TOPIC 2 – OLAP Dimensional Model and Data Warehouse Development Processes

OLAP Dimension Model

OLTP vs. OLAP

Online transaction processing systems (OLTP) : Systems that handle a company's daily operation

Online analytic processing (OLAP): Systems that enables the user to query the system and conduct an analysis in seconds

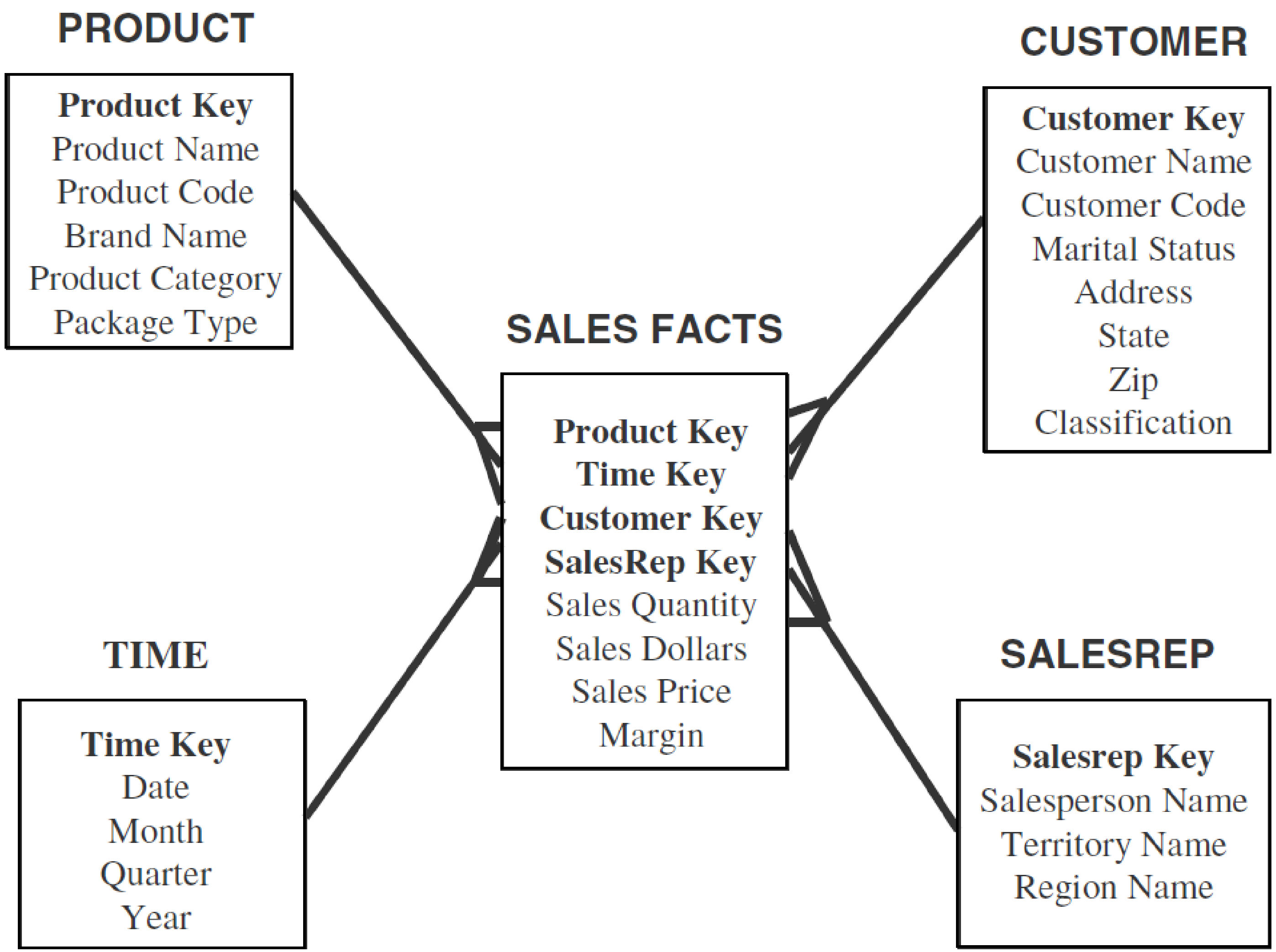
Comparison

CHARACTERISTICS	OLTP SYSTEMS	DATA WAREHOUSE
Analytical capabilities	Very low	Moderate
Data for a single session	Very limited	Small to medium size
Size of result set	Small	Large
Response time	Very fast	Fast to moderate
Data granularity	Detail	Detail and summary
Data currency	Current	Current and historical
Access method	Predefined	Predefined and ad hoc
Basic motivation	Collect and input data	Provide information
Data model	Design for data updates	Design for queries
Optimization of database	For transactions	For analysis
Update frequency	Very frequent	Generally read-only
Scope of user interaction	Single transactions	Throughout data content

Dimensional Model of OLAP

- There are two dimension models – star schema and snowflake schema
- Star schema:
  - A normalized fact table is in the middle, and de-normalized dimensions table connect to fact table
- Snowflake schema:
  - A normalized fact table is in the middle, and normalized dimensions table connect to fact table

Star Schema Model

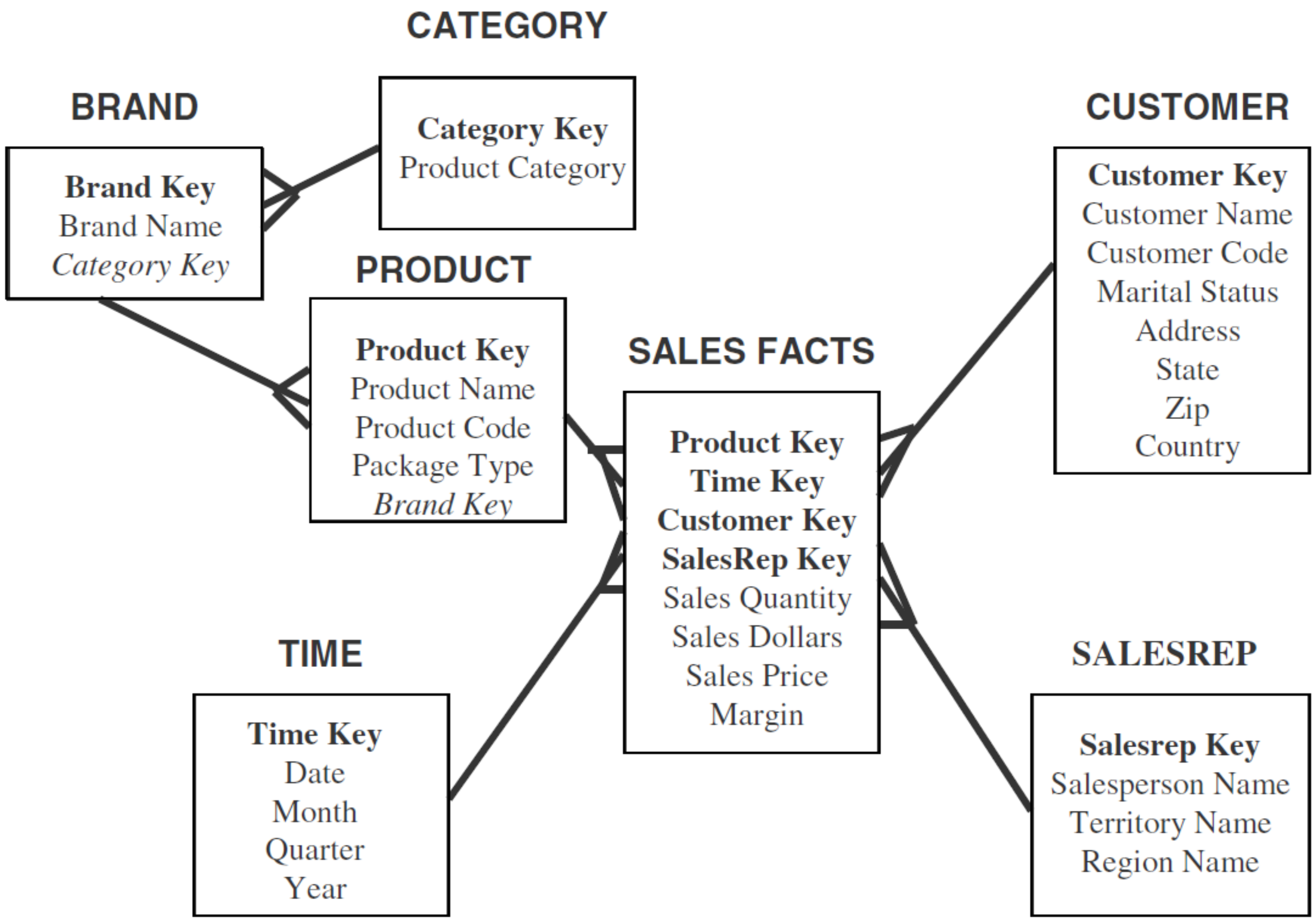


Star Schema Model

- Fast queries due to the reductions in joins required between fact and dimension tables.
- Slow to build because of denormalization in dimension tables

Dominant Schema Model in Data Warehouse!!!

Snowflake Schema Model



Snowflake Schema Model

- Trade off between flexibility and query performance
  - Slower to query but faster to load data
- More complex metadata and schema
  - Normalization in dimensional tables to avoid redundant data

Fact Table Characteristics

- Contain numerical measures for business performance
- Join with dimension tables via foreign keys
- Has composite primary key that consists of all foreign keys.

SALES FACTS

Product Key
Time Key
Customer Key
SalesRep Key
Sales Quantity
Sales Dollars
Sales Price
Margin

10

Fact Measures

A numeric values that reflect organization's performance:

- E.g., sale dollar, count of products etc.
- Associated with a specific business process
- Can be derived or aggregated data

Three types of fact measures:

- Additive: can be summed across all dimensions (sales amounts, sales price etc.)
- Semi-additive: can be summed across some dimensions (balance amount, number of customer bought products etc.)
- Non-additive: cannot be summed at all (ratio, percentage, average etc.)

Fact Table Types

Transaction fact table

- Hold the most detailed level of data

Periodic snapshot fact table

- Snapshot of transaction fact table at specific time

Accumulating snapshot fact table

- Snapshot of business lifecycle
- Constantly updated over time

Transaction Fact Table

Product Key	DateKey	CurrentQuantity
1	1/1/18	100
1	1/2/18	50
2	1/1/18	150
2	1/1/18	20
2	1/2/18	100

Periodic Snapshot Fact Table

ProductKey	DateKey	CurrentQuantity
1	1/1/18	100
1	1/2/18	50
2	1/1/18	170
2	1/2/18	100

Accumulating Snapshot Fact Table

ProductKey	ReceivedDateKey	ShippedDateKey	Quantity
1	1/1/18	1/10/18	100
2	1/1/18	1/21/18	200

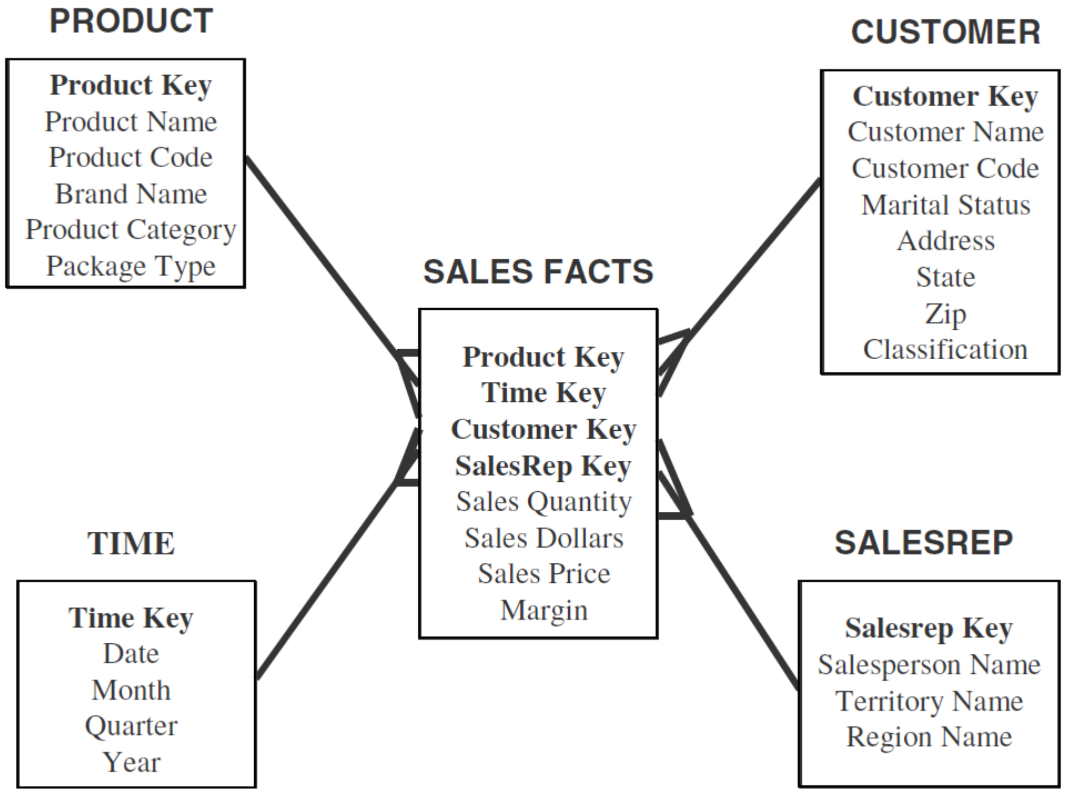
Dimensions

Provide descriptive or contextual information for the measures of a fact table

- E.g., products, customers etc.

Join to fact table with foreign keys

Link to all business processes



16

The Date Dimension

Most crucial data to data warehouse:

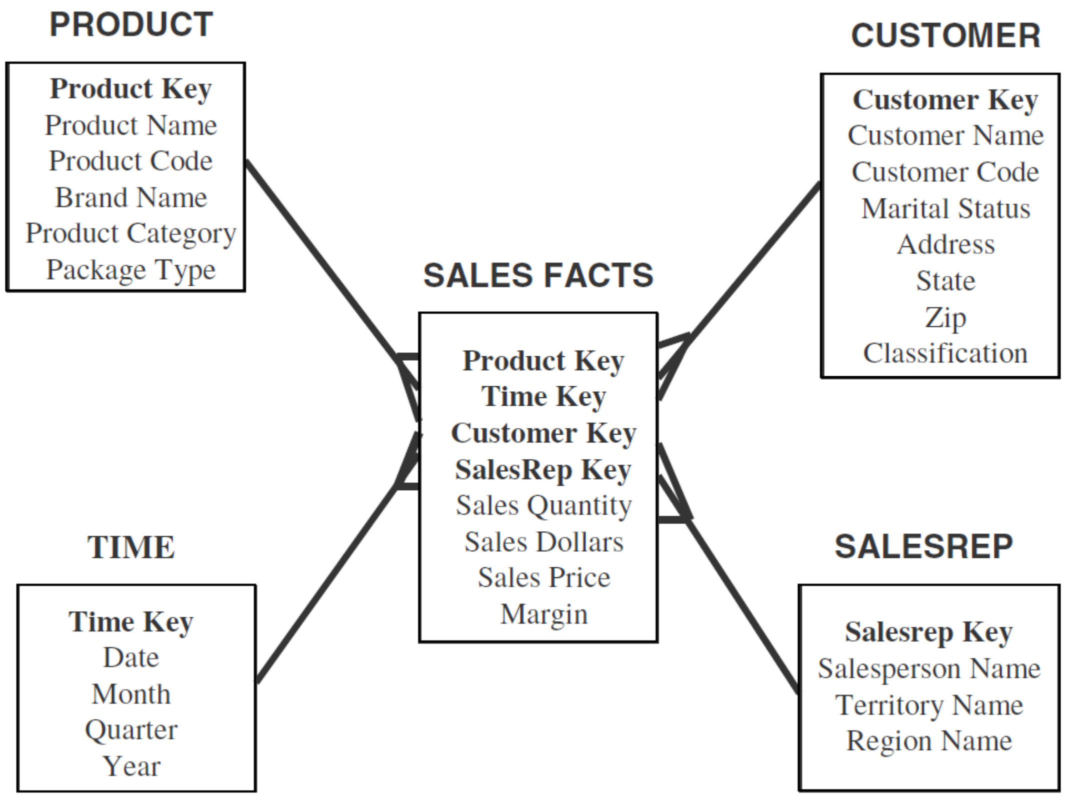
- Typically not available from OLTP
- Must extract from other sources such as transaction files
- Stored as a separated dimension in data warehouse

17

Hierarchies

Provide a way to group data within dimension table

- Address: street, city, state, country
- Time: year, quarter, month, date



Types of Dimension

There are 4 main types of frequently referred dimensions

- Conformed dimensions
- Junk dimensions
- Role playing dimensions
- Slowly changing dimensions (SCD) – Most Important

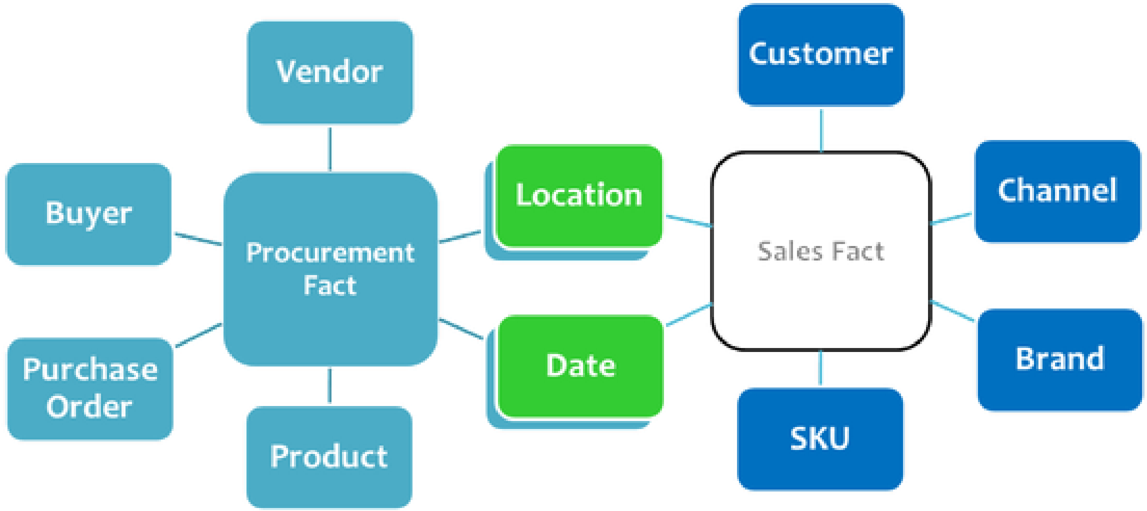
Other types of dimensions that rarely see:

- Shrunken dimensions
- Static dimensions

Conformed Dimensions

Dimensions are conformed when they has the same meaning and content when being referred from different fact table:

- E.g., date and location dimension
- Make ETL process more efficient



Source: http://wisdomschema.com/2015/06/conformed-dimensions/

Junk Dimensions

Miscellaneous attributes that don’t belong to any existing dimension

- Typically flags or indictors that describe or categorize the transaction in some way.
  - E.g., yes/no attributes, open ended comments etc. are too valuable to ignore or exclude

Four alternatives for dealing with them

- Leave them in the fact table:
  - Not ideal because it increase size of fact table
- Create a separate dimension for each attribute
  - Make data warehouse very complex
- Omit them
  - Very dangerous because they might contain valuable information
- Group them into a single junk dimension: Best approach

Junk Dimension Example

JunkID	Paymen_Method	Promoted	Prepaid	Order_Type
1	Visa	Y	N	Online
2	AMX	Y	N	Online
3	Cash	Y	Y	On-site

Degenerate Dimensions

A Dimension key in the fact table that does not link to its own dimension table, because all the interesting attributes have been placed in analytic dimensions:

- Store in fact table
- No related dimension
- Provide grouping & business meaning

Degenerate Dimension Example

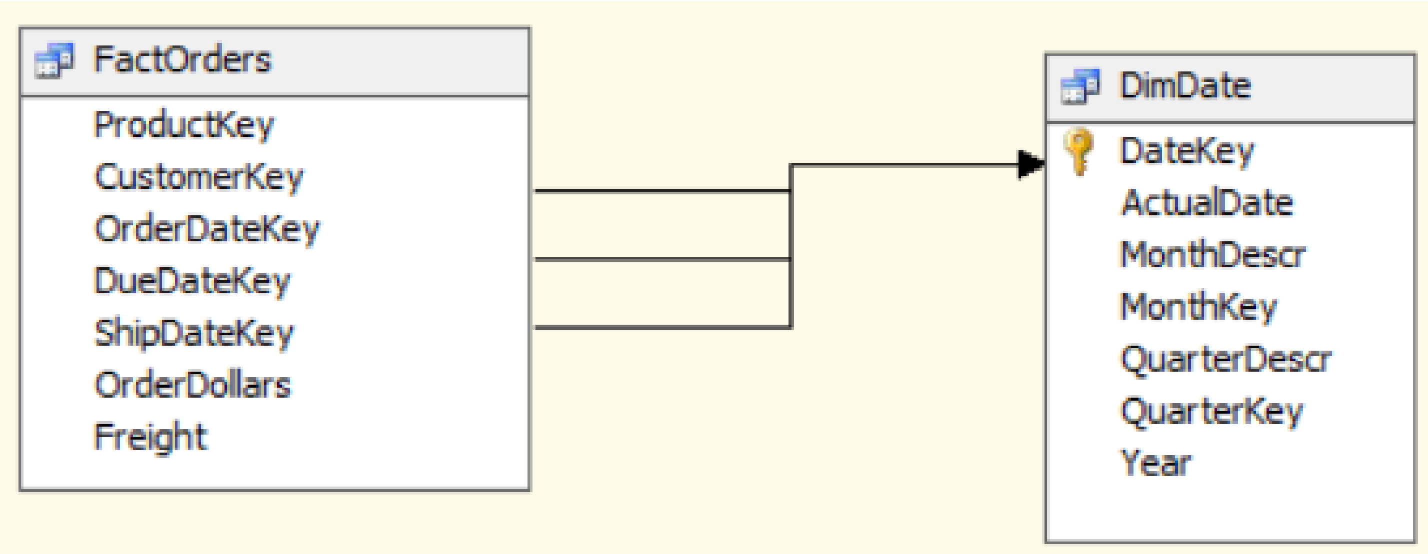
Fact Table has OrderID_DD as degenerate dimension key. They are not linked to any dimension tables but used to group data:

Product Key	DateKey	OrderID_DD	CurrentQuantity
1	1/1/18	1	100
1	1/2/18	1	50
2	1/1/18	2	150
2	1/1/18	2	20
2	1/2/18	3	100

Role-Playing Dimensions

A dimension table refers multiple times in a fact table:

- E.g., time dimension can be used for three times as OrderDate, ShipDate, and DueDate



Source: http://www.codemag.com/Article/Image/1304071/Goff_fig4.png

Slowly Changing Dimensions (SCD)

Attribute may change overtime:

- Most common issue of Data Warehouse
- Need to track the changes in Data Warehouse

For example, customer address is changed from Boston, MA to Dallas, TX. What should we do to keep track the changes?

Three Types of SCD

Type 1 SCD overwrites the existing attribute value with a new value. You don’t care about keeping track of historical values

Type 2 SCD change tracking – ETL process creates a new row in the dimension table to capture the new values of the changed item

Type 3 SCD – Similar to Type 2 SCD but only track current state and the original state; two additional attribute: SCD Start Date, SCD Initial Value

Type 1 SCD Example

Old Table:

CustomerID	CustomerState	CustomerCity
1	MA	Boston
2	MA	Boston
3	MA	Boston

New Table:

CustomerID	CustomerState	CustomerCity
1	TX	Dallas
2	TX	Dallas
3	TX	Dallas

Pros:

- Easiest way to handle SCD problem

Cons:

- All historical data is lost

Type 2 SCD Example

Old Table:

CustomerID	CustomerState	CustomerCity
1	MA	Boston

New Table:

CustomerID	CustomerState	CustomerCity
1	MA	Boston
2	TX	Dallas

Pros:

- Allows to keep historical data

Cons:

- Table size grow fast
- Complicated ETL process

Type 3 SCD Example

Old Table:

CustomerID	CustomerState	CustomerCity
1	MA	Boston

New Table:

CustomerID	OldCustomerState	OldCustomerCity	NewCustomerState	NewCustomerCity	EffectiveDate
1	MA	Boston	TX	Dallas	2/1/2017

Pros:

- Allows to keep some historical data
- Table size does not increase

Cons:

- Not be able to keep all changes

Surrogate Key

An artificial value that is unique in data warehouse.

Used in Type 2 SCD:

- Old Table

CustomerID	CustomerState	CustomerCity
1	MA	Boston

- New Table

CustomerID	CustomerState	CustomerCity
1	MA	Boston
2	TX	Dallas

How do we know customerID 1 and 2 are the same person?

Surrogate Key

Surrogate Key is created to keep track the changes:

- Old Table

CustomerID	CustomerState	CustomerCity
1	MA	Boston

- New Table

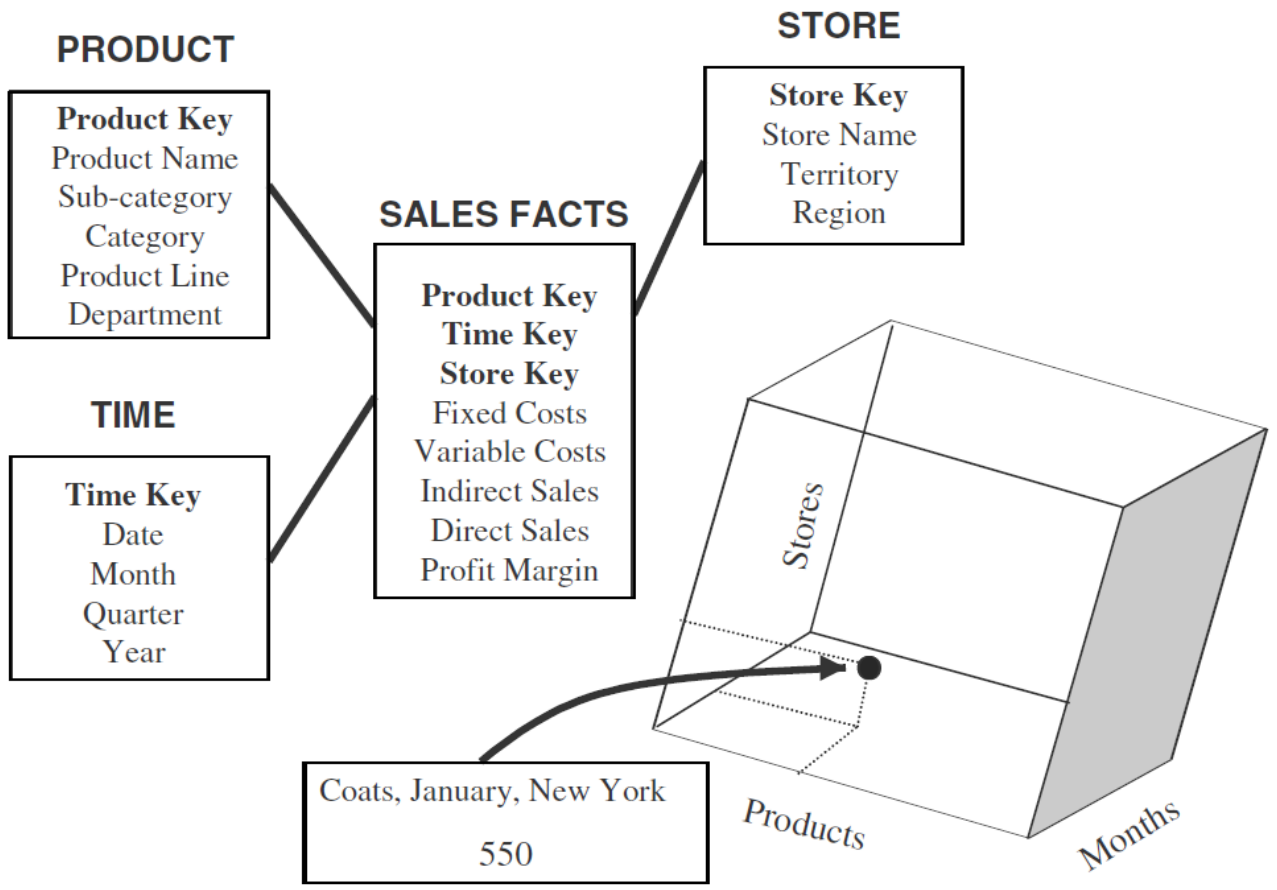
SurrogateID	CustomerID	CustomerState	CustomerCity
1	1	MA	Boston
2	1	TX	Dallas

Data Cube

Data cube is multidimensional dataset that represent data in Star Model (or Snowflake model) in data warehouse.

Data cube can have many dimensions up to 64, but we tend to look at just three at a time.

Data Cube


PRODUCT
Product Key
Product Name
Sub-category
Category
Product Line
Department
TIME
Time Key
Date
Month
Quarter
Year
SALES FACTS
Product Key
Time Key
Store Key
Fixed Costs
Variable Costs
Indirect Sales
Direct Sales
Profit Margin
STORE
Store Key
Store Name
Territory
Region
Stores
Coats, January, New York
550
Products
Months

3-D Data Cube Display

Store: New York

PAGES: STORE dimension

Products

COLUMNS: PRODUCT dimension

ROWS: TIME dimension

Months

	Hats	Coats	Jackets	Dresses	Shirts	Slacks
Jan	200	550	350	500	520	490
Feb	210	480	390	510	530	500
Mar	190	480	380	480	500	470
Apr	190	430	350	490	510	480
May	160	530	320	530	550	520
Jun	150	450	310	540	560	330
Jul	130	480	270	550	570	250
Aug	140	570	250	650	670	230
Sep	160	470	240	630	650	210
Oct	170	480	260	610	630	250
Nov	180	520	280	680	700	260
Dec	200	560	320	750	770	310

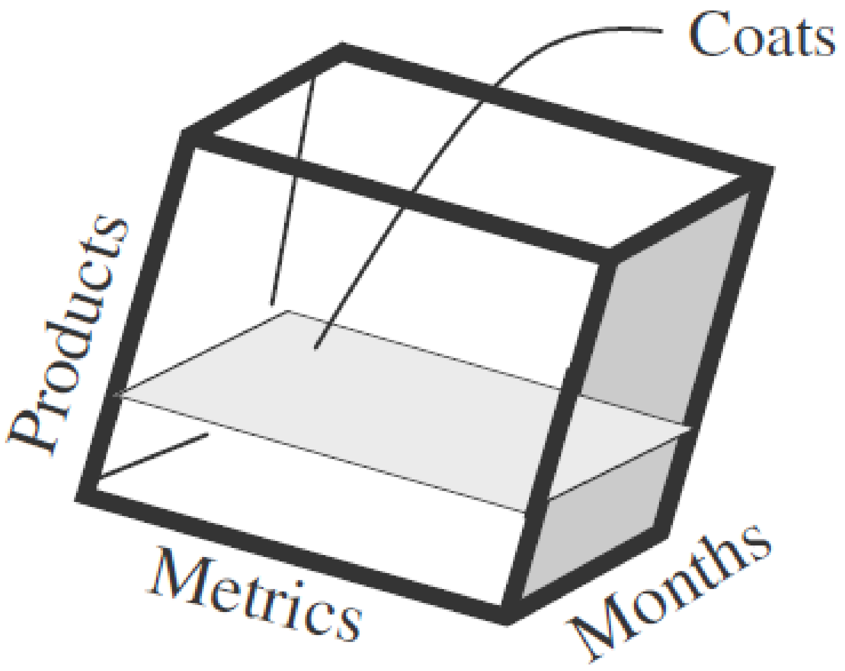
4-D Data Cube Display

PRODUCT: Coats

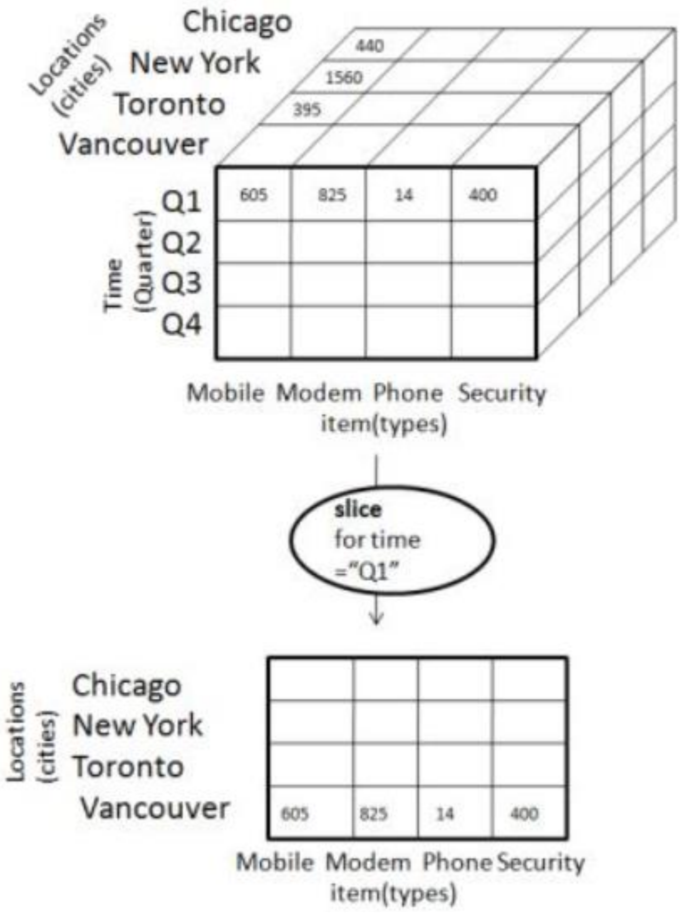
PAGES: PRODUCT dimension COLUMNS: Metrics

ROWS: TIME dimension

	Fixed Cost	Variable Cost	Indirect Sales	Direct Sales	Profit Margin
Jan	340	110	230	320	100
Feb	270	90	200	260	100
Mar	310	100	210	270	70
Apr	340	110	210	320	80
May	330	110	230	300	90
Jun	260	90	150	300	100
Jul	310	100	180	300	70
Aug	380	130	210	360	60
Sep	300	100	180	290	70
Oct	310	100	170	310	70
Nov	330	110	210	310	80
Dec	350	120	200	360	90



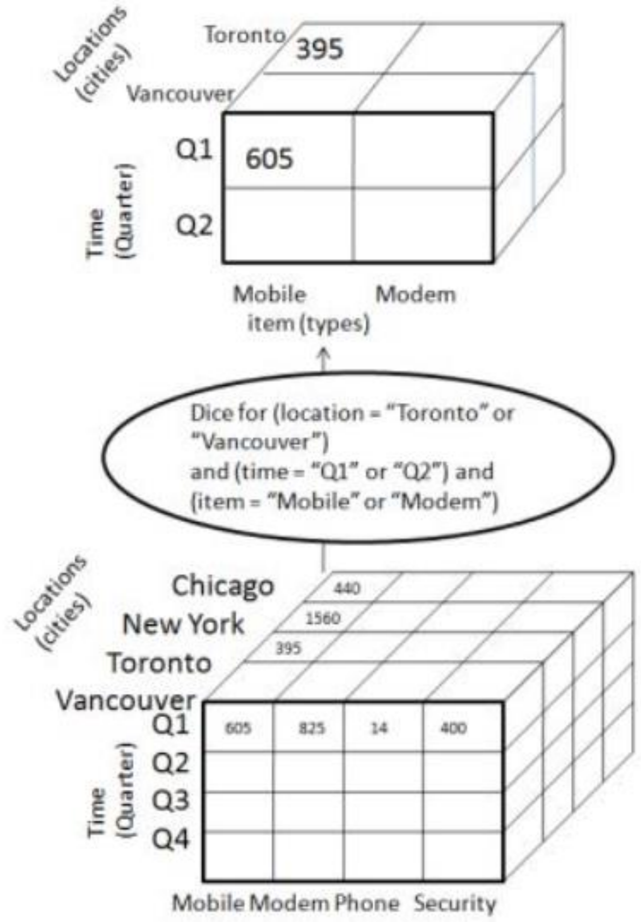
Data Cube Operation - Slice



Slice: A selection of data cube chosen by choosing a single value of one dimension

Source: https://www.tutorialspoint.com/dwh/images/slice.jpg

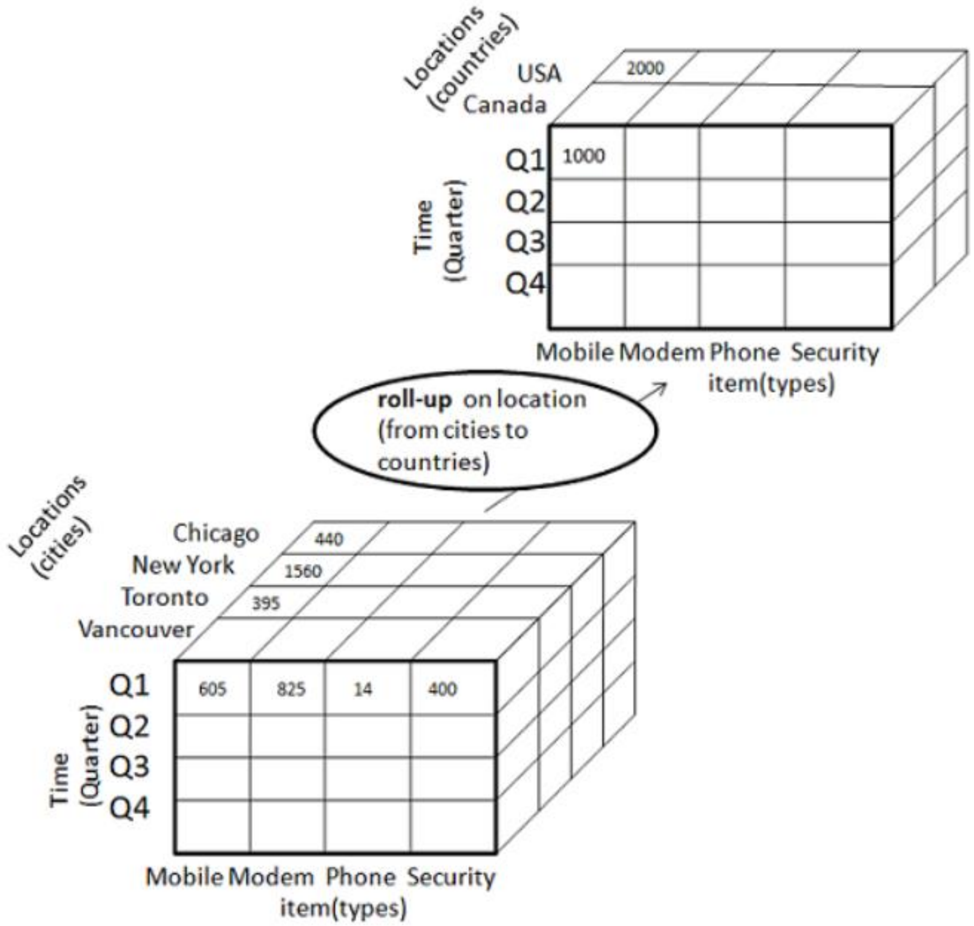
Data Cube Operation - Dice



Dice: A selection of data cube chosen by choosing a specific values of multiple dimensions

Source: https://www.tutorialspoint.com/dwh/images/dice.jpg

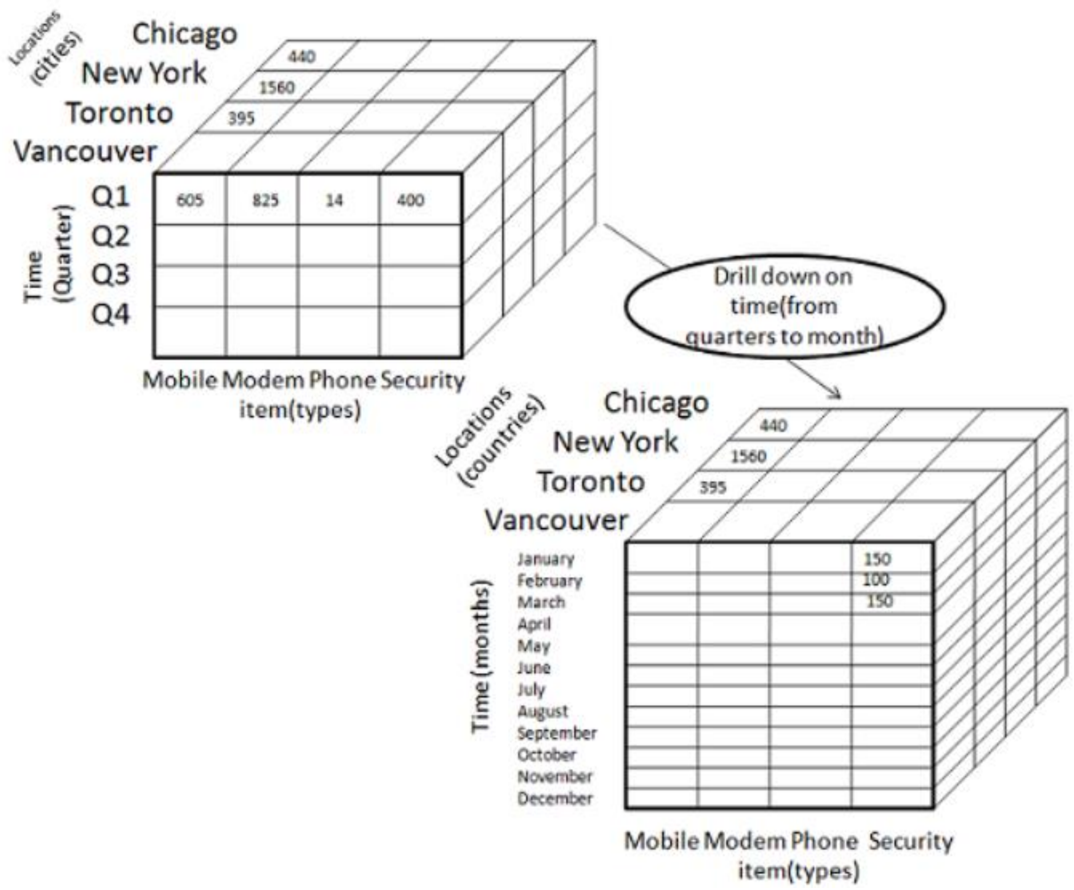
Data Cube Operation – Roll Up



Roll Up: Summarize data along one dimension

Source: https://www.tutorialspoint.com/dwh/images/rollup.jpg

Data Cube Operation – Drill Down



Drill Down: Reduce hierarchies of multiples dimensions to find detailed data summarization.

Source: https://www.tutorialspoint.com/dwh/images/drill_down.jpg

Data Warehouse Development Process

Data Warehouse Development Approaches

Kimball Model: Data mart approach

- Data marts – EDW (Bottom-up development)

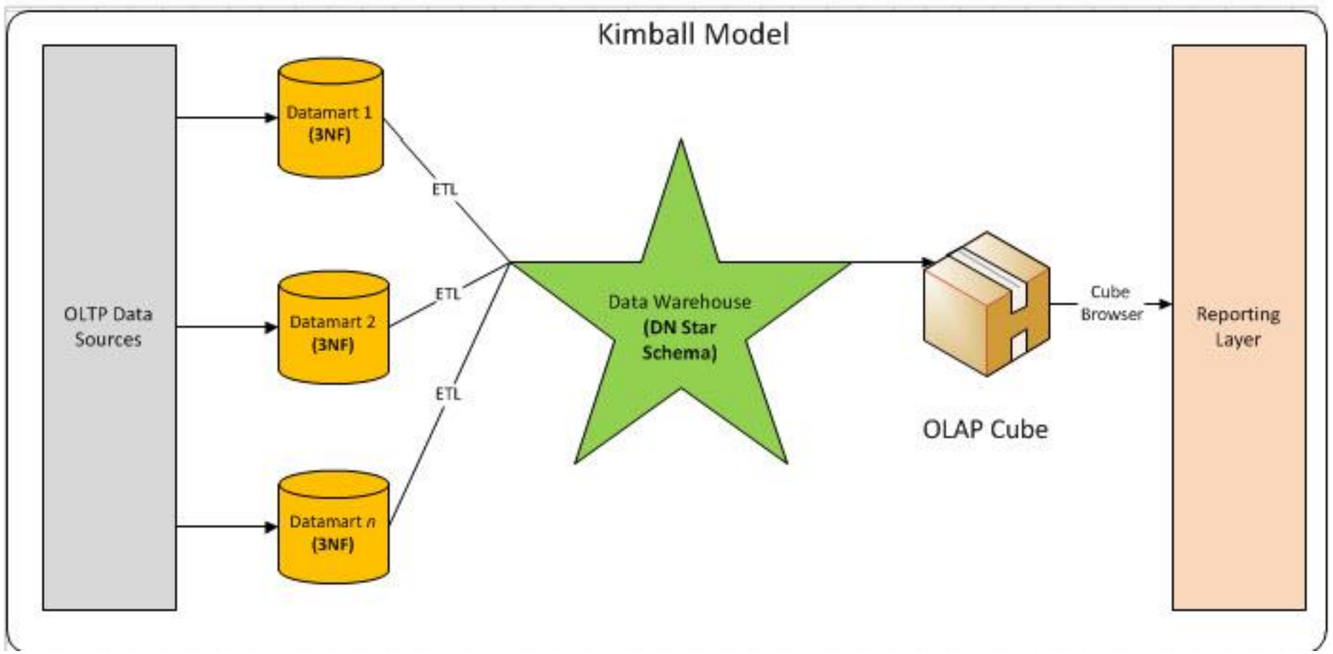
Inmon Model: EDW approach

- EDW – Data Marts (Top-down development)

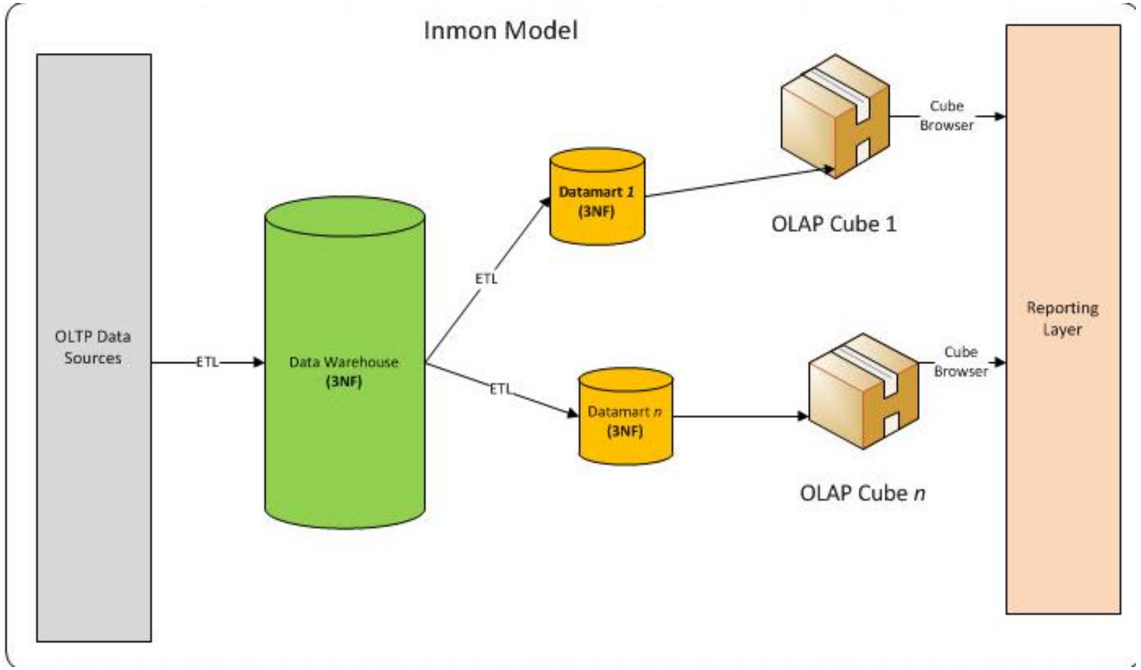
Which model is better?

- There is no one-size-fits-all strategy to data warehousing
- Depend on organization structure, and information needs.

Comparison



Kimball Model
OLTP Data Sources
Datamart 1 (3NF)
Datamart 2 (3NF)
Datamart n (3NF)
ETL
ETL
ETL
Data Warehouse (DN Star Schema)
OLAP Cube
Cube Browser
Reporting Layer

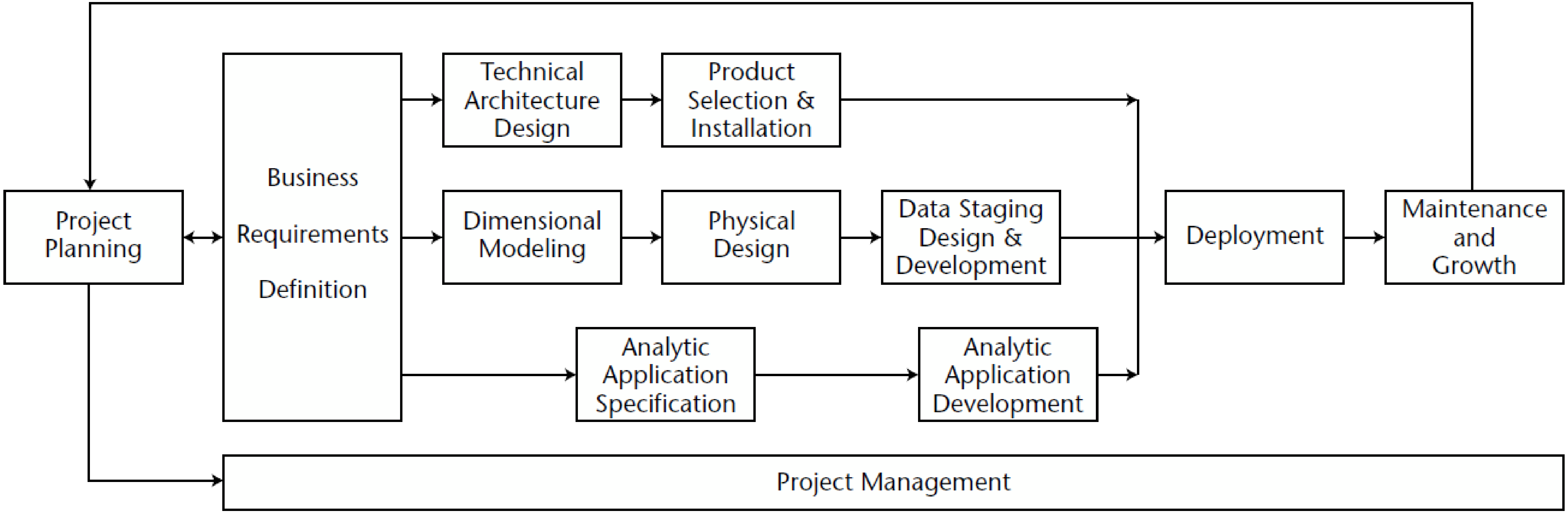


Inmon Model
OLTP Data Sources
ETL
Data Warehouse (3NF)
ETL
ETL
Datamart 1 (3NF)
Datamart n (3NF)
OLAP Cube 1
OLAP Cube n
Cube Browser
Cube Browser
Reporting Layer

Strengths and Weaknesses

- Kimball's model is more scalable and faster to build. However it is difficult to reuse for different data marts
- Inmon's model is more structured, but takes more time to build. It is easier to build data mining models.

Kimball's Data Warehouse Development Process Approach



Kimball's Approach

Project Planning:

- Define score of data warehouse projects
- Getting staffs: sponsors, users, analysts, data modelers etc.

Business Requirement Definitions: form teams to collet business requirements by interviewing or getting archival documents.

Technological Architecture Design: specify the architecture of data warehouse – data staging, data service, and metadata.

Kimball's Approach

Product Selection and Installation: determine hardware and software purchase.

Dimensional Modeling: design star or snowflake schema for data warehouse.

Physical Design: create detail model for physical database including column names, data types, and indexing.

Data Staging Design and Development: design and develop ETL process

- **Very intensive labor work**

Kimball's Approach

Analytic Application Specification: collecting sample analytic applications to establish the standards for final applications.

Analytic Application Development: develop analytics applications.

Deployment: Implement data mart, data warehouse etc.

Maintenance and Growth: provide support, and training to end-users.

Class Example: Adventure Works

A fictitious multinational manufacturer and seller of bicycles and accessories

Based on Bothell, Washington, USA and has regional sales offices in several countries

Read case study on eCollege

Basic Business Operations

Selling Products in 4 categories – bikes, components, clothing, and accessories.

Selling Products in 6 counties – US, UK, Canada, Australia, France, and Germany

Selling Products via 2 channels – Internet and Reseller

Interview Requirements

Problem: information is not available or take too much time to prepare

Major needed analytic areas:

- Sales planning
  - Growth analysis
  - Customer analysis
  - Territory analysis
- Sales performance
- Basic sales reporting
- Price lists
- Special offers
- Customer satisfaction
- International support

Success criteria

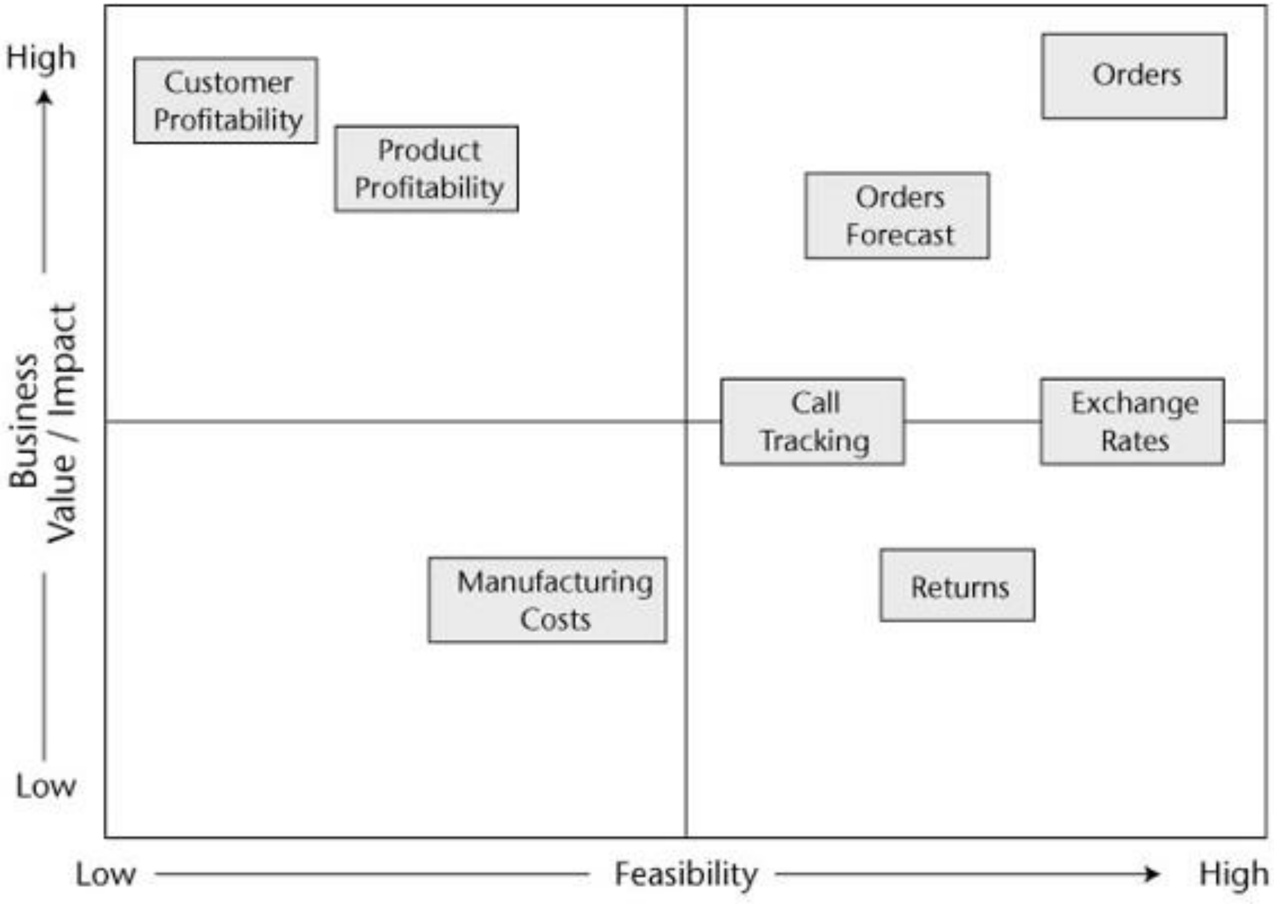
- Easy data access, flexible reporting and analyzing, and all data in one place

What’s missing? – A lot – No indication of business value

Bus Matrix

	Dimensions										
Business Process	Date	Product	Employee	Customer (Reseller)	Customer (Internet)	Sales Territory	Currency	Channel	Promotion	Call Reason	Facility
Sales Forecasting	X	X	X	X	X	X	X				
Orders	X	X	X	X	X	X	X	X	X		
Call Tracking	X	X	X	X	X	X				X	
Returns	X	X		X	X	X	X		X		X

Prioritization Grid

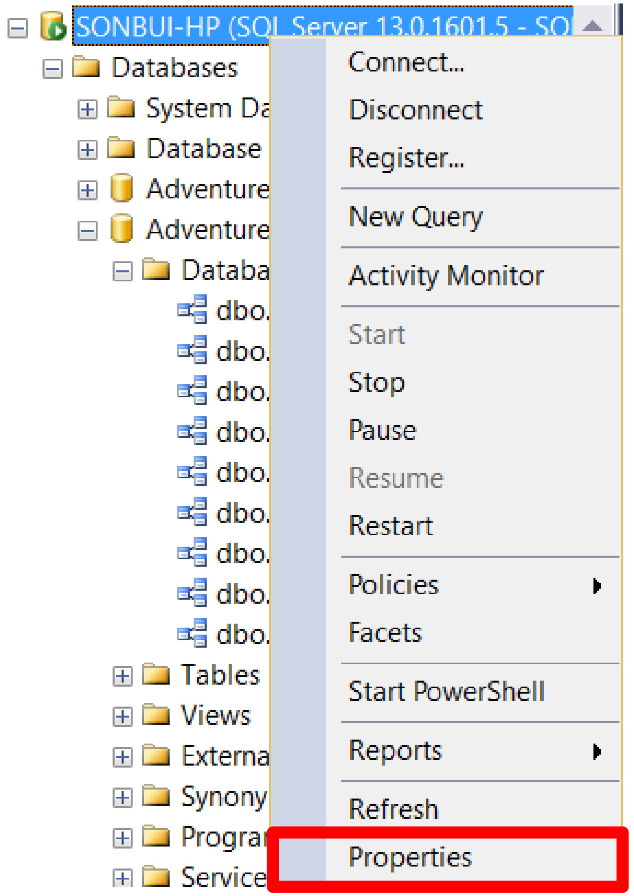


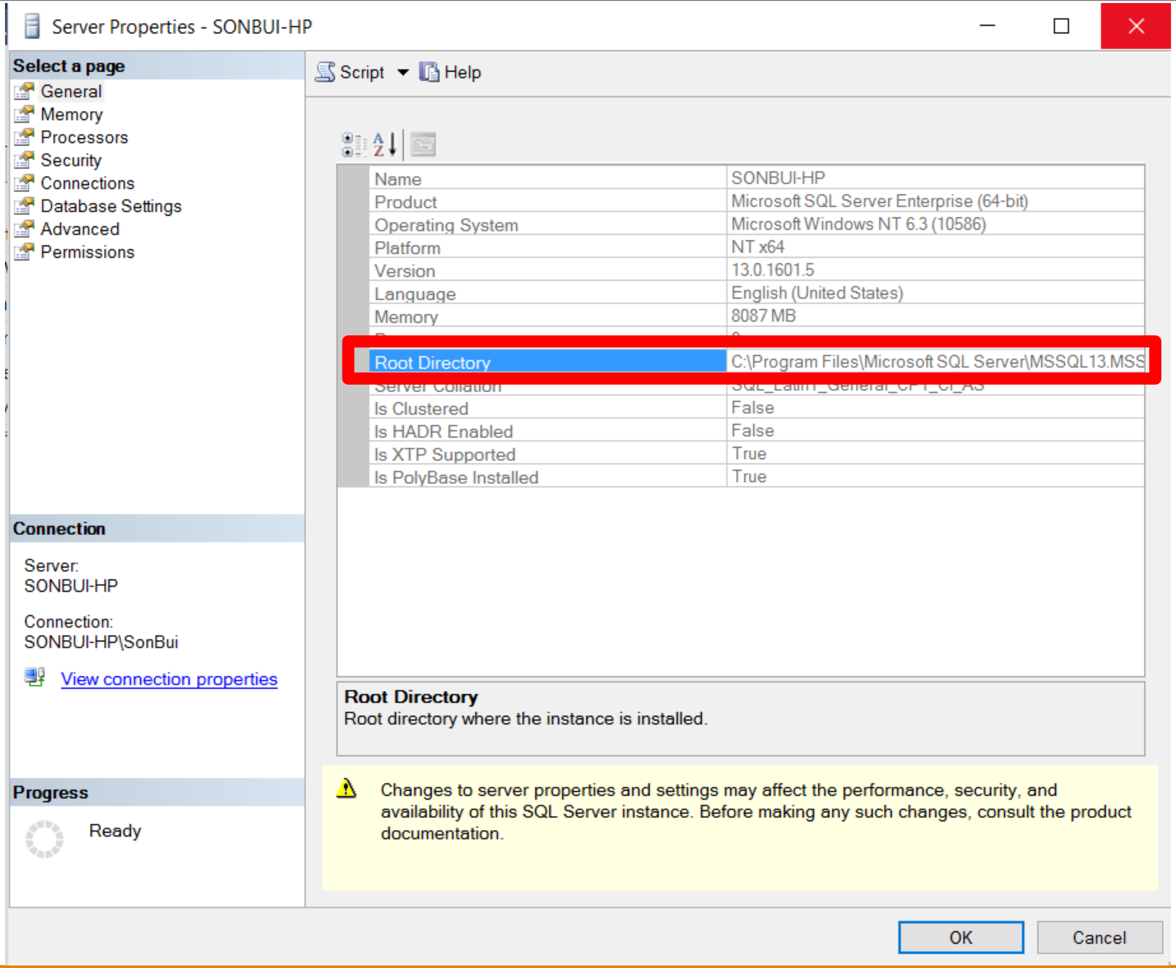
Importing Adventure Works’ OLTP into SQL Sever 2016

Step 1: Visit https://github.com/Microsoft/sql-server-samples/releases/tag/adventureworks

Step 2: Download and copy AdventureWorks2014.bak into Backup folder of your server. My root is C:\Program Files\Microsoft SQL Server\MSSQL13.MSSQLSERVER\MSSQL, then my Backup folder is C:\Program Files\Microsoft SQL Server\MSSQL13.MSSQLSERVER\MSSQL\Backup

Finding Root Directory





Importing Adventure Works' OLTP into SQL Sever 2016

Step 3: Run the script from eCollege, and make sure to change your server location

USE [master]

RESTORE DATABASE AdventureWorks2014

FROM disk= **C:\Program Files\Microsoft SQL Server\MSSQL13.MSSQLSERVER\MSSQL Backup\AdventureWorks2014.bak**'

WITH MOVE 'AdventureWorks2014_data' TO

C:\Program Files\Microsoft SQL Server\MSSQL13.MSSQLSERVER\MSSQL\DATA\AdventureWorks2014.mdf',

MOVE 'AdventureWorks2014_Log' TO ' C:\Program Files\Microsoft SQL Server\MSSQL13.MSSQLSERVER\MSSQL \DATA\AdventureWorks2014.ldf'

,REPLACE

Importing Adventure Works' OLAP into SQL Sever 2016

Step 1: Visit https://github.com/Microsoft/sql-server-samples/releases/tag/adventureworks

Step 2: Download and copy AdventureWorksDW2014.bak into Backup folder of your server. My root is C:\Program Files\Microsoft SQL Server\MSSQL13.MSSQLSERVER\MSSQL, then my Backup folder is C:\Program Files\Microsoft SQL Server\MSSQL13.MSSQLSERVER\MSSQL\Backup

Importing Adventure Works’ OLAP into SQL Sever 2016

Step 3: Run the script from eCollege, and make sure to change your server location

USE [master]

RESTORE DATABASE AdventureWorksDW2014

FROM disk= **C:\Program Files\Microsoft SQL Server\MSSQL13.MSSQLSERVER\MSSQL Backup\AdventureWorksDW2014.bak**'

WITH MOVE 'AdventureWorksDW2014_data' TO

C:\Program Files\Microsoft SQL Server\MSSQL13.MSSQLSERVER\MSSQL\DATA\AdventureWorksDW2014.mdf',

MOVE 'AdventureWorksDW2014_Log' TO ' C:\Program Files\Microsoft SQL Server\MSSQL13.MSSQLSERVER\MSSQL \DATA\AdventureWorks2DW014.ldf'

,REPLACE