SS G515 - Data Warehousing:

Dimensional Modeling

Dr. Yashvardhan Sharma

Assistant Professor, CS & IS Dept.

BITS-Pilani

Factless fact tables

- Some fact tables quite simply have no measured facts!
- Are useful to describe events and coverage, i.e. the tables contain information that something has/has not happened.
- Often used to represent many-to-many relationships
- The only thing they contain is a concatenated key, they do still however represent a focal event which is identified by the combination of conditions referenced in the dimension tables
- There are two main types of factless fact tables:
  - event tracking tables
  - coverage tables

Factless Fact Tables

- Facts are typically numeric measures
- Events which record merely the coming together of dimensional entities at a particular moment
  - Student attending a class
  - A particular product on promotion
- Can also be used to analyze what did not happen
  - Factless coverage fact table about all possibilities
  - Activity table about events that did happen
  - Subtract activity from coverage
  - Example: products that were on promotion but did not sell

Factless Fact Tables

- Case studies that employ factless fact tables
  - Retail sales
  - Order management
  - Education

Retail sales

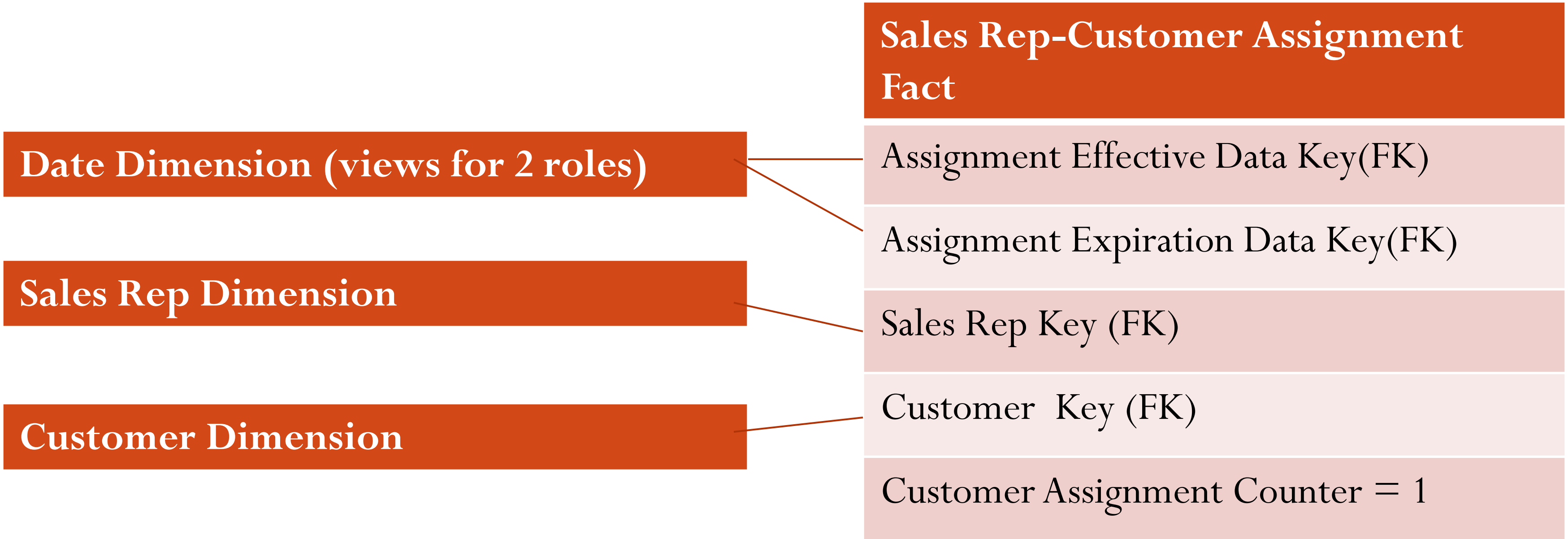
- Retail sales schema can not answer an important question – What products were on promotion but did not sell?
- Sale FT records only those SKUs that actually got sold
- Not advisable to keep those SKUs in sales FT that did not sell (it is already huge!!)
- Introduce promotion coverage fact table
  - Same keys as ales fact table
  - Grain is different
  - FT row represents a product that was on promotion regardless of whether the product sold
  - Factless fact table

Retail sales

- What products were on promotion but did not sell?
- Two step process:
  - Query the promotion coverage FFT to determine all the products that were on promotion on a given day
  - Find out all products that sold on a given day
  - Difference of these two lists!!
  - Try writing SQL query for this!

Order Management

- Customer/representative assignment
- Representatives are assigned to customers and it is not necessary that every assignment would lead to a sale



Sales Rep-Customer Assignment Fact

- Assignment Effective Data Key(FK)
- Assignment Expiration Data Key(FK)
- Sales Rep Key (FK)
- Customer Key (FK)
- Customer Assignment Counter = 1

Date Dimension (views for 2 roles)

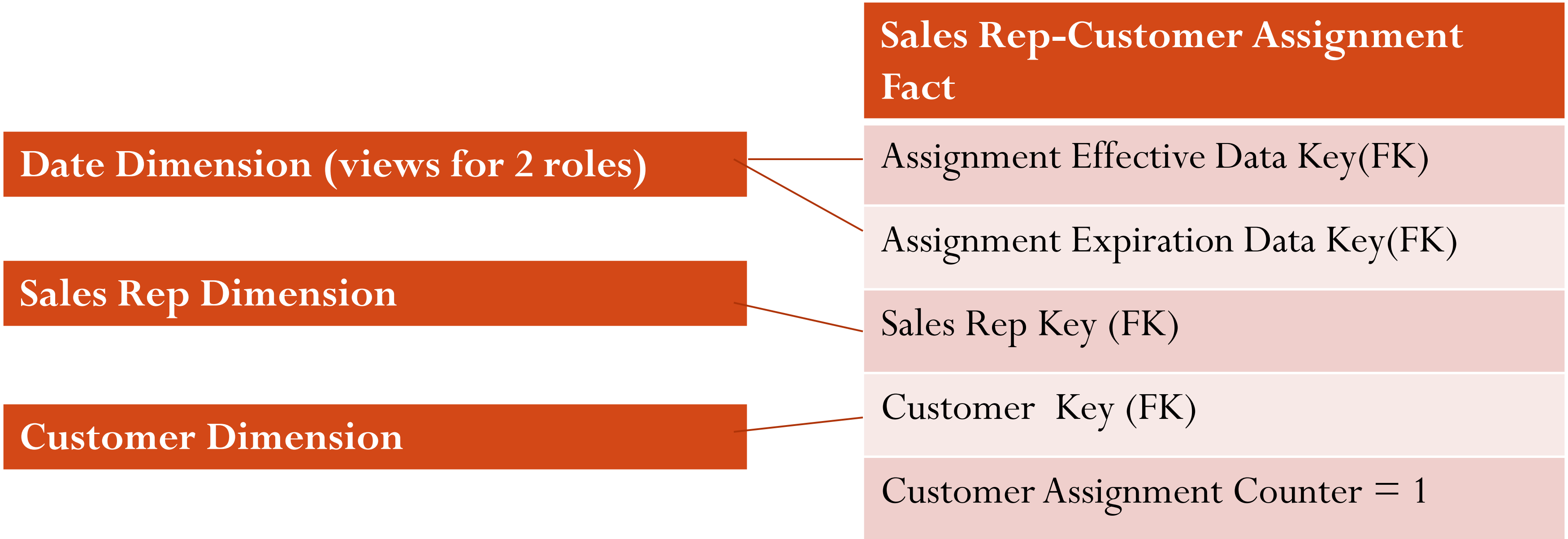
Sales Rep Dimension

Customer Dimension

Figure taken from Kibmall's book – The Data Warehouse Toolkit, 3e

Order Management

- Sales rep coverage factless fact table
- Allows us to answer queries like which assignments never resulted in sales



Date Dimension (views for 2 roles)

Sales Rep Dimension

Customer Dimension

Sales Rep-Customer Assignment Fact

Assignment Effective Data Key(FK)

Assignment Expiration Data Key(FK)

Sales Rep Key (FK)

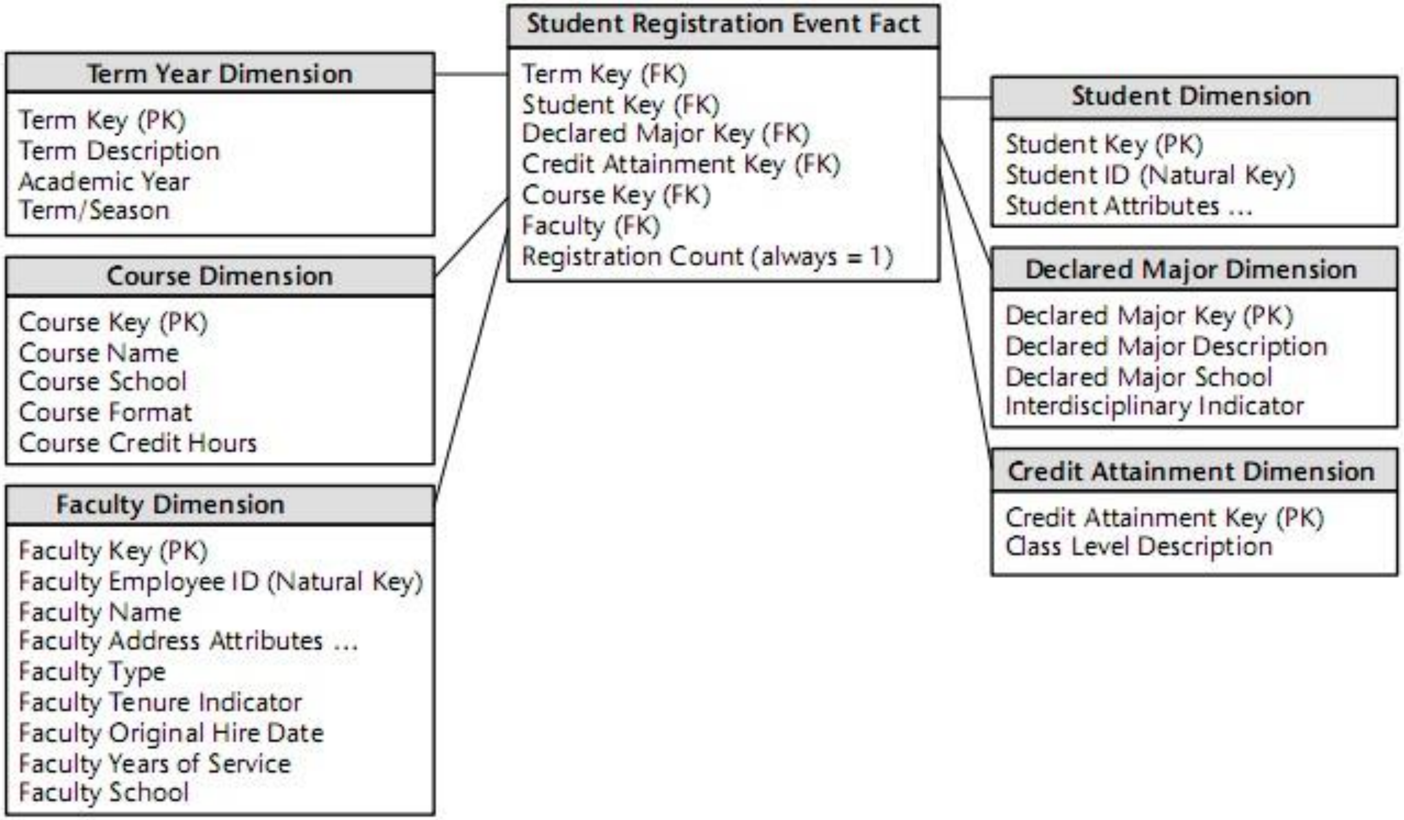
Customer Key (FK)

Customer Assignment Counter = 1

Figure taken from Kibmall's book – The Data Warehouse Toolkit, 3e

Education

- Student Registration



Student Registration Event Fact

Term Key (FK)
Student Key (FK)
Declared Major Key (FK)
Credit Attainment Key (FK)
Course Key (FK)
Faculty (FK)
Registration Count (always = 1)

Term Year Dimension

Term Key (PK)
Term Description
Academic Year
Term/Season

Course Dimension

Course Key (PK)
Course Name
Course School
Course Format
Course Credit Hours

Faculty Dimension

Faculty Key (PK)
Faculty Employee ID (Natural Key)
Faculty Name
Faculty Address Attributes ...
Faculty Type
Faculty Tenure Indicator
Faculty Original Hire Date
Faculty Years of Service
Faculty School

Student Dimension

Student Key (PK)
Student ID (Natural Key)
Student Attributes ...

Declared Major Dimension

Declared Major Key (PK)
Declared Major Description
Declared Major School
Interdisciplinary Indicator

Credit Attainment Dimension

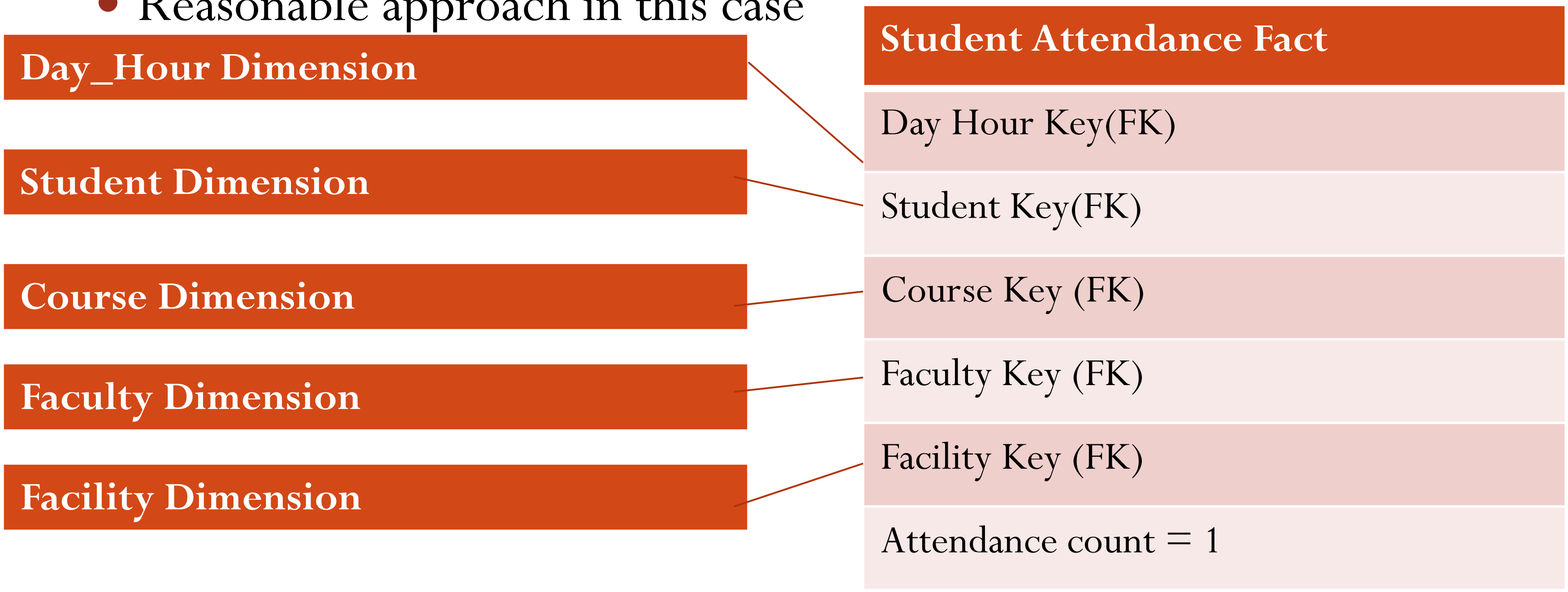
Credit Attainment Key (PK)
Class Level Description

Figure 12.2 Student registration events as a factless fact table.

Figure taken from Kibmall's book – The Data Warehouse Toolkit, 2e

Education

- Student Attendance
- What about events that did not happen?
  - Attendance count = 0 or 1
  - Ceases to be factless fact table
  - Reasonable approach in this case



Day_Hour Dimension

Student Dimension

Course Dimension

Faculty Dimension

Facility Dimension

Student Attendance Fact

Day Hour Key(FK)

Student Key(FK)

Course Key (FK)

Faculty Key (FK)

Facility Key (FK)

Attendance count = 1

Figure taken from Kibmall's book – The Data Warehouse Toolkit, 3e

Factless Fact Tables: Summary

- Records events which do not have associated facts
- Dummy fact = 1 to increase readability of SQL queries

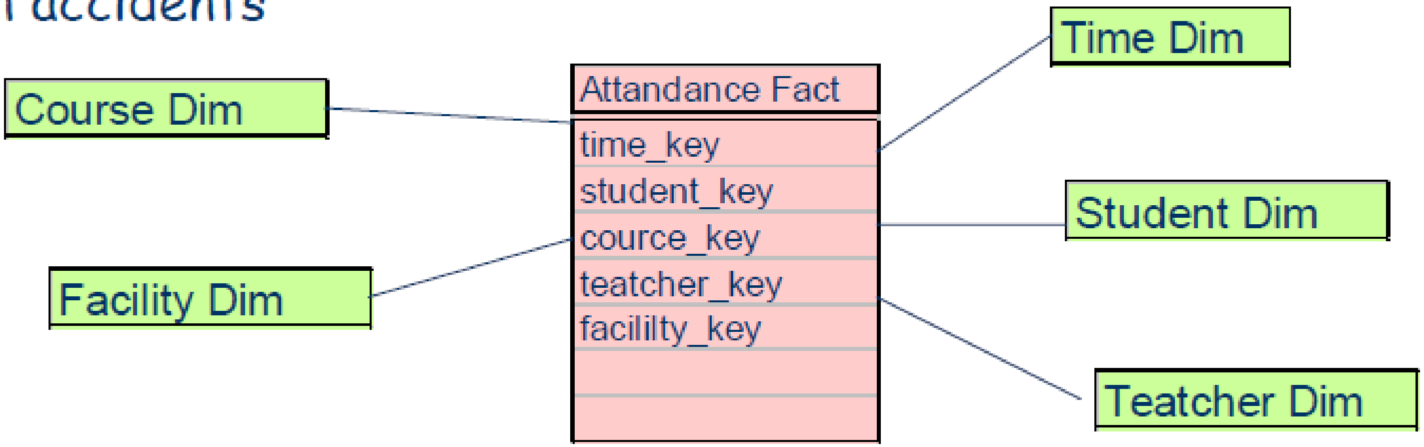
  select faculty, SUM(registration count)….

  Group By Faculty
- Used in retails sales, order management, education etc.
- In some situations, events that did not happen can also be recorded, but then the fact table ceases to be factless

Factless fact tables

Event tracking tables

- records events, e.g. records every time a student attends a course, or people involved in accidents and vehicles involved in accidents



Coverage tables

- description of something that did not happend, e.g. which product did not sell during a promotion campaign.

Changing Dimensions

- Slowly Changing Dimensions
- Rapidly Changing Dimensions
- Small Dimensions
- Monster Dimensions

Slowly Changing Dimensions

Why?

- Let’s take Sales fact table for example
- Every day more and more sales take place, hence:
- More and more rows are added to the fact table
- Very rarely are the rows in the fact table updated with changes

Also Consider...

How will we adjust the fact table when changes are made?

Why? cont'…

- Consider the dimension tables
- Compared to the fact tables, they are more stable and less volatile
- However, unlike fact tables, a dimension table does not change just through the increase of number of rows, but also through changes to the attributes themselves

- We will focus on (Slowly Changing) Dimensions

When? Good question:

- **Inside the ETL process**
- **After the ETL process, as a stored procedure**

- From what we discussed for now, we can derive these principles:
- Most dimensions are generally constant over time
- Many dimensions, through not constant over time, change slowly
- The product (<u>business</u>) key of the source record does not change
- The description and other attributes change slowly over time
- In the source OLTP system, the new values overwrite the old ones
- Overwriting of dimension table attributes is not always the appropriate option in a data warehouse
- The ways changes are made to the dimension tables depend on the types of changes and <u>what information must be preserved</u> in the DWH

How? 3 Answers:

- The usual changes to dimension tables are classified into three types
- Type 1 (Overwrite)
- Type 2 (Adding a row)
- Type 3 (Adding a column)
- We will consider the points discussed earlier when deciding which type to use

Slowly Changing Dimensions

- For example, the product or customer dimension

The assumption: the key does not change, but

some of the attributes does.

- Type 1: Overwrite the dimension record with the

new values, thereby losing history

- Type 2: Create a new additional dimension record

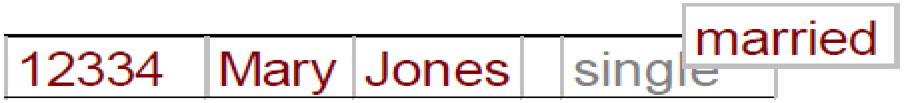
using a new value of the surrogate key

- Type 3: Create a new field in the dimension record to store the new value of the attribute

Type 1

- Overwrite the old value of an attribute with a new one

e.g.



12334	Mary	Jones		single

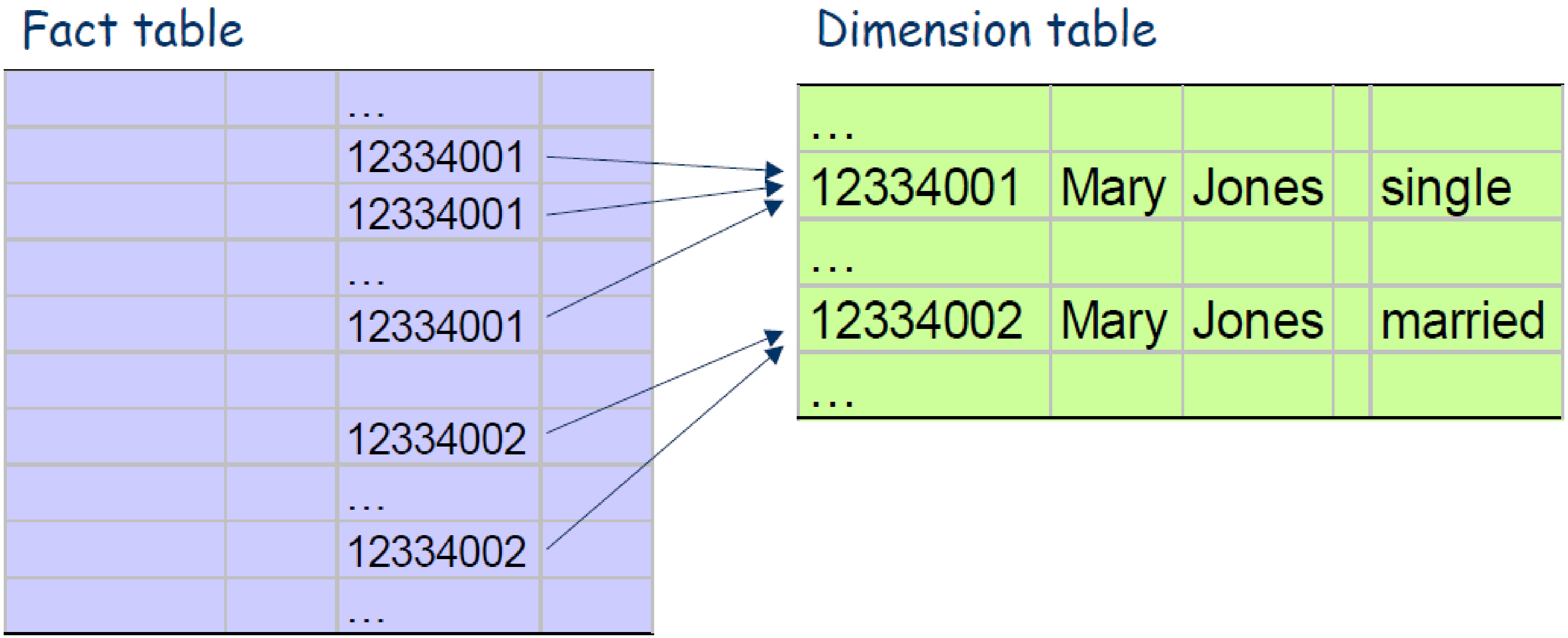
married

+ easy to implement

- avoids the real goal, which is to accurately track history

Type 2

- Create a new additional dimension record
- A generalised (surrogate) key is required (which is a responsibility of the data warehouse team)



Fact table

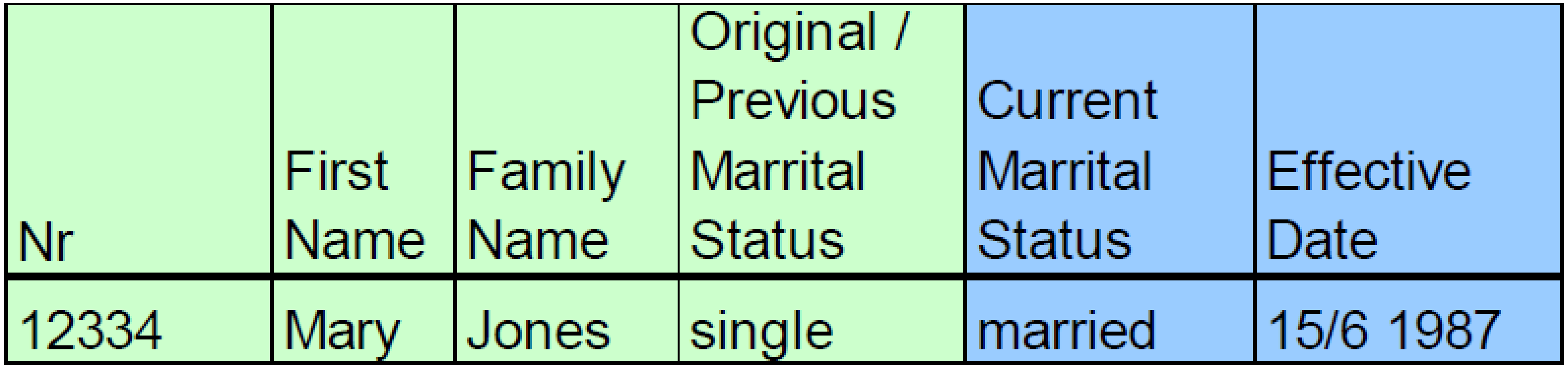
		…	
		12334001	
		12334001	
		…	
		12334001	
		12334002	
		…	
		12334002	
		…	

Dimension table

…				
12334001	Mary	Jones		single
…				
12334002	Mary	Jones		married
…				

Type 3

- Create a new field in the dimension record



Nr	First Name	Family Name	Original / Previous Marrital Status	Current Marrital Status	Effective Date
12334	Mary	Jones	single	married	15/6 1987

Surrogate Key

- A surrogate key is a unique identifier for the entity in the modeled world
- It is *not* derived from application data
- It's not meant to be shown outside the DWH

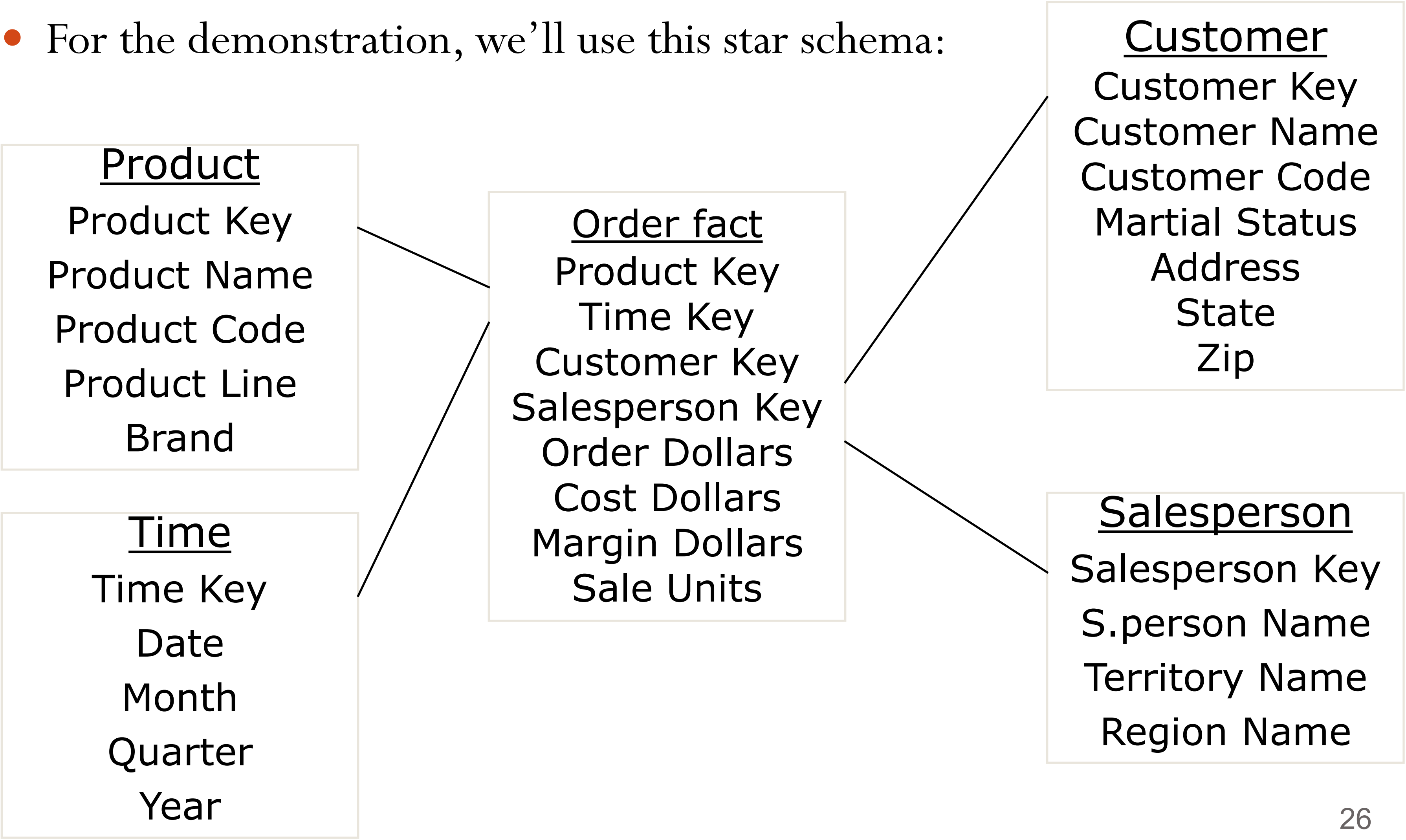
- It's only significance is to act as the primary key
- Frequently it's a sequential number (*Sequence* in Oracle or *Identity* in SQL Server)

Surrogate Key

- Having the key independent of all other columns insulates the database relationships from changes in the data values or database design (making the database more agile) and guarantees uniqueness
- For example: An employee ID is chosen as the neutral (business) key of an employee DWH. Because of a merger with another company, new employees from the merged company must be inserted. There is one employee who works in both companies…
- If the key is a compound key, joining is more expensive because there are multiple columns to compare. Surrogate keys are always contained in a single column

Our example

- For the demonstration, we'll use this star schema:



Product
Product Key
Product Name
Product Code
Product Line
Brand

Time
Time Key
Date
Month
Quarter
Year

Order fact
Product Key
Time Key
Customer Key
Salesperson Key
Order Dollars
Cost Dollars
Margin Dollars
Sale Units

Customer
Customer Key
Customer Name
Customer Code
Martial Status
Address
State
Zip

Salesperson
Salesperson Key
S.person Name
Territory Name
Region Name

Type 1 Changes

- Usually relate to corrections of errors in the source system
- For example, the customer dimension: Mickey Schreiber -> Miky Schreiber

Type 1 Changes, cont.

- General Principles for Type 1 changes:
- Usually, the changes relate to correction of errors in the source system
- Sometimes the change in the source system has no significance
- The old value in the source system needs to be discarded
- The change in the source system need not be preserved in the DWH

Also Consider…

What will happen when only the last value before the change is needed?

Applying Type 1 changes

Change Box

Customer Code:
K12356
Customer Name:
Miky Schreiber

Key Restructuring
K12356 -> 33154112

	Before	After
Customer Key:	~~33154112~~	33154112
Customer Name:	~~Mickey Schreiber~~	Miky Schreiber
Customer Code:	~~K12356~~	K12356
Martial Status:	~~Married~~	Married
Address:	~~Negba 11 ST~~	Negba 11 ST

Type 1 Changes

- Overwrite the attribute value in the dimension table row with the new value
- The old value of the attribute is not preserved
- No other changes are made in the dimension table row
- The key of this dimension table or any other key values are not affected
- Easiest to implement

Type 2 Changes

- Let's look at the martial status of Miky Schreiber
- One the DWH's requirements is to track orders by martial status (in addition to other attributes)
- All changes before 11/10/2004 will be under Martial Status = Single, and all changes after that date will be under Martial Status = Married
- We need to aggregate the orders before and after the marriage separately

- Let's make life harder:
- Miky is living in Negba st., but on 30/8/2009 he moves to Avivim st.

Type 2 Changes, cont.

- General Principles for Type 2 changes:
- They usually relate to true changes in source systems
- There is a need to preserve history in the DWH
- This type of change partitions the history in the DWH
- Every change for the same attributes must be preserved

Also Consider…

- **Must we track changes for all the attributes?**
- **For which attributes will we track changes? What are the considerations?**

Applying Type 2 changes

Change Box

Customer Code: K12356

Martial Status (11/10/2004): Married

Address (30/8/2009): Avivim st.

Key Restructuring

K12356 -> 33154112

51141234

52789342

	Before	After 11/10/2004	After 30/8/2009
Customer Key:	33154112	51141234	52789342
Customer Name:	Miky Schreiber	Miky Schreiber	Miky Schreiber
Customer Code:	K12356	K12356	K12356
Martial Status:	Single	Married	Married
Address:	Negba 11 ST	Negba 11 ST	Avivim st.

Also Consider…

- **What will happen if in addition to Address we also have State, zip code?**
- **What will happen if the customer code will change?**

Type 2 concluded

- The steps:
- Add a new dimension table row with the new value of the changed attribute
- An effective date will be included in the dimension table
- There are no changes to the original row in the dimension table
- The key of the original row is not affected
- The new row is inserted with a new surrogate key

Also Consider…

- **What is the data type of the effective date column? Must it contain both date and time?**
- **How will the surrogate key be built?**
- **Advantages? Disadvantages?**

Type 3 Changes

- Not common at all

Complex queries on type-~~2~~ 3 changes may be

- Hard to implement
- Time-consuming
- Hard to maintain

- We want to track history without lifting heavy burden
- There are many soft changes and we don't care for the "far" history

Type 3 Changes

- General Principles:
- They usually relate to “soft” or tentative changes in the source systems
- There is a need to keep track of history with old and new values of the changes attribute
- They are used to compare performances across the transition
- They provide the ability to track forward and backward

Applying Type 3 changes

Key Restructuring
RS199701 -> 12345

Salesperson ID:
RS199701
Territory Name:
Netanya
(12/1/2000)

	Before	After
Salesperson Key:	~~12345~~	12345
Salesperson Name:	~~Boris Kavkaz~~	Boris Kavkaz
Old Territory Name:	~~(null)~~	Ra’anana
Current Territory Name:	~~Ra’anana~~	Netanya
Effective Date:	~~1/1/1998~~	12/1/2000

Also Consider…

- **What is the effective date before the change?**
- **Can the old terriroty column contain null? What about the current territory?**

Type 3 concluded

- No new dimension row is needed
- The existing queries will seamlessly switch to the current value
- Any queries that need to use the old value must be revised accordingly
- The technique works best for one soft change at a time
- If there is a succession of changes, more sophisticated techniques must be advised

Conclusions

- 3 Main ways of history tracking
- Choose the way you'd like for every dimension table
- You may combine the types
- It all depends on the system's requirements