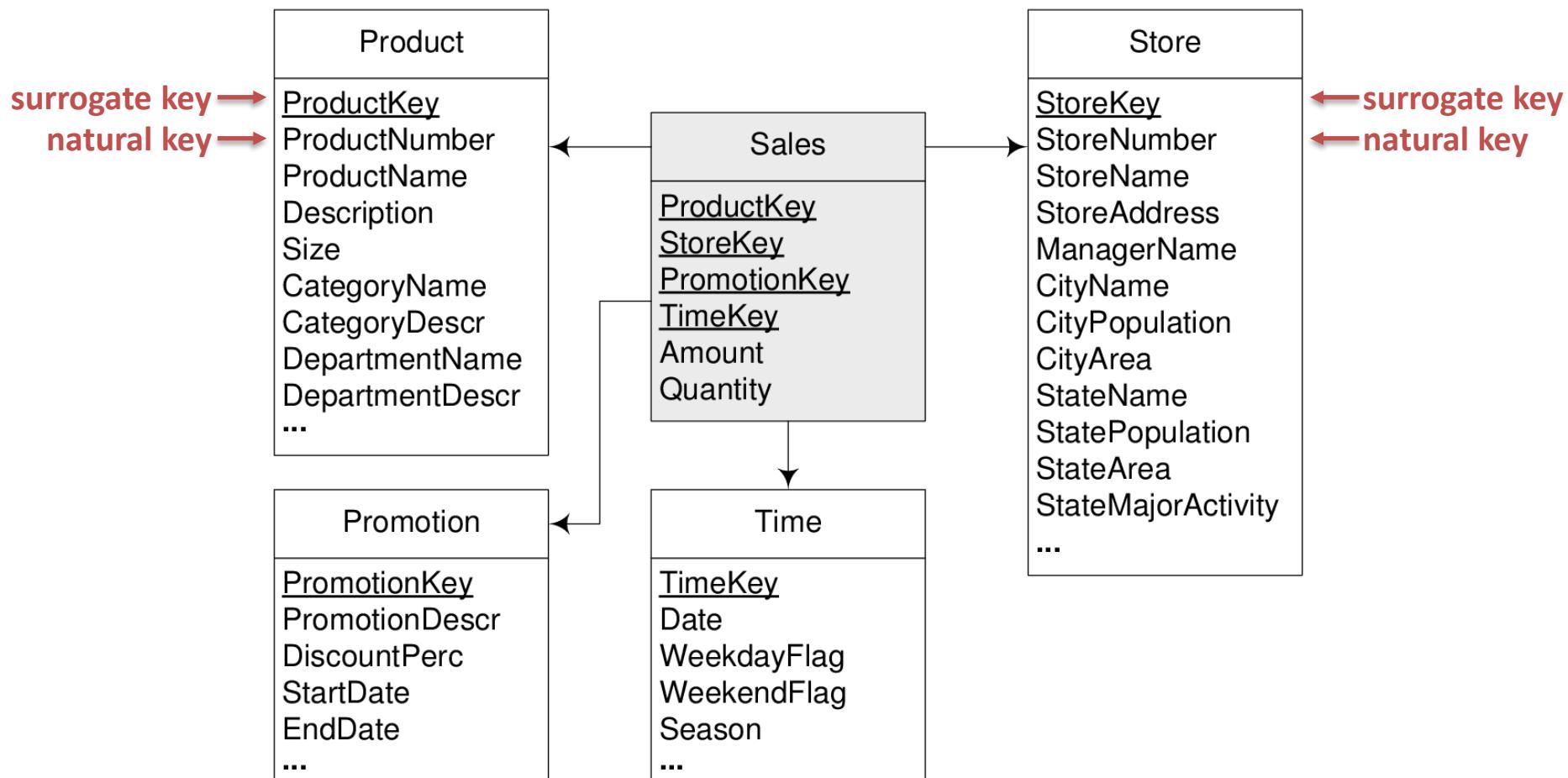


Data Analysis and Integration

Slowly changing dimensions

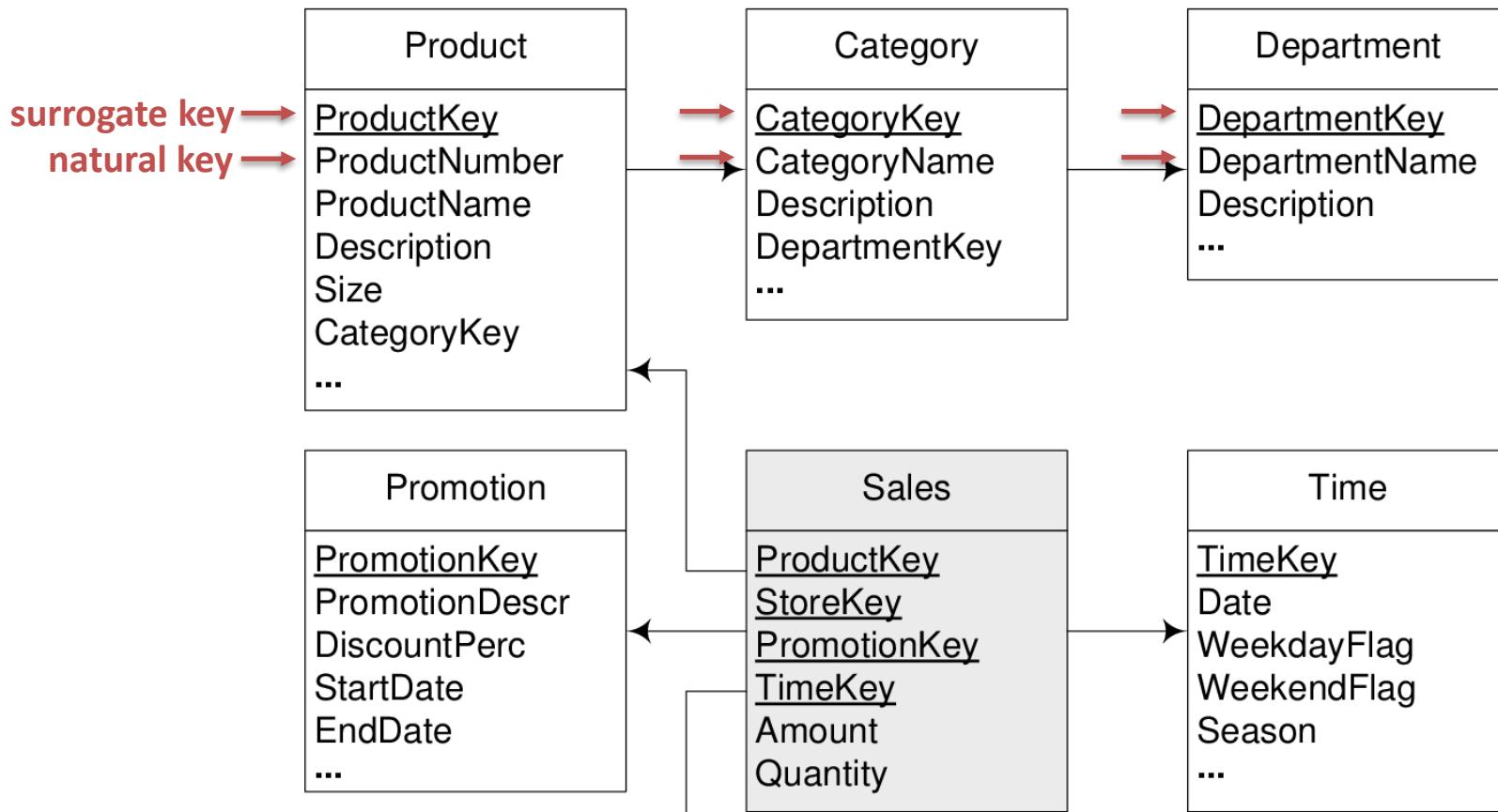
Recap: star schema

- Typically, each **dimension** has its own **surrogate key**



Recap: snowflake schema

- In snowflake, each **dimension** and **level** has its own key



Slowly changing dimensions

- What happens when attributes change?
 - e.g. a product changes to a different category
 - product p1 moves from category cat1 to cat2
- When calculating sales by product category...
 - should p1 be counted in the new or in the old category?
 - which one is correct, cat1 or cat2?
 - both can be correct, depending on time
 - older sales of p1 should be accounted in cat1
 - newer sales of p1 should be accounted in cat2
 - slowly-changing dimension

Example

- Sales fact table and product dimension table

TimeKey	EmployeeKey	CustomerKey	ProductKey	SalesAmount
t1	e1	c1	p1	100
t2	e2	c2	p1	100
t3	e1	c3	p3	100
t4	e2	c4	p4	100

ProductKey	ProductName	CategoryName	Description
p1	prod1	cat1	desc1
p2	prod2	cat1	desc1
p3	prod3	cat2	desc2
p4	prod4	cat2	desc2

Example

- Sales by employee and product category

```
SELECT      E.EmployeeKey, P.CategoryName, SUM(SalesAmount)
FROM        Sales S, Product P
WHERE       S.ProductKey = P.ProductKey
GROUP BY    E.EmployeeKey, P.CategoryName
```

EmployeeKey	CategoryName	SalesAmount
e1	cat1	100
e2	cat1	100
e1	cat2	100
e2	cat2	100

Example

- If prod1 changes to cat2

TimeKey	EmployeeKey	CustomerKey	ProductKey	SalesAmount
t1	e1	c1	p1	100
t2	e2	c2	p1	100
t3	e1	c3	p3	100
t4	e2	c4	p4	100

ProductKey	ProductName	CategoryName	Description
p1	prod1	cat1 cat2	desc1
p2	prod2	cat1 cat2	desc1
p3	prod3	cat2	desc2
p4	prod4	cat2	desc2

Example

- If prod1 changes to cat2

```
SELECT      E.EmployeeKey, P.CategoryName, SUM(SalesAmount)
FROM        Sales S, Product P
WHERE       S.ProductKey = P.ProductKey
GROUP BY    E.EmployeeKey, P.CategoryName
```

EmployeeKey	CategoryKey	SalesAmount
e1	cat2	200
e2	cat2	200

- incorrect because prod1 was cat1 when it was sold!

Slowly changing dimensions

- How to deal with SCDs
 - type 0: retain original
 - type 1: overwrite
 - type 2: add row
 - type 3: add column
 - type 4: add mini-dimension
 - type 5: add mini-dimension and foreign key
 - type 6: extends type 2 with current value
 - type 7: add foreign key to fact table

SCDs Type 0

- Type 0
 - retain original value in the DW
 - do not update with new value

ProductKey	ProductName	CategoryName	Description
p1	prod1	cat1	desc1
p2	prod2	cat1	desc1
p3	prod3	cat2	desc2
p4	prod4	cat2	desc2

SCDs Type 1

- Type 1
 - overwrite old value with new value
 - assumes the modification is due to an error in the original data
 - history of the attribute is lost

ProductKey	ProductName	CategoryName	Description
p1	prod1	cat1 cat2	desc1
p2	prod2	cat1 cat2	desc1
p3	prod3	cat2	desc2
p4	prod4	cat2	desc2

SCDs Type 2

- Type 2
 - store multiple versions of the same product
 - add two columns with validity interval (from date, to date)
 - note the use of the surrogate key for this purpose

Product Key	Product Name	Category Name	Description	From	To
p1	prod1	cat1	desc1	2010-01-01	2011-12-31
p11	prod1	cat2	desc2	2012-01-01	9999-12-31
p2	prod2	cat1	desc1	2012-01-01	9999-12-31
p3	prod3	cat2	desc2	2012-01-01	9999-12-31
p4	prod4	cat2	desc2	2012-01-01	9999-12-31

SCDs Type 2

- Type 2
 - a product participates in the fact table with as many surrogates (product keys) as there are attribute changes
 - the number of distinct product keys no longer corresponds to the number of distinct products

Product Key	Product Name	Category Name	Description	From	To
p1	prod1	cat1	desc1	2010-01-01	2011-12-31
p11	prod1	cat2	desc2	2012-01-01	9999-12-31
p2	prod2	cat1	desc1	2012-01-01	9999-12-31
p3	prod3	cat2	desc2	2012-01-01	9999-12-31
p4	prod4	cat2	desc2	2012-01-01	9999-12-31

SCDs Type 2

- Type 2
 - another variant of type 2 with an additional column

Product Key	Product Name	Category Name	Description	From	To	Row Status
p1	prod1	cat1	desc1	2010-01-01	2011-12-31	Expired
p11	prod1	cat2	desc2	2012-01-01	9999-12-31	Current
...

SCDs Type 2

- Type 2 for snowflake structure
 - e.g. product table and category table

Product Key	Product Name	Category Key
p1	prod1	c1
p2	prod2	c1
p3	prod3	c2
p4	prod4	c2

Category Key	Category Name	Description
c1	cat1	desc1
c2	cat2	desc2
c3	cat3	desc3
c4	cat4	desc4

SCDs Type 2

- Type 2 for snowflake structure
 - add two columns to product table (from date, to date)
 - here, a product can change category
 - but categories themselves do not change

Product Key	Product Name	Category Key	From	To
p1	prod1	c1	2010-01-01	2011-12-31
p11	prod1	c2	2012-01-01	9999-12-31
p2	prod2	c1	2010-01-01	9999-12-31
p3	prod3	c2	2010-01-01	9999-12-31
p4	prod4	c2	2011-01-01	9999-12-31

Category Key	Category Name	Description
c1	cat1	desc1
c2	cat2	desc2
c3	cat3	desc3
c4	cat4	desc4

SCDs Type 2

- Type 2 for snowflake structure
 - however, there might be changes in the categories
 - changes at a higher level must be propagated to lower levels
 - e.g. change in category description

Product Key	Product Name	Category Key	From	To
p1	prod1	c1	2010-01-01	2011-12-31
p11	prod1	c11	2012-01-01	9999-12-31
p2	prod2	c1	2010-01-01	9999-12-31
p3	prod3	c2	2010-01-01	9999-12-31
p4	prod4	c2	2011-01-01	9999-12-31

Category Key	Category Name	Description	From	To
c1	cat1	desc1	2010-01-01	2011-12-31
c11	cat1	desc11	2012-01-01	9999-12-31
c2	cat2	desc2	2012-01-01	9999-12-31
c3	cat3	desc3	2010-01-01	9999-12-31
c4	cat4	desc4	2010-01-01	9999-12-31

SCDs Type 3

- Type 3
 - additional column for each attribute that might change
 - e.g. new category, new description
 - stores only the current version and the previous one
 - the two most recent versions

Product Key	Product Name	Category Name	New Category	Description	New Description
p1	prod1	cat1	cat2	desc1	desc2
p2	prod2	cat1	Null	desc1	Null
p3	prod3	cat2	Null	desc2	Null
p4	prod4	cat2	Null	desc2	Null

Slowly changing dimensions

- Real-world DW software and tools
 - typically, provide support for SCDs of type 1 to type 3
 - however, large dimension tables with many recorded changes decrease the performance of join operations
 - there are more sophisticated types of SCDs to address those cases, but more difficult to implement
 - types 4 to 7

SCDs Type 4

- Type 4
 - for attributes that change frequently, create a **mini-dimension**
 - e.g. sales ranking and price range of a product

Product FeaturesKey	Sales Ranking	Price Range
pf1	1	1–100
pf2	2	1–100
...
pf200	7	500–600

- one row for each unique combination of SalesRanking and PriceRange (not one row per product)

SCDs Type 4

- Type 4
 - add ProductFeaturesKey to fact table
 - sales ranking and price range when the product was sold

TimeKey	EmployeeKey	CustomerKey	ProductKey	SalesAmount	ProductFeaturesKey
t1	e1	c1	p1	100	pf2
t2	e2	c2	p1	100	pf2

Product FeaturesKey	Sales Ranking	Price Range
pf1	1	1–100
pf2	2	1–100
...
pf200	7	500–600

SCDs Type 4

- Type 4
 - if the sales ranking of product p1 goes up, subsequent sales will be entered with pf1

TimeKey	EmployeeKey	CustomerKey	ProductKey	SalesAmount	ProductFeaturesKey
t1	e1	c1	p1	100	pf2
t2	e2	c2	p1	100	pf2
t3	e3	c3	p1	50	pf1



Product FeaturesKey	Sales Ranking	Price Range
pf1	1	1–100
pf2	2	1–100
...
pf200	7	500–600

SCDs Type 5

- Type 5
 - extension of type 4 with foreign key being added to the dimension table

Product Key	Product Name	CurrentProductFeaturesKey
p1	prod1	pf1
...

- makes it easier to analyze the current features of a product
 - e.g. roll-up based on current product features

SCDs Type 5

- Type 5
 - CurrentProductFeaturesKey is a type 1 attribute
 - must be overwritten when the product features change

Product Key	Product Name	CurrentProduct FeaturesKey
p1	prod1	pf1
...

- however, the fact table includes ProductFeaturesKey at the time of the sales
 - possibly different from CurrentProductFeaturesKey in the dimension table

SCDs Type 6

- Type 6
 - extension of type 2 with additional column for current value
 - e.g. current value for the category of each product

Product Key	Product Name	Category Key	From	To	Current CategoryKey
p1	prod1	c1	2010-01-01	2011-12-31	c11
p11	prod1	c11	2012-01-01	9999-12-31	c11
p2	prod2	c1	2010-01-01	9999-12-31	c1
p3	prod3	c2	2010-01-01	9999-12-31	c2
p4	prod4	c2	2011-01-01	9999-12-31	c2

SCDs Type 6

- Type 6
 - use CategoryKey to group by original categories
 - product category when the product was sold
 - use CurrentCategoryKey to group by current categories

Product Key	Product Name	Category Key	From	To	Current CategoryKey
p1	prod1	c1	2010-01-01	2011-12-31	c11
p11	prod1	c11	2012-01-01	9999-12-31	c11
p2	prod2	c1	2010-01-01	9999-12-31	c1
p3	prod3	c2	2010-01-01	9999-12-31	c2
p4	prod4	c2	2011-01-01	9999-12-31	c2

SCDs Type 7

- Type 7
 - used if there are several attributes for which we need to support both current and historical perspectives
 - type 6 would require one additional column in the dimension table for each of those attributes
 - instead, type 7 stores **natural key** in fact table

TimeKey	EmployeeKey	CustomerKey	ProductKey	ProductName	SalesAmount
t1	e1	c1	p1	prod1	100
t2	e2	c2	p11	prod1	100
t3	e1	c3	p3	prod3	100
t4	e2	c4	p4	prod4	100

SCDs Type 7

- Type 7
 - use ProductKey for historical analysis
 - i.e. based on product values when the products were sold
 - use ProductName to analyze by current values
 - needs an additional view (see next slide)

TimeKey	EmployeeKey	CustomerKey	ProductKey	Product Name	SalesAmount
t1	e1	c1	p1	prod1	100
t2	e2	c2	p11	prod1	100
t3	e1	c3	p3	prod3	100
t4	e2	c4	p4	prod4	100

SCDs Type 7

- Type 7
 - use ProductName to analyze by current values
 - for this purpose, use a view with the current values

Product Name	Category Key
prod1	c2
prod2	c1
prod3	c2
prod4	c2

- current values correspond to the current version of each product (with $\text{from_date} \leq \text{current_date} < \text{to_date}$)

SCDs Type 7

- Type 7
 - a variant using only the **surrogate key**
 - avoids two foreign keys for product in fact table
 - use a view with current category for each product, where product is now identified by surrogate key

Product Key	Current CategoryKey
p1	c11
p11	c11
p2	c1
p3	c2
p4	c2

Slowly changing dimensions

Type 0	no change of value
Type 1	overwrite value
Type 2	add new row with validity interval
Type 3	add new column to preserve current and previous value
Type 4	add mini-dimension for frequently changing attributes
Type 5	add mini-dimension along with FK in base dimension
Type 6	add new row with validity interval plus additional column for current value
Type 7	add new row with validity interval plus view with current values by natural key or surrogate key