Data Warehousing
Dimension Changes

Esteban Zimányi

ezimanyi@ulb.ac.be

Slides by Toon Calders

ULB

ECOLE
POLYTECHNIQUE
DE BRUXELLES

What have we seen last time?

- Properties of measures
  - Additive along a dimension
  - Aggregable
- Properties of aggregation operators
  - Distributive
  - Algebraic
  - Holistic

What have we seen last time?

Continent	Country	City	Amount
Europe	Belgium	Brussels	5
		Antwerp	3
	Germany	Berlin	2
North-America	USA	Chicago	1
		Tampa	8

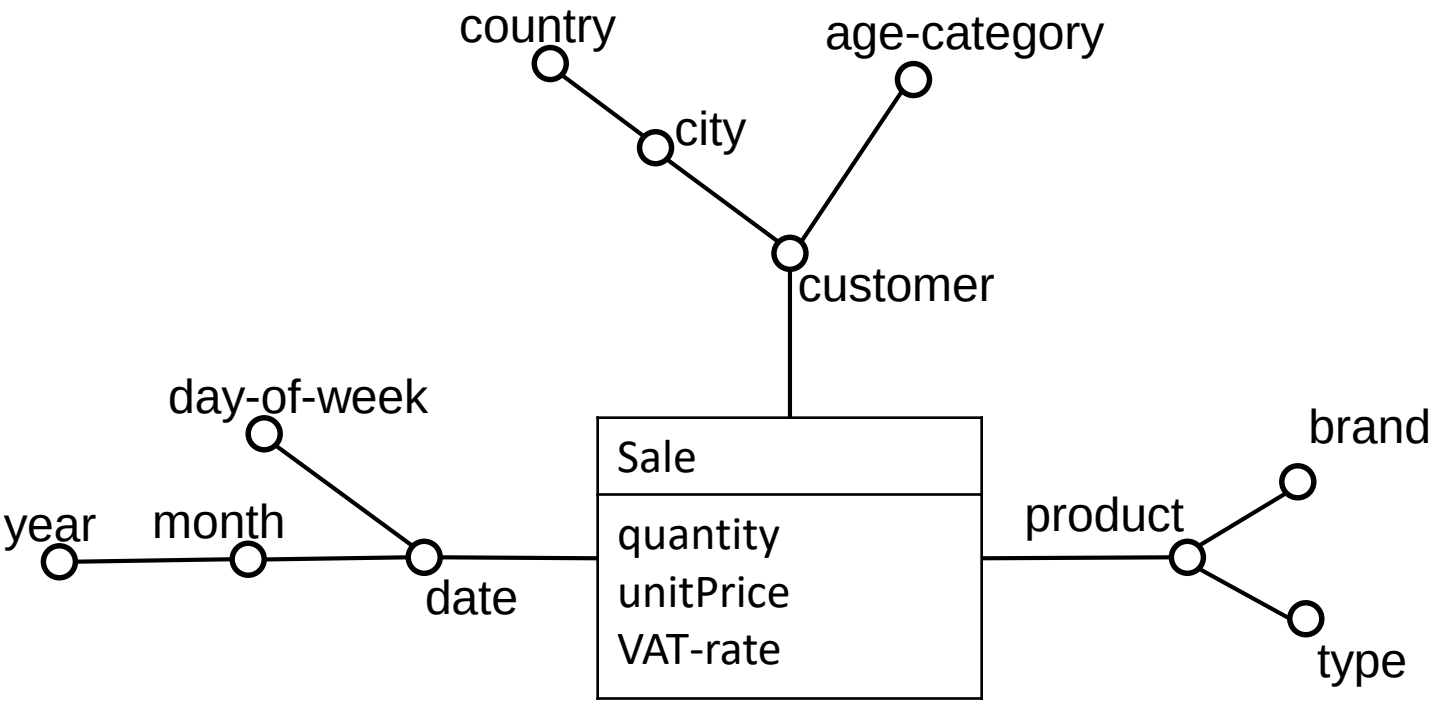
Properties of measures: Distributivity

Continent	Country	SUM(Amount)
Europe	Belgium	8
	Germany	2
North-America	USA	9

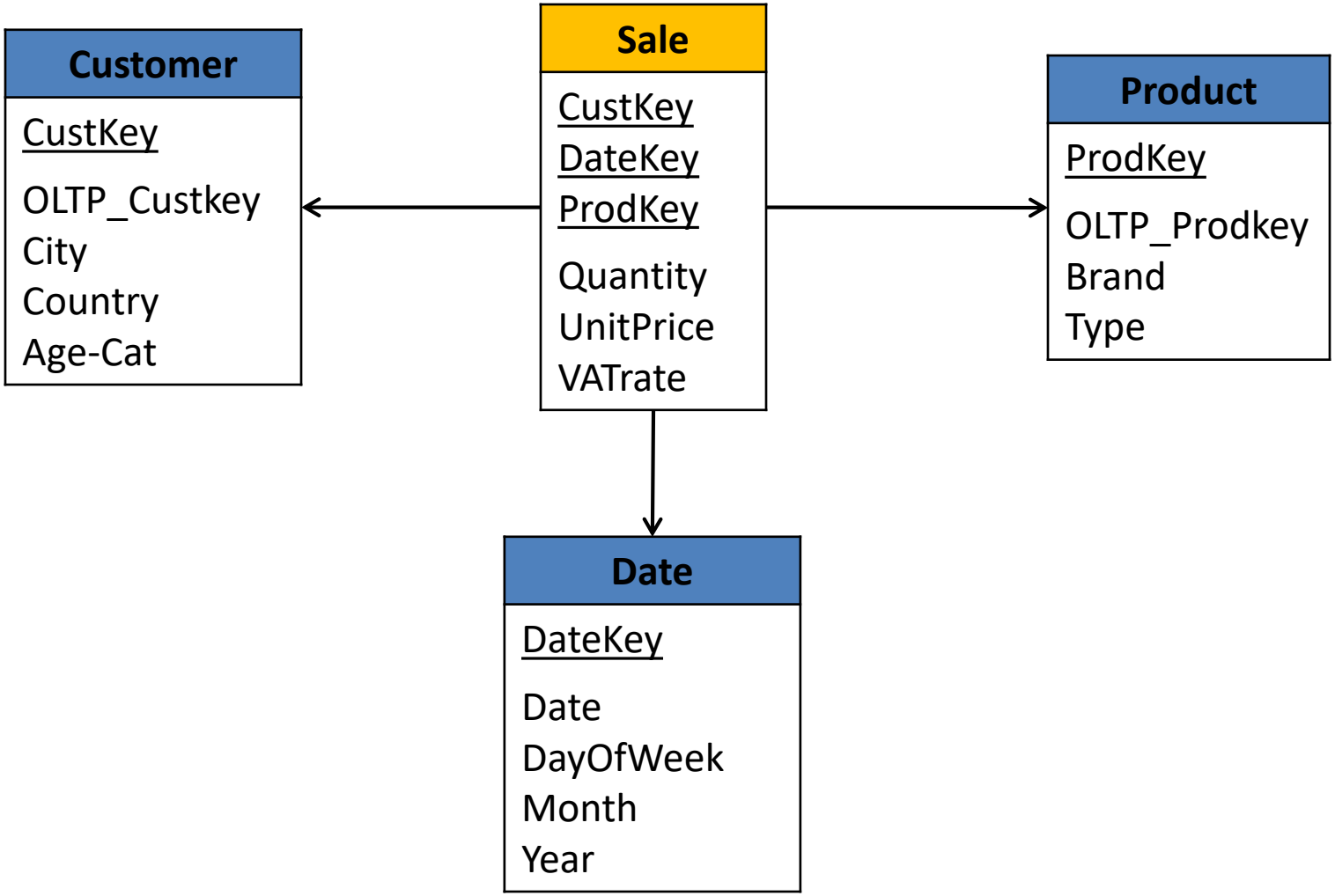
Continent	Sum(Amount)
Europe	10
North-America	9

Sum(Amount)
19

What have we seen last time?



What have we seen last time?



What have we seen last time?

Star schema

- Dimension tables are not normalized
  - Use surrogate key
- Dimensions such as Date are materialized
- Key for the fact table consists of the foreign keys to the dimension tables

Snowflake schema

- has (partially) normalized dimensions

What have we seen last time?

- Ways to deal with the different conceptual modeling constructions
  - Non-standard hierarchies
  - Multiple arcs
  - Cross-dimensional attributes

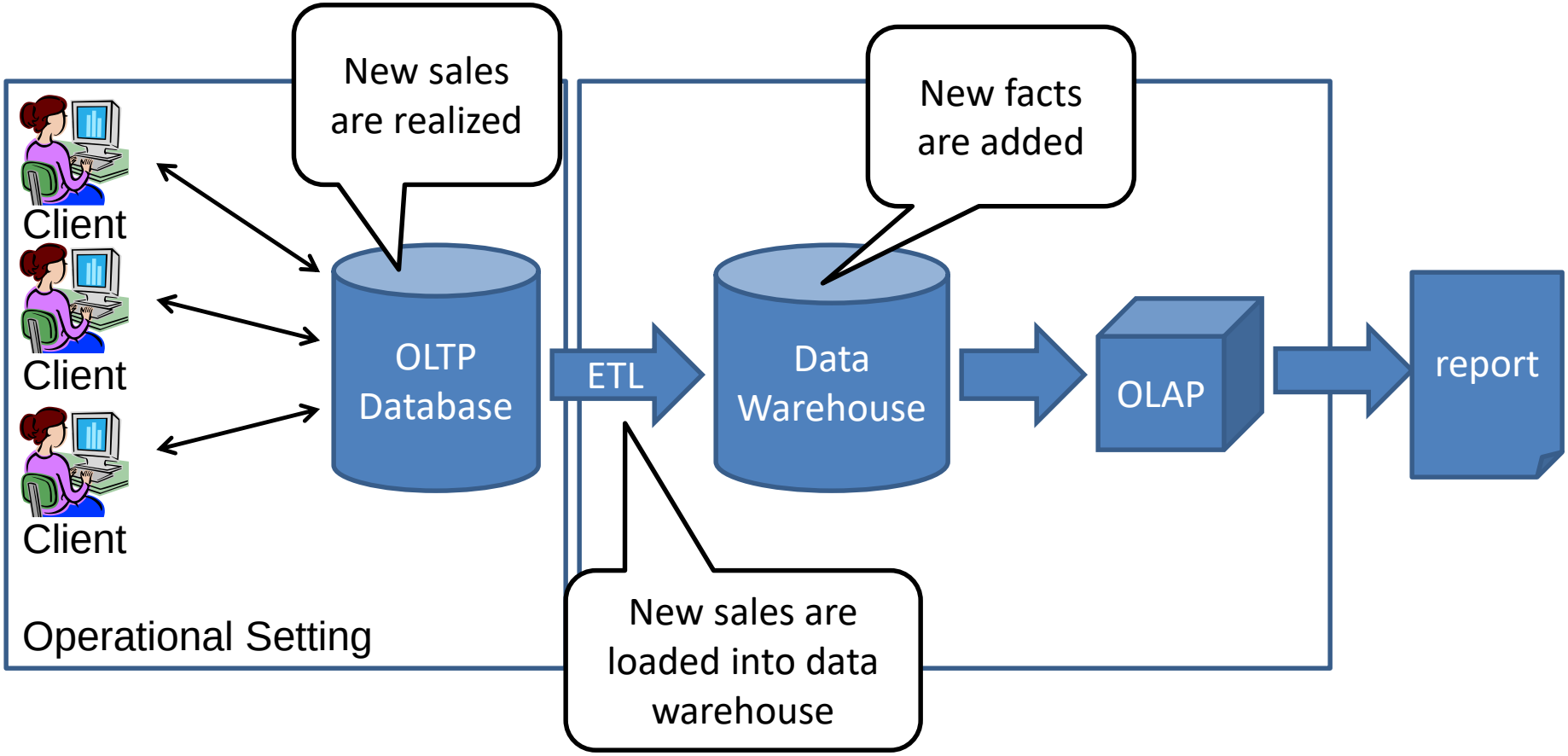
Outline

- Dealing with changing dimensions
  - Slowly Changing Dimensions
    - Type 1, 2, and 3
  - Rapidly changing dimensions
    - Type 4: Mini dimension
- Specific dimension types
  - Junk dimension
  - Outriggers
  - Degenerate dimension
  - Time and Data Quality dimensions

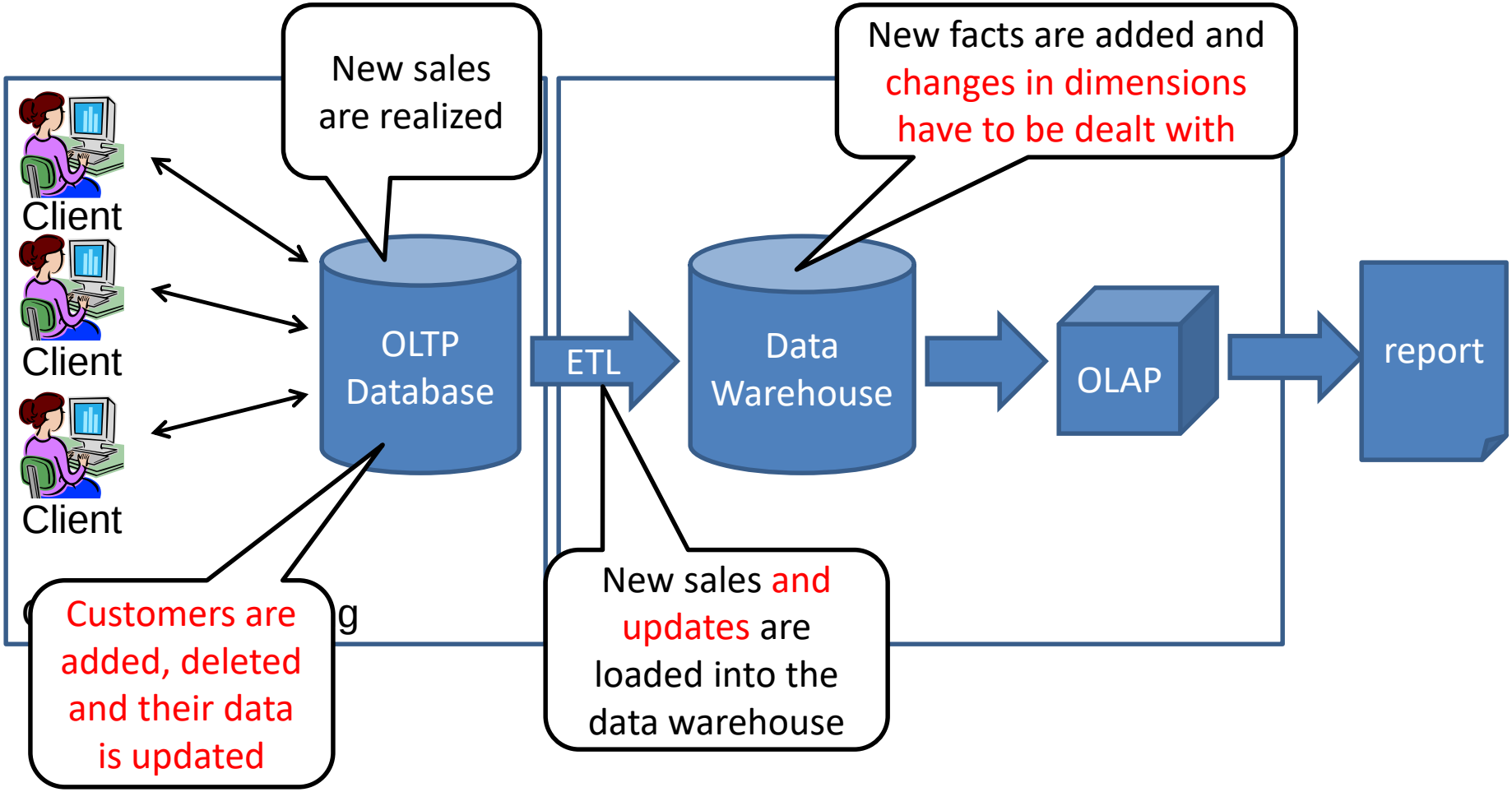
Changes in Dimensions

- Dimensions are not stable over time
  - New rows can be inserted
  - Existing rows can be updated
- We will see techniques for handling changing dimensions
  - Slowly changing dimensions
  - Rapidly changing dimensions

Idealized Picture



More Realistic Picture



What is the problem?

2000

Customer

CID	Name	Address
001	John	Dallas
002	Mary	Dallas
003	Pete	New York

Sales

CID	Product	Price
001	Gun	5$
002	Beef	20$
003	Lava lamp	150$

2010

Customer

CID	Name	Address
001	John	New York
002	Mary	New York
004	Mark	Dallas

Sales

CID	Product	Price
001	Gun	15$
004	Pork	15$
004	Lava lamp	160$

What is the problem?

Customer

CID	Name	Address
001	John	Dallas / New York
002	Mary	Dallas / New York
003	Pete	New York
004	Mark	Dallas

Sales

CID	PID	Date	Amount
001	G	2000	5$
001	G	2010	15$
002	B	2000	20$
003	L	2000	150$
004	P	2010	15$
004	L	2010	160$

Product

Date

Outline

- Dealing with changing dimensions
  - Slowly Changing Dimensions
    - Type 1, 2, and 3
  - Rapidly changing dimensions
    - Mini dimension
- Specific dimension types
  - Junk dimension
  - Outriggers
  - Degenerate dimension
  - Time and Data Quality dimensions

Different Types of Handling

"Standardized" types of handling changes

- Type 1: No special handling
- Type 2: Versioning dimension values
  - 2A. Special facts
  - 2B. Time stamping
- Type 3: Capturing the previous and the current value

Type 1 - Updating

- Type 1 is updating the value
  - Suitable in case of mistakes
  - For dimensions with static attributes; e.g., last name; date of birth

CID	Name	Address
001	John	NY
002	Mary	New York
003	Pete	NY

CID	Name	Address
001	John	New York
002	Mary	New York
003	Pete	New York
004	Mark	Dallas

Correct inconsistency in the city names of John and Pete
Mark is a new client

Type 2 - Versioning

- Whenever there is a change, create a new version of the affected row
  - Need for surrogate key!

SID	CID	Name	Address
1	001	John	Dallas
2	002	Mary	Dallas
3	003	Pete	New York

SID	CID	Name	Address
1	001	John	Dallas
2	002	Mary	Dallas
3	003	Pete	New York
4	001	John	New York
5	002	Mary	New York
6	004	Mark	Dallas

John and Mary move to New York
Mark is a new client

Type 2 – Valid Time

- It may be useful to know at what point a certain version was valid

- Different ways to store the valid time of a version
  - 2A: use time dimension and special facts
  - 2B: time stamping of rows

Type 2A

- Use special facts and time dimension of facts

Customer

SID	CID	Name	Address
1	001	John	Dallas
2	002	Mary	Dallas
3	003	Pete	New York
4	001	John	New York
5	002	Mary	New York
6	004	Mark	Dallas

Sales

SID	PID	Date	Amount
1	G	D1	5$
2	G	D1	15$
2	B	D2	20$
3	L	D3	150$
4	P	D4	15$
6	L	D4	160$
5	-	D5	-

Special fact to store date of change

Type 2B (More popular one)

- Keep valid time as explicit attributes

Customer

SID	CID	Name	Address	Start	End	Valid
1	001	John	Dallas	D1	D2	
2	002	Mary	Dallas	D1	D3	
3	003	Pete	New York	D2	-	**X**
4	001	John	New York	D2	-	**X**
5	002	Mary	New York	D3	-	**X**
6	004	Mark	Dallas	D5	-	**X**

Type 3 – Limited Versions

- Foresee limited number of changes
  - Add attribute for every change
- Advantage:
  - If the change itself is useful information
    - How does calling volume change if people move?
  - No need to chain primary events
- Various disadvantages:
  - When did the change happen?
  - What if there are more changes?

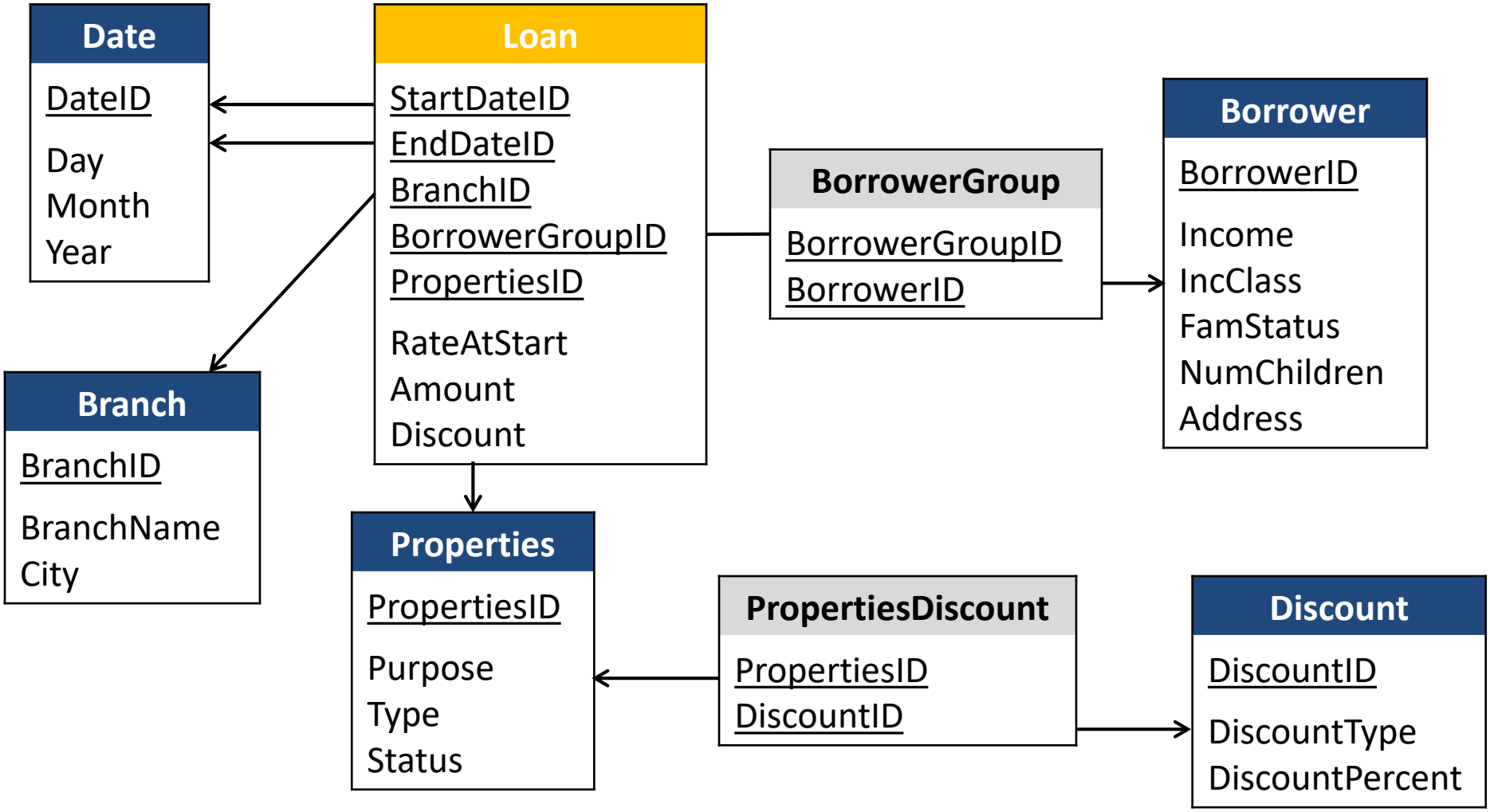
Type 3 – Limited Versions

Customer

SID	CID	Name	OldAddress	NewAddress
1	001	John	Dallas	New York
2	002	Mary	Dallas	New York
3	003	Pete	New York	New York
6	004	Mark	Dallas	Dallas

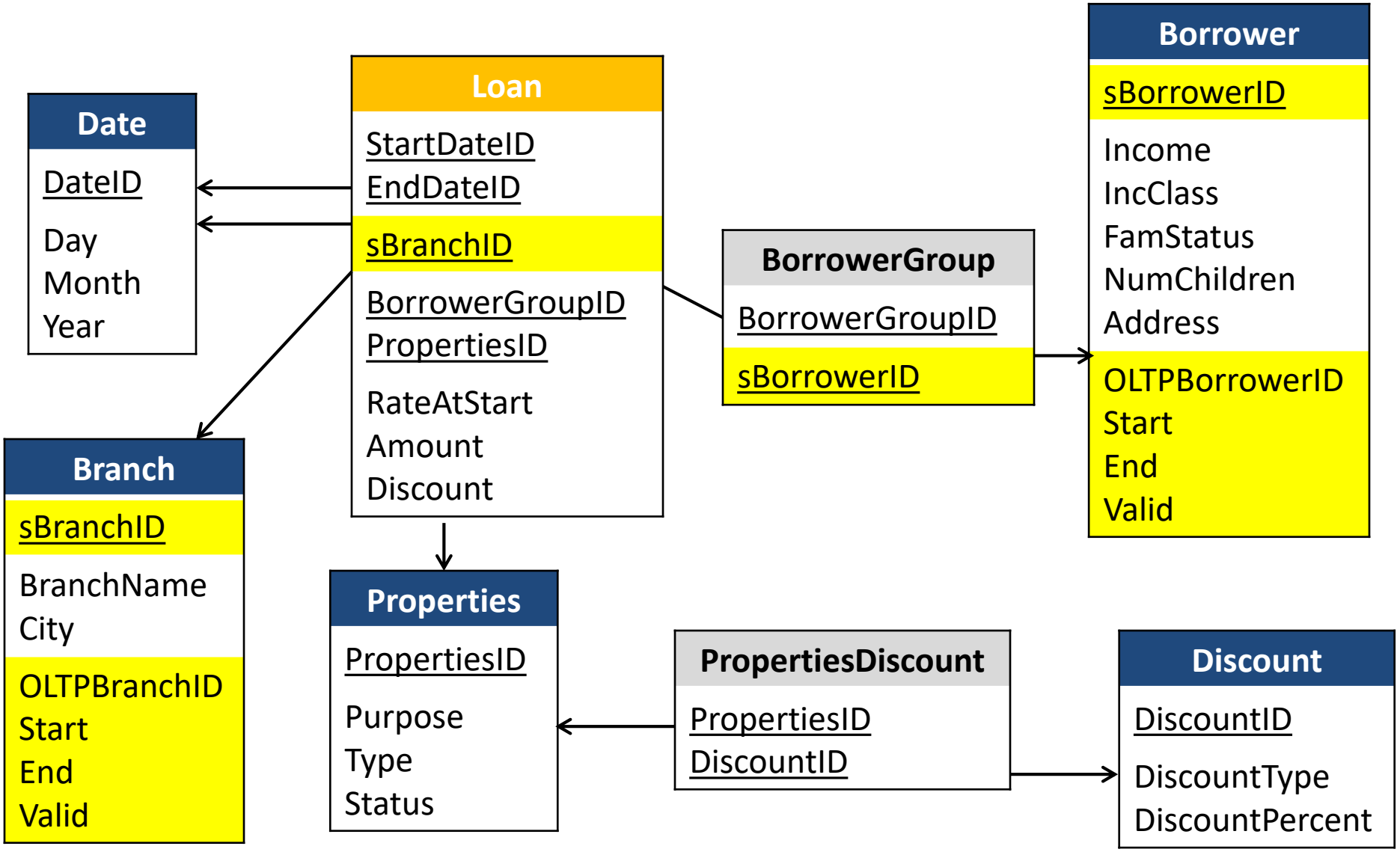
ONLY TYPE 1 CHANGES

Example: SCD



Example: SCD

TYPE 2 CHANGES for brand and borrower



Example: SCD

Borrower

sBID	OLTP_BID	Address	…	Valid
S1	001	Brussels	…	X
S2	002	Brussels	…	X

Borrower_Group

BGroup	sBID
G1	S1
G1	S2

Loan

BGroup	Date	…	Amount
G1	D1	…	1000

Example: SCD

002 moves

Borrower

sBID	OLTP_BID	Address	…	Valid
S1	001	Brussels	…	X
S2	002	Brussels	…	
S3	002	Antwerp	…	X

001 and 002 get a new loan

Borrower_Group

BGroup	sBID
G1	S1
G1	S2
G2	S1
G2	S3

Loan

BGroup	Date	…	Amount
G1	D1	…	1000
G2	D2	…	7000

SCD and Bridge Table

dimCustomer

sCID	Key	VIP	Adress
SC1	C1	Y	Brussels
SC2	C2	N	Antwerp
SC3	C3	N	Brussels

bridgeAccountCustomer

sAID	sCID
SA1	SC1
SA1	SC2
SA2	SC3

dimAccount

sAID	OLTPkey	Type	Premium
SA1	001	Savings	Y
SA2	002	Debit	N

sAID	…	Amount
SA1	…	1000
SA2	…	2000

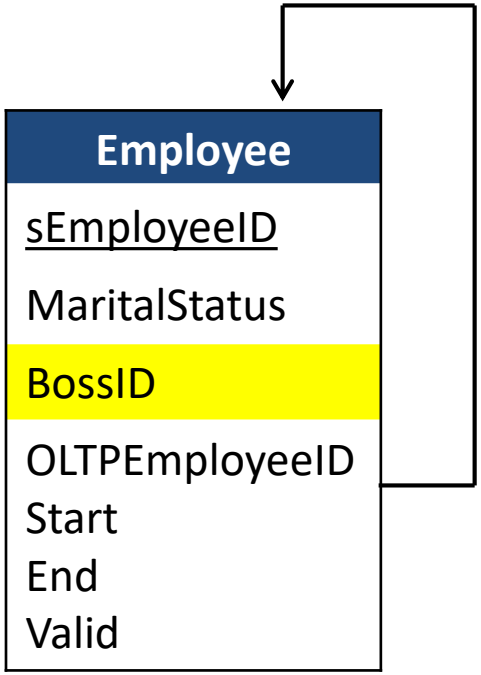
- Changes possible to:
  - Account
  - Customer
  - Relation between Account and Customer

SCD and Bridge Table

- Solution: add valid start and end time to bridge table; Customer and Account versioned
- New tuples are added to the bridge table whenever:
  - Customer changes
  - Account changes
  - Relation between accounts and customers changes

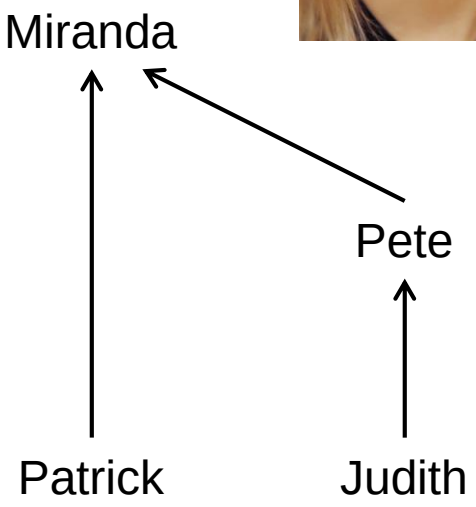
Type-2 and Recursive Hierarchy

- Be very careful when to apply type-2 changes; consider the following Employee dimension with an unbalanced hierarchy expressing who is whose boss.
  - What happens if the CEO of the company gets married?



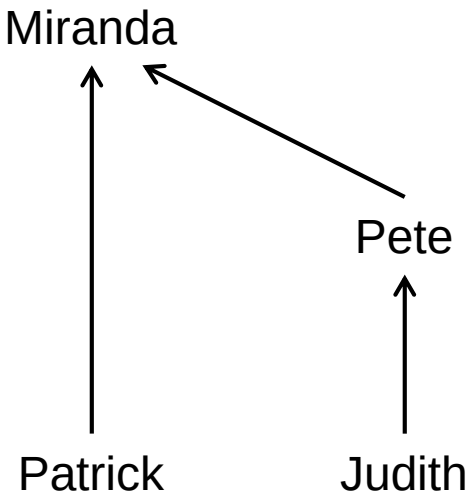
Type-2 and Recursive Hierarchy

sID	OLTPkey	BossID	MStat	Start	End	Val
1	miranda	none	single	D1	-	X
2	patrick	1	married	D1	-	X
3	pete	1	single	D1	-	X
4	judith	3	married	D1	-	X



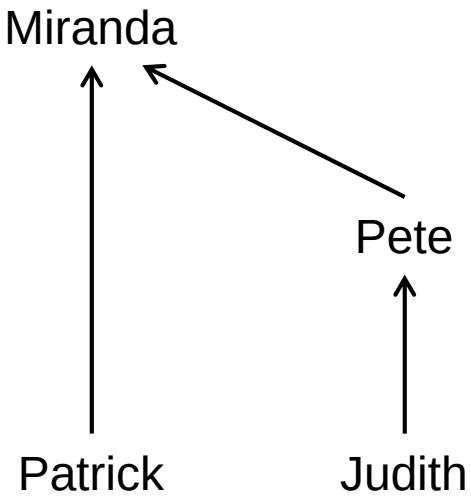
Type-2 and Recursive Hierarchy

sID	OLTPkey	BossID	MStat	Start	End	Val
1	miranda	none	single	D1	-	X
2	patrick	1	married	D1	-	X
3	pete	1	single	D1	-	X
4	judith	3	married	D1	-	X



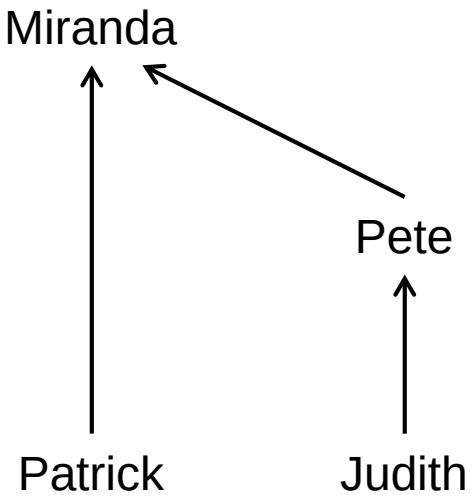
Type-2 and Recursive Hierarchy

sID	OLTPkey	BossID	MStat	Start	End	Val
1	miranda	none	single	D1	D2	
5	miranda	none	married	D2	-	X
2	patrick	1	married	D1	-	X
3	pete	1	single	D1	-	X
4	judith	3	married	D1	-	X



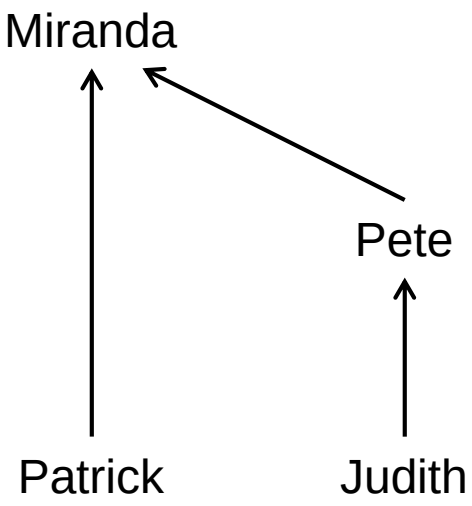
Type-2 and Recursive Hierarchy

sID	OLTPkey	BossID	MStat	Start	End	Val
1	miranda	none	single	D1	D2	
5	miranda	none	married	D2	-	X
2	patrick	1	married	D1	D2	
6	patrick	5	married	D2	-	X
3	pete	1	single	D1	D2	
7	pete	5	single	D2	-	X
4	judith	3	married	D1	-	X



Type-2 and Recursive Hierarchy

sID	OLTPkey	BossID	MStat	Start	End	Val
1	miranda	none	single	D1	D2	
5	miranda	none	married	D2	-	X
2	patrick	1	married	D1	D2	
6	patrick	5	married	D2	-	X
3	pete	1	single	D1	D2	
7	pete	5	single	D2	-	X
4	judith	3	married	D1	D2	
8	judith	7	married	D2	-	X



Outline

- Dealing with changing dimensions
  - Slowly Changing Dimensions
    - Type 1, 2, and 3
  - Rapidly changing dimensions
    - Type 4: Mini dimension
- Specific dimension types
  - Junk dimension
  - Outriggers
  - Degenerate dimension
  - Time and Data Quality dimensions

Rapidly Changing Dimensions

- Some attributes may change frequently
  - Many dimensional attributes in a changing dimension of Type 2 results in many repeated values when there is a change
  - Some attributes never change, but are duplicated nevertheless

Customer
<u>SID</u>
CID
Fname Lname Dob Gender MaritalStatus NumChildren
CreditScoreGroup BuyingStatusGroup IncomeGroup EducationGroup

Example: Changes

Customer
<u>SID</u>
CID
Fname
Lname
Dob
Gender
Address
City
Country
MaritalStatus
NumChildren
CreditScoreGroup
BuyingStatusGroup
IncomeGroup
EducationGroup

D1

Customer
<u>SID</u>
CID
Fname
Lname
Dob
Gender
Address
City
Country
MaritalStatus
NumChildren
CreditScoreGroup
BuyingStatusGroup
IncomeGroup
EducationGroup

D2

Customer
<u>SID</u>
CID
Fname
Lname
Dob
Gender
Address
City
Country
MaritalStatus
NumChildren
CreditScoreGroup
BuyingStatusGroup
IncomeGroup
EducationGroup

D3

Customer
<u>SID</u>
CID
Fname
Lname
Dob
Gender
Address
City
Country
MaritalStatus
NumChildren
CreditScoreGroup
BuyingStatusGroup
IncomeGroup
EducationGroup

D4

Solution Type 4: Split Dimension

Customer

SID

CID

Fname
Lname
Dob
Gender
MaritalStatus
NumChildren

CreditScoreGroup
BuyingStatusGroup
IncomeGroup
EducationGroup

Customer

SID

CID

Fname
Lname
Dob
Gender

Demographics

DemID

MaritalStatus
NumChildren

CreditScoreGroup
BuyingStatusGroup
IncomeGroup
EducationGroup

Add as a FK in the fact table; not in customer dimension!

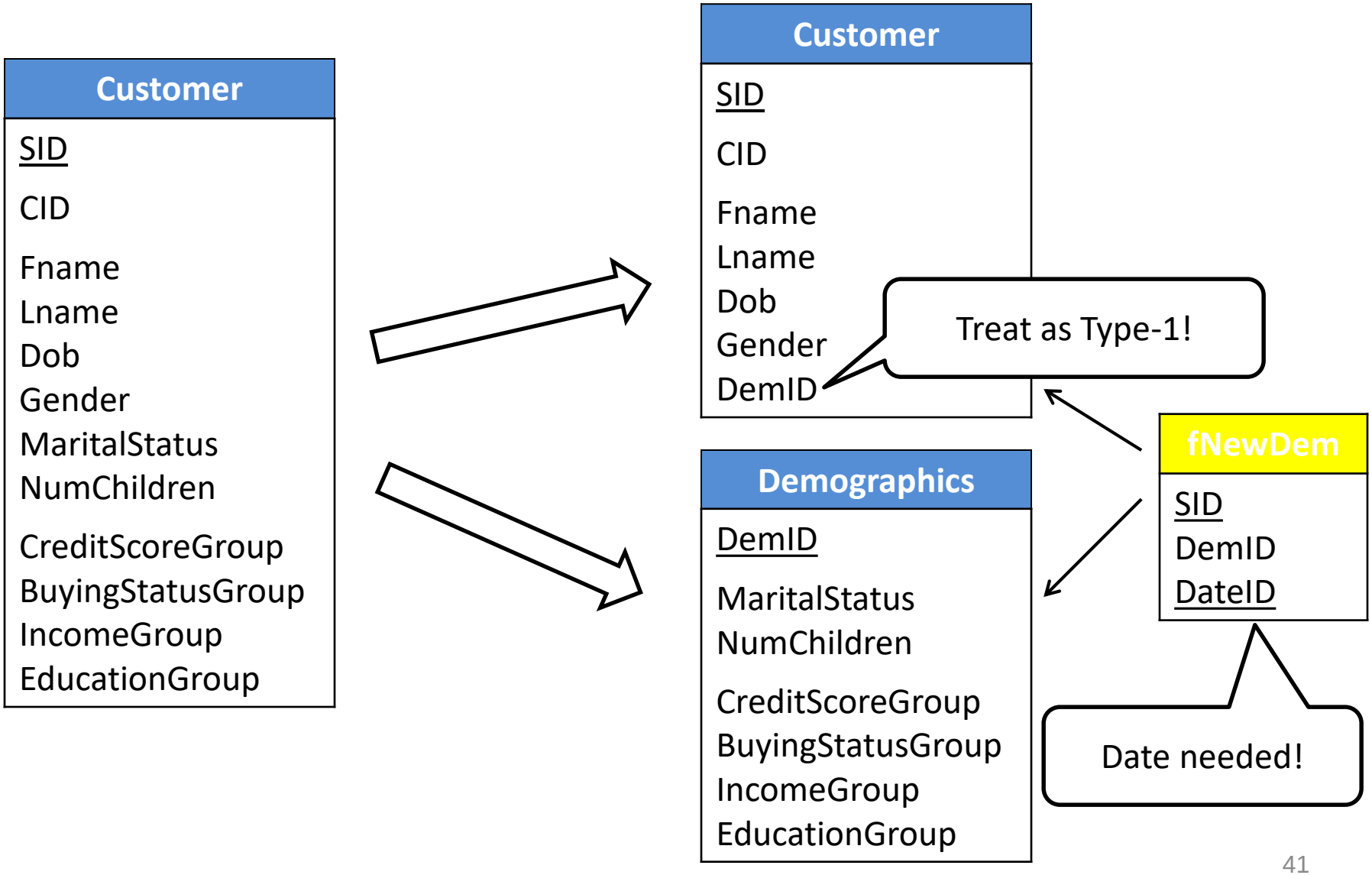
Mini-Dimension

- Suppose frequently changing attributes have small domains
  - We could force this situation by discretizing some attributes with many values
- We can fully populate the dimension with all possible combinations of values
  - No type-II changes to the mini-dimension!
  - Demography is not updated; a fact is about a customer with a new demography

Mini-Dimension

- What if we need to keep the changes in the dimensions?
  - Make an empty fact. Caveat: what with COUNT?
  - Can be considered a new kind of fact that needs to be stored → add a separate measureless fact table. Include additional identifying attribute, f.i. date/time as one of the dimensions (why?)
- Often surrogate key of the most recent mini-dimension value is added to dimension table

Solution Type 4: Split Dimension



Example: Mini-Dimension

Model a dimension Customer that stores the following information:

- **passport number**
- **first name, middle name, last name**
- **date of birth and gender**
- **street, number, postal code, city, country**

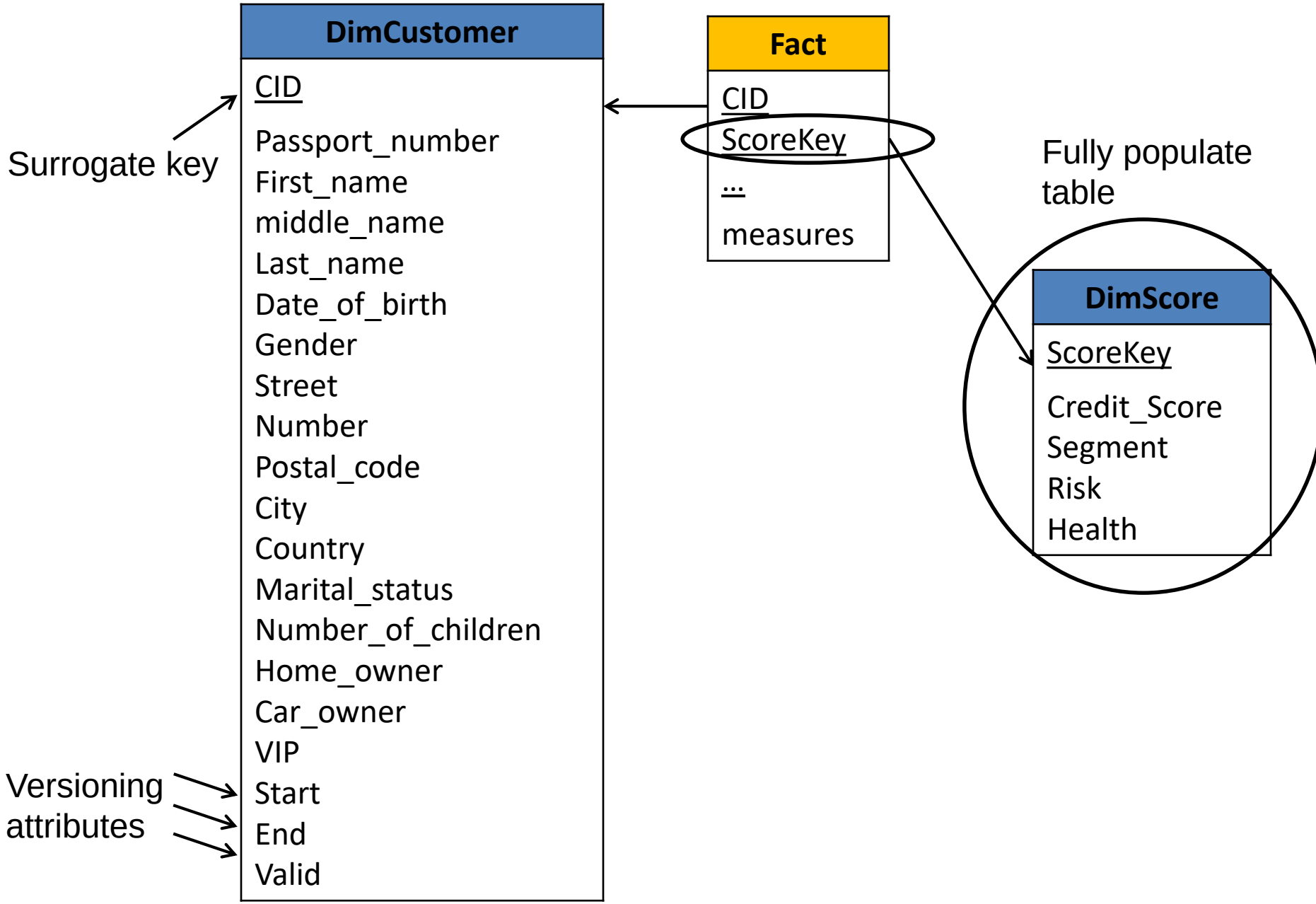
No changes

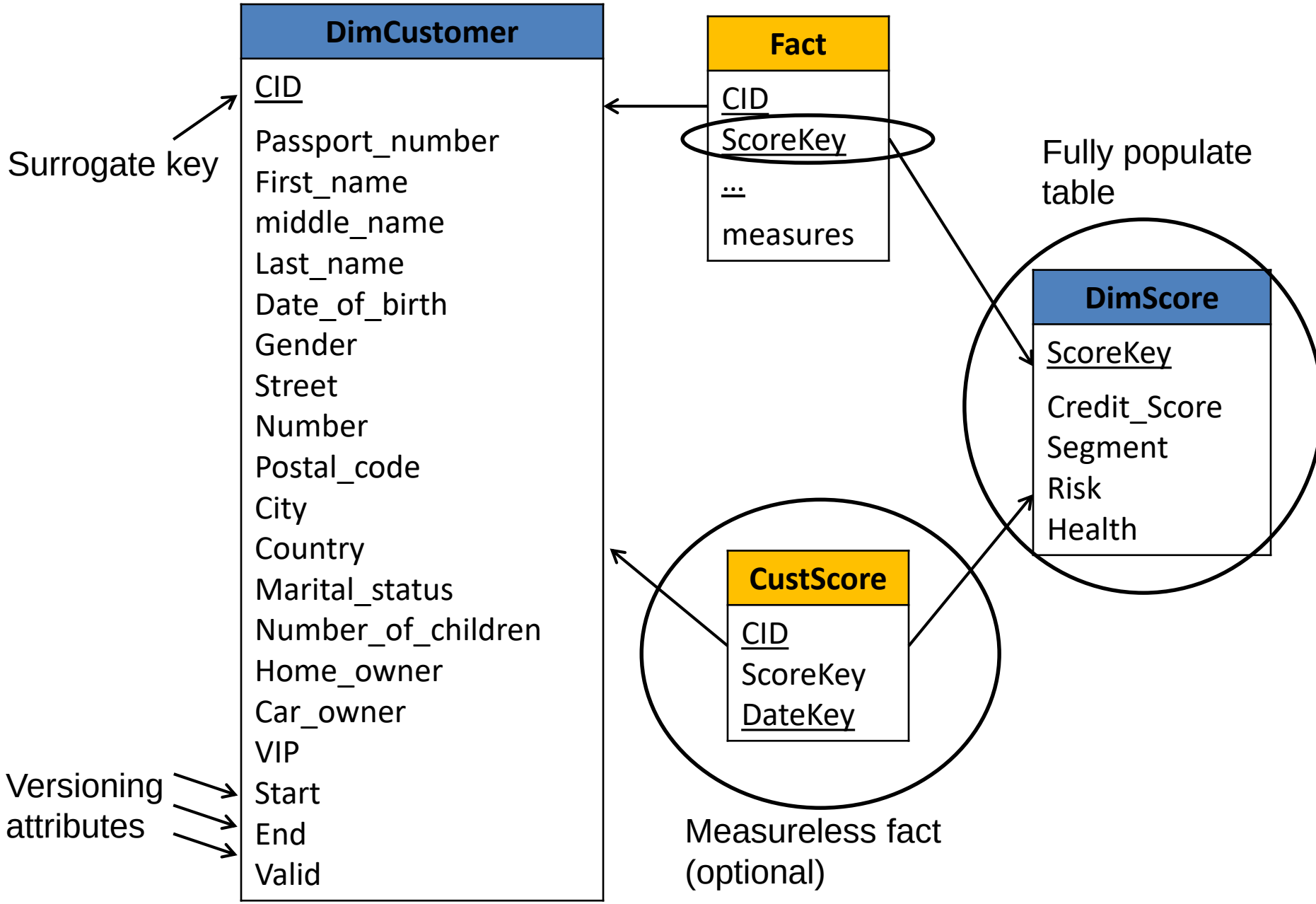
- **marital status (single, married, divorced, widowed)**
- **number of children**
- **whether or not he or she is a home owner**
- **whether or not he or she owns a car**
- **whether or not he or she is a VIP client**

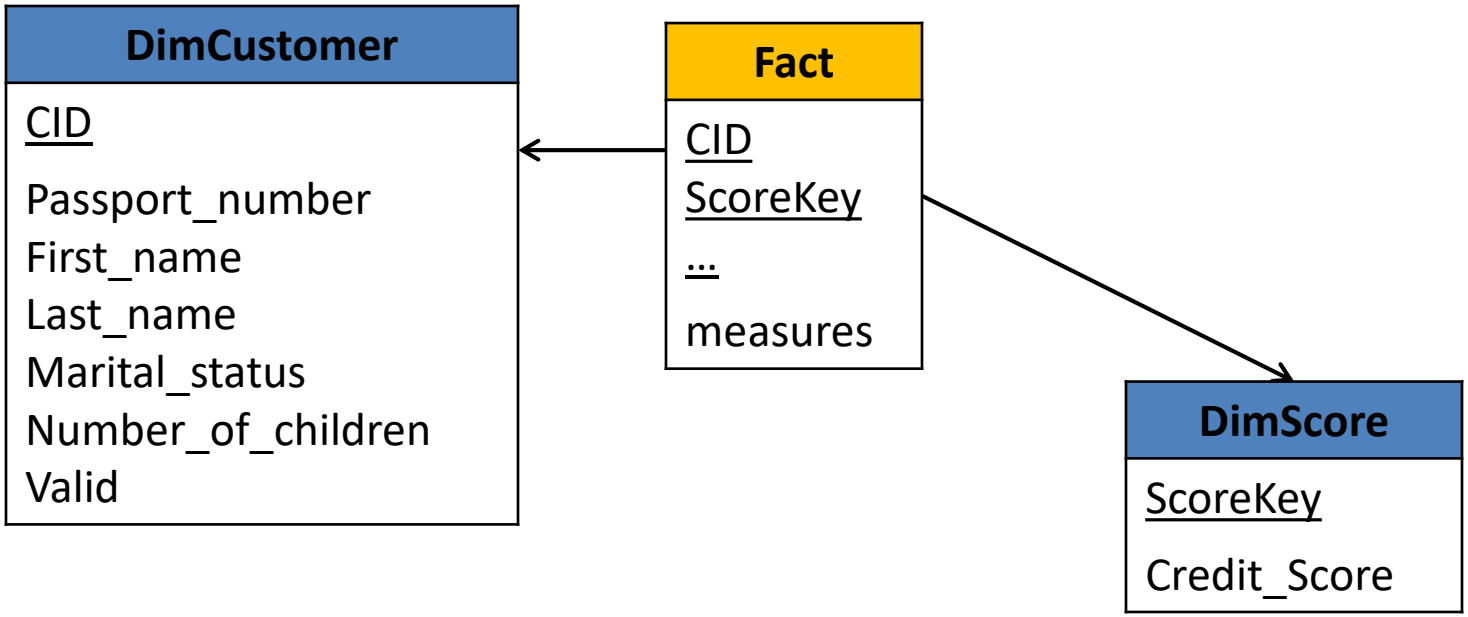
Some changes

- **credit score (A+, A, B, C, or D)**
- **market segment (a number from 1 to 8)**
- **risk categorization as a car driver**
  (number 1 to 10; starts at 5, decreases 1
- **per year, increases if person has an accident)**
- **health categorization (low risk, medium, or high risk)**

Frequent changes


DimCustomer
CID
Passport_number
First_name
middle_name
Last_name
Date_of_birth
Gender
Street
Number
Postal_code
City
Country
Marital_status
Number_of_children
Home_owner
Car_owner
VIP
Start
End
Valid
Surrogate key
Versioning attributes
Fact
CID
ScoreKey
...
measures
Fully populate table
DimScore
ScoreKey
Credit_Score
Segment
Risk
Health


DimCustomer
CID
Passport_number
First_name
middle_name
Last_name
Date_of_birth
Gender
Street
Number
Postal_code
City
Country
Marital_status
Number_of_children
Home_owner
Car_owner
VIP
Start
End
Valid
Surrogate key
Versioning attributes
Fact
CID
ScoreKey
...
measures
Fully populate table
DimScore
ScoreKey
Credit_Score
Segment
Risk
Health
CustScore
CID
ScoreKey
DateKey
Measureless fact (optional)



CID	passport	fname	sname	Marital stat	nChildren	Valid
…	…	…	…	…	…	…

CID	ScoreKey	…	measure
…	…	…	…

ScoreKey	credit
SK1	A+
…	…
SK8	DF

Example: Mini-Dimension

- A new customer with passport number 1234 and name Jan Janssens is added to the database. He is single, has no children and has a perfect credit score A+.

CID	passport	fname	sname	Marital stat	nChildren	Valid
…	…	…	…	…	…	…
C01	1234	Jan	Janssens	single	0	Y

CID	ScoreKey	…	measure
…	…	…	…

ScoreKey	credit
SK1	A+
…	…
SK8	DF

Example: Mini-Dimension

- Customer 1234 makes a sale; measure is 5

CID	passport	fname	sname	Marital stat	nChildren	Valid
…	…	…	…	…	…	…
C01	1234	Jan	Janssens	single	0	Y

CID	ScoreKey	…	measure
…	…	…	…
C01	SK1	…	5

ScoreKey	credit
SK1	A+
…	…
SK8	DF

Example: Mini-Dimension

- The customer with passport number 1234 gets married.

CID	passport	fname	sname	Marital stat	nChildren	Valid
…	…	…	…	…	…	…
C01	1234	Jan	Janssens	single	0	N
C02	1234	Jan	Janssens	married	0	Y

CID	ScoreKey	…	measure
…	…	…	…
C01	SK1	…	5

ScoreKey	credit
SK1	A+
…	…
SK8	DF

Example: Mini-Dimension

- The name of customer 1234 is corrected to Jan Jansens (one s removed from last name).

CID	passport	fname	sname	Marital stat	nChildren	Valid
…	…	…	…	…	…	…
C01	1234	Jan	Jansens	single	0	N
C02	1234	Jan	Jansens	married	0	Y

CID	ScoreKey	…	measure
…	…	…	…
C01	SK1	…	5

ScoreKey	credit
SK1	A+
…	…
SK8	DF

Example: Mini-Dimension

- Customer 1234 becomes the father of twins. His credit score drops to B. He makes a sale.

CID	passport	fname	sname	Marital stat	nChildren	Valid
…	…	…	…	…	…	…
C01	1234	Jan	Jansens	single	0	N
C02	1234	Jan	Jansens	married	0	N
C03	1234	Jan	Jansens	married	2	Y

CID	ScoreKey	…	measure
…	…	…	…
C01	SK1	…	5
C03	SK3	…	3

ScoreKey	credit
SK1	A+
…	…
SK8	DF

Outline

- Dealing with changing dimensions
  - Slowly Changing Dimensions
    - Type 1, 2, and 3
  - Rapidly changing dimensions
    - Mini dimension
- Specific dimension types
  - Outriggers
  - Degenerate dimension
  - Junk dimension
  - Time and Data Quality dimensions

Outrigger

- Dimension referred to by another dimension
  - Not exactly the same as a snow-flake

Example:

Date that an employee joined the company can be stored by referring to the date dimension

Degenerate Dimension

- Dimension without any content

CustID	ProdID	DateID	Transaction	quantity	Price
001	P003	D101	1	2	5.10
001	P001	D101	1	3	3.24
002	P002	D101	2	6	7.99
003	P003	D102	3	1	2.13
003	P005	D102	3	2	8.15

- Transaction is needed to be able to aggregate data at transaction level
- Dimension table is not necessary

Junk Dimension

- Many small dimensions combined into one
  - "Junk drawer"
  - Typically flags; promotion; how-displayed, …
- Combine into one dimension; fully populate

JunkID	Packed	Shipped	Delivered	Returned	Refunded
001	N	N	N	N	N
002	N	N	N	N	Y
003	N	N	N	Y	N
004	N	N	N	Y	Y
…	…	…	…	…	…
032	Y	Y	Y	Y	Y

Date/Time Dimension

- Date-time dimensions can become extremely large
  - Enormous number of possible combinations
  - Either get from data (expensive) or generate all possibilities (infeasible)
- Therefore: usually split into Date dimension at granularity day and a Time-of-day dimension
  - Limited number of dates
  - Only 1440 minutes in a day

Data Quality Dimension

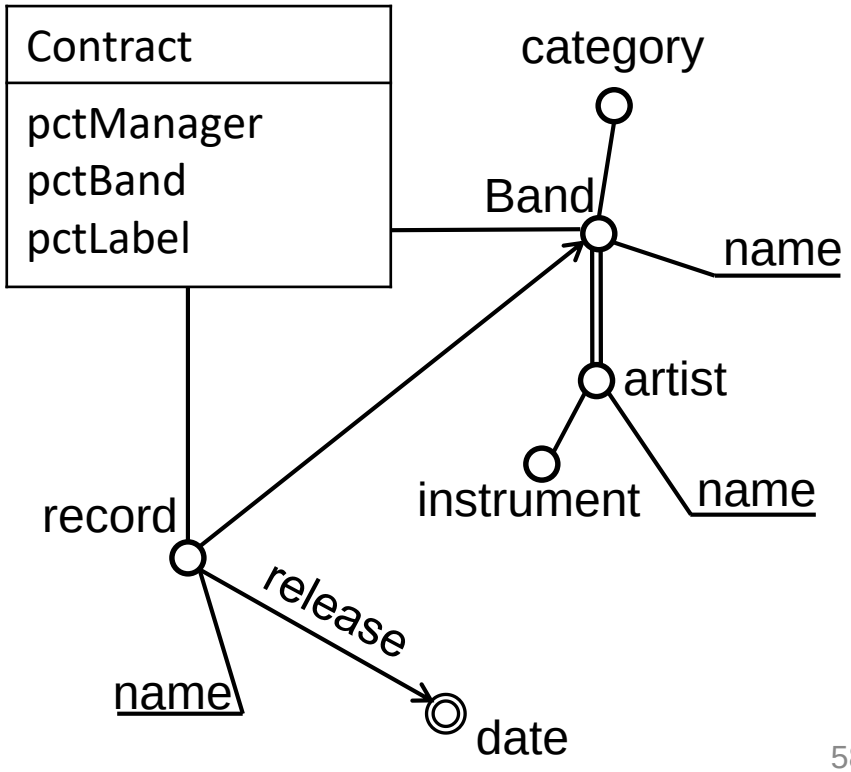
- Is sometimes added to comment on the quality of a fact
  - Normal value
  - Out-of-bounds value
  - Unlikely value
  - Verified value
  - Unverified value
  - …

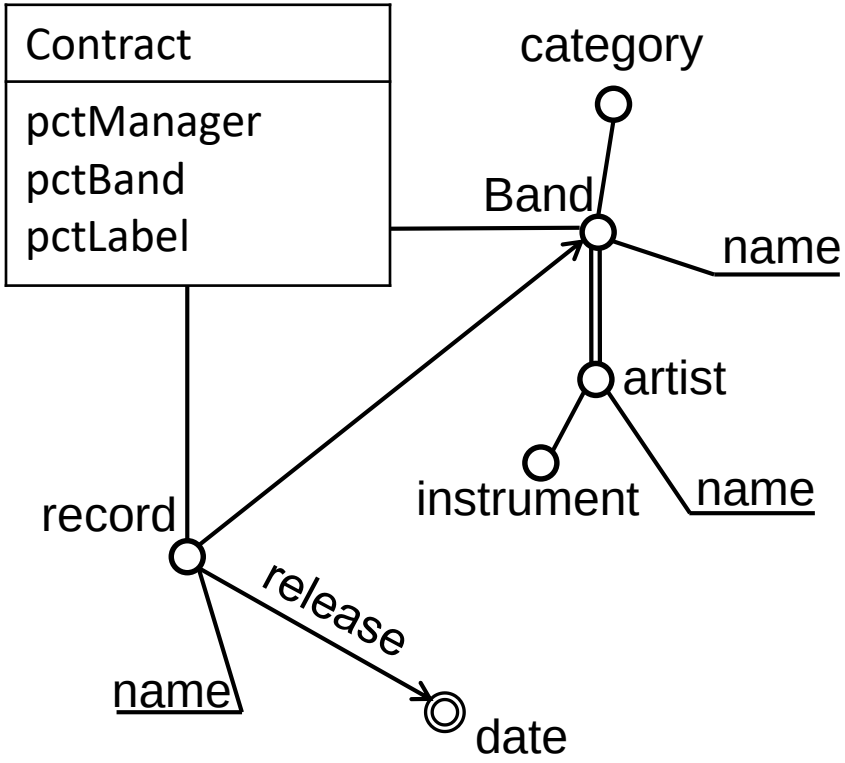
Summary

- Different techniques to store changes in dimensions
  - Type 1: update
  - Type 2: keep versions
  - Type 3: limited changes
  - Type 4: dimension splitting
- Special types of dimensions
  - Outrigger, degenerate, junk
  - Date, data quality

Exercise

- Devise a relational schema that can accommodate changes to the following attributes:
  - Band name
  - Band members
  - Instrument




Contract
pctManager
pctBand
pctLabel
category
Band
name
artist
instrument
name
record
release
name
date