Introduction to Big Data

ECAP456

Dr. Rajni Bhalla

Associate Professor

Learning Outcomes

After this lecture, you will be able to

- understand what is data format.
- learn data model and tips for creating effective big data models.
- understand data mart and types of data mart.
- differentiate between data warehouse and data mart.

Introduction

```
"rules": {
  "align": [false,
    "parameters",
    "arguments",
    "statements"],
  "ban": [true,
    ["angular", "forEach"]
  ],
  "class-name": true,
  "comment-format": [false,
    "check-space",
    "check-lowercase"
  ],
```

JSON

```
- <menu>
  - <area text="Welcome" file="index.html">
      <submenuitem text="New in Scribus 1.5" file="readme.html"/>
      <submenuitem text="Specification" file="specs.html"/>
    </area>
  - <area text="Documentation" file="intro.html">
    - <submenuitem text="Introduction" file="documentation.html">
        <submenuitem text="Editorial Notes" file="editorial.html"/>
        <submenuitem text="About the Team" file="about1.html"/>
      </submenuitem>
    - <submenuitem text="Setup" file="config.html">
        <submenuitem text="Configuring Scribus" file="settings1.html"/>
        <submenuitem text="Hyphenation and Spellchecking" file="hyphenator.html"/>
        <submenuitem text="Font Setup" file="fonts1.html"/>
        <submenuitem text="Fonts in Depth" file="fonts2.html"/>
      </submenuitem>
    - <submenuitem text="Scribus Basics" file="about2.html">
        <submenuitem text="Quick Start Guide" file="qsg.html"/>
        <submenuitem text="Command Line Reference" file="cli.html"/>
        <submenuitem text="Keyboard Shortcuts" file="keys.html"/>
        <submenuitem text="Mouse Shortcuts" file="mouse.html"/>
        <submenuitem text="Document Information" file="docinfo.html"/>
        <submenuitem text="Working with Frames" file="WwFrames.html"/>
        <submenuitem text="Working with Text" file="WwText.html"/>
        <submenuitem text="Text Properties" file="TextProp.html"/>
        <submenuitem text="Search and Replace" file="SearchReplace.html"/>
        <submenuitem text="Working with Styles" file="WwStyles.html"/>
        <submenuitem text="Working with Images" file="WwImages.html"/>
```

XML

Introduction

```
%YAML 1.2
---
YAML: YAML Ain't Markup Language

What It Is: YAML is a human friendly data serialization
  standard for all programming languages.

YAML Resources:
  YAML 1.2 (3rd Edition): http://yaml.org/spec/1.2/spec.html
  YAML 1.1 (2nd Edition): http://yaml.org/spec/1.1/
  YAML 1.0 (1st Edition): http://yaml.org/spec/1.0/
  YAML Issues Page: https://github.com/yaml/yaml/issues
  YAML Mailing List: yaml-core@lists.sourceforge.net
  YAML IRC Channel: "#yaml on irc.freenode.net"
  YAML Cookbook (Ruby): http://yaml4r.sourceforge.net/cookbook/ (local)
  YAML Reference Parser: http://yaml.org/ypaste/

Projects:
  C/C++ Libraries:
  - libyaml       # "C" Fast YAML 1.1
  - Syck          # (dated) "C" YAML 1.0
  - yaml-cpp      # C++ YAML 1.2 implementation
  Ruby:
  - psych         # libyaml wrapper (in Ruby core for 1.9.2)
  - RbYaml        # YAML 1.1 (PyYaml Port)
  - yaml4r        # YAML 1.0, standard library syck binding
  Python:
  - PyYaml        # YAML 1.1, pure python and libyaml binding
  - PySyck        # YAML 1.0, syck binding
  Java:
  - JvYaml        # Java port of RbYaml
  - SnakeYAML     # Java 5 / YAML 1.1
  - YamlBeans     # To/from JavaBeans
  - JYaml         # Original Java Implementation
```

YAML

Introduction

```xsd
<xs:simpleType name="Money">
  <xs:restriction base="xs:decimal">
    <xs:totalDigits value="13" />
    <xs:fractionDigits value="2" />
    <xs:minInclusive value="0.00" />
    <xs:maxInclusive value="99999999999.99" />
  </xs:restriction>
</xs:simpleType>

<xs:element name="Envelope">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Deposit" minOccurs="1" maxOccurs="1">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="ClientId" type="xs:unsignedLong" />
            <xs:element name="Account" type="xs:unsignedShort" />
            <xs:element name="Currency" type="xs:string" />
            <xs:element name="TotalSum" type="xs:Money" />
            <xs:eleme
            <xs:element name="Cheques" />
```

Type 'http://www.w3.org/2001/XMLSchema:Money' is not declared.

XSD

```markup
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE recipe PUBLIC "-//Happy-Monkey//DTD RecipeBook//EN"
"http://www.happy-monkey.net/recipebook/recipebook.dtd">

<recipe>

    <title>Peanut-butter On A Spoon</title>

    <ingredientlist>
        <ingredient>Peanut-butter</ingredient>
    </ingredientlist>

    <preparation>
        Stick a spoon in a jar of peanut-butter,
        scoop and pull out a big glob of peanut-butter.
    </preparation>

</recipe>
```

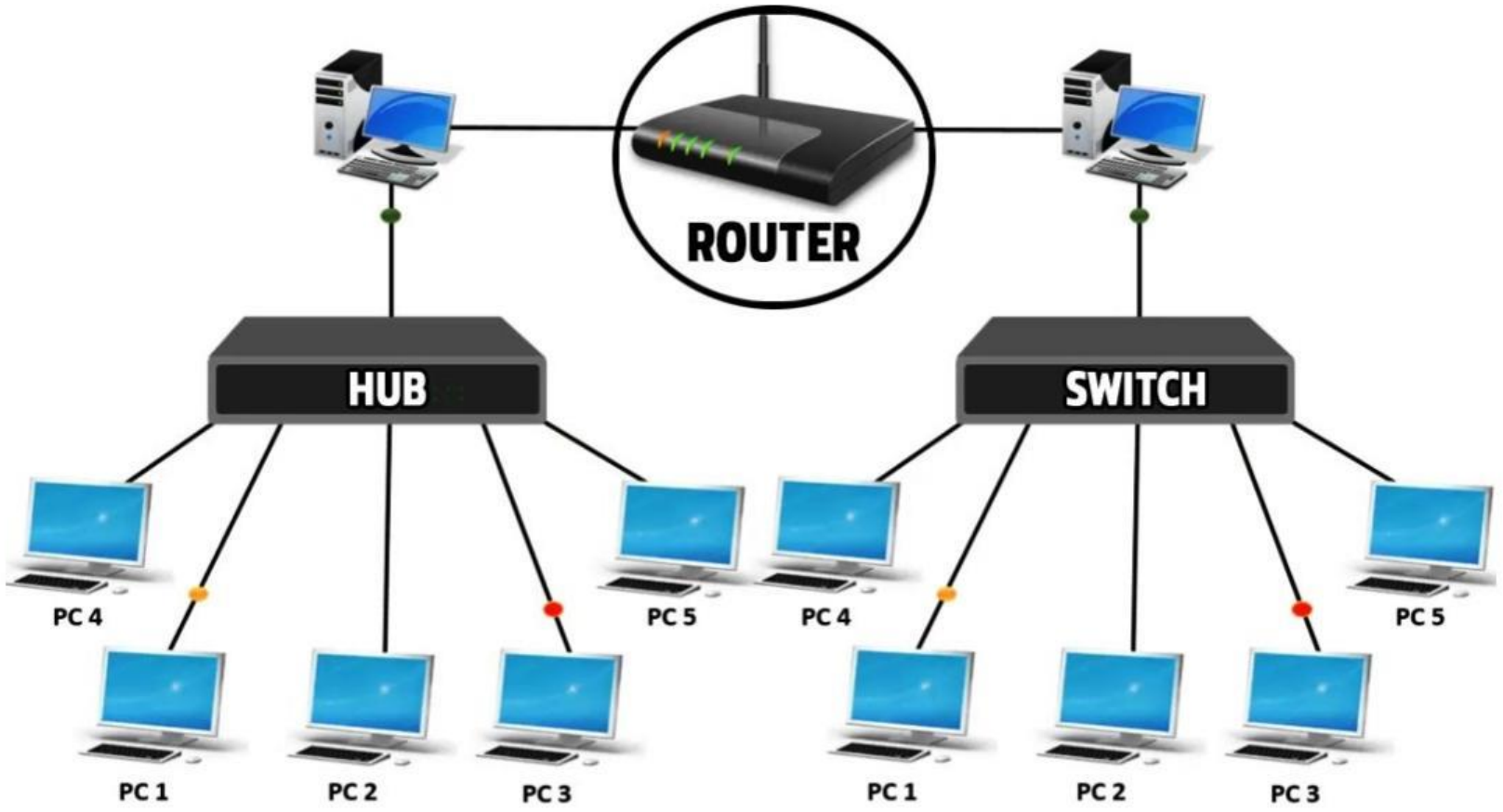
Markup Language

Introduction

Router

Switch

Introduction



Introduction to Data Formats

- A computer programmer typically uses a wide variety of tools to store and work with data in the programs they build.
- They may use simple variables (single value), arrays (multiple values), hashes (key-value pairs), or even custom objects built in the syntax of the language they're using.

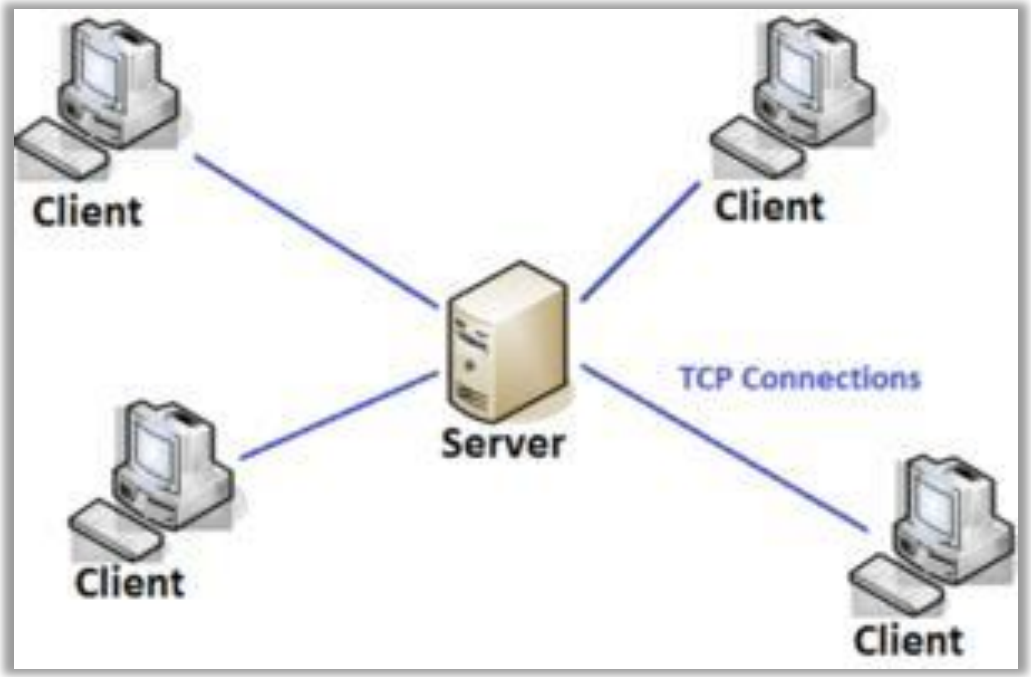
Data Format

- Portable format is required.
- Another program may have to communicate with this program in a similar way, and the programs may not even be written in the same language, as is often the case with something like traditional client-server communications.

Data Format

- This is all perfectly standard within the confines of the software being written.
- However, sometimes a more abstract, portable format is required.
- For instance, a non-programmer may need to move data in and out of these programs.

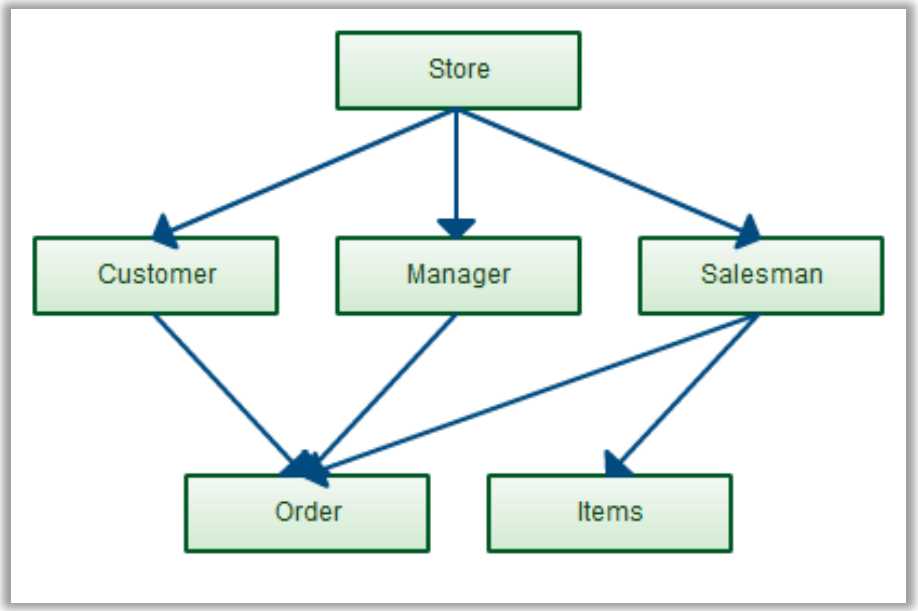
Data Format



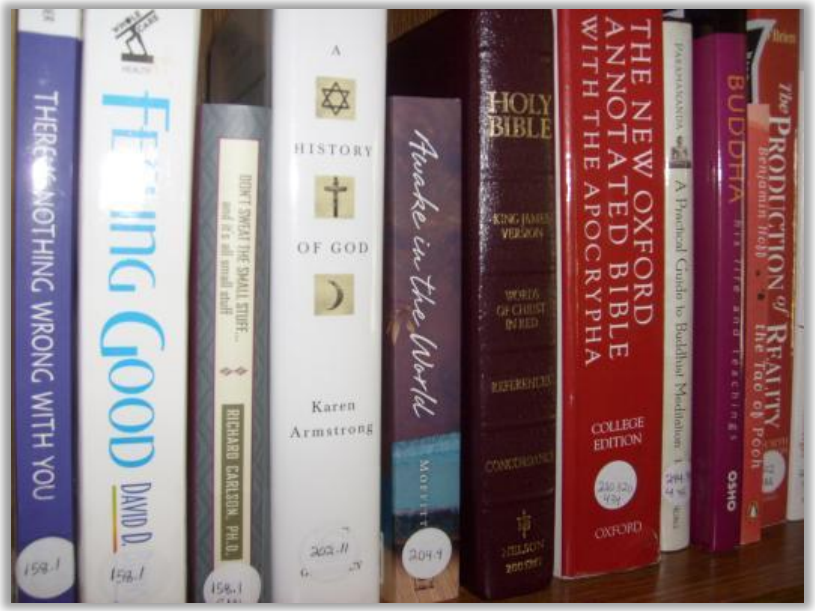
Traditional client-server communications

What is Data Model?

What is a Data Model?



Organize and Store data



Dewey Decimal System organizes the books

Bad programmers worry about the code. Good programmers worry about data structures and their relationships.

Linus Torvalds

Data Model

- A data model explicitly determines the structure of data.
- Data models are specified in a data modeling(link is external) notation, which is often graphical in form.

Data Model

- A data model can be sometimes referred to as a data structure(link is external), especially in the context of programming languages(link is external).

How is Data-Model Built?

- A data model is built using components that act as abstractions of real-world things.
- The simplest data model consists of entities and relationships.

How is Data-Model Built?

- As work on the data model progresses, additional detail and complexity are added, including attributes, domains, constraints, keys, cardinality, requirements, relationships—and importantly, definitions of everything in the data model.
- If we want to understand the data we have—and how to use it—a foundational model is required.

Benefits of Appropriate Models and Storage Environments to Big Data

Performance

Cost

Efficiency

Quality

Benefits of Appropriate Models and Storage Environments to Big Data

Performance

Cost

Efficiency

Quality

Benefits of Appropriate Models and Storage Environments to Big Data

Performance

Cost

Efficiency

Quality

Benefits of Appropriate Models and Storage Environments to Big Data

Performance

Cost

Efficiency

Quality

Therefore, it is without question that a big data system requires high-quality data modeling methods for organizing and storing data, allowing us to reach the optimal balance of performance, cost, efficiency, and quality.

Tips for Creating Effective Big Data Models

- Don't try to impose traditional modeling techniques on big data.
- Design a system, not a schema.
- Look for big data modeling tools.
- Focus on data that is core to your business.
- Deliver quality data.
- Look for key inroads into the data.

Tips for Creating Effective Big Data Models

- Don't try to impose traditional modeling techniques on big data
- Design a system, not a schema
- Look for big data modeling tools
- Focus on data that is core to your business
- Deliver quality data
- Look for key inroads into the data

Tips for Creating Effective Big Data Models

- Don't try to impose traditional modeling techniques on big data
- Design a system, not a schema
- Look for big data modeling tools
- Focus on data that is core to your business
- Deliver quality data
- Look for key inroads into the data

Tips for Creating Effective Big Data Models

- Don't try to impose traditional modeling techniques on big data
- Design a system, not a schema
- Look for big data modeling tools
- Focus on data that is core to your business
- Deliver quality data
- Look for key inroads into the data

Tips for Creating Effective Big Data Models

- Don't try to impose traditional modeling techniques on big data
- Design a system, not a schema
- Look for big data modeling tools
- Focus on data that is core to your business
- Deliver quality data
- Look for key inroads into the data

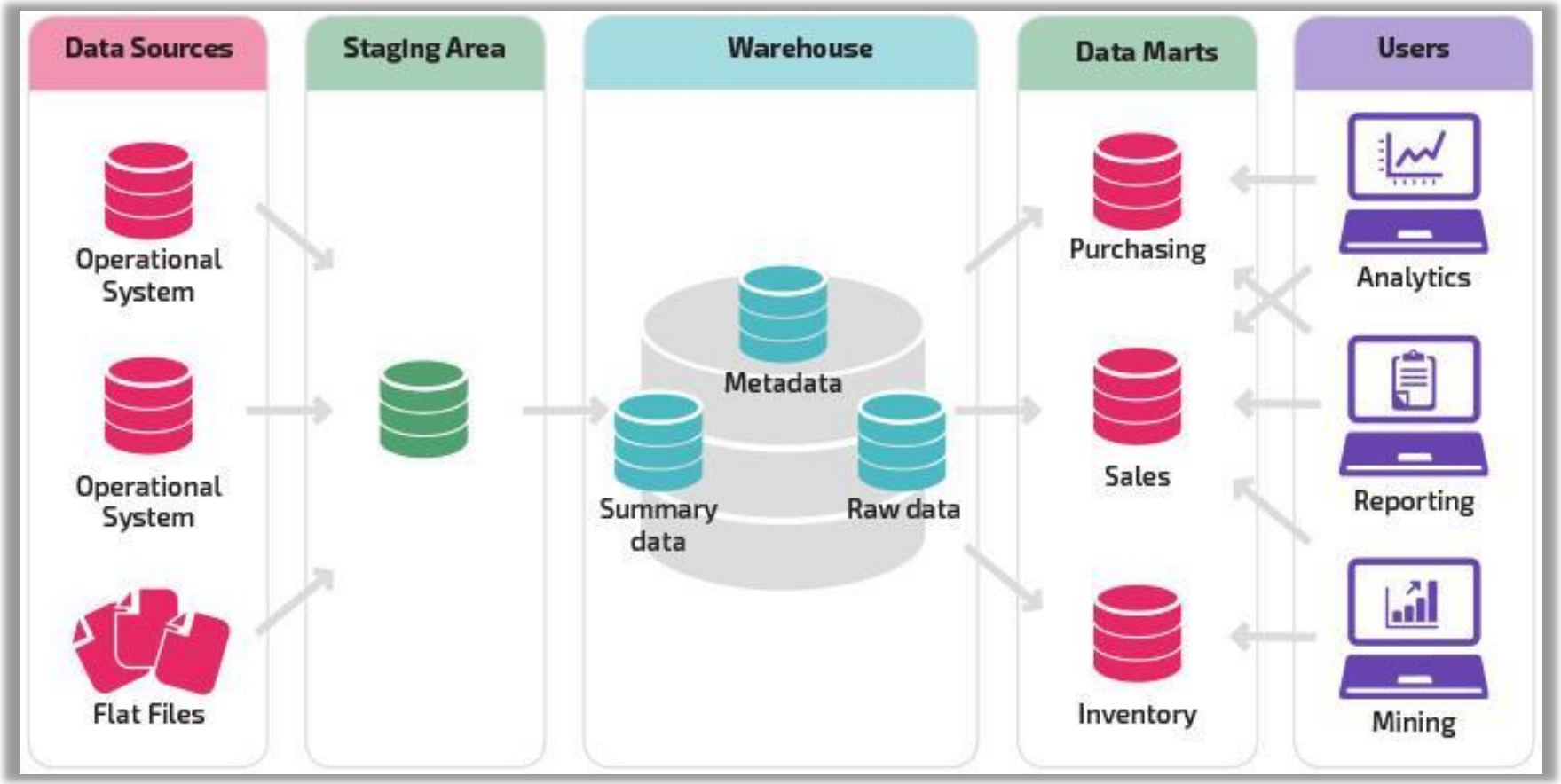
Tips for Creating Effective Big Data Models

- Don't try to impose traditional modeling techniques on big data
- Design a system, not a schema
- Look for big data modeling tools
- Focus on data that is core to your business
- Deliver quality data
- Look for key inroads into the data

What is Data Mart

- It is a data store which is designed for a particular department of an organization, or data mart is a subset of Datawarehouse that is usually oriented to a specific purpose.

Example


Data Sources
Staging Area
Warehouse
Data Marts
Users
Operational System
Operational System
Flat Files
Metadata
Summary data
Raw data
Purchasing
Sales
Inventory
Analytics
Reporting
Mining

Reasons for Using Data Mart

- Easy access of frequent data.
- Improved end user response time.
- Easy creation of data marts.
- Less cost in building the data mart.

Different Types of Data Mart

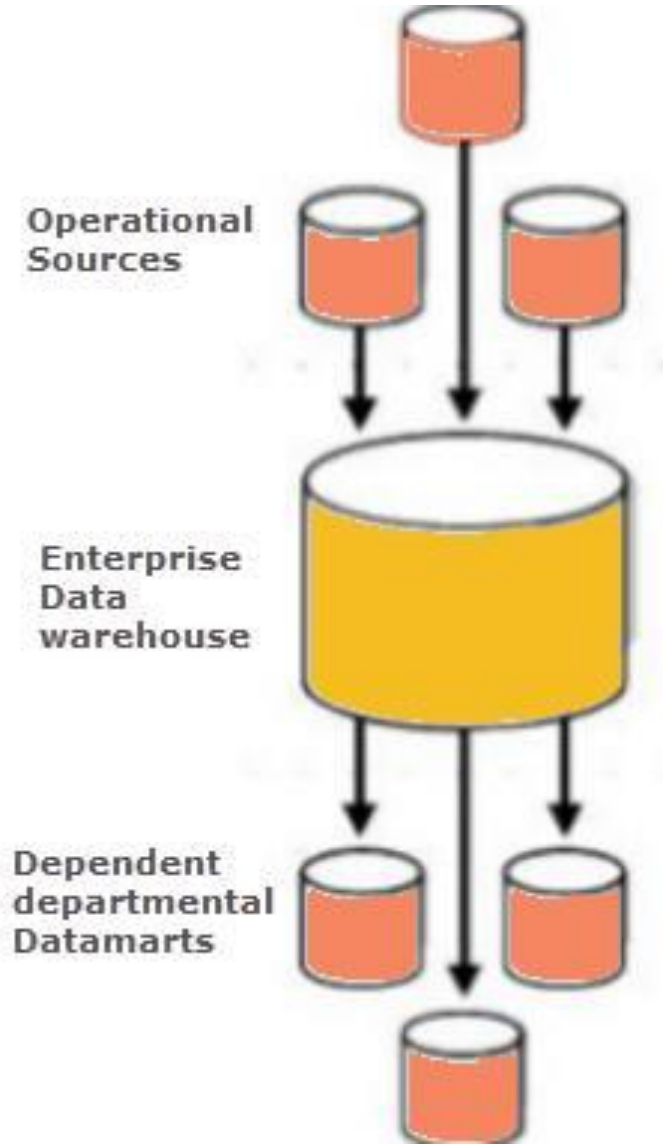
Dependent Data Mart

Independent Data Mart

Hybrid Data Mart

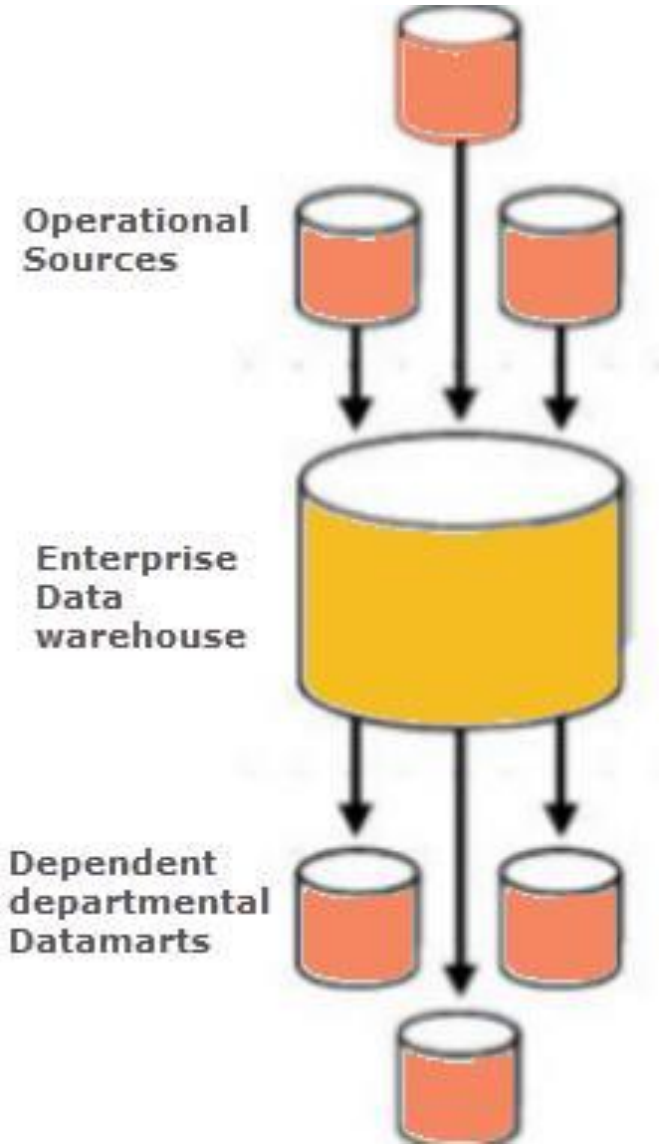
Dependent Data Mart

In this the data mart is built by drawing data from central data warehouse that already exists.



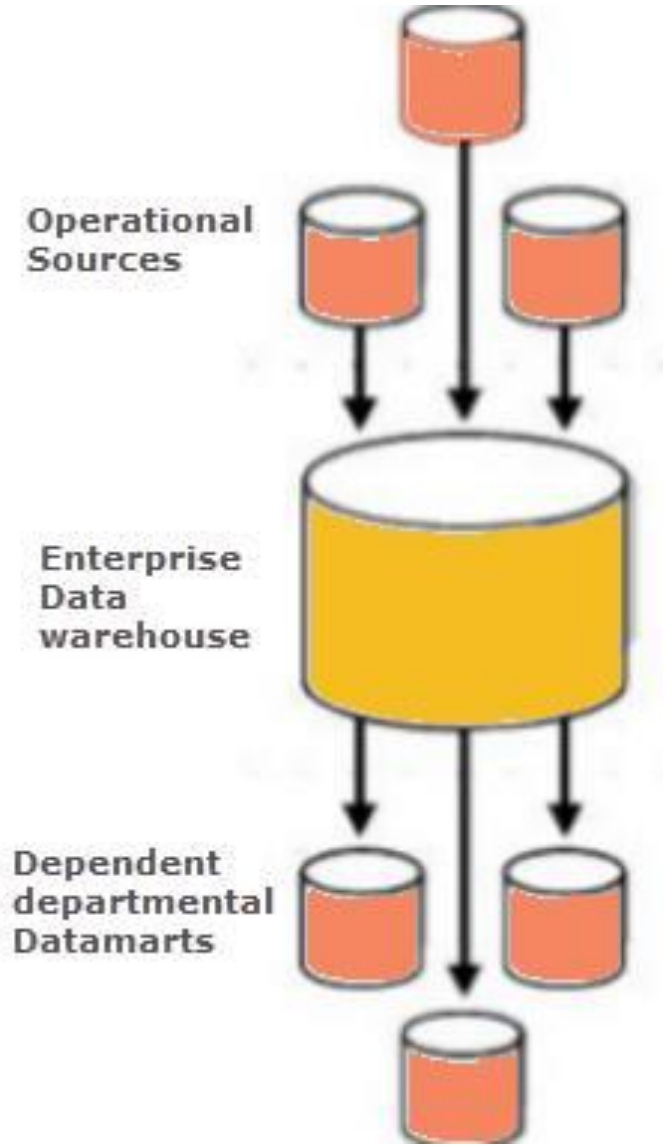
Dependent Data Mart

It is one of the data mart example which offers the benefit of centralization.



Dependent Data Mart

If you need to develop one or more physical data marts, then you need to configure them as dependent data marts.



Dependent Data Mart

Dependent data mart can be built in two different ways: -

a. Either where a user can access both the data mart and data warehouse, depending on need, or where access is limited only to the data mart

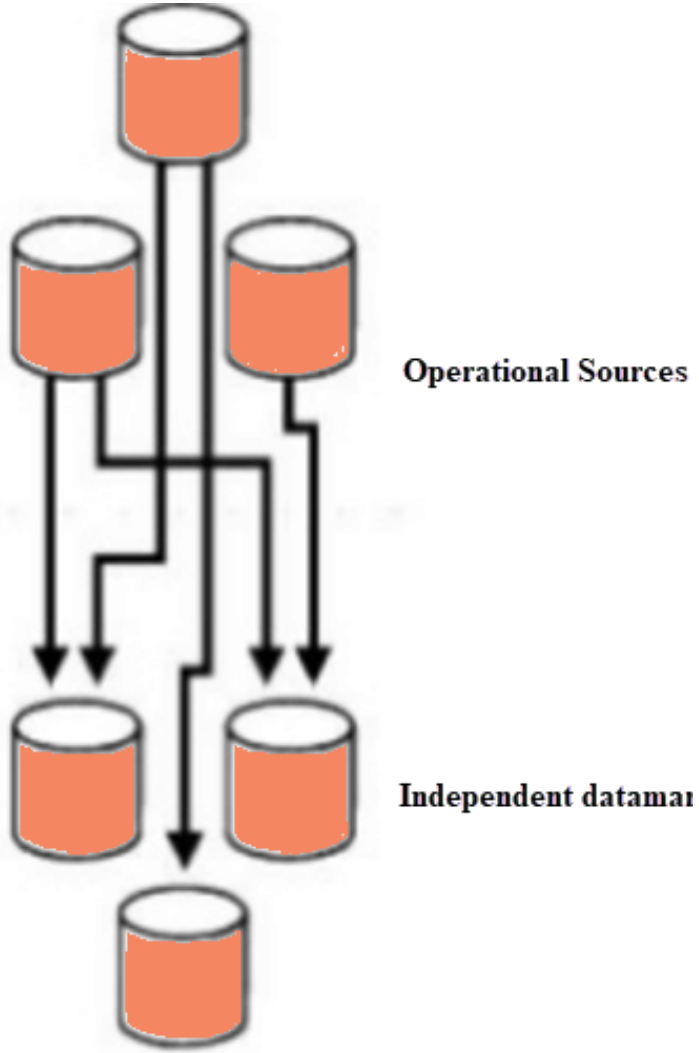
Dependent Data Mart

Dependent data mart can be built in two different ways: -

b. The second approach is not optimal as it produces sometimes referred to as a data junkyard. In the data junkyard, all data begins with a common source, but they are scrapped, and mostly junked.

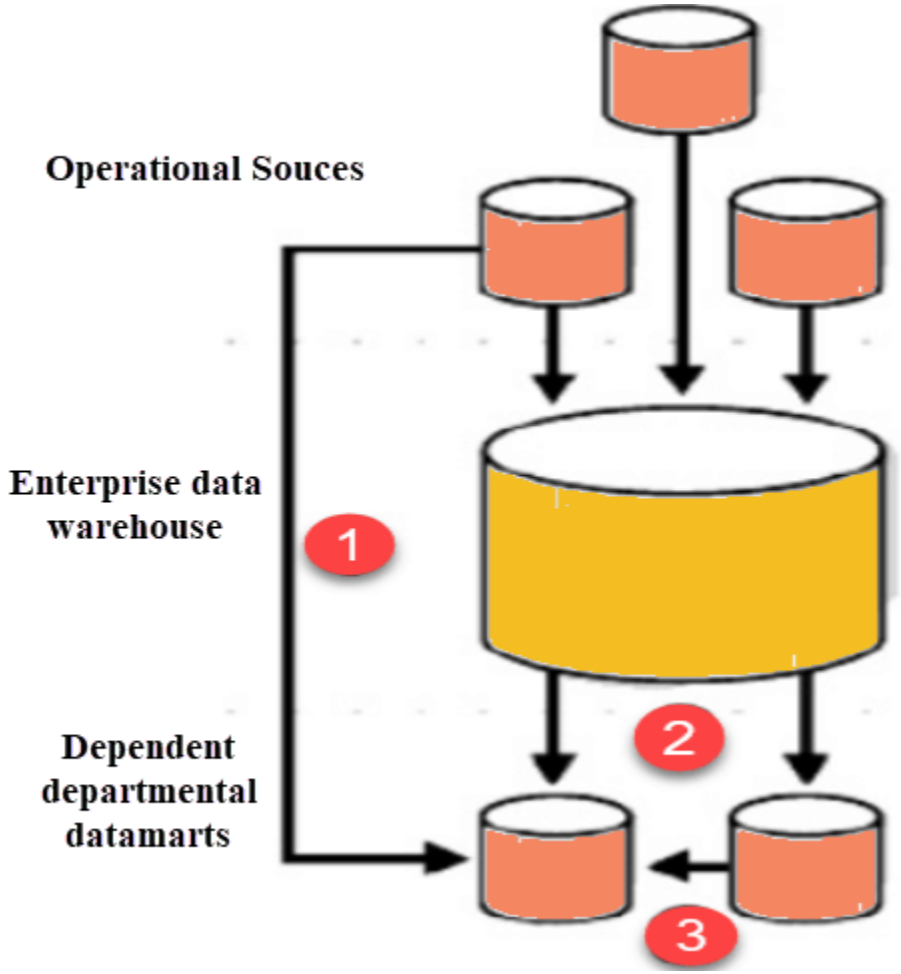
Independent Data Mart

In this, the data mart is built by drawing from operational or external sources of data or both.



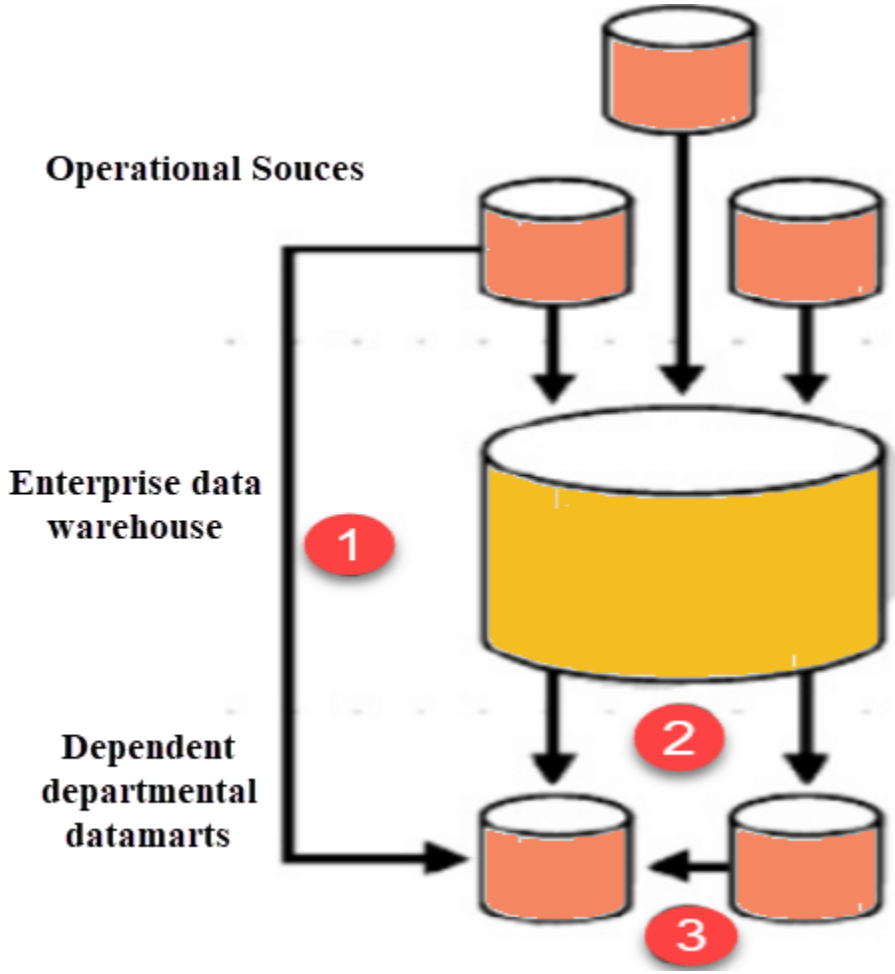
Hybrid Data Mart

A hybrid data mart combines input from sources apart from Data warehouse.



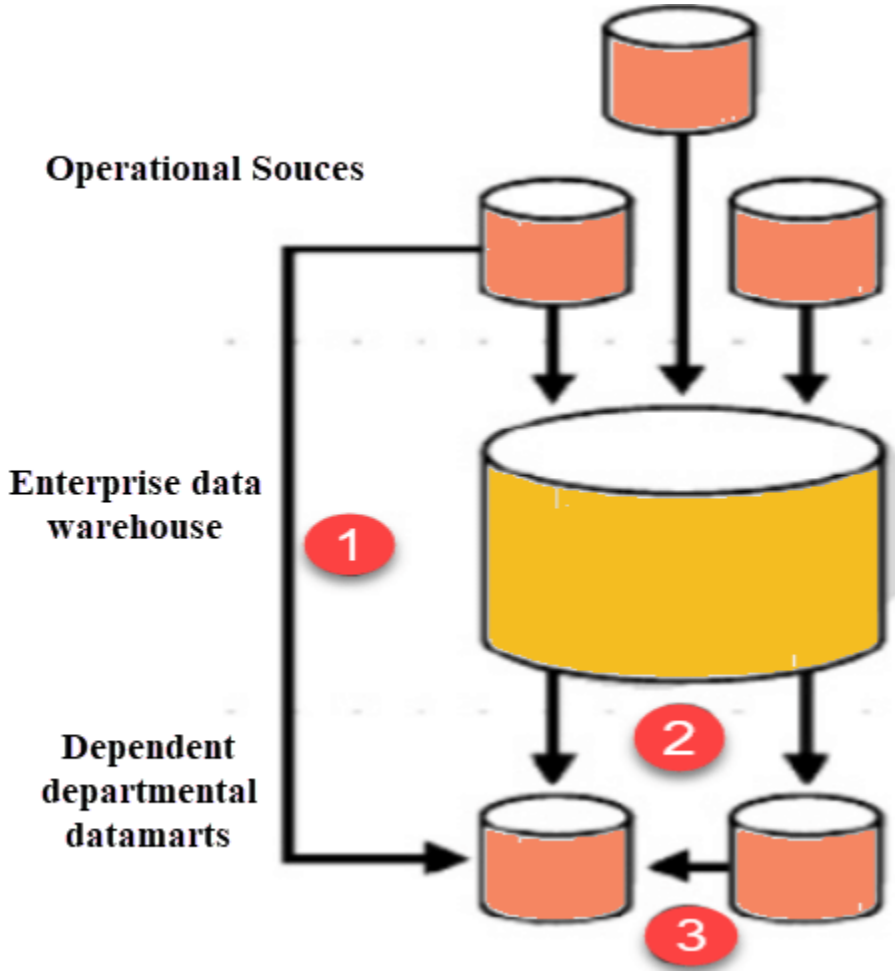
Hybrid Data Mart

It also requires least data cleansing effort.



Hybrid Data Mart

Hybrid Data mart also supports large storage structures, and it is best suited for flexible for smaller data-centric applications.



Advantages of Data mart

- Simpler, more focused & flexible.
- Low cost for both h/w and s/w.
- Faster and cheaper to build.
- Stores data closer that enhances performance.

Advantages of Data mart

- Simpler, more focused & flexible.
- Low cost for both h/w and s/w.
- Faster and cheaper to build.
- Stores data closer that enhances performance.

Advantages of Data mart

- Simpler, more focused & flexible.
- Low cost for both h/w and s/w.
- Faster and cheaper to build.
- Stores data closer that enhances performance.

Advantages of Data mart

- Simpler, more focused & flexible.
- Low cost for both h/w and s/w.
- Faster and cheaper to build.
- Stores data closer that enhances performance.

Disadvantages of data mart

- Many a times enterprises create too many disparate and unrelated data marts without much benefit. It can become a big hurdle to maintain.
- Data Mart cannot provide company-wide data analysis as their data set is limited.
- Unorganized development

Disadvantages of data mart

- Many a times enterprises create too many disparate and unrelated data marts without much benefit. It can become a big hurdle to maintain.
- Data Mart cannot provide company-wide data analysis as their data set is limited.
- Unorganized development

Disadvantages of data mart

- Many a times enterprises create too many disparate and unrelated data marts without much benefit. It can become a big hurdle to maintain.
- Data Mart cannot provide company-wide data analysis as their data set is limited.
- Unorganized development

Disadvantages of data mart

- Increase in datamart size leads to problems such as performance degradation, data inconsistency.
- Big hurdle to maintain.
- Data set is limited.
- Unorganized development
- Increase in datamart size leads to problems such as performance degradation, data inconsistency.

Disadvantages of data mart

- Increase in datamart size leads to problems such as performance degradation, data inconsistency.
- Big hurdle to maintain.
- Data set is limited.
- Unorganized development
- Increase in datamart size leads to problems such as performance degradation, data inconsistency.

Disadvantages of data mart

- Increase in datamart size leads to problems such as performance degradation, data inconsistency.
- Big hurdle to maintain.
- Data set is limited.
- Unorganized development
- Increase in datamart size leads to problems such as performance degradation, data inconsistency.

Disadvantages of data mart

- Increase in datamart size leads to problems such as performance degradation, data inconsistency.
- Big hurdle to maintain.
- Data set is limited.
- Unorganized development
- Increase in datamart size leads to problems such as performance degradation, data inconsistency.

Disadvantages of data mart

- Increase in datamart size leads to problems such as performance degradation, data inconsistency.
- Big hurdle to maintain.
- Data set is limited.
- Unorganized development
- Increase in datamart size leads to problems such as performance degradation, data inconsistency.

Difference between Data warehouse and Data mart

Data Warehouse	Data Mart
Data warehouse is a Centralized system.	While it is a Decentralized system
In data warehouse, lightly denormalization takes place.	While in Data mart, highly denormalization takes place
Data warehouse is top-down model.	While it is a bottom-up model.
To built a warehouse is difficult.	While to build a mart is easy

Difference between Data warehouse and Data mart

Data Warehouse	Data Mart
In data warehouse, Fact constellation schema is used.	While in this, Star schema and snowflake schema are used.
Data Warehouse is flexible.	While it is not flexible.
Data Warehouse is the data-oriented in nature.	While it is the project-oriented in nature.
Data Warehouse has long life.	While data-mart has short life than warehouse.

Difference between Data warehouse and Data mart

Data Warehouse	Data Mart
In Data Warehouse, Data are contained in detail form.	While in this, data are contained in summarized form.
Data Warehouse is vast in size.	While data mart is smaller than warehouse.


That's all for now...