~~	~~29~~
~~103~~	~~SE8568~~	~~SE~~	~~Alice Winter~~	~~32~~
104	MN3785	MN	Mary Jones	24
105	MN9448	MN	Peter McAlister	47

Anomalies

To address anomalies, database are usually normalized and the data is separated into master data and transactional data.

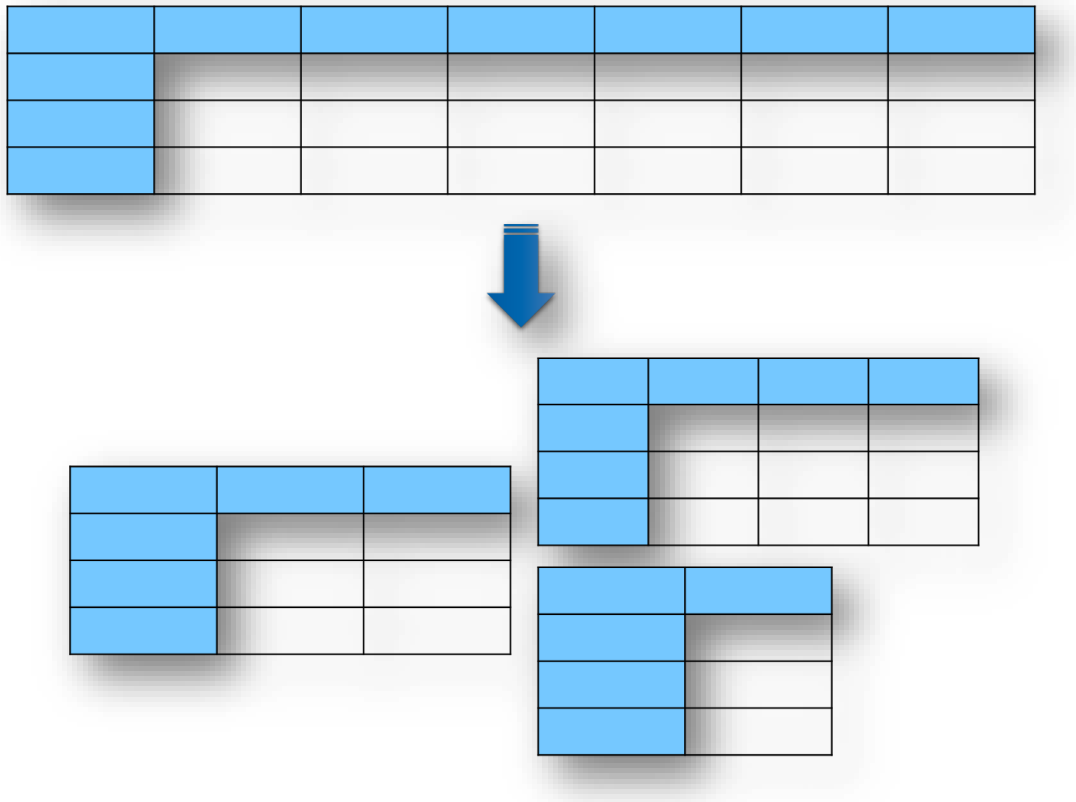
The standard procedure is to break down large tables into a set of smaller tables



In addition, normalization and data separation helps to simplify data queries.

Content

- Relations
- Keys
- Anomalies
- Normalization
- Redundancy
- Referential Integrity



Normalization

Functional Dependency

A **functional dependency (FD)** exists, if the value of an attribute depends on another attribute. Of special interest are functional dependencies on the key attributes.

ID	EmployeeNr	Department	Name	Age
101	MN0345	MN	John Smith	52
102	SE2376	SE	Peter Johnson	29
103	SE8568	SE	Alice Winter	32
104	MN3785	MN	Mary Jones	24
105	MN9448	MN	Peter McAlister	47

functional dependency (FD)

Normalization

Partial Dependency

A **partial dependency (PD)** exists, if an attribute is logically dependent on only one part of a composite key.

composite key: (Name + ZIP)

ID	EmployeeNr	Department	Name	Age	ZIP	City
101	MN0345	MN	John Smith	52	8008	Zürich
102	SE2376	SE	Peter Johnson	29	8006	Zürich
103	SE8568	SE	Alice Winter	32	9000	St.Gallen
104	MN3785	MN	Mary Jones	24	3002	Bern
105	MN9448	MN	Peter McAlister	47	3004	Bern

partial dependency (PD)

Normalization

Transitive Dependency

A **transitive dependency (TD)** exists, if an attribute is logically dependent on another attribute that is not the primary key or part of the primary key. It's dependency on the primary key is through another attribute.

primary key: EmployeeNr

ID	EmployeeNr	Department	Name	Age	ZIP	City
101	MN0345	MN	John Smith	52	8008	Zürich
102	SE2376	SE	Peter Johnson	29	8006	Zürich
103	SE8568	SE	Alice Winter	32	9000	St.Gallen
104	MN3785	MN	Mary Jones	24	3002	Bern
105	MN9448	MN	Peter McAlister	47	3004	Bern

transient dependency (FD)

Normalization

First Normal Form (1NF)

For a table to be in the First Normal Form, it has to satisfy the following four rules:

- Only single (atomic) valued attributes
- Values stored in a column are of the same data type
- All the columns in a table have unique names
- The order in which data is stored does not matter

ID	EmployeeNr	Department	Name	Age	ZIP	City
101	MN0345	MN	John Smith	52	8008	Zürich
102	SE2376	SE	Peter Johnson	29	8006	Zürich
103	SE8568	SE	Alice Winter	32	9000	St.Gallen
104	MN3785	MN	Mary Jones	24	3002	Bern
105	MN9448	MN	Peter McAlister	47	3004	Bern

ID	EmployeeNr	Department	FamilyName	FirstName	Age	ZIP	City
101	MN0345	MN	Smith	John	17	8008	Zürich
102	SE2376	SE	Johnson	Peter	29	8006	Zürich
103	SE8568	SE	Winter	Alice	9	9000	St.Gallen
104	MN3785	MN	Jones	Mary	4	3002	Bern
105	MN9448	MN	McAlister	Peter	12	3004	Bern

Normalization

Second Normal Form (2NF)

For a table to be in the Second Normal Form, it has to satisfy the following two rules:

- Satisfies 1NF
- No partial dependency

ID	EmployeeNr	Department	FamilyName	FirstName	Age	ZIP	City
101	MN0345	MN	Smith	John	17	8008	Zürich
102	SE2376	SE	Johnson	Peter	29	8006	Zürich
103	SE8568	SE	Winter	Alice	9	9000	St.Gallen
104	MN3785	MN	Jones	Mary	4	3002	Bern
105	MN9448	MN	McAlister	Peter	12	3004	Bern

Normalization

Second Normal Form (2NF)

For a table to be in the Second Normal Form, it has to satisfy the following two rules:

- Satisfies 1NF
- No partial dependency → addresses update anomalies

Department	FamilyName	FirstName	Age	ZIP	City
MN	Smith	John	17	8008	Zürich
SE	Johnson	Peter	29	8006	Zürich
SE	Winter	Alice	9	9000	St.Gallen
MN	Jones	Mary	4	3002	Bern
MN	McAlister	Peter	12	3004	Bern

composite key: (Name + ZIP)

Department	FamilyName	FirstName	Age	ZIP
MN	Smith	John	17	8008
SE	Johnson	Peter	29	8006
SE	Winter	Alice	9	9000
MN	Jones	Mary	4	3002
MN	McAlister	Peter	12	3004

composite key: (Name + ZIP)

ZIP	City
8008	Zürich
8006	Zürich
9000	St.Gallen
3002	Bern
3004	Bern

Normalization

Third Normal Form (3NF)

For a table to be in the Second Normal Form, it has to satisfy the following two rules:

- Satisfies 2NF
- No transitive dependency → addresses update anomalies

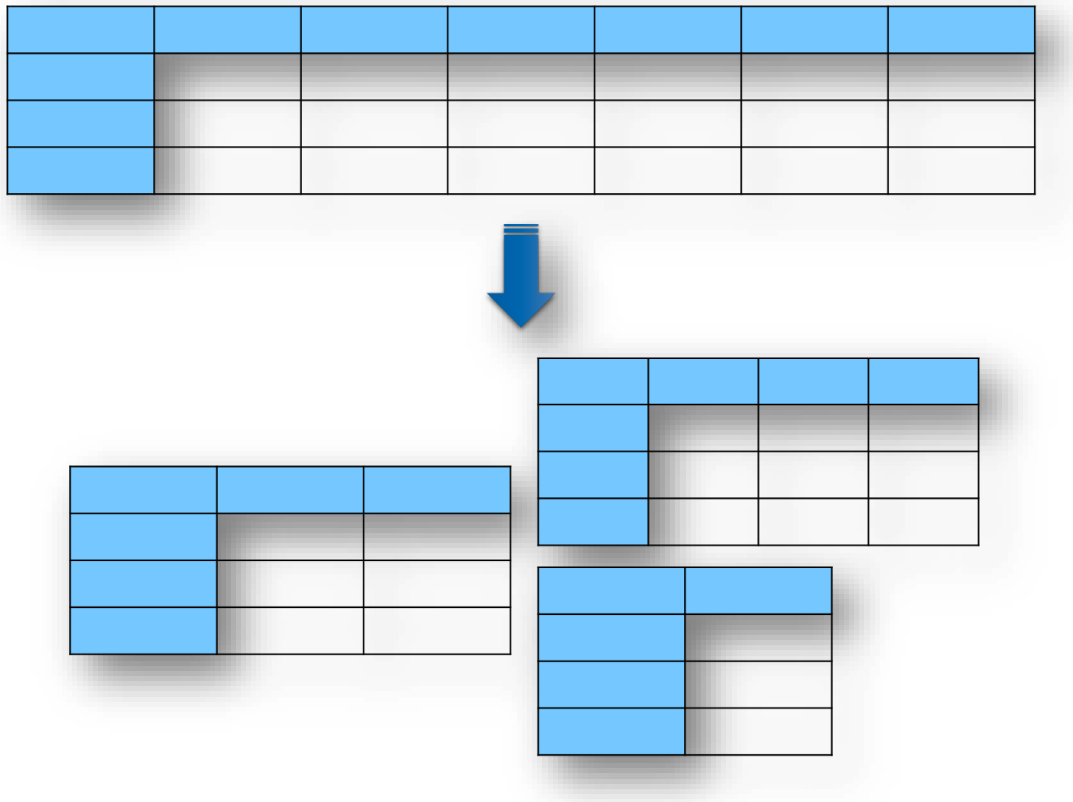
ID	Department	Name	Age	ZIP	City
101	MN	John Smith	52	8008	Zürich
102	SE	Peter Johnson	29	8006	Zürich
103	SE	Alice Winter	32	9000	St.Gallen
104	MN	Mary Jones	24	3002	Bern
105	MN	Peter McAlister	47	3004	Bern

ID	Department	Name	Age	ZIP
101	MN	John Smith	52	8008
102	SE	Peter Johnson	29	8006
103	SE	Alice Winter	32	9000
104	MN	Mary Jones	24	3002
105	MN	Peter McAlister	47	3004

ZIP	City
8008	Zürich
8006	Zürich
9000	St.Gallen
3002	Bern
3004	Bern

Content

- Relations
- Keys
- Anomalies
- Normalization
- **Redundancy**
- **Referential** Integrity



Redundancy

"Reducing the number of times you must enter each item has the side benefit of reducing data entry errors. Each item is stored only once, so even if you do make an error, you must correct only that one entry."

"Relational databases: The Inspiration Behind the Theory," *Tech Republic*
April 2, 2003

Redundancy

Redundancy is the multiple occurrence of the same data in a database.

This has two major drawbacks:

- It requires unnecessarily more physical storage space.
- If the data alters, every occurrence has to be updated (manually?)

To minimize redundancy, relational databases are **normalized**.

In addition, the data is usually separated into **master data** and **transactional data**.

EmployeeNr	Department	Name	Age
MN0345	MN	John Smith	52
SE2376	SE	Peter Johnson	29
SE8568	SE	Alice Winter	32
MN3785	MN	Mary Jones	24
MN9448	MN	Peter McAlister	47

Redundancy

Master Data and Transactional Data

EmployeeNr	Department	Name	Age
MN0345	MN	John Smith	52
SE2376	SE	Peter Johnson	29
SE8568	SE	Alice Winter	32
MN3785	MN	Mary Jones	24
MN9448	MN	Peter McAlister	47

Master Data

Master data is data that is very rarely changes. Master data usually serves as look-up data and is often referred to by other records. Typical master data are categories (profession, religion, sex, etc.) or assets (machines, employees, etc.).

→ addresses update anomalies

Transactional Data

Transactional data is data that is updated on a regular basis or new records are added very frequently. Compared to master data, which can be considered as look-up data, transactional data can be considered as observational data. Furthermore, transactional data usually contain one or more FK of master data. Typical transactional data are accounting data, reservations, orders, etc.

EmployeeNr	Department	Name	Age
MN0345	1	John Smith	52
SE2376	2	Peter Johnson	29
SE8568	2	Alice Winter	32
MN3785	1	Mary Jones	24
MN9448	1	Peter McAlister	47

ID	Department
1	MN
2	SE

Content

- Relations
- Keys
- Anomalies
- Normalization
- Redundancy
- Referential Integrity

Referential Integrity

Referential integrity is an aspect of a relational database to ensure data consistency. To achieve referential integrity, two rules are enforced by the database-management system:

- A record, which is supposed to contain a FK, can only be created if a valid FK is set.
- A record, whose PK is used as a FK in another record, cannot be deleted.

employee

EmployeeNr	Department	Name	Age
MN0345	MN	John Smith	52
SE2376	SE	Peter Johnson	29
SE8568	SE	Alice Winter	32
MN3785	MN	Mary Jones	24
MN9448	MN	Peter McAlister	47

project

ProjectNr	PM	Budget
P870	MN3785	50’000
P348	MN0345	120’000
P101	MN0345	1’000

Referential Integrity

Referential integrity is an aspect of a relational database to ensure data consistency. To achieve referential integrity, two rules are enforced by the database-management system:

- A record, which is supposed to contain a FK, can only be created if a valid FK is set. → addresses insert anomalies
- A record, whose PK is used as a FK in another record, cannot be deleted.

employee

EmployeeNr	Department	Name	Age
MN0345	MN	John Smith	52
SE2376	SE	Peter Johnson	29
SE8568	SE	Alice Winter	32
MN3785	MN	Mary Jones	24
MN9448	MN	Peter McAlister	47

project

ProjectNr	PM	Budget
P870	MN3785	50’000
P348	MN0345	120’000
P101	MN0345	1’000
P205	*	25’000

**FK is required*

Referential Integrity

Referential integrity is an aspect of a relational database to ensure data consistency. To achieve referential integrity, two rules are enforced by the database-management system:

- A record, which is supposed to contain a FK, can only be created if a valid FK is set.
- A record, whose PK is used as a FK in another record, cannot be deleted. → addresses delete anomalies

employee

EmployeeNr	Department	Name	Age
MN0345	MN	John Smith	52
SE2376	SE	Peter Johnson	29
SE8568	SE	Alice Winter	32
MN3785	MN	Mary Jones	24
MN9448	MN	Peter McAlister	47

project

ProjectNr	PM	Budget
P870	MN3785	50’000
P348	MN0345	120’000
P101	MN0345	1’000

PK is used as FK in another table.
Record cannot be deleted.