Introduction

School of Information Studies
Syracuse University

Agenda

- Dimensional model designs in detail
- Slowly changing dimensions
- Rapidly changing dimensions
- Advanced dimensional modeling concepts
- Walkthrough: Detailed dimensional modeling worksheet

Where Are We?

We covered:

- Requirements analysis

We learned how to:

- Turn business processes into dimensional models

High level

What we will cover

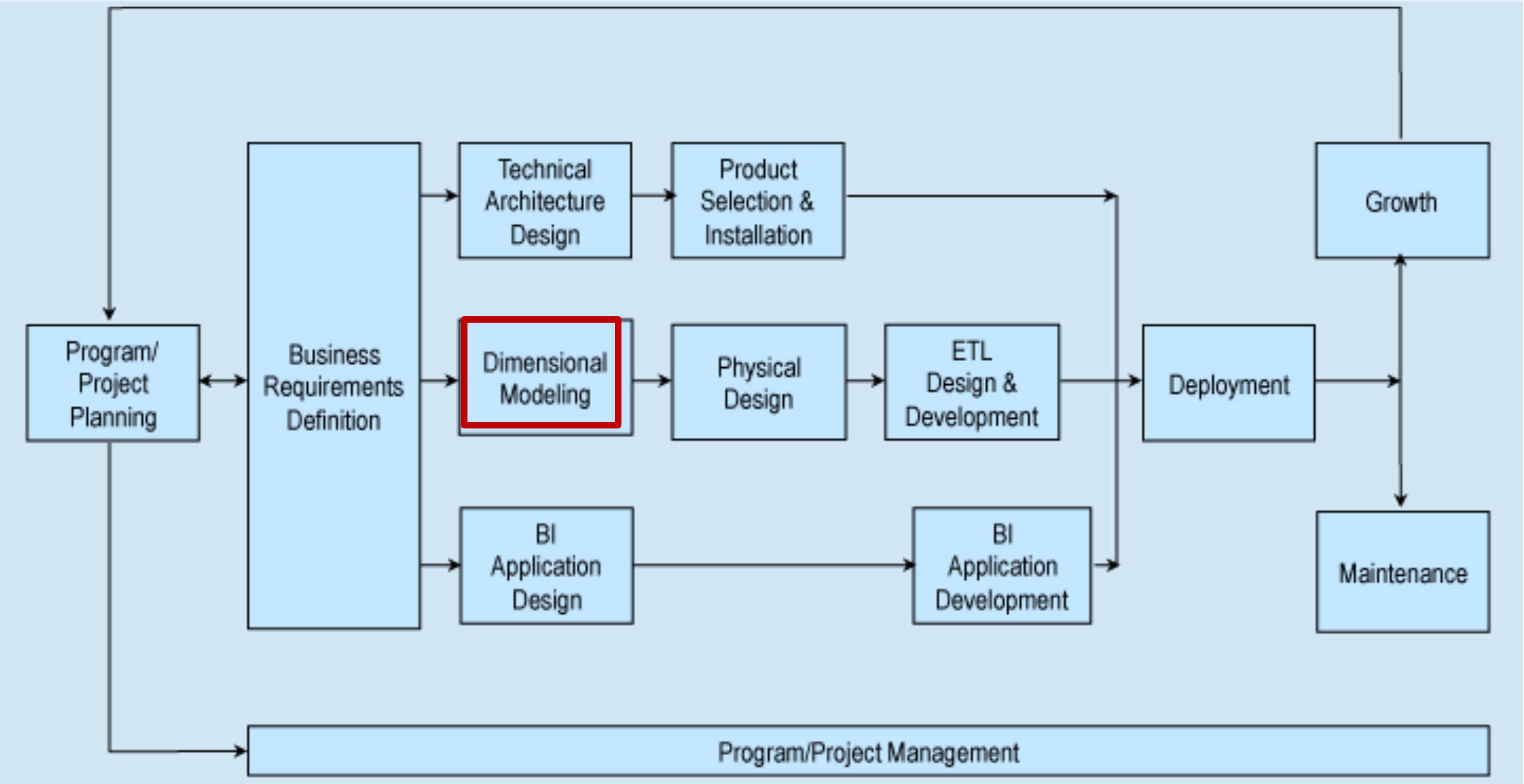
- Dimensional modeling

We'll learn how to

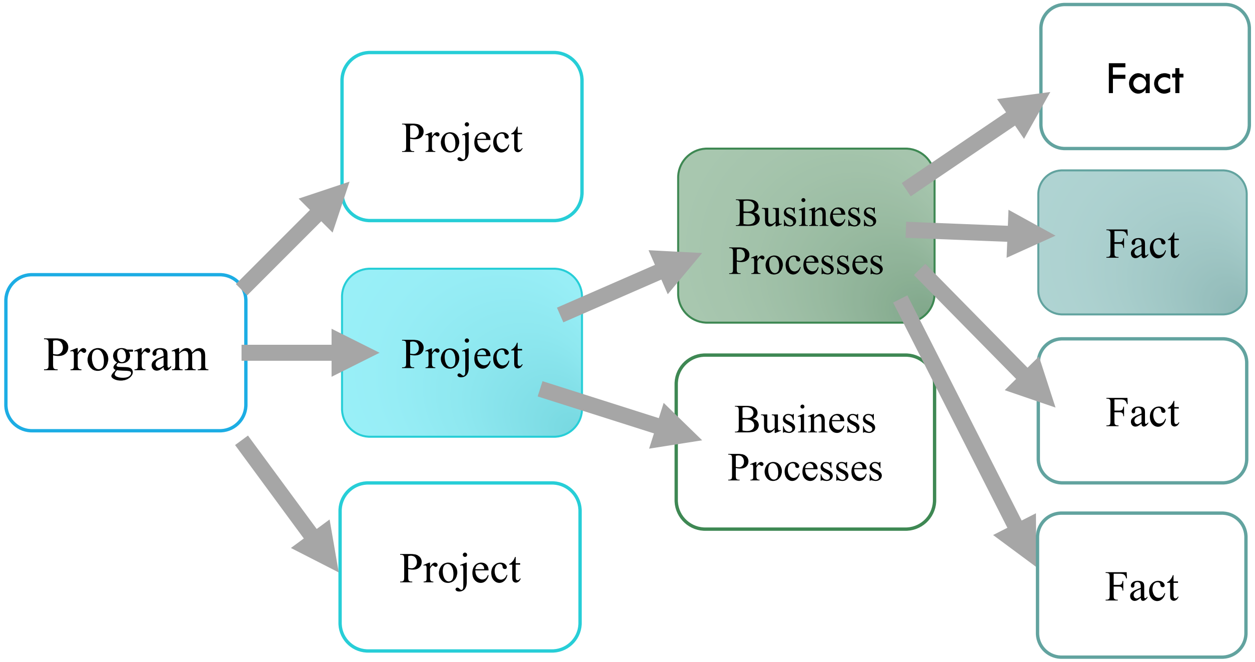
- Design and implement dimensional models in relational databases (ROLAP)

Detailed

Kimball Lifecycle


Program/ Project Planning
Business Requirements Definition
Technical Architecture Design
Product Selection & Installation
Dimensional Modeling
Physical Design
ETL Design & Development
BI Application Design
BI Application Development
Deployment
Growth
Maintenance
Program/Project Management

Kimball Method



Example: Northwind DW Program | Sales | Monthly Sales Snapshot | Amount Sold

Terminology Translator: Requirements Analysis vs. Design and Implementation

Requirements Analysis		Design and Implementation
Business Process	→	Fact Table
Fact	→	Column in Fact Table
Dimension	→	Dimension Table
Dimension Model	→	Star Schema
Business Processes “Uses” a Dimension	→	Foreign Key

Essentials

School of Information Studies
Syracuse University

Dimensional Model Design

Now that you have dimensional models, it is time to focus on how to build the relational structures to support them.

Dimensional Modeling

A logical design technique for structuring data with the following objectives:

1. **Intuitive**: easy for business users to understand
2. **Fast**: excellent query performance

- ✓ Think of a dimensional model as a fact table + the dimensions it requires.
- ✓ Dimensional models are implemented in the relational DBMS as star schemas (ROLAP). They exist in MOLAP databases as cubes.

(Also called a "data mart.")

Components of the Dimensional Model

Fact table: A database table of quantifiable performance measurements (**facts**), originating from an OLTP system's business processes. Has FK's to each of the dimensions.

- **E.g.,** sales amount, days to ship, quantity on hand

Dimension table: a table of contexts for the facts.

- **E.g.,** date/time, location, customer, product

Attribute: a characteristic of a dimension.

- **E.g.,** product: name, category, department

Star schema: connections among facts and dimensions that define a business process.

- **E.g.,** sales, inventory management

Star Schema: Relational Answer to the DM

The star schema

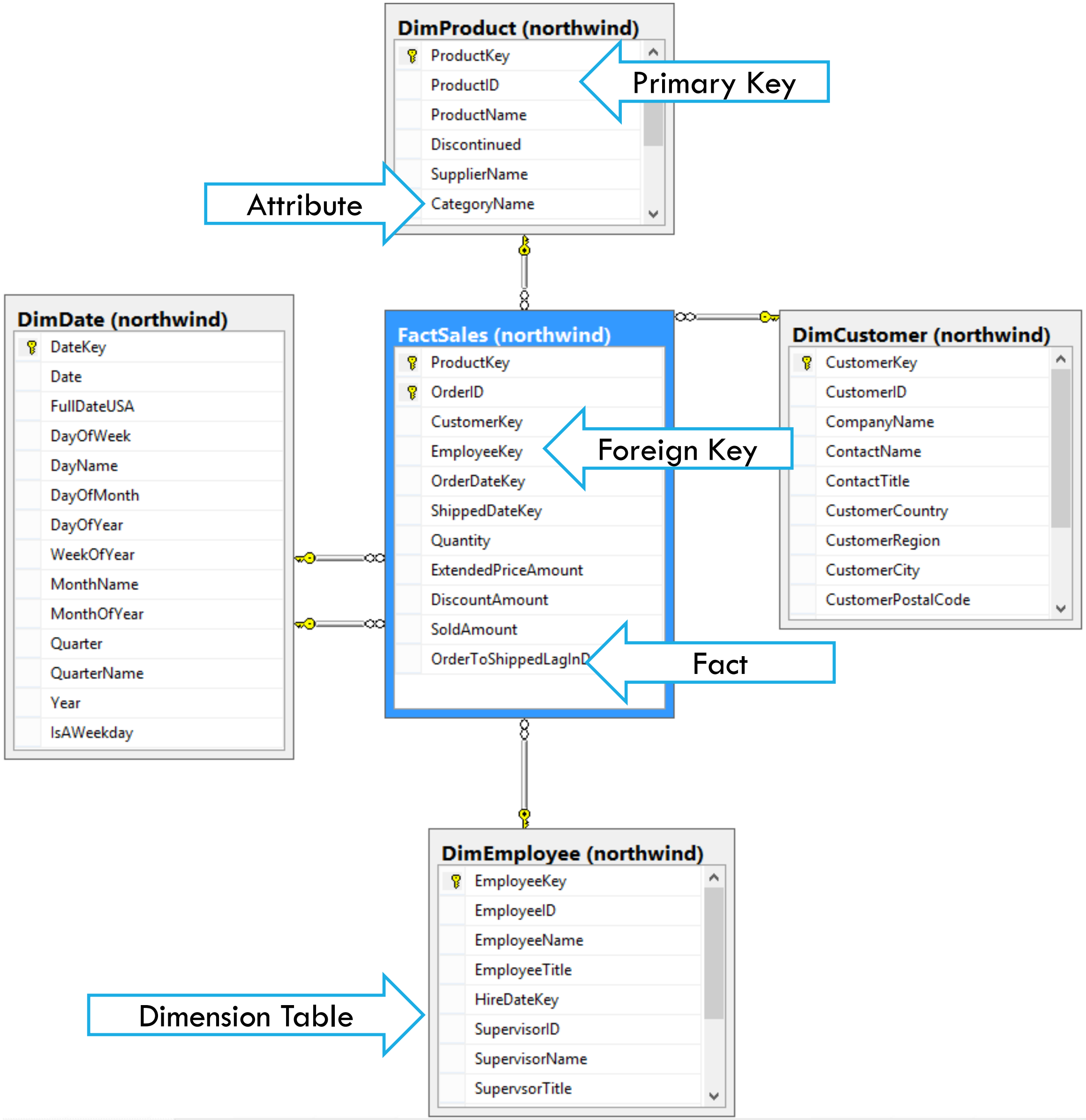
is a relational database

implementation

of a

dimensional

model.



DimProduct (northwind)
- ProductKey
- ProductID
- ProductName
- Discontinued
- SupplierName
- CategoryName

DimDate (northwind)
- DateKey
- Date
- FullDateUSA
- DayOfWeek
- DayName
- DayOfMonth
- DayOfYear
- WeekOfYear
- MonthName
- MonthOfYear
- Quarter
- QuarterName
- Year
- IsAWeekday

FactSales (northwind)
- ProductKey
- OrderID
- CustomerKey
- EmployeeKey
- OrderDateKey
- ShippedDateKey
- Quantity
- ExtendedPriceAmount
- DiscountAmount
- SoldAmount
- OrderToShippedLagInD

DimCustomer (northwind)
- CustomerKey
- CustomerID
- CompanyName
- ContactName
- ContactTitle
- CustomerCountry
- CustomerRegion
- CustomerCity
- CustomerPostalCode

DimEmployee (northwind)
- EmployeeKey
- EmployeeID
- EmployeeName
- EmployeeTitle
- HireDateKey
- SupervisorID
- SupervisorName
- SupervsorTitle

Rules of Fact Table Design

School of Information Studies
Syracuse University

Rules of Fact Table Design

1. The **primary key** of your fact table uses the minimum number columns possible **and no surrogate keys**.
   (It should be made up of FKs and degenerate dimensions.)
2. **Referential integrity** is a must. Every **foreign key** in the fact table must have a value.
3. Avoid NULLs in the foreign key by using **flags**, which are special values in place of NULL.
   E.g., "no shopper card" in customer dimension
4. The **granularity** of your fact table should be at the lowest, most detailed atomic grain captured by the business process. **When in doubt, consult the source system.**
5. Each **fact** should be additive across all dimensions, or re-designed to be as additive as possible.
6. Each **fact** must be of the of the same **granularity**. **One row in the fact table means ________?**

Solution: Fact Table Design

School of Information Studies
Syracuse University

Solution: Fact Table Design

Poor choice of FK

Fact not at table grain

Stat Key **(PK)**	Player ID	Game ID	Shot Attempts	Shots Made	Points	Pts This Season	Shooting Pct
1	Jordan	1	3	2	5	5	0.667
2	Jordan	2	7	6	12	17	0.583
3	Miller	1	2	1	2	2	0.500
4	Miller	2	5	3	9	11	0.600
5	Miller	1	2	1	2	13	0.500

Poor selection of PK: 3 and 5 are the same entity.

Non-additive fact

Factless Fact Tables

Business processes that do not generate quantifiable measurements

- **E.g.,** student attendance, college admissions

Can be easily converted into traditional fact tables by adding an attribute, Count, which is always equal to 1

Consider adding facts for when the event did not happen

- Helps to perform aggregations
  - **E.g.,** attendance percent present or absent versus class size

Rules of Dimension Table Design

School of Information Studies
Syracuse University

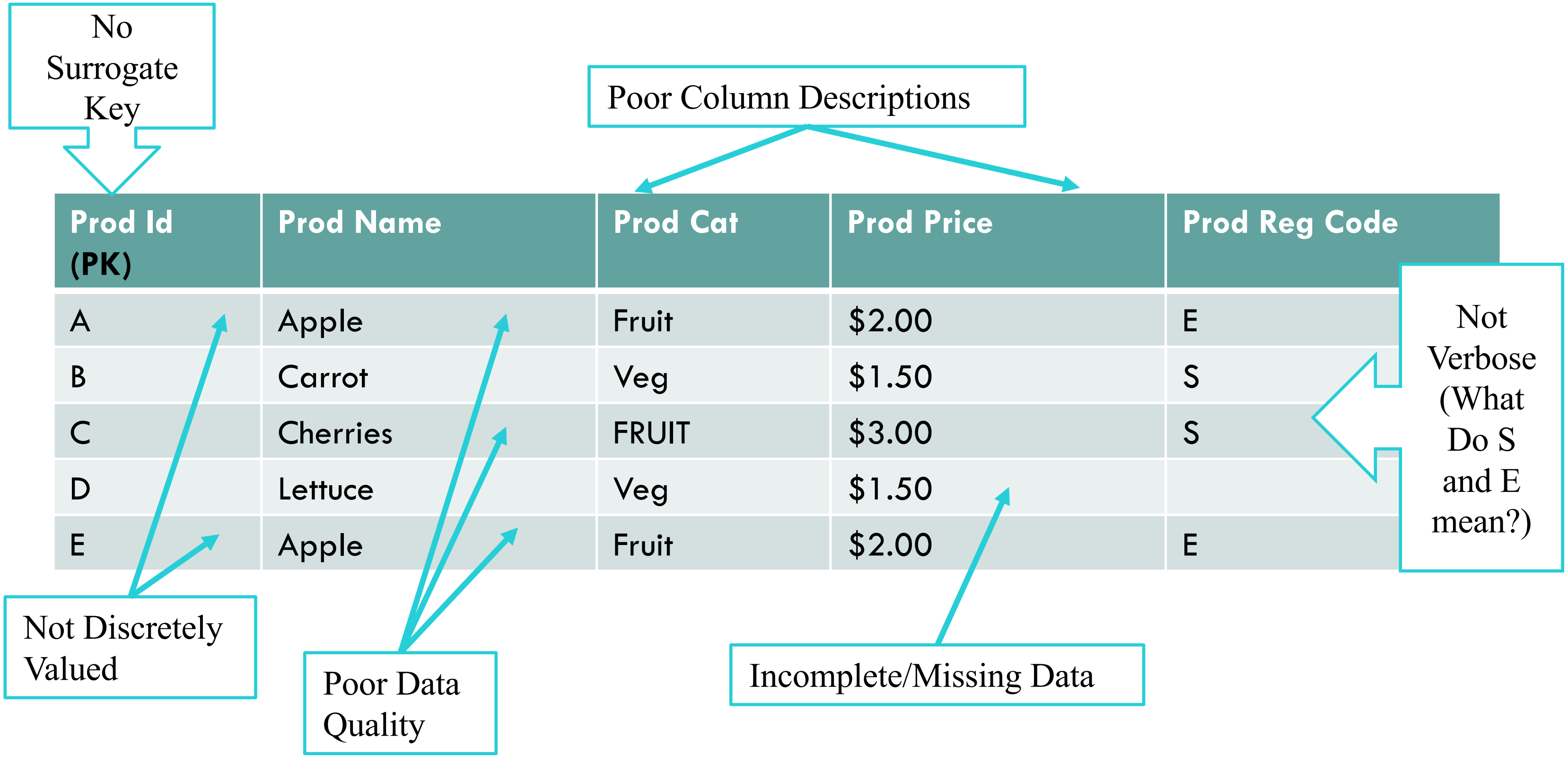
Rules of Dimension Table Design

1. **Verbose attribute values:** Should be as descriptive as possible.
2. **Descriptive columns:** Should be easy to tell what the column means.
3. **Complete:** No NULL/empty values in any of the attributes.
4. **Discretely valued:** One business entity value per row.
5. **Quality Assured:** Data are clean and consistent.
6. Should always contain a **business/natural key**, or **legacy PK** from source system.
7. Always have a **surrogate primary key.** You do not introduce a dependency on an external key.

Solution: Dimension Table Design

School of Information Studies
Syracuse University

What's Wrong With This Dimension?



Prod Id (PK)	Prod Name	Prod Cat	Prod Price	Prod Reg Code
A	Apple	Fruit	$2.00	E
B	Carrot	Veg	$1.50	S
C	Cherries	FRUIT	$3.00	S
D	Lettuce	Veg	$1.50	
E	Apple	Fruit	$2.00	E

No Surrogate Key

Poor Column Descriptions

Not Verbose (What Do S and E mean?)

Not Discretely Valued

Poor Data Quality

Incomplete/Missing Data

The Dimension Table Key

- **Surrogate keys** (identities, sequences, e.g., 1,2,3,…) are used for the **primary key constraint**.
- They yield best performance for the star schema.
  - Integers make for efficient joins.
  - Smaller indexes in fact table.
  - More rows per block in the fact table.
- They have no dependency on primary key in operational source data.
  - Makes it easier to deal with changes to the source data than reflect time variance in the data warehouse.
- Dimension table requires a **natural key or business key** to identify a unique row.
  - Incoming facts must be matched to the row in the dimension. This is called the **surrogate key pipeline.**

Introduction

School of Information Studies
Syracuse University

Slowly Changing Dimensions

Dimensional data changes **infrequently**, but when it does you need a strategy for addressing the change.

- **E.g.,** what happens when a customer has a new address or an employee has a name change?

Three Popular Strategies

- ✓ Type 1: Overwrite the existing attribute
- ✓ Type 2: Add a new dimension row
- ✓ Type 3: Add a new dimension attribute

These strategies are not mutually exclusive and can be combined within a single dimension.

Type 1

School of Information Studies
Syracuse University

Type 1: Overwrite

Appropriate for:

- Correcting mistakes or errors in data.
- Changes where historical associations do not matter.
- The old value has no significance.

Not appropriate for:

- Preserving point-in-time history. If the previous value matters, **don't use this strategy**. You are **rewriting history**.

E.g., employee name changes, corrections to data quality, natural key edits such as an e-mail address

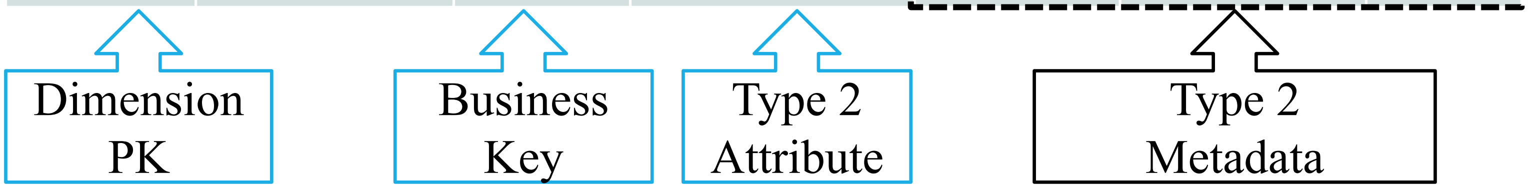
Type 2

School of Information Studies
Syracuse University

Type 2: Add New Dimension Row

- Most popular strategy, as it preserves history.
- Natural/business key is repeated.
- Old and new values are stored along with effective dates and indicator of which row is “current.”

Product Key (PK)	Product Description	Product Code	Department	Effective Date	Expiration Date	Curren t Row
11981	Stapler, Red	ST901	Accessories	4/7/2010	9/1/2011	N
20342	Stapler, Red	ST901	Supplies	9/2/2011	3/31/2013	N
45393	Stapler, Red	ST901	Office Supplies	4/1/2013	12/31/9999	Y


Dimension PK

Business Key

Type 2 Attribute

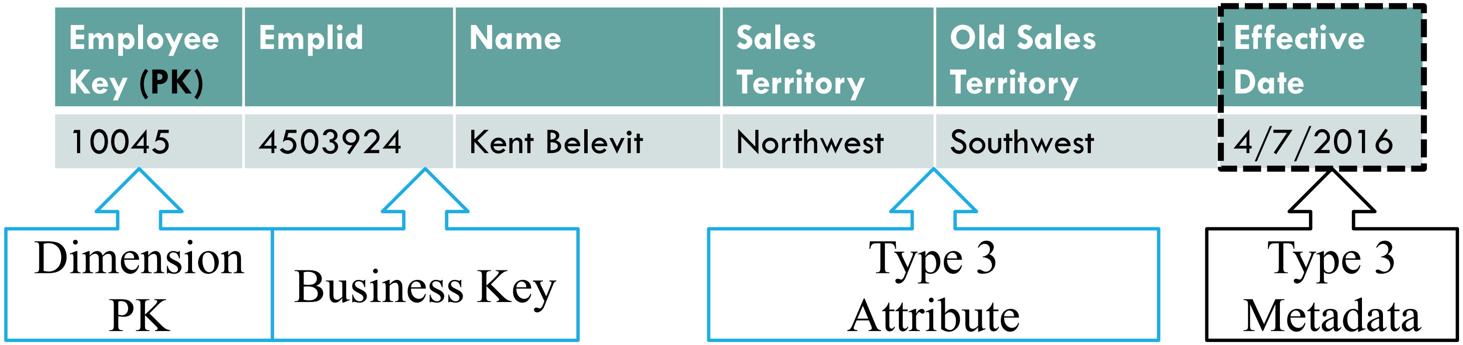
Type 2 Metadata

Type 3

School of Information Studies
Syracuse University

Type 3: Add a New Dimension Attribute

- Infrequently used, preserves history.
- Useful for **soft changes** where users might want to choose between the old and new attribute or need to access both values for a time.
- The **new value** is written to the **existing column**, and the **old value** is stored in a **new column**, so queries do not have to be rewritten.
  - **E.g.,** redistricting sales territories, recharting accounting codes



Employee Key **(PK)**	Emplid	Name	Sales Territory	Old Sales Territory	Effective Date
10045	4503924	Kent Belevit	Northwest	Southwest	4/7/2016

Dimension PK

Business Key

Type 3 Attribute

Type 3 Metadata

How Rapid Is "Rapid"?

School of Information Studies
Syracuse University

How Rapid Is "Rapid"?

- Rapid is loosely defined as **"not often and with no consistency."**
- A customer street address?
  - How often do you move?
  - Not often and with no consistency
  - Slowly changing dimension
- Your age in years?
  - Age changes annually… not often.
  - Age changes with consistency… updates for all rows in the dimension.
  - Rapidly changing dimension.
- Customer shirt color at the time of product purchase?
  - Changes often… we wear different shirts all the time!
  - Not consistent.
  - Rapidly changing dimension.

Attributes tied to the business process are always RCDs.

Degenerate Dimensions

School of Information Studies
Syracuse University

Degenerate Dimensions

- **Degenerate dimensions** are dimension attributes we store in the **fact table**, because:
  - They change too frequently to remain in their own dimension, or
  - There's too many of them for their own a dimension.
    Examples: order number, flight number, customer age, product quantity on hand
- Some degenerate dimensions are business keys. They allow us to **drill through** to operational data, in the ODS. They usually end up as part of the **primary key** of the fact table.
- **Any attribute in the fact table that is not a dimension key or fact is considered a degenerate dimension.**

Example: Degenerate Dimensions

Order Number (PK)	Product Key (PK, FK)	Order Date Key (FK)	Order Qty	Order Total	Product Qty On Hand
11005	99001	20170101	1	$15.99	5
11005	99002	20170101	1	$0.99	12
11006	99001	20170105	10	$159.90	2
11006	99002	20170105	5	$4.95	3

Several business processes (orders, resupply, returns) change this value so we write it to the fact table at the time of event.

This is a business key, not a fact or dimension key.

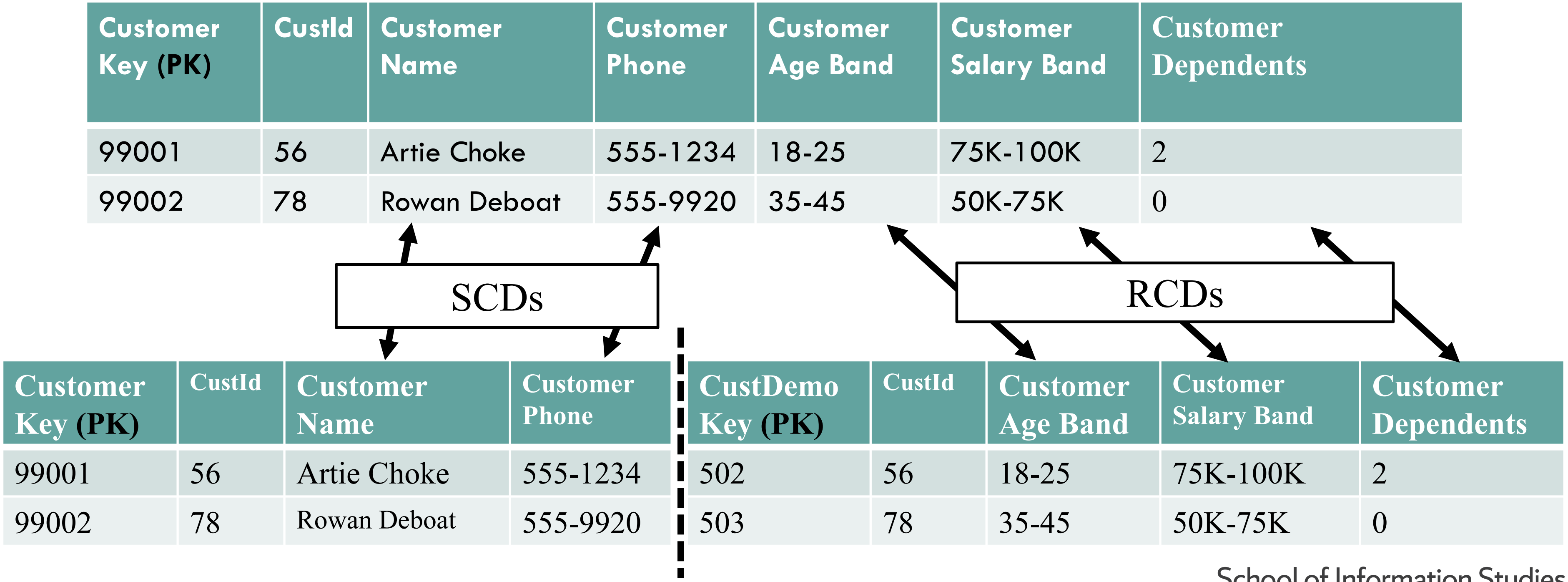
Product Key (PK)	Product Description	Product Code	Department	Product Qty On Hand
99001	Stapler, Red	ST901	Accessories	??
99002	#2 Pencil	PN902	Supplies	??

Mini-dimensions

School of Information Studies
Syracuse University

Mini-dimensions

- If attributes change frequently, consider placing them in their own "mini-dimensions."
- Most effective when you have **banded values that change over time**, demographic survey data, or ranges of discrete values.



Customer Key (PK)	CustId	Customer Name	Customer Phone	Customer Age Band	Customer Salary Band	Customer Dependents
99001	56	Artie Choke	555-1234	18-25	75K-100K	2
99002	78	Rowan Deboat	555-9920	35-45	50K-75K	0

SCDs

RCDs

Customer Key (PK)	CustId	Customer Name	Customer Phone
99001	56	Artie Choke	555-1234
99002	78	Rowan Deboat	555-9920

CustDemo Key (PK)	CustId	Customer Age Band	Customer Salary Band	Customer Dependents
502	56	18-25	75K-100K	2
503	78	35-45	50K-75K	0

Conformed Dimensions

School of Information Studies
Syracuse University

Conformed Dimensions

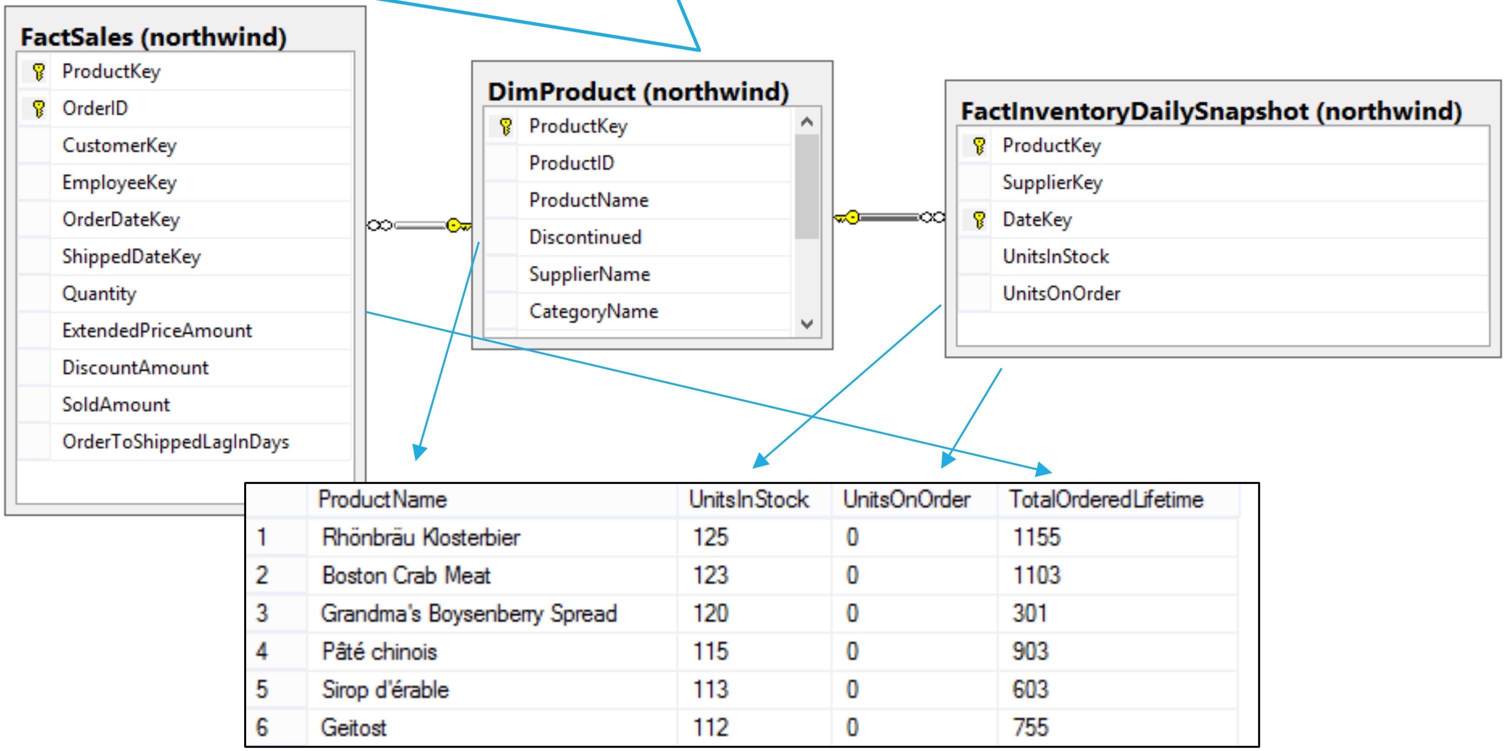
- These are master or common reference dimensions and a key part of the enterprise bus technical architecture.
- Shared across business processes (fact tables) in the DW.
- Reusable, can be used for **drill across**, where you combine facts across a common dimension.
- Lower time to develop next star schema.
- Contain a superset of attributes required by all fact tables.

Two types of conformed dimensions:

- Identical dimensions: exactly the same dimensions (e.g., dates)
- Perfect subset of an existing dimension

Example: Conformed Dimension and Drill Across

Single **DimProduct** dimension table is used in two fact table star schemas. This is enterprise bus technical architecture!



FactSales (northwind)

- ProductKey
- OrderID
- CustomerKey
- EmployeeKey
- OrderDateKey
- ShippedDateKey
- Quantity
- ExtendedPriceAmount
- DiscountAmount
- SoldAmount
- OrderToShippedLagInDays

DimProduct (northwind)

- ProductKey
- ProductID
- ProductName
- Discontinued
- SupplierName
- CategoryName

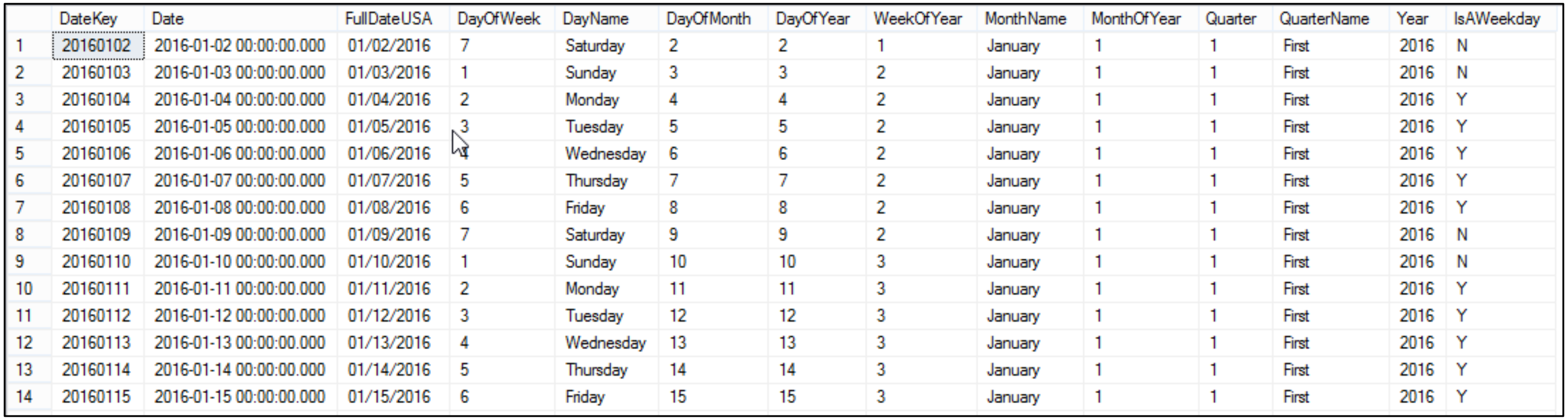
FactInventoryDailySnapshot (northwind)

- ProductKey
- SupplierKey
- DateKey
- UnitsInStock
- UnitsOnOrder

	ProductName	UnitsInStock	UnitsOnOrder	TotalOrderedLifetime
1	Rhönbräu Klosterbier	125	0	1155
2	Boston Crab Meat	123	0	1103
3	Grandma's Boysenberry Spread	120	0	301
4	Pâté chinois	115	0	903
5	Sirop d'érable	113	0	603
6	Geitost	112	0	755

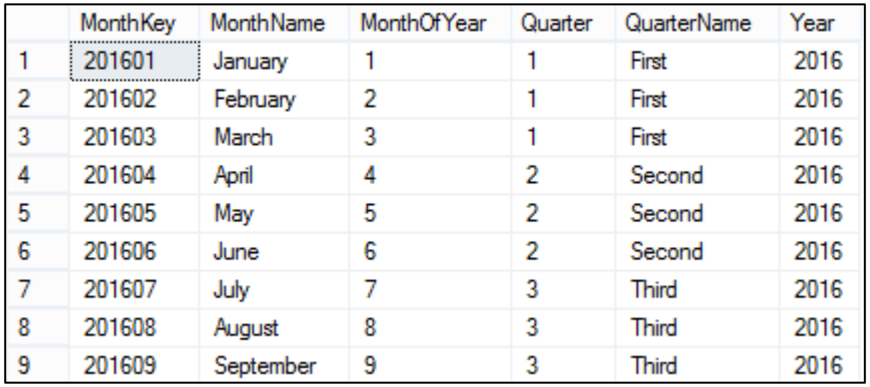
Example: Conformed Subset

DimDate



	DateKey	Date	FullDateUSA	DayOfWeek	DayName	DayOfMonth	DayOfYear	WeekOfYear	MonthName	MonthOfYear	Quarter	QuarterName	Year	IsAWeekday
1	20160102	2016-01-02 00:00:00.000	01/02/2016	7	Saturday	2	2	1	January	1	1	First	2016	N
2	20160103	2016-01-03 00:00:00.000	01/03/2016	1	Sunday	3	3	2	January	1	1	First	2016	N
3	20160104	2016-01-04 00:00:00.000	01/04/2016	2	Monday	4	4	2	January	1	1	First	2016	Y
4	20160105	2016-01-05 00:00:00.000	01/05/2016	3	Tuesday	5	5	2	January	1	1	First	2016	Y
5	20160106	2016-01-06 00:00:00.000	01/06/2016	4	Wednesday	6	6	2	January	1	1	First	2016	Y
6	20160107	2016-01-07 00:00:00.000	01/07/2016	5	Thursday	7	7	2	January	1	1	First	2016	Y
7	20160108	2016-01-08 00:00:00.000	01/08/2016	6	Friday	8	8	2	January	1	1	First	2016	Y
8	20160109	2016-01-09 00:00:00.000	01/09/2016	7	Saturday	9	9	2	January	1	1	First	2016	N
9	20160110	2016-01-10 00:00:00.000	01/10/2016	1	Sunday	10	10	3	January	1	1	First	2016	N
10	20160111	2016-01-11 00:00:00.000	01/11/2016	2	Monday	11	11	3	January	1	1	First	2016	Y
11	20160112	2016-01-12 00:00:00.000	01/12/2016	3	Tuesday	12	12	3	January	1	1	First	2016	Y
12	20160113	2016-01-13 00:00:00.000	01/13/2016	4	Wednesday	13	13	3	January	1	1	First	2016	Y
13	20160114	2016-01-14 00:00:00.000	01/14/2016	5	Thursday	14	14	3	January	1	1	First	2016	Y
14	20160115	2016-01-15 00:00:00.000	01/15/2016	6	Friday	15	15	3	January	1	1	First	2016	Y

DimMonth



	MonthKey	MonthName	MonthOfYear	Quarter	QuarterName	Year
1	201601	January	1	1	First	2016
2	201602	February	2	1	First	2016
3	201603	March	3	1	First	2016
4	201604	April	4	2	Second	2016
5	201605	May	5	2	Second	2016
6	201606	June	6	2	Second	2016
7	201607	July	7	3	Third	2016
8	201608	August	8	3	Third	2016
9	201609	September	9	3	Third	2016

DimDate → One row per day
DimMonth → One row per month/year

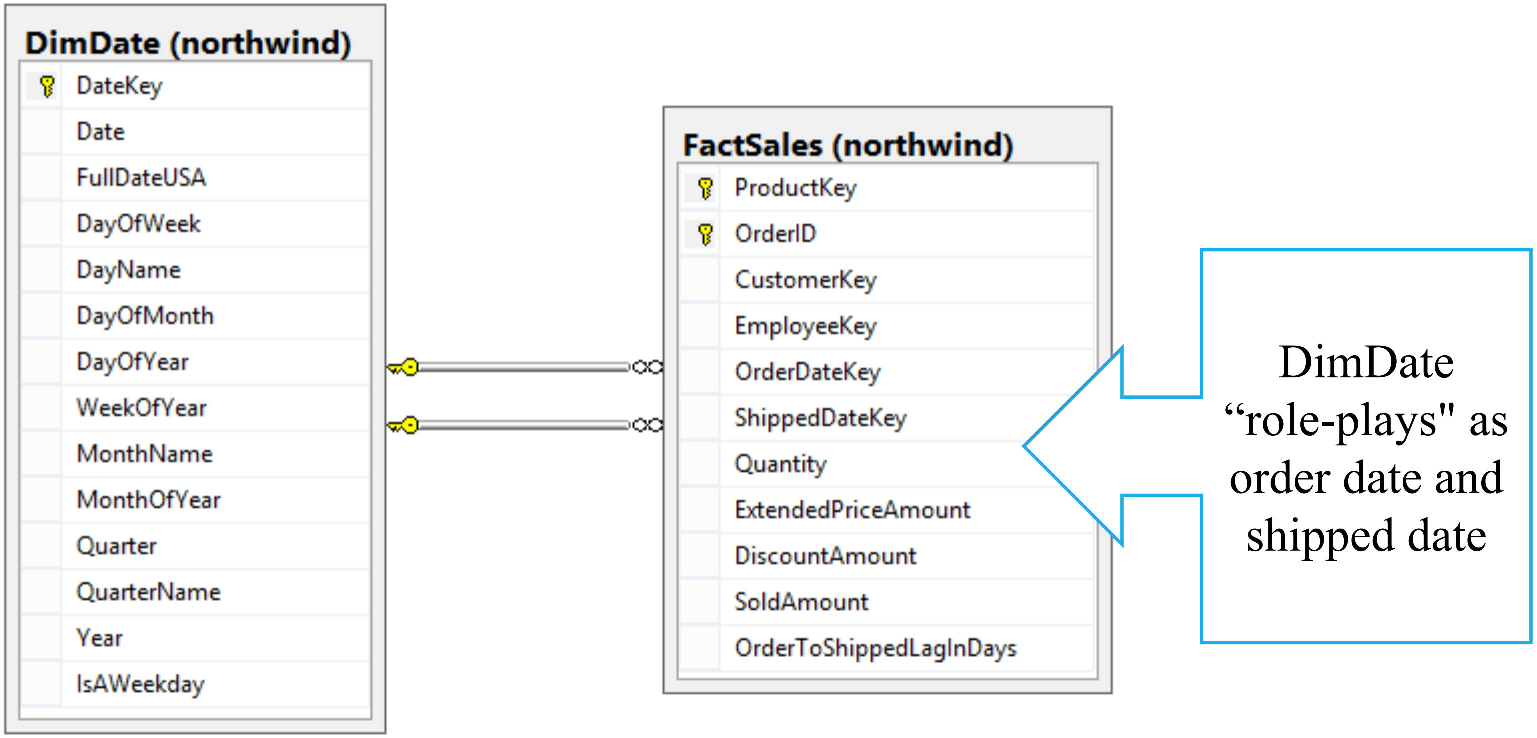
Role-Playing Dimensions

School of Information Studies
Syracuse University

Role-Playing Dimensions

- The same physical dimension plays more than one logical dimensional **role**.
- This is common among the date dimension.
- Stored in the same physical table, just aliased as a view.
- Implemented as multiple FKs in the fact table to the same dimension table.
- Examples:
  - **Date:** order date, shipping date, delivery date ➔ same date
  - **Address:** ship to, bill to ➔ same address dimension
  - **Airport:** arrival, departure ➔ same airport dimension

Example: Role-Playing Dimension



DimDate (northwind)

- DateKey
- Date
- FullDateUSA
- DayOfWeek
- DayName
- DayOfMonth
- DayOfYear
- WeekOfYear
- MonthName
- MonthOfYear
- Quarter
- QuarterName
- Year
- IsAWeekday

FactSales (northwind)

- ProductKey
- OrderID
- CustomerKey
- EmployeeKey
- OrderDateKey
- ShippedDateKey
- Quantity
- ExtendedPriceAmount
- DiscountAmount
- SoldAmount
- OrderToShippedLagInDays

DimDate “role-plays" as order date and shipped date

Date and Time Dimension

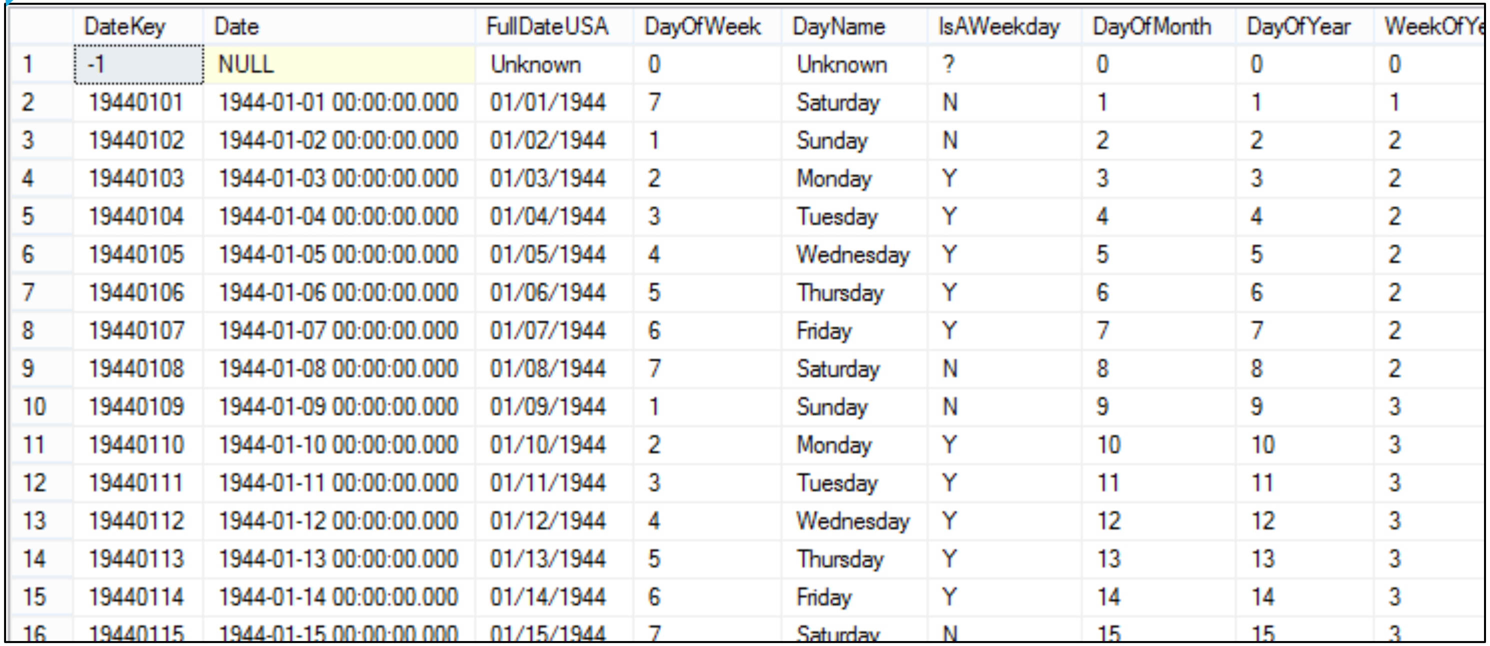
School of Information Studies
Syracuse University

Date and Time Dimensions

- Just about every fact table has a date and/or time dimension.
- This is the most common of **conformed dimensions** and are often **role-playing** in several fact tables.
- Usually **generated programmatically** during the ETL process or **imported** from a spreadsheet.
- Acceptable to use PK in the form YYYYMMDD.
- If you need time of day, use a separate dimension for time.
- Time of day should be used only if there are **meaningful textual descriptions** of time.
  - **E.g.,** lunch, dinner, first shift, second shift, etc.
- Elapsed times intervals are **facts**, not attributes.
  - **E.g.,** minutes between when order was received and shipped

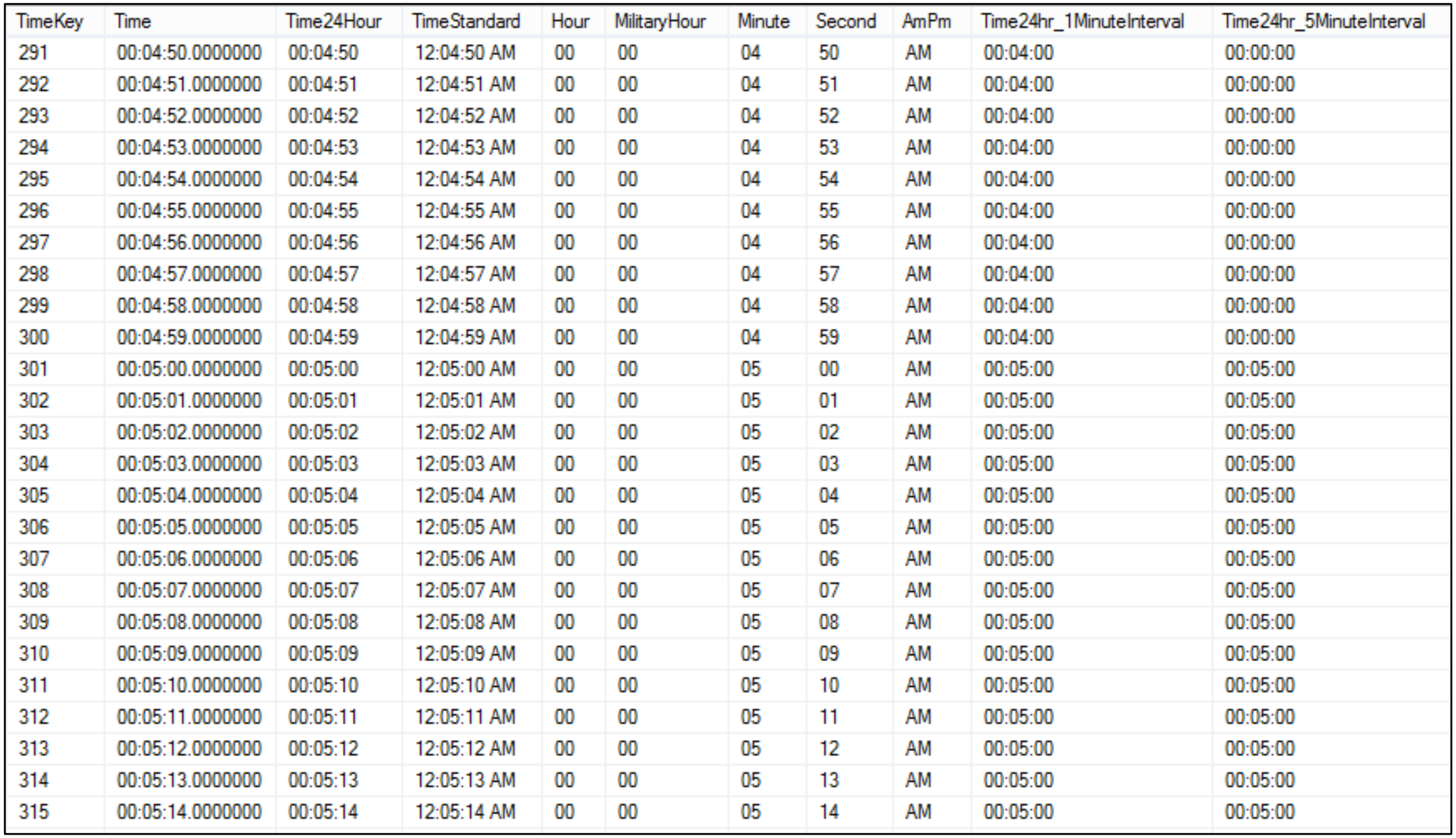
Example: Date Dimension

Unknown member



	DateKey	Date	FullDateUSA	DayOfWeek	DayName	IsAWeekday	DayOfMonth	DayOfYear	WeekOfYe
1	-1	NULL	Unknown	0	Unknown	?	0	0	0
2	19440101	1944-01-01 00:00:00.000	01/01/1944	7	Saturday	N	1	1	1
3	19440102	1944-01-02 00:00:00.000	01/02/1944	1	Sunday	N	2	2	2
4	19440103	1944-01-03 00:00:00.000	01/03/1944	2	Monday	Y	3	3	2
5	19440104	1944-01-04 00:00:00.000	01/04/1944	3	Tuesday	Y	4	4	2
6	19440105	1944-01-05 00:00:00.000	01/05/1944	4	Wednesday	Y	5	5	2
7	19440106	1944-01-06 00:00:00.000	01/06/1944	5	Thursday	Y	6	6	2
8	19440107	1944-01-07 00:00:00.000	01/07/1944	6	Friday	Y	7	7	2
9	19440108	1944-01-08 00:00:00.000	01/08/1944	7	Saturday	N	8	8	2
10	19440109	1944-01-09 00:00:00.000	01/09/1944	1	Sunday	N	9	9	3
11	19440110	1944-01-10 00:00:00.000	01/10/1944	2	Monday	Y	10	10	3
12	19440111	1944-01-11 00:00:00.000	01/11/1944	3	Tuesday	Y	11	11	3
13	19440112	1944-01-12 00:00:00.000	01/12/1944	4	Wednesday	Y	12	12	3
14	19440113	1944-01-13 00:00:00.000	01/13/1944	5	Thursday	Y	13	13	3
15	19440114	1944-01-14 00:00:00.000	01/14/1944	6	Friday	Y	14	14	3
16	19440115	1944-01-15 00:00:00.000	01/15/1944	7	Saturday	N	15	15	3

Example: Time Dimension



TimeKey	Time	Time24Hour	TimeStandard	Hour	MilitaryHour	Minute	Second	AmPm	Time24hr_1MinuteInterval	Time24hr_5MinuteInterval
291	00:04:50.0000000	00:04:50	12:04:50 AM	00	00	04	50	AM	00:04:00	00:00:00
292	00:04:51.0000000	00:04:51	12:04:51 AM	00	00	04	51	AM	00:04:00	00:00:00
293	00:04:52.0000000	00:04:52	12:04:52 AM	00	00	04	52	AM	00:04:00	00:00:00
294	00:04:53.0000000	00:04:53	12:04:53 AM	00	00	04	53	AM	00:04:00	00:00:00
295	00:04:54.0000000	00:04:54	12:04:54 AM	00	00	04	54	AM	00:04:00	00:00:00
296	00:04:55.0000000	00:04:55	12:04:55 AM	00	00	04	55	AM	00:04:00	00:00:00
297	00:04:56.0000000	00:04:56	12:04:56 AM	00	00	04	56	AM	00:04:00	00:00:00
298	00:04:57.0000000	00:04:57	12:04:57 AM	00	00	04	57	AM	00:04:00	00:00:00
299	00:04:58.0000000	00:04:58	12:04:58 AM	00	00	04	58	AM	00:04:00	00:00:00
300	00:04:59.0000000	00:04:59	12:04:59 AM	00	00	04	59	AM	00:04:00	00:00:00
301	00:05:00.0000000	00:05:00	12:05:00 AM	00	00	05	00	AM	00:05:00	00:05:00
302	00:05:01.0000000	00:05:01	12:05:01 AM	00	00	05	01	AM	00:05:00	00:05:00
303	00:05:02.0000000	00:05:02	12:05:02 AM	00	00	05	02	AM	00:05:00	00:05:00
304	00:05:03.0000000	00:05:03	12:05:03 AM	00	00	05	03	AM	00:05:00	00:05:00
305	00:05:04.0000000	00:05:04	12:05:04 AM	00	00	05	04	AM	00:05:00	00:05:00
306	00:05:05.0000000	00:05:05	12:05:05 AM	00	00	05	05	AM	00:05:00	00:05:00
307	00:05:06.0000000	00:05:06	12:05:06 AM	00	00	05	06	AM	00:05:00	00:05:00
308	00:05:07.0000000	00:05:07	12:05:07 AM	00	00	05	07	AM	00:05:00	00:05:00
309	00:05:08.0000000	00:05:08	12:05:08 AM	00	00	05	08	AM	00:05:00	00:05:00
310	00:05:09.0000000	00:05:09	12:05:09 AM	00	00	05	09	AM	00:05:00	00:05:00
311	00:05:10.0000000	00:05:10	12:05:10 AM	00	00	05	10	AM	00:05:00	00:05:00
312	00:05:11.0000000	00:05:11	12:05:11 AM	00	00	05	11	AM	00:05:00	00:05:00
313	00:05:12.0000000	00:05:12	12:05:12 AM	00	00	05	12	AM	00:05:00	00:05:00
314	00:05:13.0000000	00:05:13	12:05:13 AM	00	00	05	13	AM	00:05:00	00:05:00
315	00:05:14.0000000	00:05:14	12:05:14 AM	00	00	05	14	AM	00:05:00	00:05:00

Junk Dimensions

School of Information Studies
Syracuse University

Junk Dimensions

- Miscellaneous low-cardinality flags and text attributes that are **tied to the business process** and do not fit within any other dimension.
- **Do not** make a dimension for each one. This overcomplicates the amount of joins in the star schema.
- Instead place them in their own "Junk" dimension

Terms Key (PK)	Payment Terms
1	Net 10
2	Net 15

Mode Key (PK)	Payment Terms
1	Fax
2	Phone
3	Web

Ship Key (PK)	Ship Mode
1	Air
2	Freight

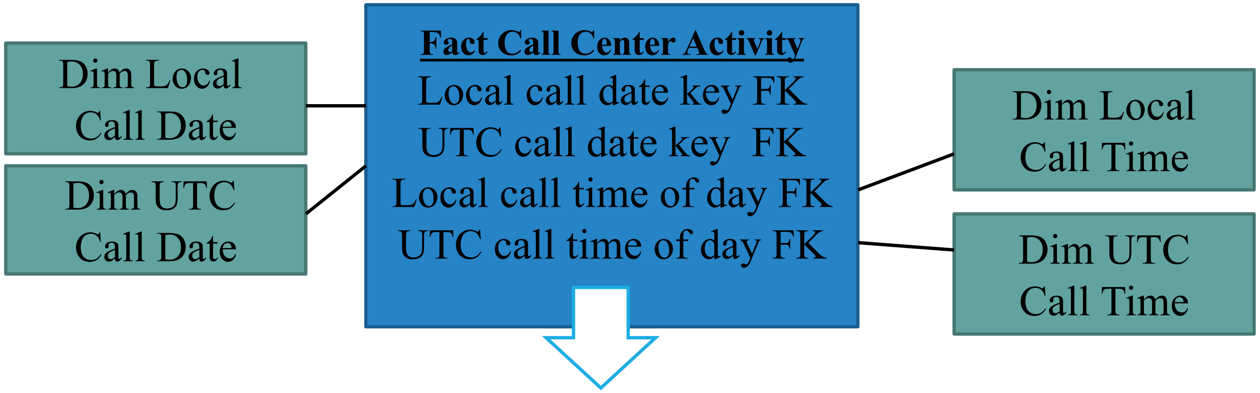
Invoice Junk Key (PK)	Payment Terms	Order Mode	Ship Mode
1	Net 10	Web	Freight
2	Net 10	Web	Air
3	Net 10	Fax	Freight
4	Net 10	Fax	Air
5	Net 10	Phone	Freight
6	Net 10	Phone	Air
7	Net 15	Web	Freight
8	Net 15	Web	Air

Handling Time Zones

School of Information Studies
Syracuse University

How Do You Handle Time Zones?

- Express time in coordinated universal time (UTC).
- Express in local time, too.
- Other options: Use a single time zone (for example, UTC) to express all times in this zone.



Dim Local Call Date

Dim UTC Call Date

Fact Call Center Activity
Local call date key FK
UTC call date key FK
Local call time of day FK
UTC call time of day FK

Dim Local Call Time

Dim UTC Call Time

Call ID (PK)	Local Call Date Key	UTC Call Date Key	Local Call Time Key	UTC Call Time Key
994039	20170404	20170405	09:35	1:35