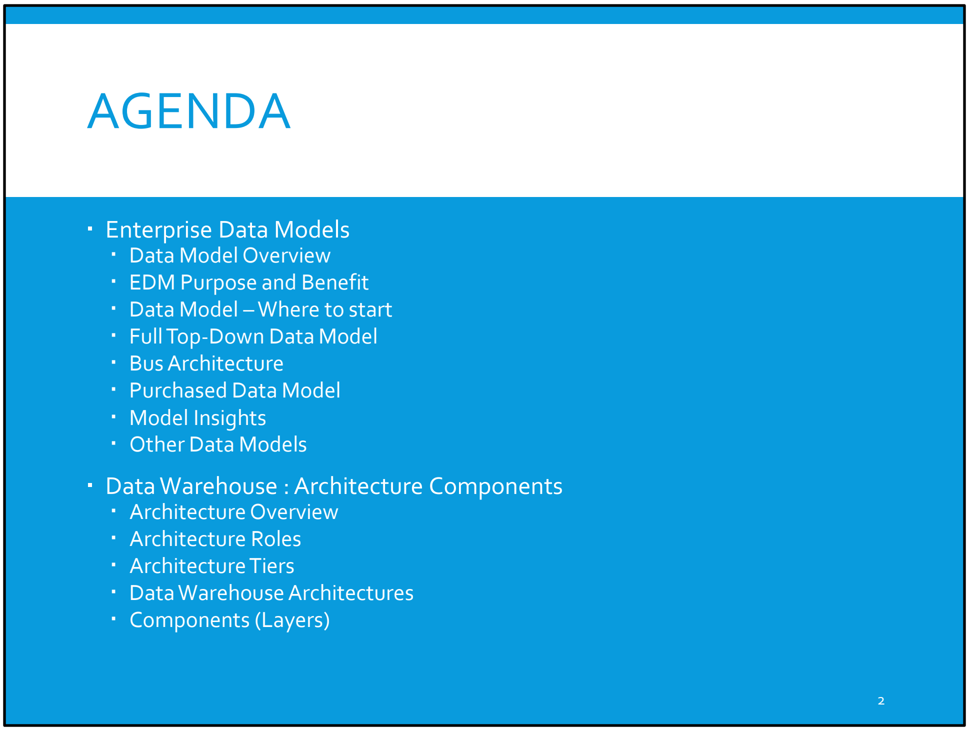
Colorado Technical University

Colorado Technical University
Instructor: Dr. John Conklin
Unit 3 – Load Data



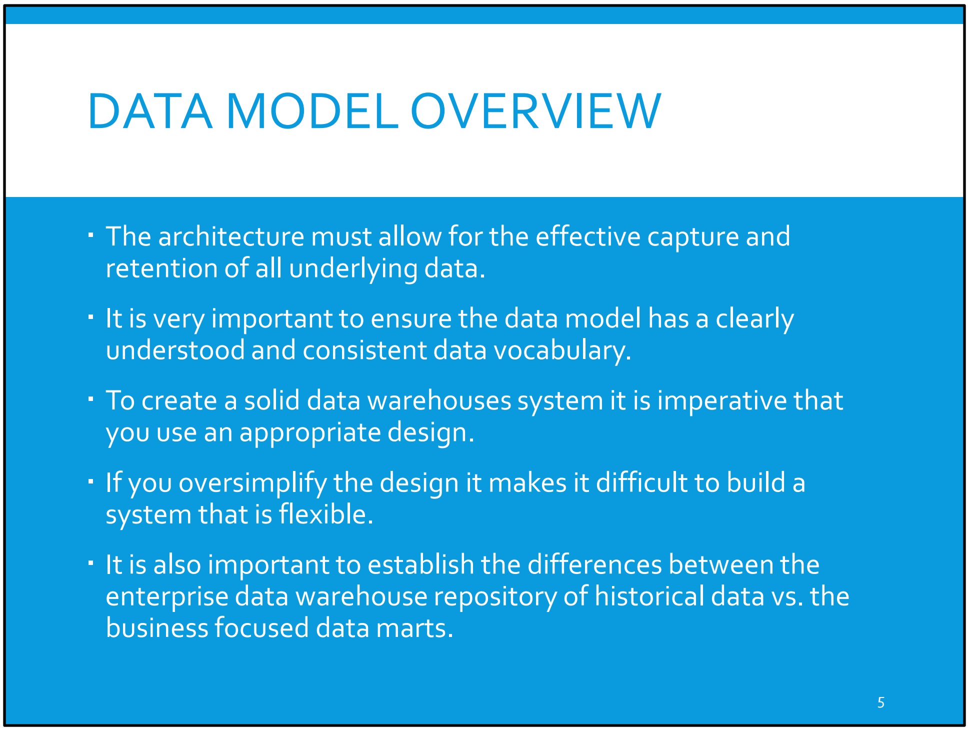
AGENDA

- Enterprise Data Models
  - Data Model Overview
  - EDM Purpose and Benefit
  - Data Model – Where to start
  - Full Top-Down Data Model
  - Bus Architecture
  - Purchased Data Model
  - Model Insights
  - Other Data Models
- Data Warehouse : Architecture Components
  - Architecture Overview
  - Architecture Roles
  - Architecture Tiers
  - Data Warehouse Architectures
  - Components (Layers)

AGENDA (CONT.)

- Data Warehouse : Architecture Components (cont.)
  - Implementation Approaches
- ETL and Data Quality
  - Architecture
  - Operational Data Store (ODS)
  - Source Systems
  - Transformation and Staging
  - Loading

ENTERPRISE DATA MODELS



DATA MODEL OVERVIEW

- The architecture must allow for the effective capture and retention of all underlying data.
- It is very important to ensure the data model has a clearly understood and consistent data vocabulary.
- To create a solid data warehouses system it is imperative that you use an appropriate design.
- If you oversimplify the design it makes it difficult to build a system that is flexible.
- It is also important to establish the differences between the enterprise data warehouse repository of historical data vs. the business focused data marts.

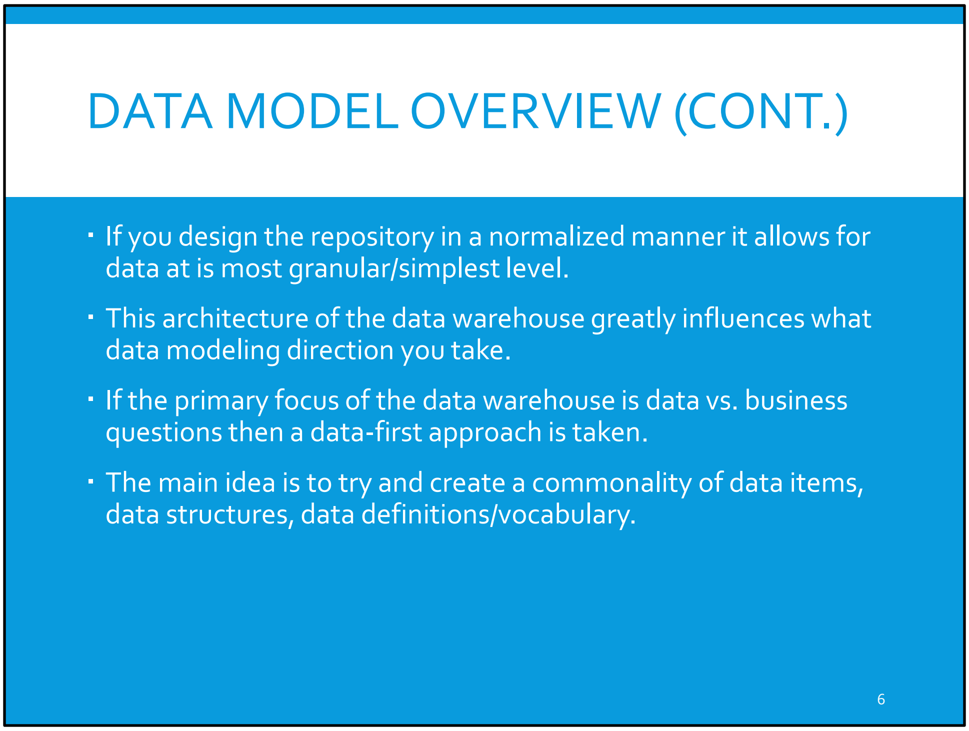
The architecture must allow for the effective capture and retention of all underlying data and must be organized in a way that is conducive to this activity.

It is very important to ensure the data model has a clearly understood and consistent data vocabulary, no matter what model is used.

To create a solid data warehouses system it is imperative that you use an appropriate design, and is the key to a usable and accepted business solution.

If you oversimplify the design it makes it difficult to build a system that is flexible and has an open environment that can grow with future efforts.

It is also important to establish the differences between the enterprise data warehouse repository of historical data vs. the business focused data marts.



DATA MODEL OVERVIEW (CONT.)

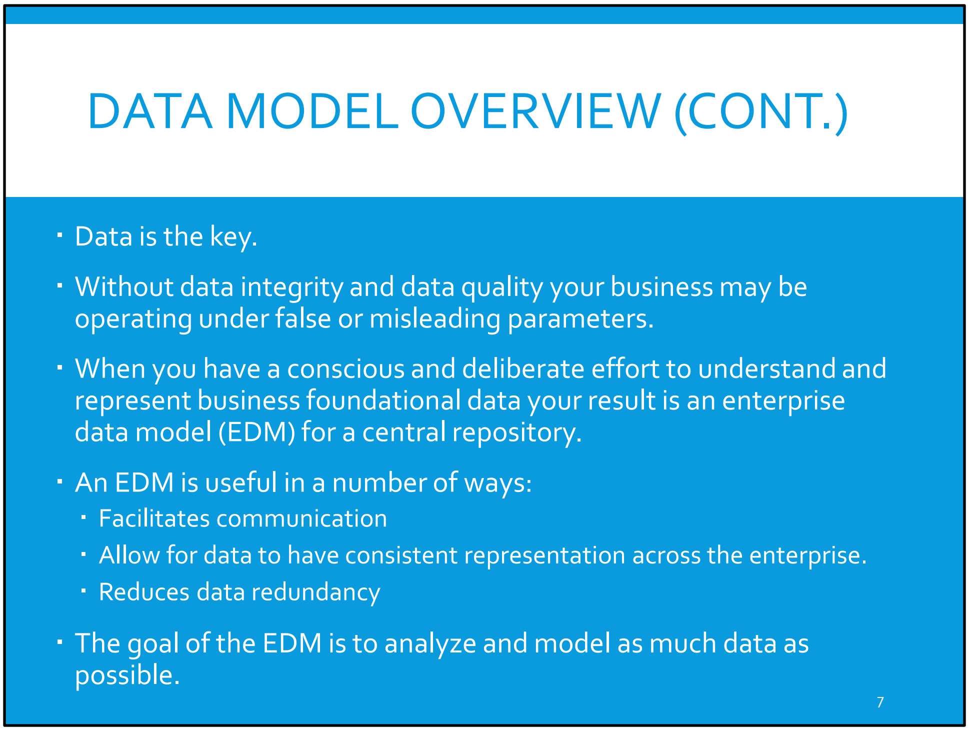
- If you design the repository in a normalized manner it allows for data at is most granular/simplest level.
- This architecture of the data warehouse greatly influences what data modeling direction you take.
- If the primary focus of the data warehouse is data vs. business questions then a data-first approach is taken.
- The main idea is to try and create a commonality of data items, data structures, data definitions/vocabulary.

If you design the repository in a normalized manner it allows for data at is most granular/simplest level, which in turns allows for full flexibility of all fundamental data components and how they are related.

This architecture of the data warehouse greatly influences what data modeling direction you take.

If the primary focus of the data warehouse is data vs. business questions then a data-first approach is taken (aka -> Repository Architecture)

The main idea is to try and create a commonality of data items, data structures, data definitions/vocabulary, thus limiting the business and department data-sharing incompatibilities.



DATA MODEL OVERVIEW (CONT.)

- Data is the key.
- Without data integrity and data quality your business may be operating under false or misleading parameters.
- When you have a conscious and deliberate effort to understand and represent business foundational data your result is an enterprise data model (EDM) for a central repository.
- An EDM is useful in a number of ways:
  - Facilitates communication
  - Allow for data to have consistent representation across the enterprise.
  - Reduces data redundancy
- The goal of the EDM is to analyze and model as much data as possible.

Data is the key.

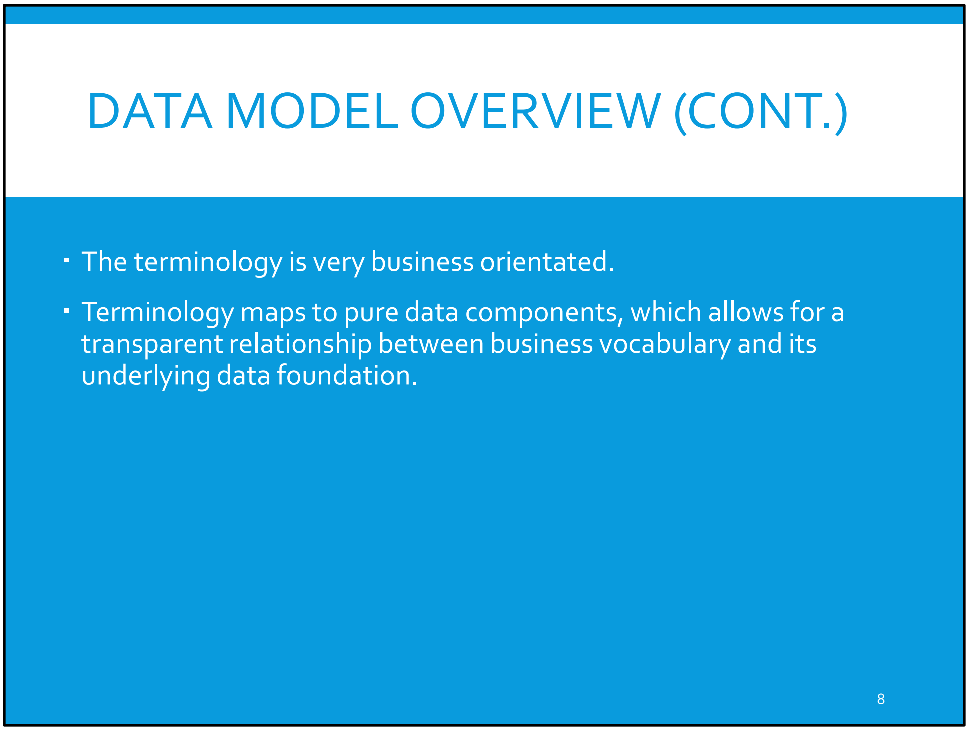
Without data integrity and data quality your business may be operating under false or misleading parameters and limiting growth potential.

When you have a conscious and deliberate effort to understand and represent business foundational data your result is an enterprise data model (EDM) for a central repository.

An EDM is useful in a number of ways:

- Facilitates communication – IT & Business
- Modeling standards allow for data to have consistent representation across the enterprise.
- Reduces data redundancy

The goal of the EDM is to analyze and model as much data as possible, which will result in an enterprise vocabulary and conformed data structures.



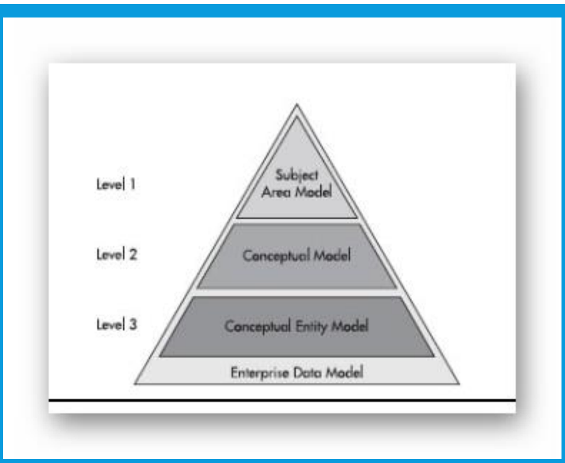
DATA MODEL OVERVIEW (CONT.)

- The terminology is very business orientated.
- Terminology maps to pure data components, which allows for a transparent relationship between business vocabulary and its underlying data foundation.

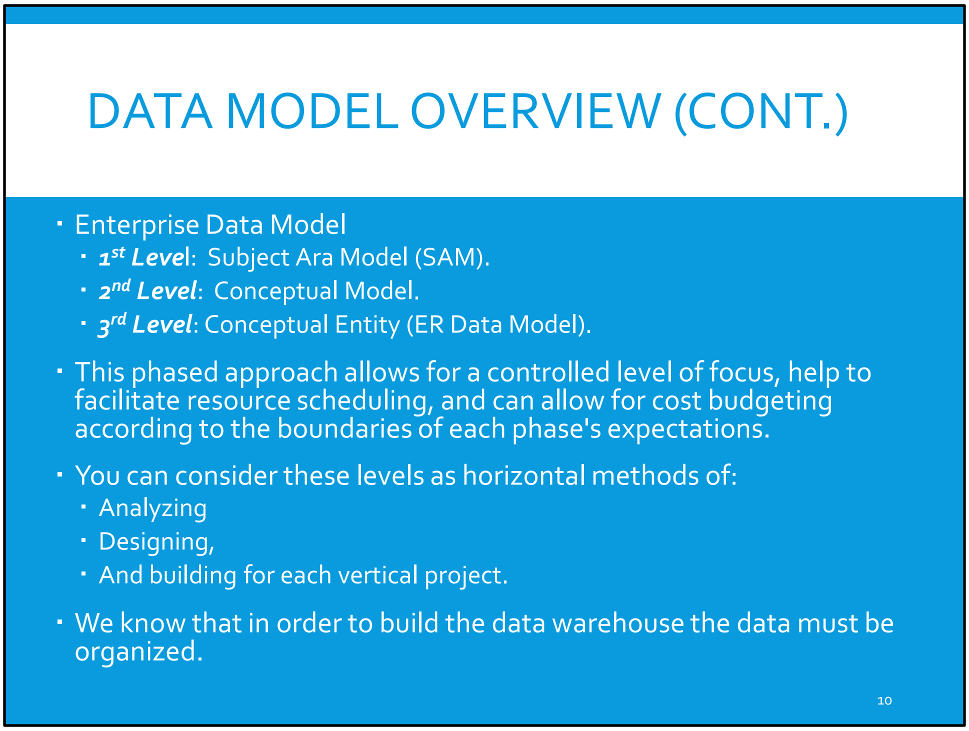
At the beginning the terminology is very business orientated.

Eventually this terminology maps to pure data components, which allows for a transparent relationship between business vocabulary and its underlying data foundation.

DATA MODEL OVERVIEW (CONT.)



(Laberge, 2011)



DATA MODEL OVERVIEW (CONT.)

- Enterprise Data Model
  - ***1st Leve***l: Subject Ara Model (SAM).
  - ***2nd Level***: Conceptual Model.
  - ***3rd Level***: Conceptual Entity (ER Data Model).
- This phased approach allows for a controlled level of focus, help to facilitate resource scheduling, and can allow for cost budgeting according to the boundaries of each phase's expectations.
- You can consider these levels as horizontal methods of:
  - Analyzing
  - Designing,
  - And building for each vertical project.
- We know that in order to build the data warehouse the data must be organized.

10

Enterprise Data Model

1st Level: Subject Ara Model (SAM). Breaks down the organizations business into 10 to 25 chunks.

2nd Level: Conceptual Model. Further breaks downs business into more notable components.

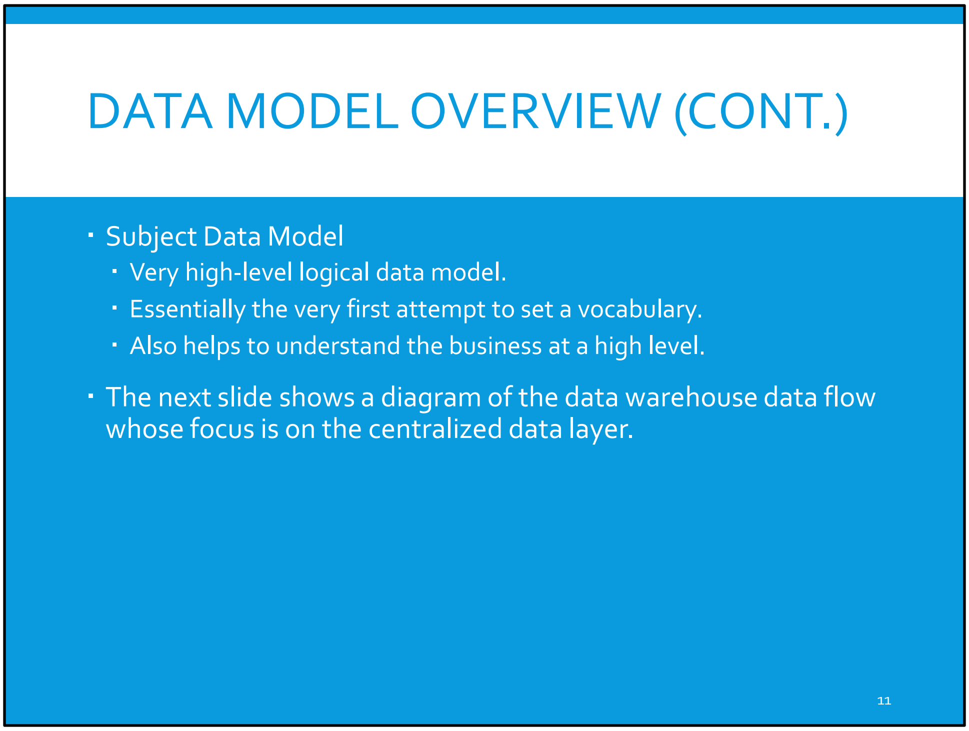
3rd Level: Conceptual Entity (ER Data Model). Continues to subdividing the conceptual model into a model that is usable for the data modelers.

Having this phased approach allows for a controlled level of focus, help to facilitate resource scheduling, and can allow for cost budgeting according to the boundaries of each phase's expectations.

You can consider these levels as horizontal methods of:

- Analyzing
- Designing,
- And building for each vertical project.

With all this in mind we know that in order to build the data warehouse the data must be organized.



DATA MODEL OVERVIEW (CONT.)

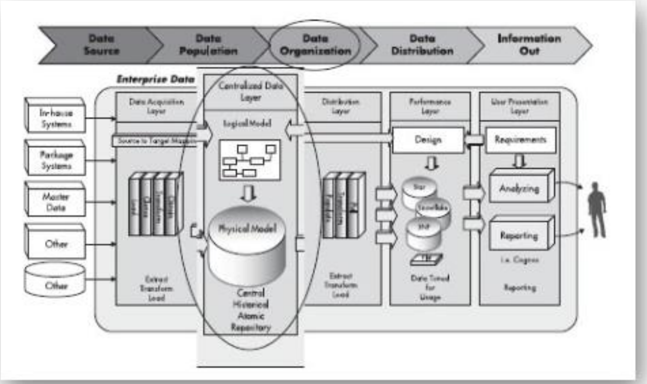
- Subject Data Model
  - Very high-level logical data model.
  - Essentially the very first attempt to set a vocabulary.
  - Also helps to understand the business at a high level.
- The next slide shows a diagram of the data warehouse data flow whose focus is on the centralized data layer.

Subject Data Model

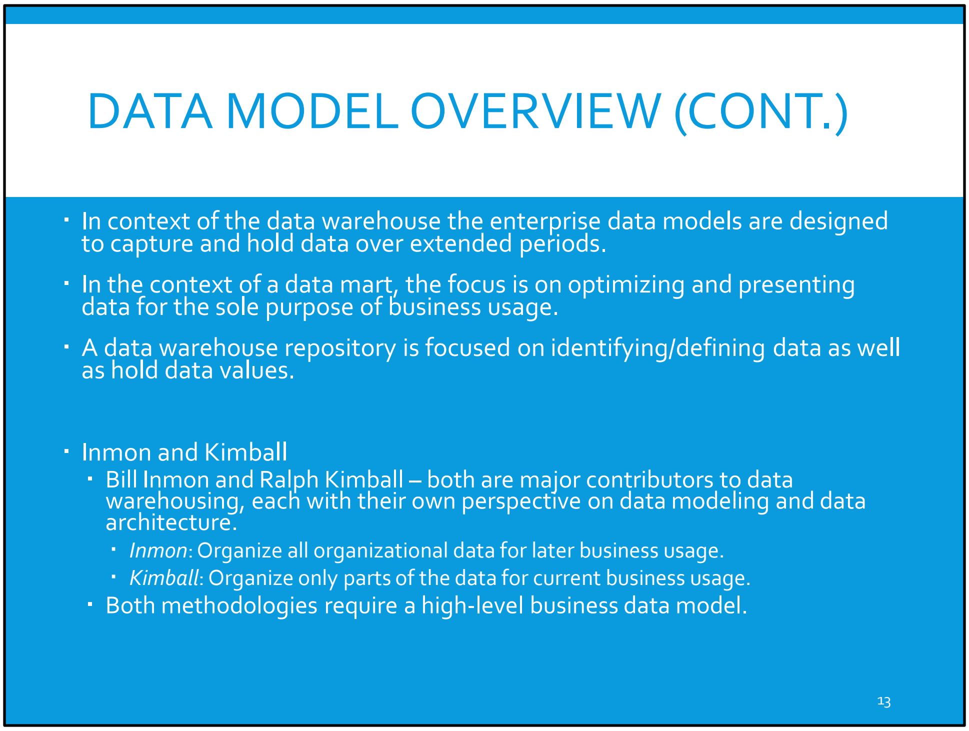
- Very high-level logical data model.
- Essentially the very first attempt to set a vocabulary.
- Also helps to understand the business at a high level.

The next slide shows a diagram of the data warehouse data flow whose focus is on the centralized data layer. Here the idea is that data is to be centralized in terminology, definition, structure and values, which then allows for a common enterprise-wide understanding of data.

DATA MODEL OVERVIEW (CONT.)



(Laberge, 2011)



DATA MODEL OVERVIEW (CONT.)

- In context of the data warehouse the enterprise data models are designed to capture and hold data over extended periods.
- In the context of a data mart, the focus is on optimizing and presenting data for the sole purpose of business usage.
- A data warehouse repository is focused on identifying/defining data as well as hold data values.
- Inmon and Kimball
  - Bill Inmon and Ralph Kimball – both are major contributors to data warehousing, each with their own perspective on data modeling and data architecture.
    - *Inmon*: Organize all organizational data for later business usage.
    - *Kimball*: Organize only parts of the data for current business usage.
  - Both methodologies require a high-level business data model.

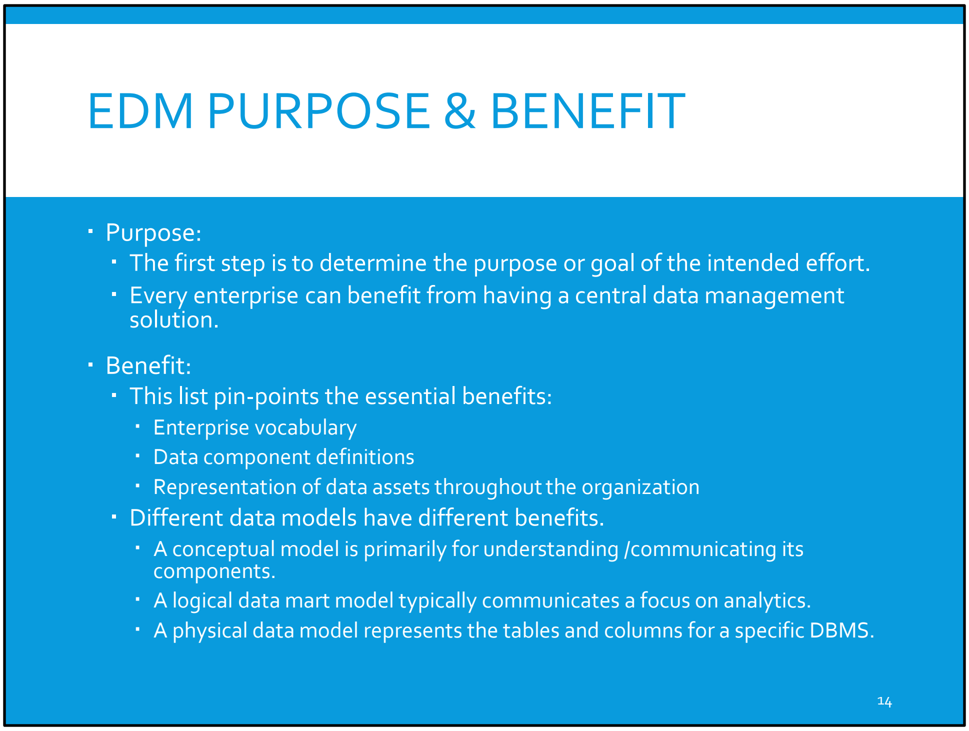
In context of the data warehouse the enterprise data models are designed to capture and hold data over extended periods.

In the context of a data mart, the focus is on optimizing and presenting data for the sole purpose of business usage.

A data warehouse repository is focused on identifying/defining data as well as hold data values that are to be extracted at a later time into a structure that is more appropriate for specific business usage.

Inmon and Kimball

- Bill Inmon and Ralph Kimball – both are major contributors to data warehousing, each with their own perspective on data modeling and data architecture.
  - *Inmon*: His view is to organize all organizational data for later business usage.
  - *Kimball*: His view is to organize only parts of the data for current business usage.
- Both methodologies require a high-level business data model and the consideration of the data as an enterprise asset.



EDM PURPOSE & BENEFIT

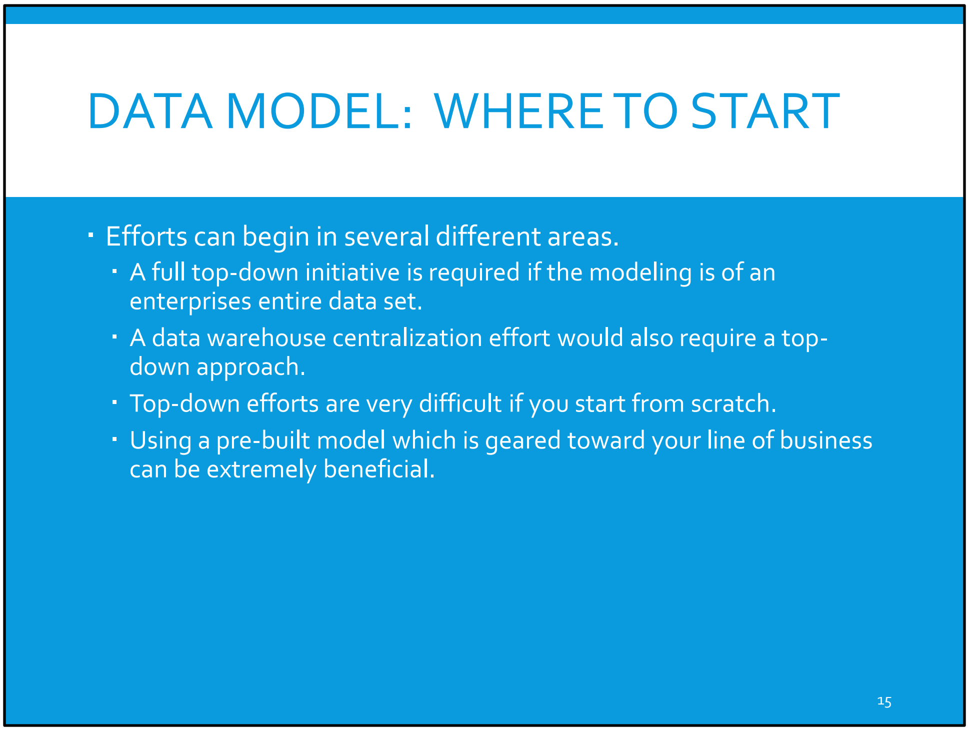
- Purpose:
  - The first step is to determine the purpose or goal of the intended effort.
  - Every enterprise can benefit from having a central data management solution.
- Benefit:
  - This list pin-points the essential benefits:
    - Enterprise vocabulary
    - Data component definitions
    - Representation of data assets throughout the organization
  - Different data models have different benefits.
    - A conceptual model is primarily for understanding /communicating its components.
    - A logical data mart model typically communicates a focus on analytics.
    - A physical data model represents the tables and columns for a specific DBMS.

Purpose:

- As with any other data processing effort, the first step is to determine the purpose or goal of the intended effort.
- Every enterprise can benefit from having a central data management solution, where the key is to plan in focused stages given the timeline, budget and resources.

Benefit:

- This list pin-points the essential benefits:
  - Consistent enterprise vocabulary
  - Consistent data component definitions
  - Consistent representation of data assets throughout the organization
- Different data models have different benefits.
  - A conceptual model is primarily for understanding /communicating its components.
  - A logical data mart model typically communicates a focus on analytics.
  - A physical data model represents the tables and columns for a specific DBMS



DATA MODEL: WHERE TO START

- Efforts can begin in several different areas.
  - A full top-down initiative is required if the modeling is of an enterprises entire data set.
  - A data warehouse centralization effort would also require a top-down approach.
  - Top-down efforts are very difficult if you start from scratch.
  - Using a pre-built model which is geared toward your line of business can be extremely beneficial.

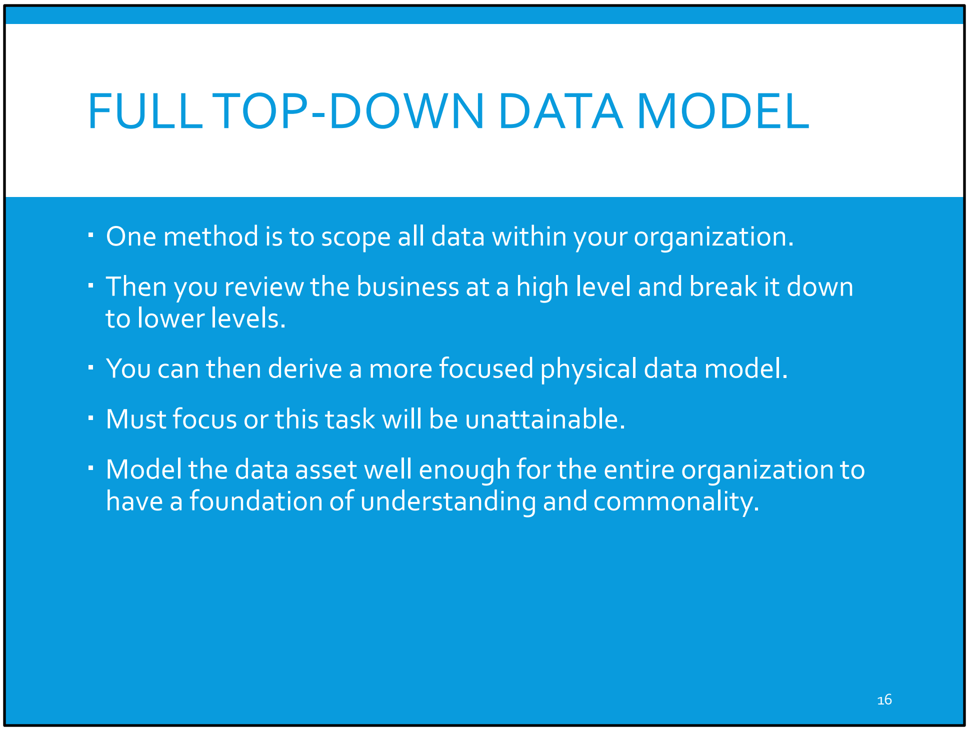
Efforts can begin in several different areas, depending on the data model expertise within the internal data modeling team.

- A full top-down initiative is required if the modeling is of an enterprises entire data set.

- A data warehouse centralization effort would also require a top-down approach.

- Top-down efforts are very difficult if you start from scratch.

- Using a pre-built model which is geared toward your line of business can be extremely beneficial.



FULL TOP-DOWN DATA MODEL

- One method is to scope all data within your organization.
- Then you review the business at a high level and break it down to lower levels.
- You can then derive a more focused physical data model.
- Must focus or this task will be unattainable.
- Model the data asset well enough for the entire organization to have a foundation of understanding and commonality.

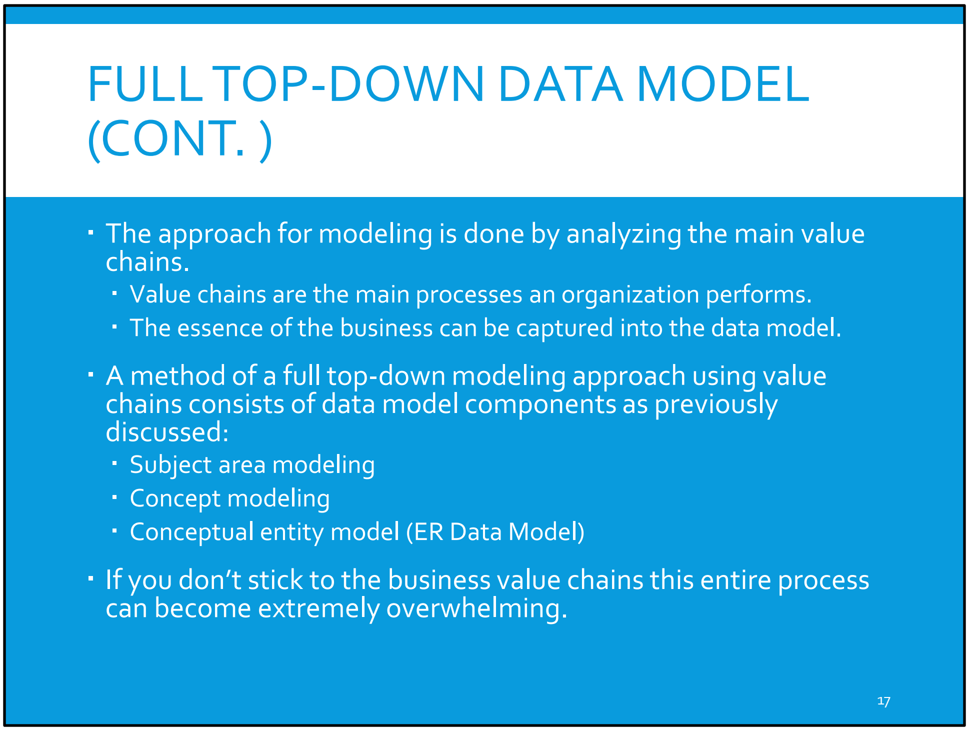
- One method is to scope all data within your organization, which can take years to accomplish.

- Then you review the business at a high level and break it down to lower levels until you reach an entity relationship model.

- From this logical data model you can then derive a more focused physical data model.

- To undertake this effort for the entire enterprise you must focus or this task will be unattainable.

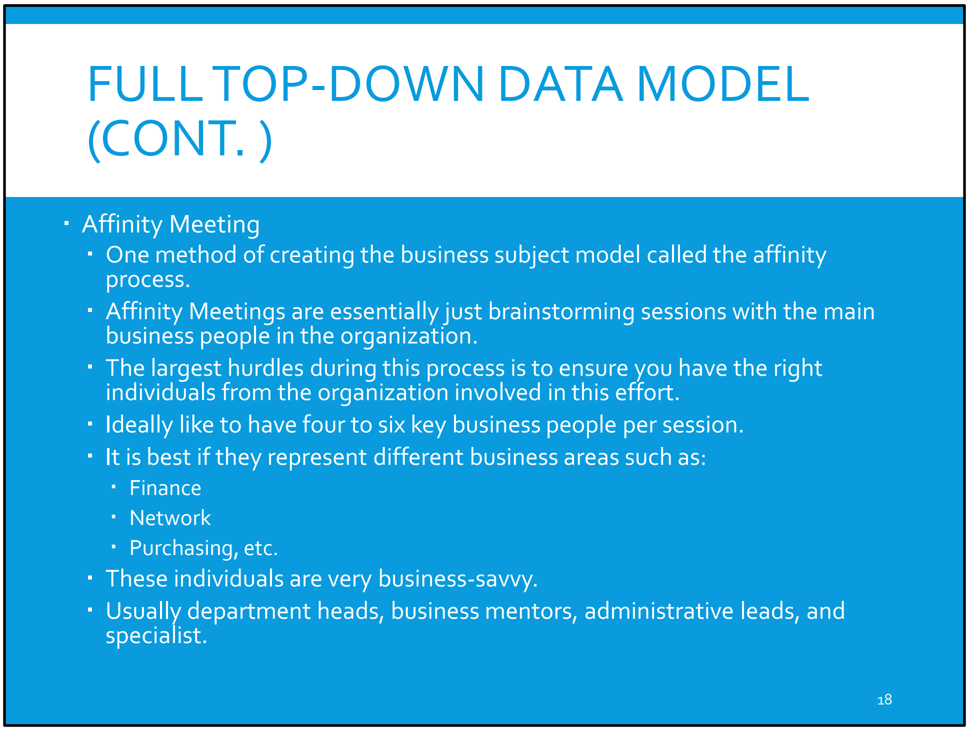
- The idea here is to model the data asset well enough for the entire organization to have a foundation of understanding and commonality.



FULL TOP-DOWN DATA MODEL (CONT.)

- The approach for modeling is done by analyzing the main value chains.
  - Value chains are the main processes an organization performs.
  - The essence of the business can be captured into the data model.
- A method of a full top-down modeling approach using value chains consists of data model components as previously discussed:
  - Subject area modeling
  - Concept modeling
  - Conceptual entity model (ER Data Model)
- If you don’t stick to the business value chains this entire process can become extremely overwhelming.

- The approach for modeling the organization is done by analyzing the main value chains.
  - Value chains are the main processes an organization performs, which in essence is the core of the business.
  - From these views the essence of the business can be captured into the data model.
- A method of a full top-down modeling approach using value chains consists of data model components as previously discussed:
  - Subject area modeling
  - Concept modeling
  - Conceptual entity model (ER Data Model)
- If you don’t stick to the business value chains this entire process can become extremely overwhelming.



FULL TOP-DOWN DATA MODEL (CONT.)

- Affinity Meeting
  - One method of creating the business subject model called the affinity process.
  - Affinity Meetings are essentially just brainstorming sessions with the main business people in the organization.
  - The largest hurdles during this process is to ensure you have the right individuals from the organization involved in this effort.
  - Ideally like to have four to six key business people per session.
  - It is best if they represent different business areas such as:
    - Finance
    - Network
    - Purchasing, etc.
  - These individuals are very business-savvy.
  - Usually department heads, business mentors, administrative leads, and specialist.

Affinity Meeting

- There is one method of creating the business subject model called the affinity process.

- Affinity Meetings are essentially just brainstorming sessions with the main business people in the organization. This is usually members of the upper management of the organization.

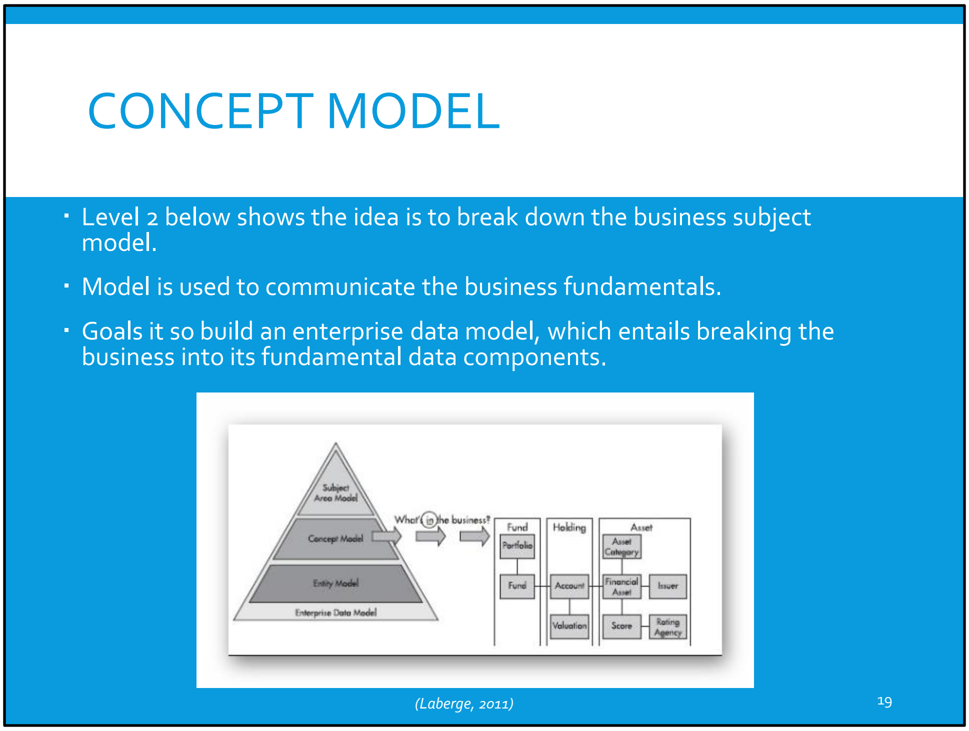
- One of the largest hurdles during this process is to ensure you have the right individuals from the organization involved in this effort.

- You would ideally like to have four to six key business people per session.

- It is best if they represent different business areas such as:
  - Finance
  - Network
  - Purchasing, etc.

- These individuals are very business-savvy, not supporting staff or IT.

- They are usually department heads, business mentors, administrative leads, and specialist.



CONCEPT MODEL

- Level 2 below shows the idea is to break down the business subject model.
- Model is used to communicate the business fundamentals.
- Goals it so build an enterprise data model, which entails breaking the business into its fundamental data components.

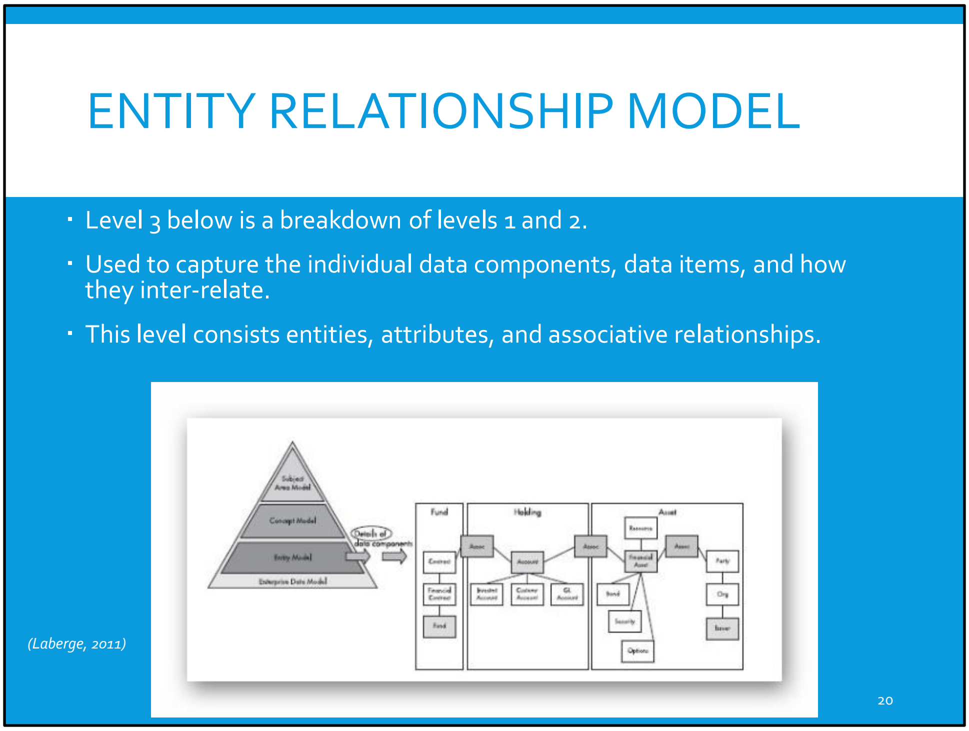
(Laberge, 2011)

19

- Level 2 below shows the idea is to break down the business subject model into more comprehensive components.

- This model is used to communicate the business fundamentals and how the business operates from business and data perspective.

- The goals it so build an enterprise data model, which entails breaking the business into its fundamental data components.



ENTITY RELATIONSHIP MODEL

- Level 3 below is a breakdown of levels 1 and 2.
- Used to capture the individual data components, data items, and how they inter-relate.
- This level consists entities, attributes, and associative relationships.

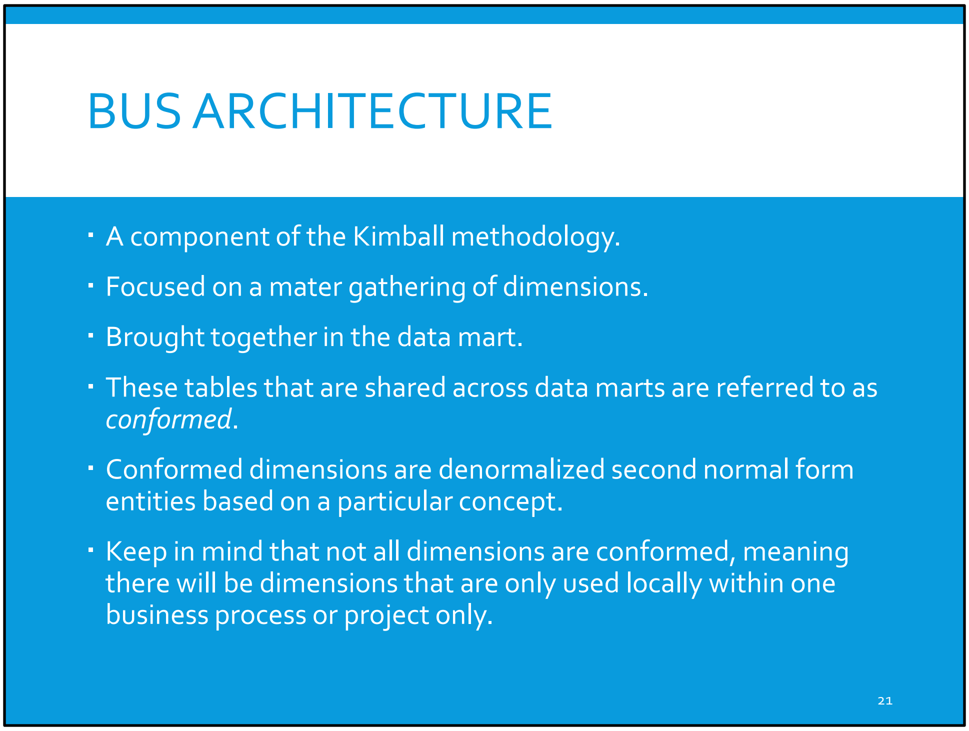
(Laberge, 2011)

20

- Level 3 below is a breakdown of levels 1 and 2 into the business data components at a detailed level.

- Used throughout an organization to capture the individual data components, data items, and how they inter-relate.

- This level consists entities, attributes, and associative relationships.



BUS ARCHITECTURE

- A component of the Kimball methodology.
- Focused on a mater gathering of dimensions.
- Brought together in the data mart.
- These tables that are shared across data marts are referred to as *conformed*.
- Conformed dimensions are denormalized second normal form entities based on a particular concept.
- Keep in mind that not all dimensions are conformed, meaning there will be dimensions that are only used locally within one business process or project only.

- This is a component of the Kimball methodology which is used in creating an enterprise data design for a specific application.

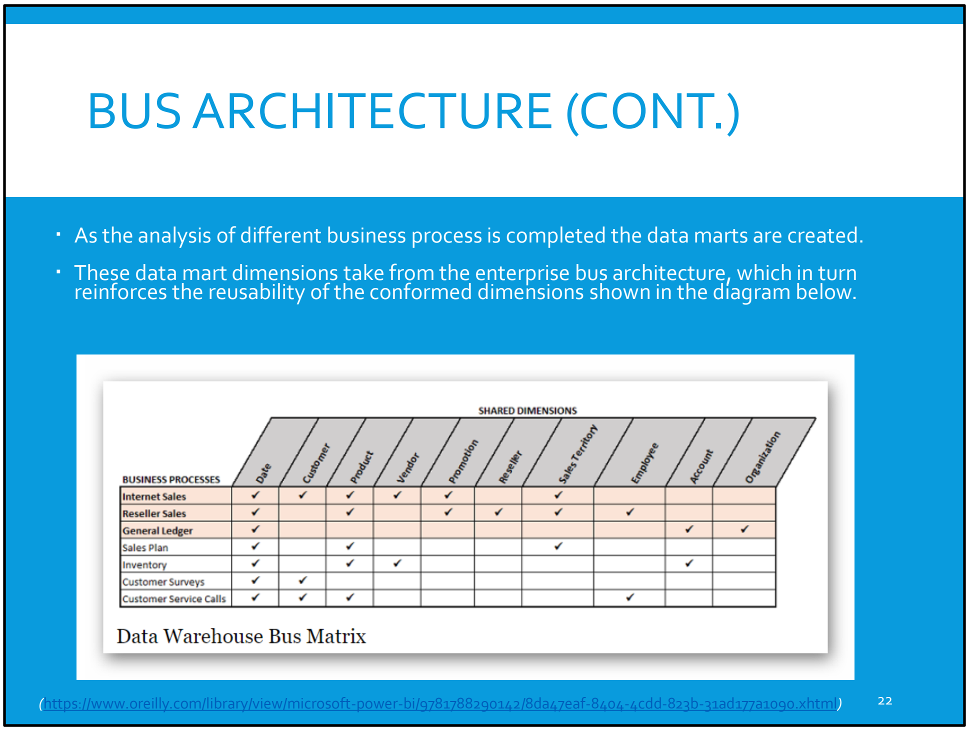
- This idea is focused on a mater gathering of dimensions which can be used and reused throughout the organization to analyze business data.

- These dimension and fact tables are brought together in the data mart.

- These tables that are shared across data marts are referred to as *conformed*. These conformed dimensions then together form the bus architecture.

- These conformed dimensions are denormalized second normal form entities based on a particular concept.

- Keep in mind that not all dimensions are conformed, meaning there will be dimensions that are only used locally within one business process or project only.



BUS ARCHITECTURE (CONT.)

- As the analysis of different business process is completed the data marts are created.
- These data mart dimensions take from the enterprise bus architecture, which in turn reinforces the reusability of the conformed dimensions shown in the diagram below.

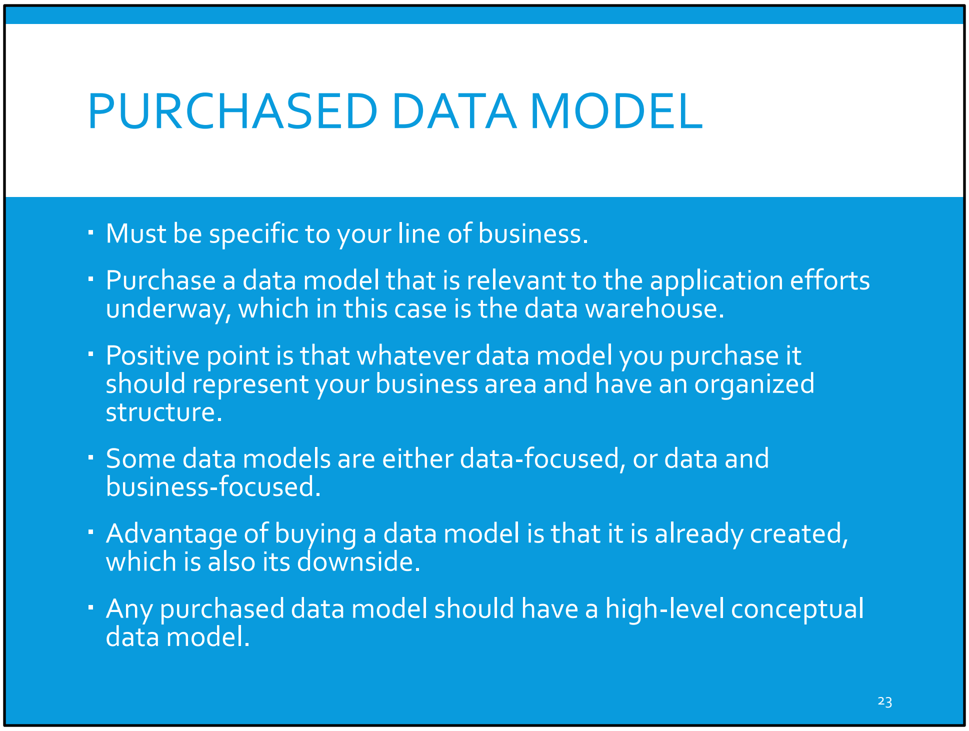
	SHARED DIMENSIONS									
BUSINESS PROCESSES	Date	Customer	Product	Vendor	Promotion	Reseller	Sales Territory	Employee	Account	Organization
Internet Sales	✓	✓	✓	✓	✓		✓			
Reseller Sales	✓		✓		✓	✓	✓	✓		
General Ledger	✓								✓	✓
Sales Plan	✓		✓				✓			
Inventory	✓		✓	✓					✓	
Customer Surveys	✓	✓								
Customer Service Calls	✓	✓	✓					✓		

Data Warehouse Bus Matrix

(https://www.oreilly.com/library/view/microsoft-power-bi/9781788290142/8da47eaf-8404-4cdd-823b-31ad177a1090.xhtml)

- As the analysis of different business process is completed the data marts are created.

- These data mart dimensions take from the enterprise bus architecture, which in turn reinforces the reusability of the conformed dimensions shown in the diagram below.

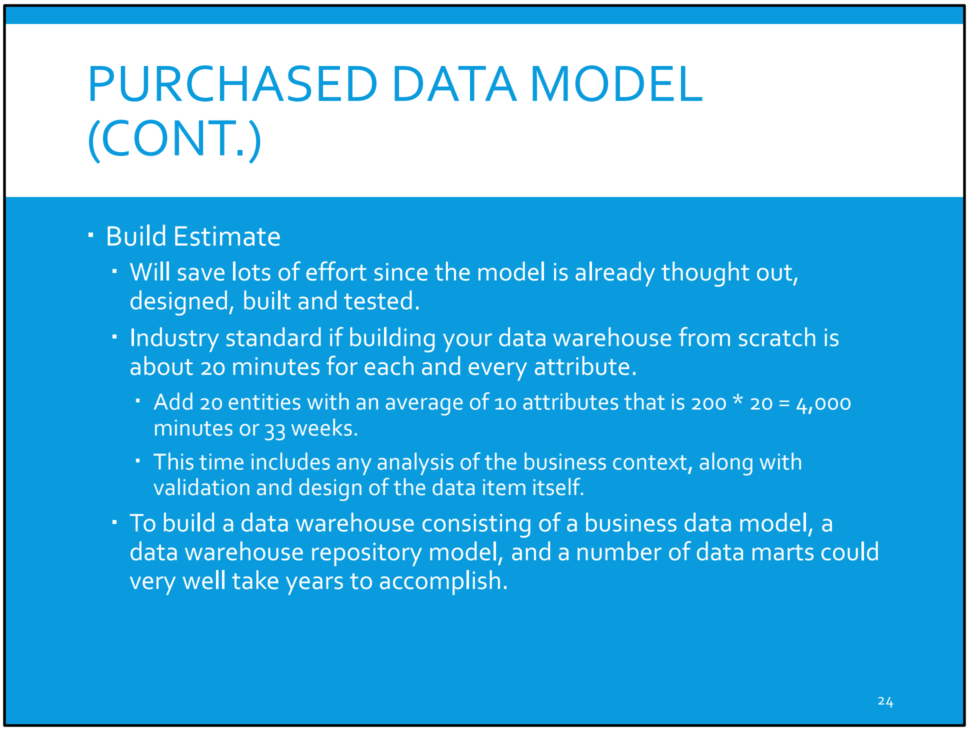


PURCHASED DATA MODEL

- Must be specific to your line of business.
- Purchase a data model that is relevant to the application efforts underway, which in this case is the data warehouse.
- Positive point is that whatever data model you purchase it should represent your business area and have an organized structure.
- Some data models are either data-focused, or data and business-focused.
- Advantage of buying a data model is that it is already created, which is also its downside.
- Any purchased data model should have a high-level conceptual data model.

23

- Must be specific to your line of business, and current project.
- You would want to purchase a data model that is relevant to the application efforts underway, which in this case is the data warehouse.
- The main positive point is that whatever data model you purchase it should represent your business area and have an organized structure.
- Some data models are either data-focused, or data and business-focused.
- The great advantage of buying a data model is that it is already created, which is also its downside.
- Any purchased data model should have a high-level conceptual data model representing the key points within the model and how they relate to the business.



PURCHASED DATA MODEL (CONT.)

- Build Estimate
  - Will save lots of effort since the model is already thought out, designed, built and tested.
  - Industry standard if building your data warehouse from scratch is about 20 minutes for each and every attribute.
    - Add 20 entities with an average of 10 attributes that is 200 * 20 = 4,000 minutes or 33 weeks.
    - This time includes any analysis of the business context, along with validation and design of the data item itself.
  - To build a data warehouse consisting of a business data model, a data warehouse repository model, and a number of data marts could very well take years to accomplish.

Build Estimate

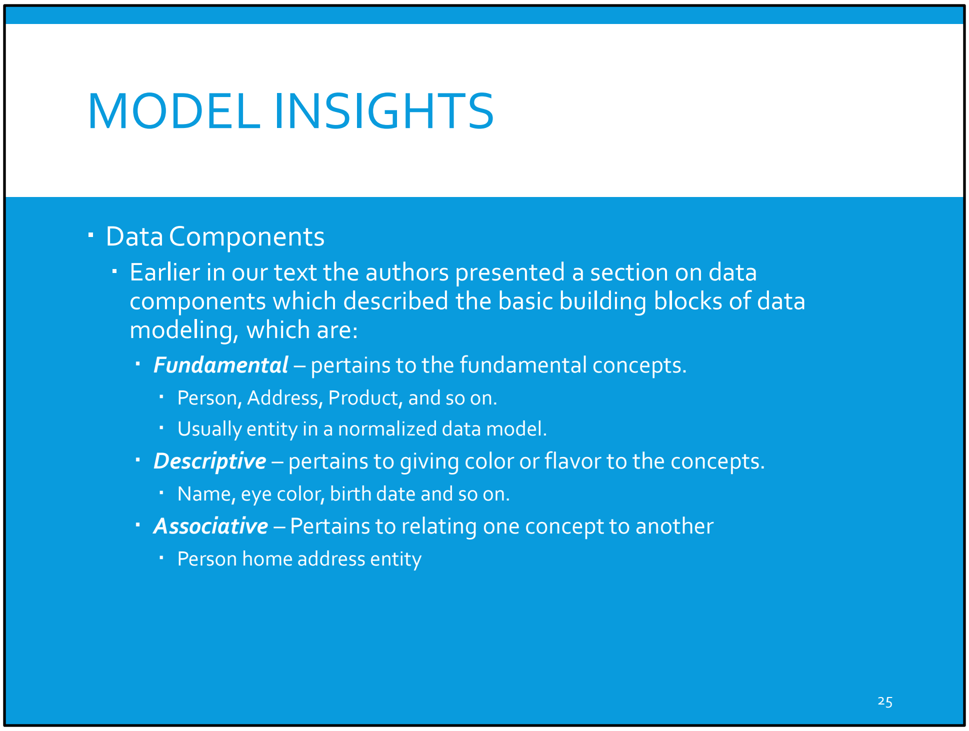
- Using a purchased model will save lots of effort since the model is already thought out, designed, built and tested.

- The industry standard if building your data warehouse from scratch is about 20 minutes for each and every attribute.

  - So to add 20 entities with an average of 10 attributes that is 200 * 20 = 4,000 minutes or 33 weeks.

  - This time includes any analysis of the business context, along with validation and design of the data item itself.

- So in essence to build a data warehouse consisting of a business data model, a data warehouse repository model, and a number of data marts could very well take years to accomplish.



MODEL INSIGHTS

- Data Components
  - Earlier in our text the authors presented a section on data components which described the basic building blocks of data modeling, which are:
    - ***Fundamental*** – pertains to the fundamental concepts.
      - Person, Address, Product, and so on.
      - Usually entity in a normalized data model.
    - ***Descriptive*** – pertains to giving color or flavor to the concepts.
      - Name, eye color, birth date and so on.
    - ***Associative*** – Pertains to relating one concept to another
      - Person home address entity

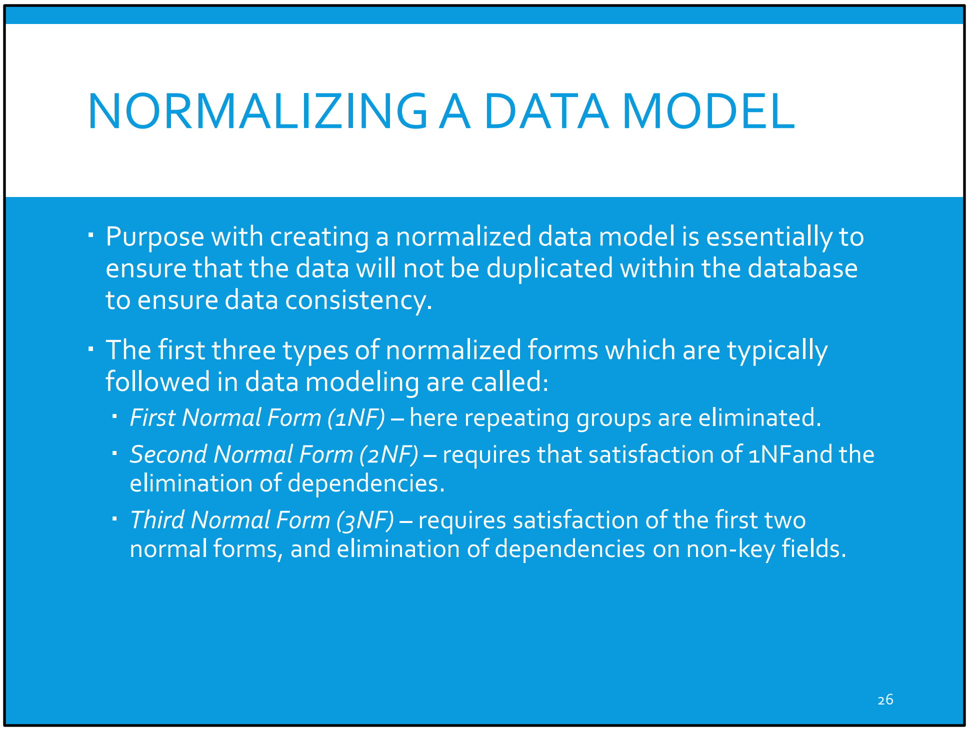
Data Components

- Earlier in our text the authors presented a section on data components which described the basic building blocks of data modeling, which are:

- ***Fundamental*** – pertains to the fundamental concepts.
  - Person, Address, Product, and so on.
  - Usually entity in a normalized data model.

- ***Descriptive*** – pertains to giving color or flavor to the concepts.
  - Name, eye color, birth date and so on.

- ***Associative*** – Pertains to relating one concept to another
  - Person home address entity



NORMALIZING A DATA MODEL

- Purpose with creating a normalized data model is essentially to ensure that the data will not be duplicated within the database to ensure data consistency.
- The first three types of normalized forms which are typically followed in data modeling are called:
  - *First Normal Form (1NF)* – here repeating groups are eliminated.
  - *Second Normal Form (2NF)* – requires that satisfaction of 1NFand the elimination of dependencies.
  - *Third Normal Form (3NF)* – requires satisfaction of the first two normal forms, and elimination of dependencies on non-key fields.

26

- The purpose with creating a normalized data model is essentially to ensure that the data will not be duplicated within the database to ensure data consistency.

- The first three types of normalized forms which are typically followed in data modeling are called:

- *First Normal Form (1NF)* – here repeating groups are eliminated and each row has its own unique identifier (key).

- *Second Normal Form (2NF)* – requires that satisfaction of 1NFand the elimination of dependencies on a partial key by putting the fields into a separate table.

- *Third Normal Form (3NF)* – requires satisfaction of the first two normal forms, and elimination of dependencies on non-key fields by putting them into a separate table. Here all non key fields are dependent on the key entirely.

NORMALIZING A DATA MODEL (CONT.)

Figure 8-11 ***Unnormalized design***

This figure shows a table called ***Person*** which has two columns. Putting these both together in the same table means that if the Company Name is to be used in another context, other than in the Person table, there would be redundancy.

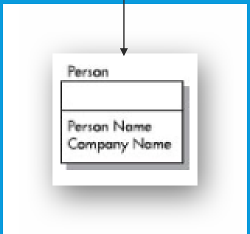
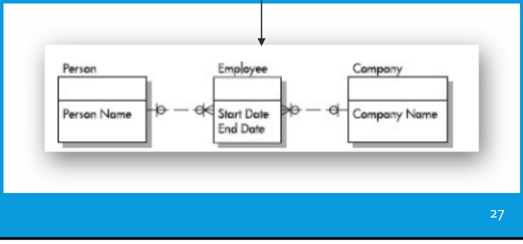
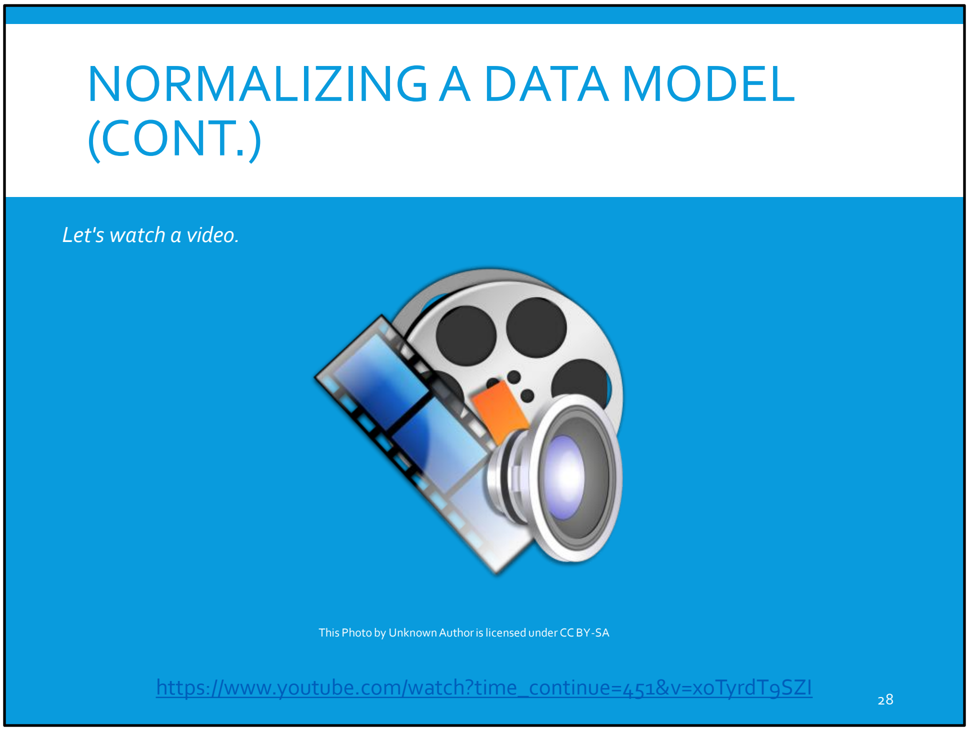


Figure 8-12 N*ormalized design*

This figure shows ***Person*** and ***Company*** tables along with an associative table. This associative tables shows the relationships between the ***Person*** and ***Company*** relationship.



(Laberge, 2011)

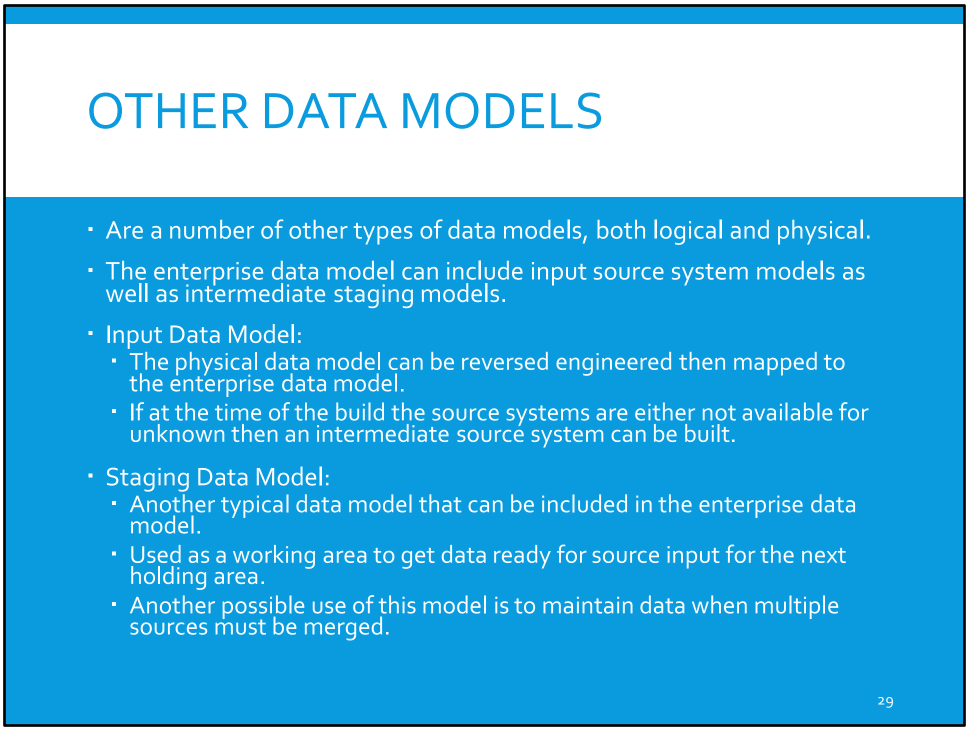


NORMALIZING A DATA MODEL (CONT.)

Let's watch a video.

This Photo by Unknown Author is licensed under CC BY-SA

https://www.youtube.com/watch?time_continue=451&v=xoTyrdT9SZI



OTHER DATA MODELS

- Are a number of other types of data models, both logical and physical.
- The enterprise data model can include input source system models as well as intermediate staging models.
- Input Data Model:
  - The physical data model can be reversed engineered then mapped to the enterprise data model.
  - If at the time of the build the source systems are either not available for unknown then an intermediate source system can be built.
- Staging Data Model:
  - Another typical data model that can be included in the enterprise data model.
  - Used as a working area to get data ready for source input for the next holding area.
  - Another possible use of this model is to maintain data when multiple sources must be merged.

- There are a number of other types of data models, both logical and physical.

- The enterprise data model can include input source system models as well as intermediate staging models.

- Input Data Model:

  - In lieu of an absolute source system data model the physical data model can be reversed engineered then mapped to the enterprise data model.

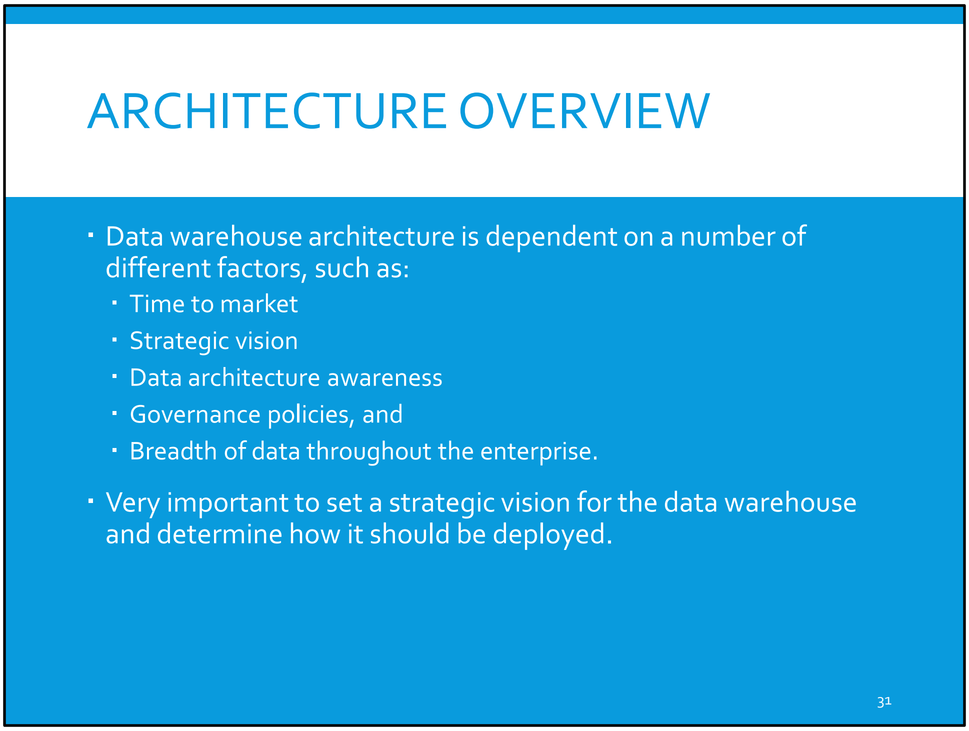
  - If at the time of the build the source systems are either not available for unknown then an intermediate source system can be built.

- Staging Data Model:
  - This is another typical data model that can be included in the enterprise data model.

  - This is used as a working area to get data ready for source input for the next holding area.

  - Another possible use of this model is to maintain data when multiple sources must be merged.

DATA WAREHOUSE ARCHITECTURE: COMPONENTS



ARCHITECTURE OVERVIEW

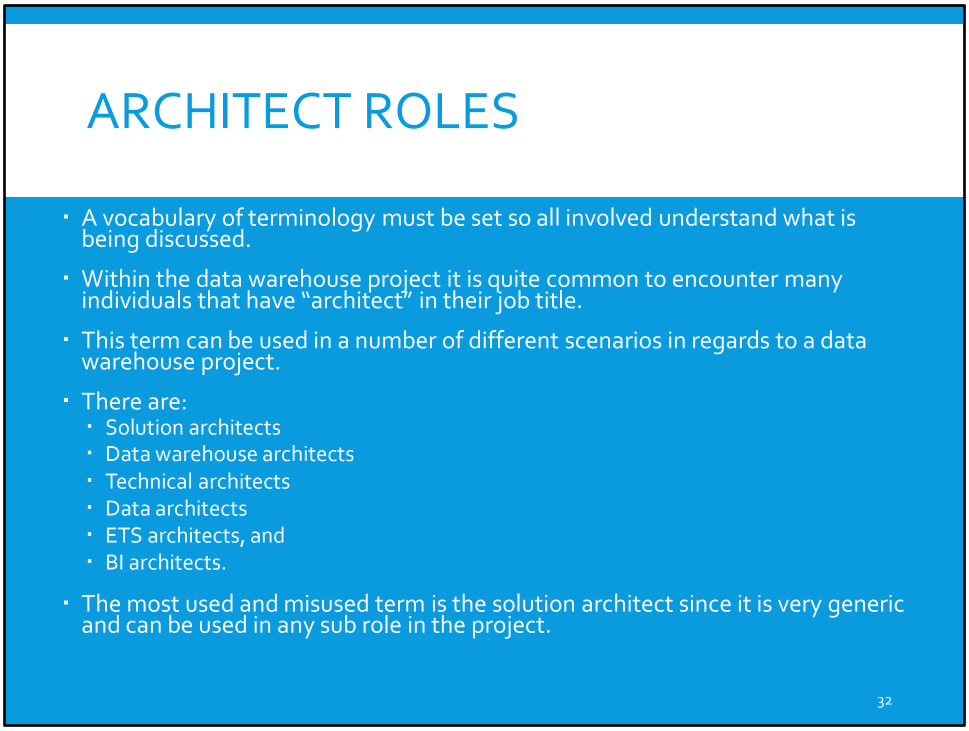
- Data warehouse architecture is dependent on a number of different factors, such as:
  - Time to market
  - Strategic vision
  - Data architecture awareness
  - Governance policies, and
  - Breadth of data throughout the enterprise.
- Very important to set a strategic vision for the data warehouse and determine how it should be deployed.

31

The data warehouse architecture is dependent on a number of different factors, such as:

- Time to market
- Strategic vision
- Data architecture awareness
- Governance policies, and
- Breadth of data throughout the enterprise.

In all cases it is very important to set a strategic vision for the data warehouse and determine how it should be deployed.



ARCHITECT ROLES

- A vocabulary of terminology must be set so all involved understand what is being discussed.
- Within the data warehouse project it is quite common to encounter many individuals that have “architect” in their job title.
- This term can be used in a number of different scenarios in regards to a data warehouse project.
- There are:
  - Solution architects
  - Data warehouse architects
  - Technical architects
  - Data architects
  - ETS architects, and
  - BI architects.
- The most used and misused term is the solution architect since it is very generic and can be used in any sub role in the project.

32

- As a fundamental element of a data governance initiative, a vocabulary of terminology must be set so all involved understand what is being discussed.
- Within the data warehouse project it is quite common to encounter many individuals that have “architect” in their job title.
- This term can be used in a number of different scenarios in regards to a data warehouse project.
- There are:
  - Solution architects
  - Data warehouse architects
  - Technical architects
  - Data architects
  - ETS architects, and
  - BI architects.
- The most used and misused term is the solution architect since it is very generic and can be used in any sub role in the project.

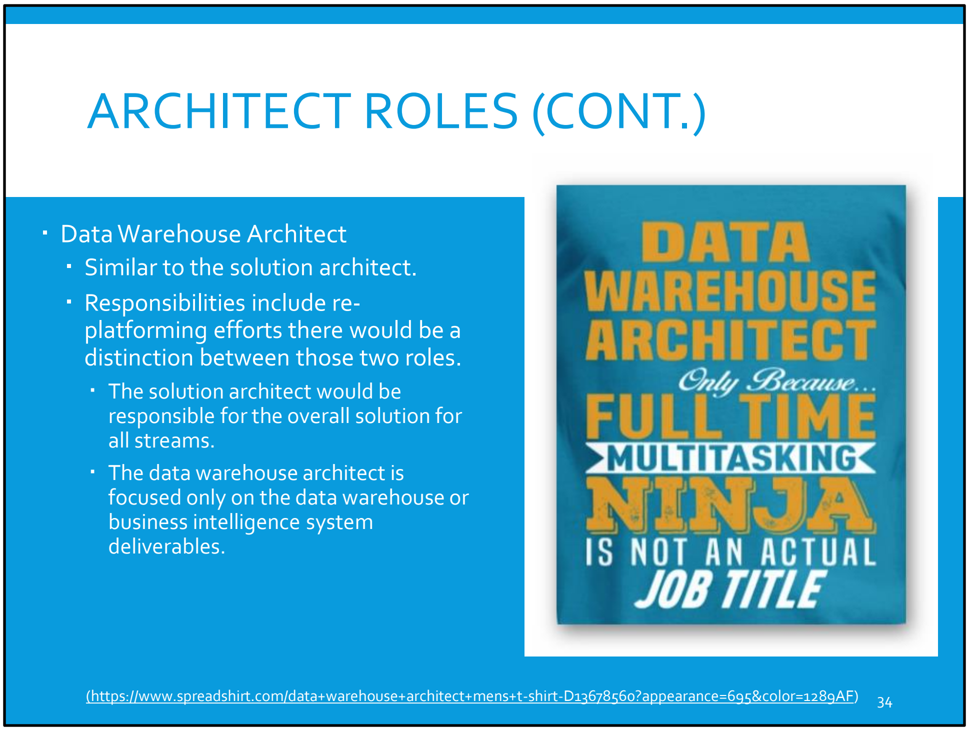
ARCHITECT ROLES (CONT.)

- Solution Architect
  - Responsible for coordinating overall design for a data warehouse.
  - Works with the project manager.
  - Sometimes referred to as the *information architect* or *data warehouse architect.*
  - Includes all technical areas with a goal of making the selected solution come to life.

(https://www.computercareers.org/what-is-the-solution-architect-career-path/)

Solution Architect

- Responsible for coordinating overall design for a data warehouse or business intelligence system.
- Works with the project manager to deliver the overall solution from the requirements to output usage.
- Sometimes referred to as the *information architect* or *data warehouse architect.*
- Includes all technical areas with a goal of making the selected solution come to life.



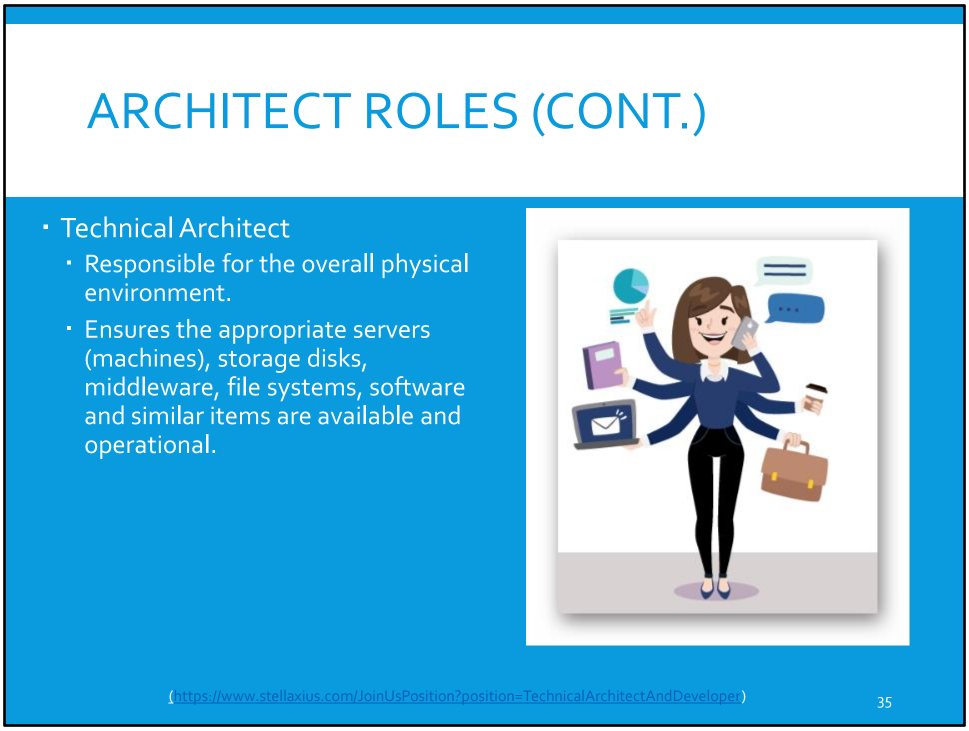
ARCHITECT ROLES (CONT.)

- Data Warehouse Architect
  - Similar to the solution architect.
  - Responsibilities include re-platforming efforts there would be a distinction between those two roles.
    - The solution architect would be responsible for the overall solution for all streams.
    - The data warehouse architect is focused only on the data warehouse or business intelligence system deliverables.

(https://www.spreadshirt.com/data+warehouse+architect+mens+t-shirt-D13678560?appearance=695&color=1289AF) 34

Data Warehouse Architect

- This role is similar to the solution architect.
- Responsibilities include re-platforming efforts there would be a distinction between those two roles.
- The solution architect would be responsible for the overall solution for all streams.
- The data warehouse architect is focused only on the data warehouse or business intelligence system deliverables.



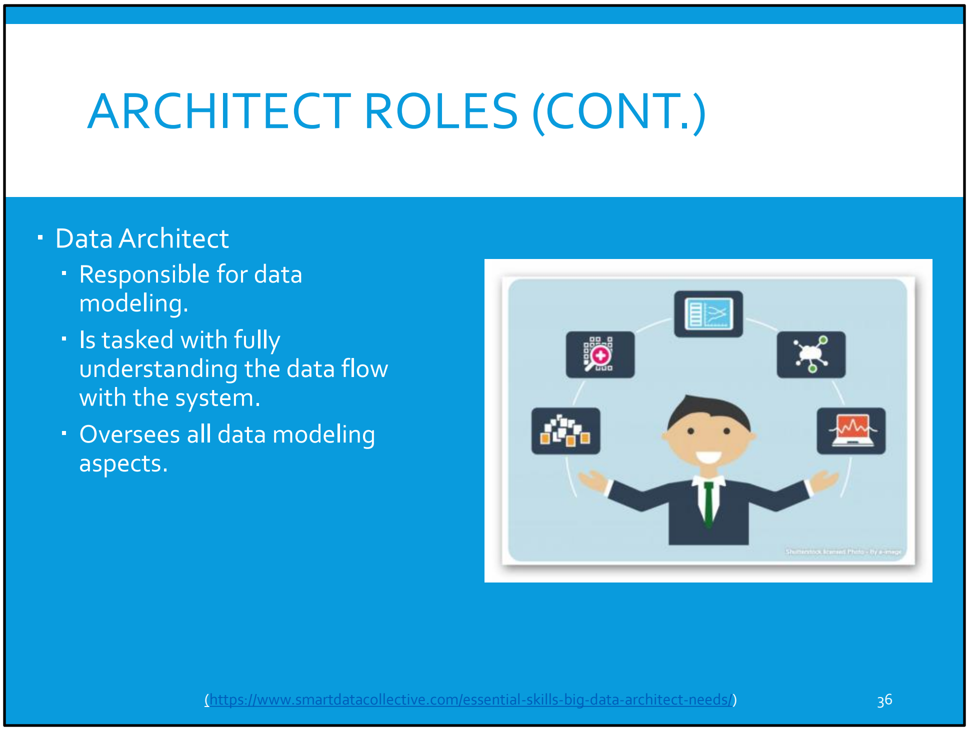
ARCHITECT ROLES (CONT.)

- Technical Architect
  - Responsible for the overall physical environment.
  - Ensures the appropriate servers (machines), storage disks, middleware, file systems, software and similar items are available and operational.

(https://www.stellaxius.com/JoinUsPosition?position=TechnicalArchitectAndDeveloper)

Technical Architect

- Responsible for the overall physical environment.
- Ensures the appropriate servers (machines), storage disks, middleware, file systems, software and similar items are available and operational.



ARCHITECT ROLES (CONT.)

- Data Architect
  - Responsible for data modeling.
  - Is tasked with fully understanding the data flow with the system.
  - Oversees all data modeling aspects.

(https://www.smartdatacollective.com/essential-skills-big-data-architect-needs/)

36

Data Architect

- Responsible for data modeling, and possibly oversees the database administration.
- Is tasked with fully understanding the data flow with the system, and will take the lead on leading the governance effort.
- Oversees all data modeling aspects and typically guides the ETL and BI architectures.

ARCHITECT ROLES (CONT.)

- ETL Architect
  - Responsible for all aspects of planning and designing the ETL.
  - Also include the extraction from the repository and distribution to the data marts if the architecture holds a realized data repository.

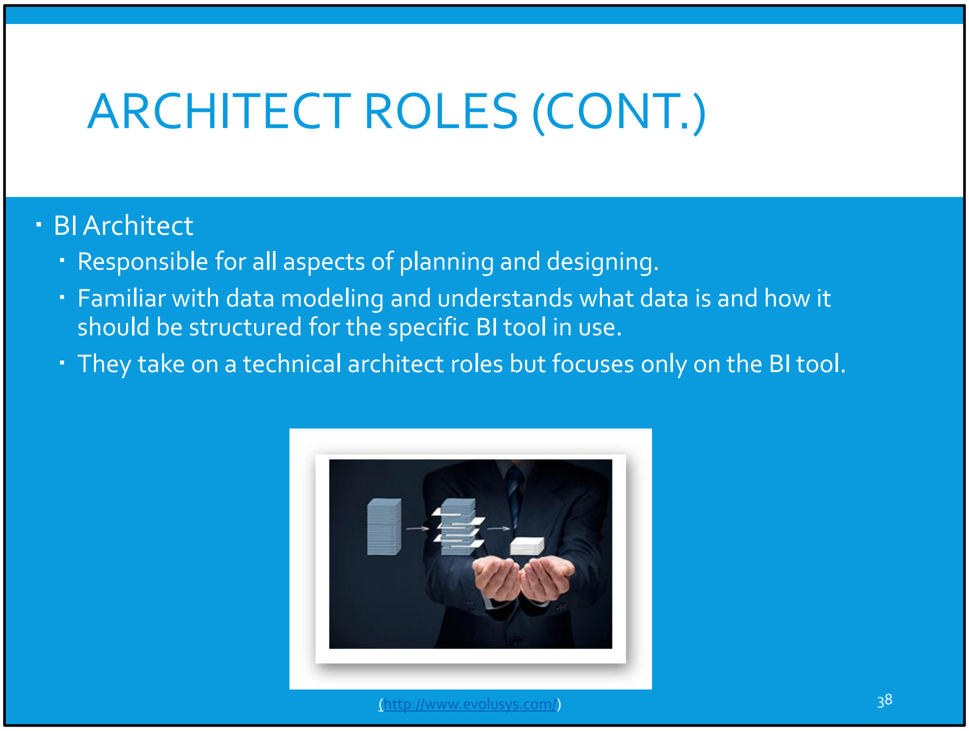
(http://www.evolusys.com/)

37

ETL Architect

- Responsible for all aspects of planning and designing the ETL, which involves source capture, source data transformation and loading into the database.

- Tasks also include the extraction from the repository and distribution to the data marts if the architecture holds a realized data repository.



ARCHITECT ROLES (CONT.)

- BI Architect
  - Responsible for all aspects of planning and designing.
  - Familiar with data modeling and understands what data is and how it should be structured for the specific BI tool in use.
  - They take on a technical architect roles but focuses only on the BI tool.

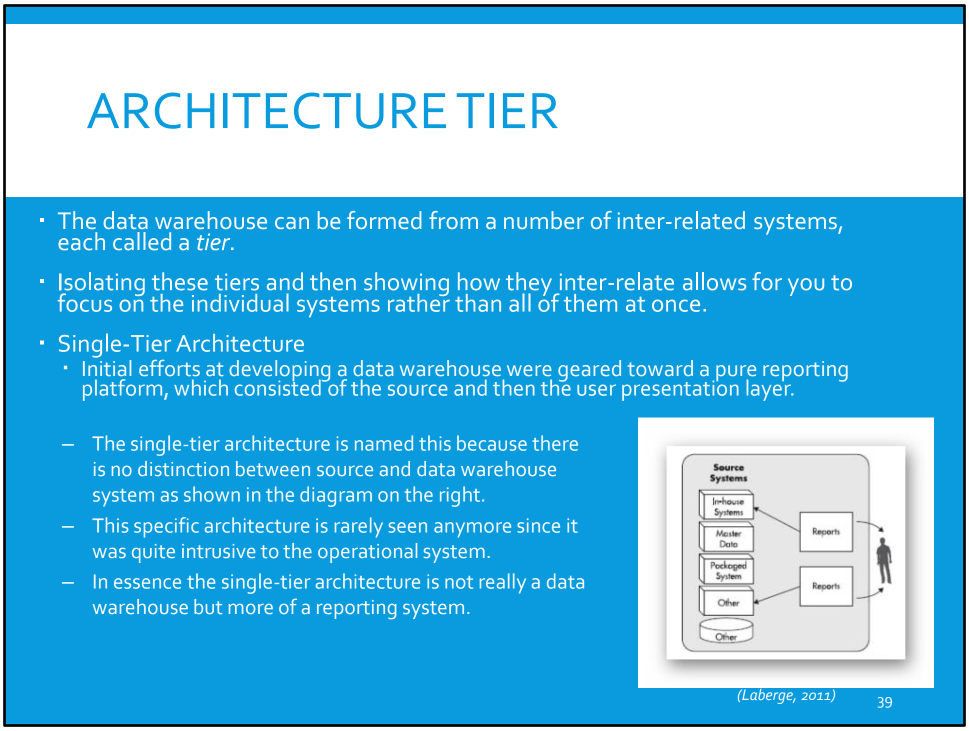
(http://www.evolusys.com/)

BI Architect

- Responsible for all aspects of planning and designing the end-user reporting and usage environment.

- They are familiar with data modeling and understands what data is and how it should be structured for the specific BI tool in use.

- In many instances they take on a technical architect roles but focuses only on the BI tool.


Source Systems
In-house Systems
Master Data
Packaged System
Other
Other
Reports
Reports

ARCHITECTURE TIER

- The data warehouse can be formed from a number of inter-related systems, each called a *tier*.
- Isolating these tiers and then showing how they inter-relate allows for you to focus on the individual systems rather than all of them at once.
- Single-Tier Architecture
  - Initial efforts at developing a data warehouse were geared toward a pure reporting platform, which consisted of the source and then the user presentation layer.

– The single-tier architecture is named this because there is no distinction between source and data warehouse system as shown in the diagram on the right.

– This specific architecture is rarely seen anymore since it was quite intrusive to the operational system.

– In essence the single-tier architecture is not really a data warehouse but more of a reporting system.

(Laberge, 2011)

39

- The data warehouse can be formed from a number of inter-related systems, each called a *tier*.

- Isolating these tiers and then showing how they inter-relate allows for you to focus on the individual systems rather than all of them at once.

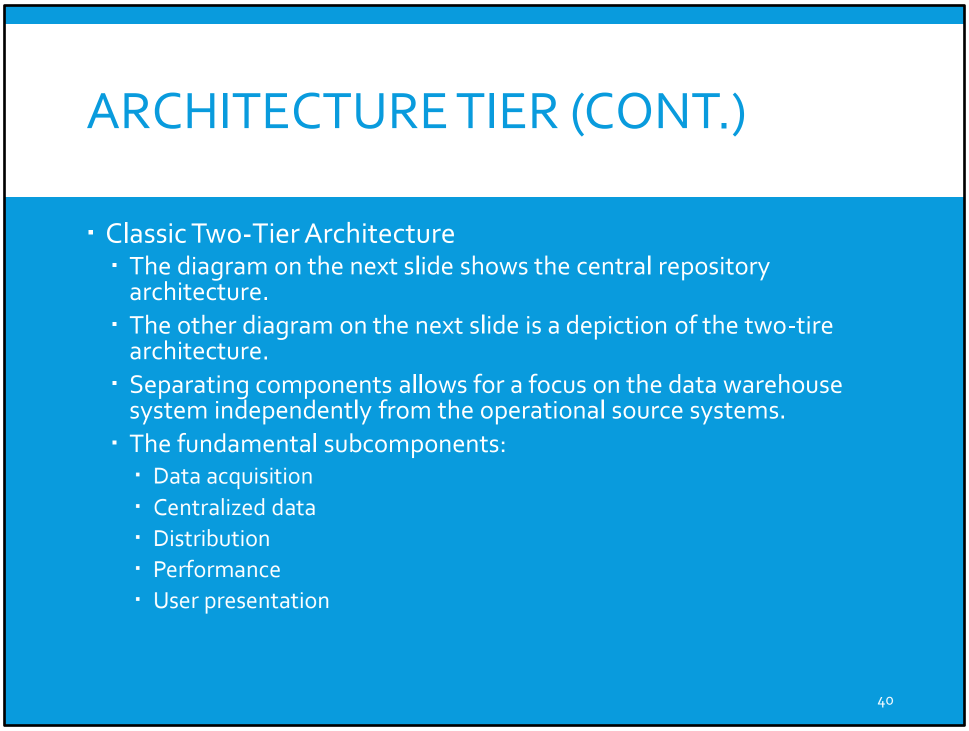
Single-Tier Architecture

- Initial efforts at developing a data warehouse were geared toward a pure reporting platform, which consisted of the source and then the user presentation layer.

- The single-tier architecture is named this because there is no distinction between source and data warehouse system as shown in the diagram on the right.

- This specific architecture is rarely seen anymore since it was quite intrusive to the operational system.

- In essence the single-tier architecture is not really a data warehouse but more of a reporting system.



ARCHITECTURE TIER (CONT.)

- Classic Two-Tier Architecture
  - The diagram on the next slide shows the central repository architecture.
  - The other diagram on the next slide is a depiction of the two-tire architecture.
  - Separating components allows for a focus on the data warehouse system independently from the operational source systems.
  - The fundamental subcomponents:
    - Data acquisition
    - Centralized data
    - Distribution
    - Performance
    - User presentation

Classic Two-Tier Architecture

- The diagram on the next slide shows the central repository architecture, which is the classic two-tier data warehouse architecture which consist of source systems and a data ware house.

- The other diagram on the next slide is a depiction of the two-tire architecture which shows the distinct differences between the source and data warehouse systems.

- Separating components allows for a focus on the data warehouse system independently from the operational source systems.

- The fundamental subcomponents in the central repository are:

  - Data acquisition layer

  - Centralized data layer

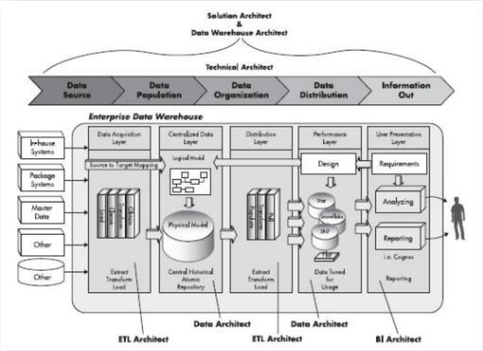
  - Distribution layer

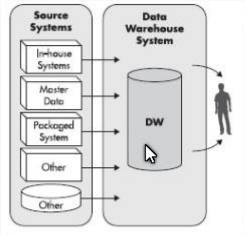
  - Performance layer

  - User presentation layer

ARCHITECTURE TIER (CONT.)

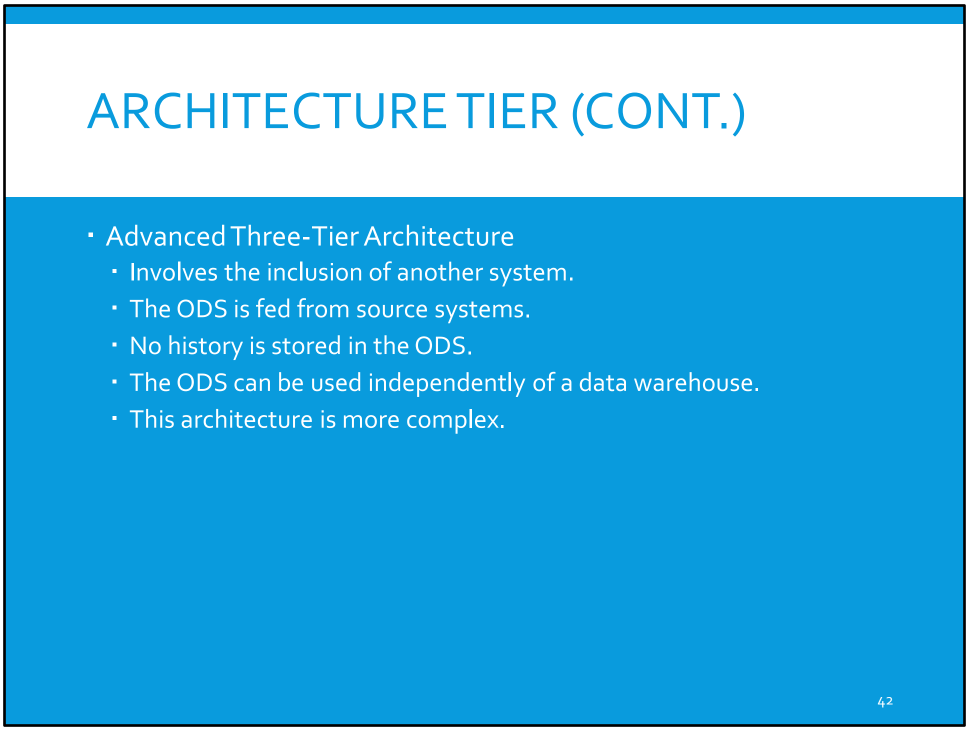
Data Warehouse Architecture





(Laberge, 2011)

Two-Tier Architecture



ARCHITECTURE TIER (CONT.)

- Advanced Three-Tier Architecture
  - Involves the inclusion of another system.
  - The ODS is fed from source systems.
  - No history is stored in the ODS.
  - The ODS can be used independently of a data warehouse.
  - This architecture is more complex.

42

Advanced Three-Tier Architecture

- Involves the inclusion of another system, typically the operational data store (ODS).

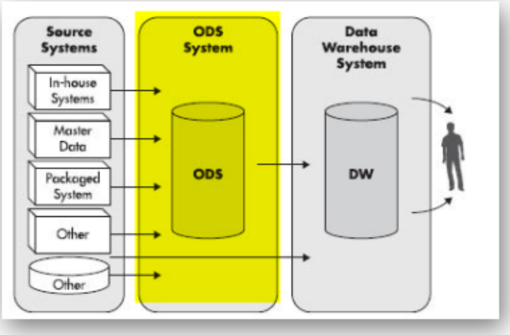
- The ODS is fed from source systems and is typically the current view of customer data.

- Typically no history is stored in the ODS.

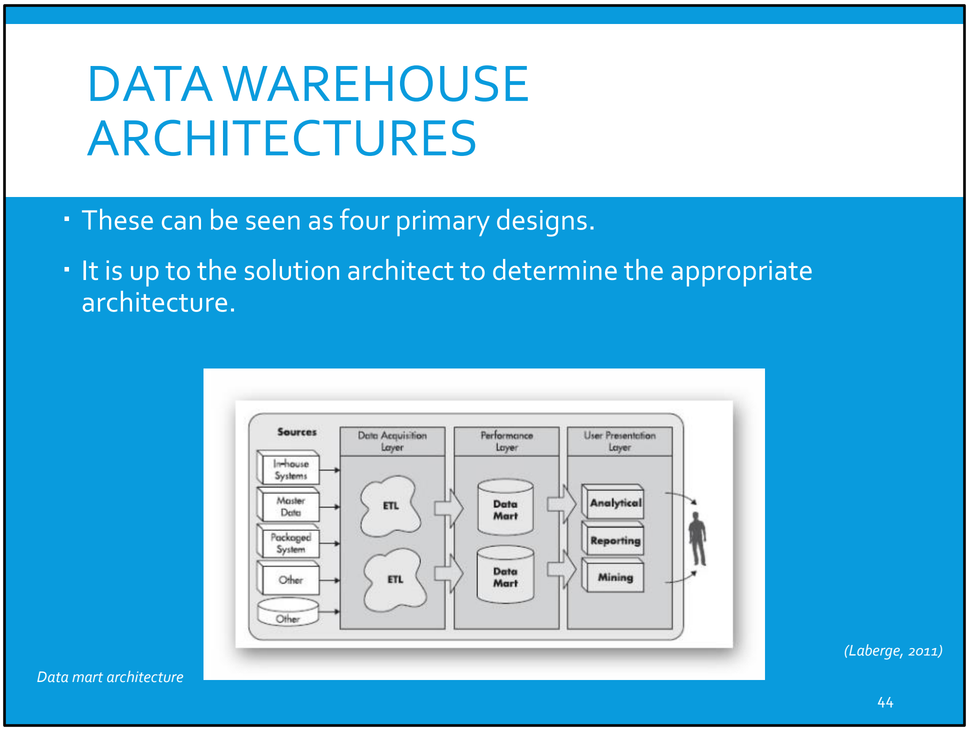
- The ODS can be used independently of a data warehouse system but is usually created in conjunction with the data warehouse using data those structures and definitions.

- This architecture is more complex given the need to align the ODS to the DW via data governance and overall performance structures.

ARCHITECTURE TIER (CONT.)



(Laberge, 2011)



DATA WAREHOUSE ARCHITECTURES

- These can be seen as four primary designs.
- It is up to the solution architect to determine the appropriate architecture.

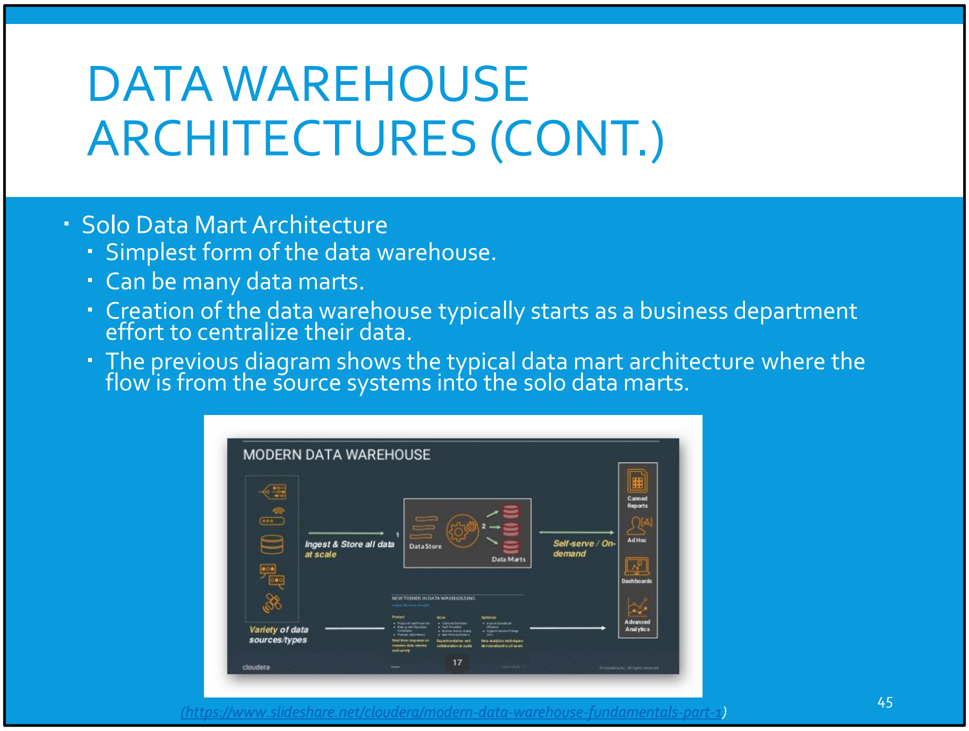
Data mart architecture

(Laberge, 2011)

44

- These can be seen as four primary designs, each of which we will discuss below.

- It is up to the solution architect to determine the appropriate architecture to support the requirements while also ensuring a flexible design in implemented which allows for future unknown requirements.



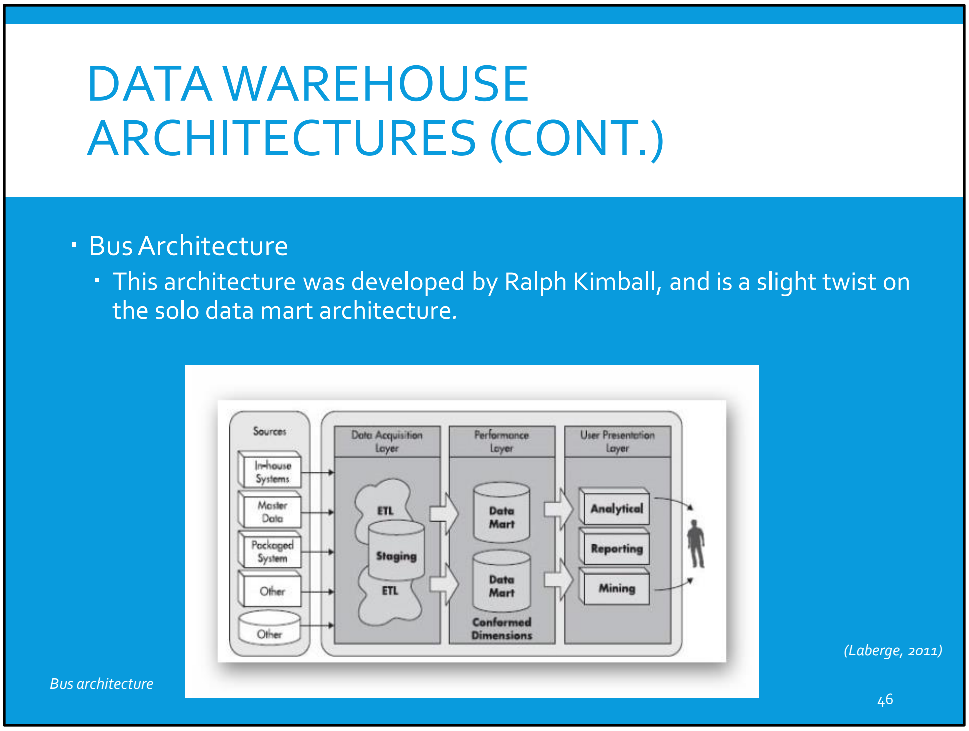
DATA WAREHOUSE ARCHITECTURES (CONT.)

- Solo Data Mart Architecture
  - Simplest form of the data warehouse.
  - Can be many data marts.
  - Creation of the data warehouse typically starts as a business department effort to centralize their data.
  - The previous diagram shows the typical data mart architecture where the flow is from the source systems into the solo data marts.

(https://www.slideshare.net/cloudera/modern-data-warehouse-fundamentals-part-1)

Solo Data Mart Architecture

- This is the simplest form of the data warehouse.
- There can be many data marts that are simply not connected in any way.
- The creation of the data warehouse typically starts as a business department effort to centralize their data into one database for reporting purposes.
- The previous diagram shows the typical data mart architecture where the flow is from the source systems into the solo data marts.



DATA WAREHOUSE ARCHITECTURES (CONT.)

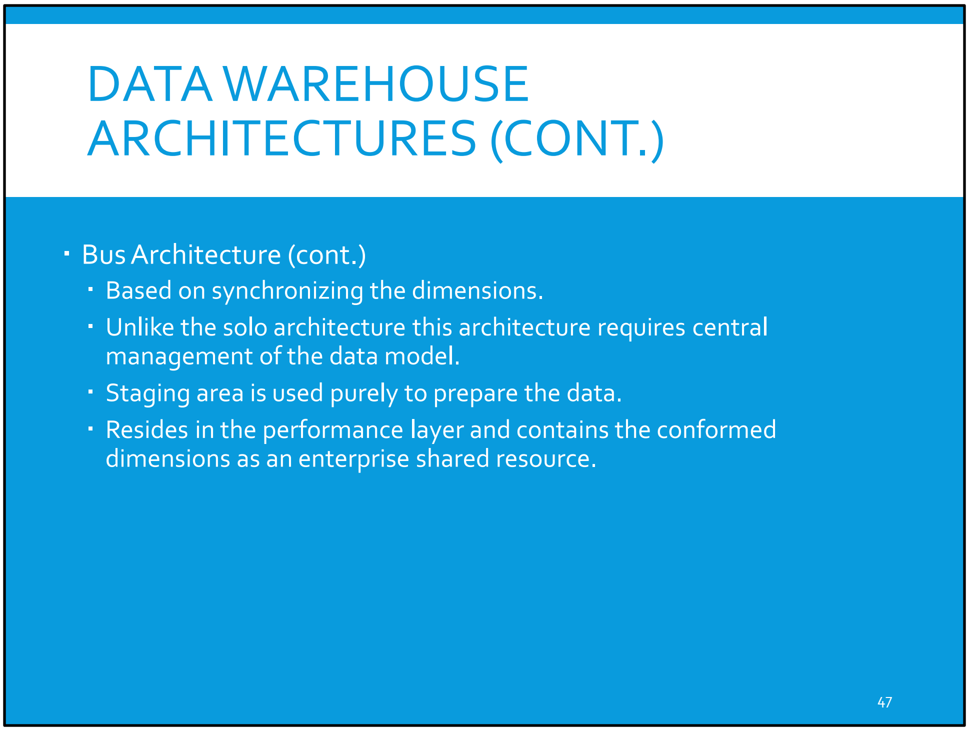
- Bus Architecture
  - This architecture was developed by Ralph Kimball, and is a slight twist on the solo data mart architecture.

Bus architecture

(Laberge, 2011)

Bus Architecture

- This architecture, pictured below, was developed by Ralph Kimball, and is a slight twist on the solo data mart architecture and is called *conformed dimensions* or *bus architecture.*



DATA WAREHOUSE ARCHITECTURES (CONT.)

- Bus Architecture (cont.)
  - Based on synchronizing the dimensions.
  - Unlike the solo architecture this architecture requires central management of the data model.
  - Staging area is used purely to prepare the data.
  - Resides in the performance layer and contains the conformed dimensions as an enterprise shared resource.

Bus Architecture (cont.)

- This design is based on synchronizing the dimensions across the enterprise.

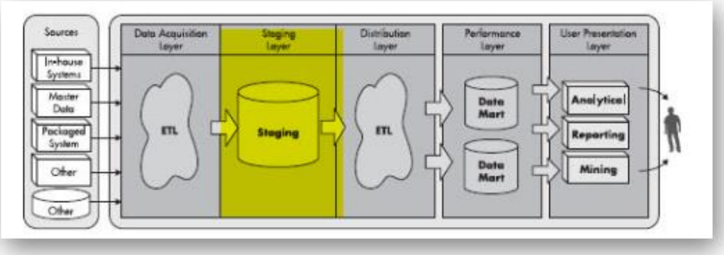
- Unlike the solo architecture previously discussed this architecture requires central management of the data model to help ensure that the dimensions are indeed conformed, for this process a staging area is typically required to prepare the data for distribution.

- The staging area is used purely to prepare the data and is not meant to act as a repository in the central repository architecture sense.

- The bus architecture resides in the performance layer and contains the conformed dimensions as an enterprise shared resource.

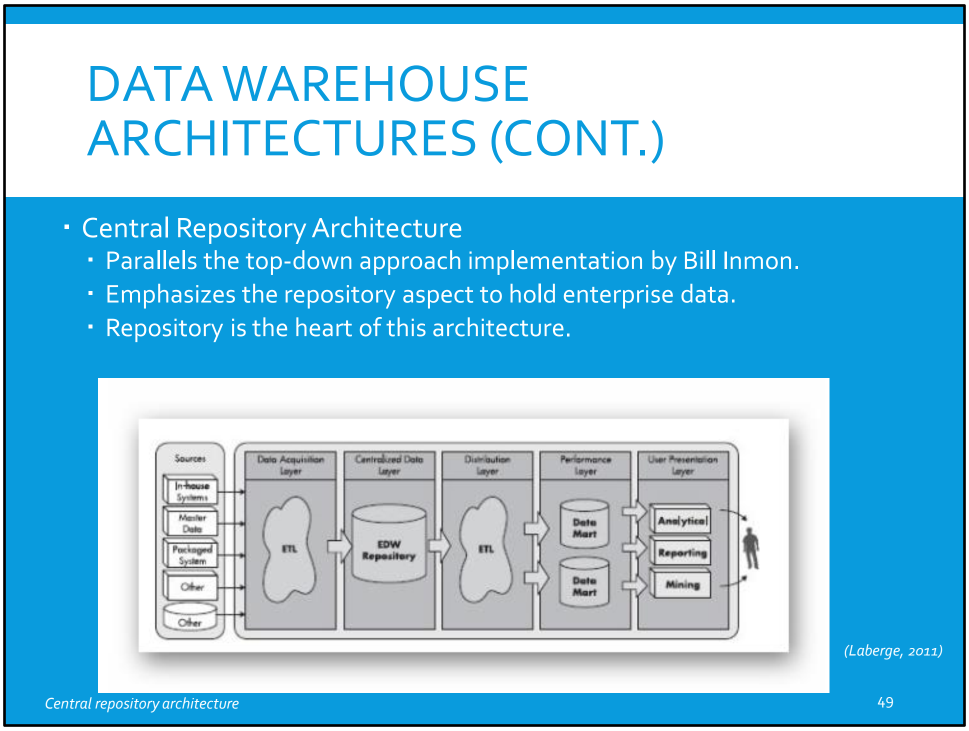
DATA WAREHOUSE ARCHITECTURES (CONT.)

- Bus Architecture (cont.)



Full bus architecture – emphasizing the staging area.

(Laberge, 2011)



DATA WAREHOUSE ARCHITECTURES (CONT.)

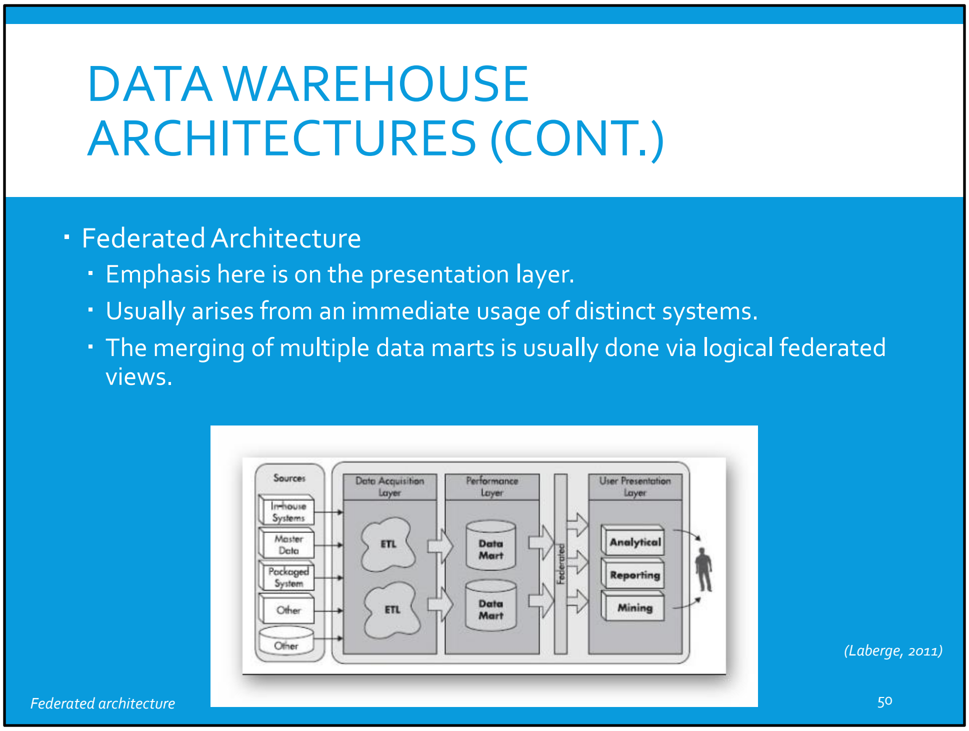
- Central Repository Architecture
  - Parallels the top-down approach implementation by Bill Inmon.
  - Emphasizes the repository aspect to hold enterprise data.
  - Repository is the heart of this architecture.

(Laberge, 2011)

Central repository architecture

Central Repository Architecture

- This architecture parallels the top-down approach implementation by Bill Inmon.
- It emphasizes the repository aspect to hold enterprise data in a normalized way, with full history over time.
- The repository is the heart of this architecture and emphasizes flexibility and scability.



DATA WAREHOUSE ARCHITECTURES (CONT.)

- Federated Architecture
  - Emphasis here is on the presentation layer.
  - Usually arises from an immediate usage of distinct systems.
  - The merging of multiple data marts is usually done via logical federated views.

(Laberge, 2011)

Federated architecture

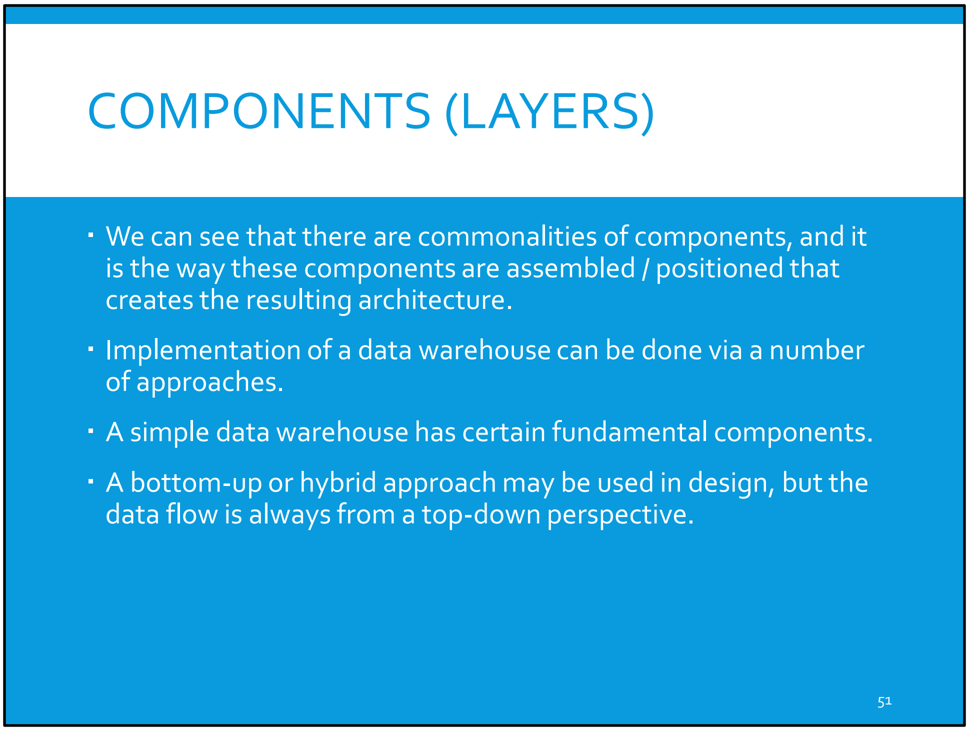
50

Federated Architecture

- The main emphasis here is on the presentation layer.

- This architecture usually arises from an immediate usage of distinct systems, most likely when two departments wish to integrate, or when an organization is merged with another.

- The merging of multiple data marts is usually done via logical federated views, which results in usage with minimum redesigning of any existing systems. Maintenance can be difficult and costly.



COMPONENTS (LAYERS)

- We can see that there are commonalities of components, and it is the way these components are assembled / positioned that creates the resulting architecture.
- Implementation of a data warehouse can be done via a number of approaches.
- A simple data warehouse has certain fundamental components.
- A bottom-up or hybrid approach may be used in design, but the data flow is always from a top-down perspective.

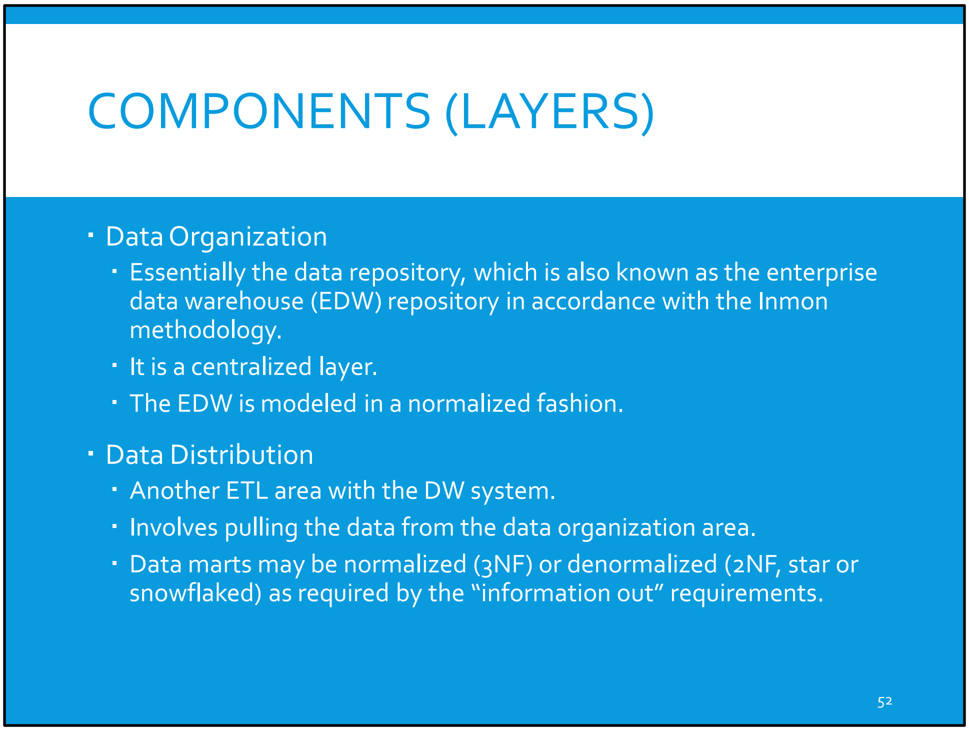
51

- From looking at the different data warehouse architectures we can see that there are commonalities of components, and it is the way these components are assembled / positioned that creates the resulting architecture.

- Note that implementation of a data warehouse can be done via a number of approaches.

- A simple data warehouse has certain fundamental components, which are viewed from the top-down data flow architecture.

- A bottom-up or hybrid approach may be used in design, but the data flow is always from a top-down perspective.



COMPONENTS (LAYERS)

- Data Organization
  - Essentially the data repository, which is also known as the enterprise data warehouse (EDW) repository in accordance with the Inmon methodology.
  - It is a centralized layer.
  - The EDW is modeled in a normalized fashion.
- Data Distribution
  - Another ETL area with the DW system.
  - Involves pulling the data from the data organization area.
  - Data marts may be normalized (3NF) or denormalized (2NF, star or snowflaked) as required by the "information out" requirements.

52

Data Organization

- This is essentially the data repository, which is also known as the enterprise data warehouse (EDW) repository in accordance with the Inmon methodology.

- It is a centralized layer which holds lots of data for a long period of time.

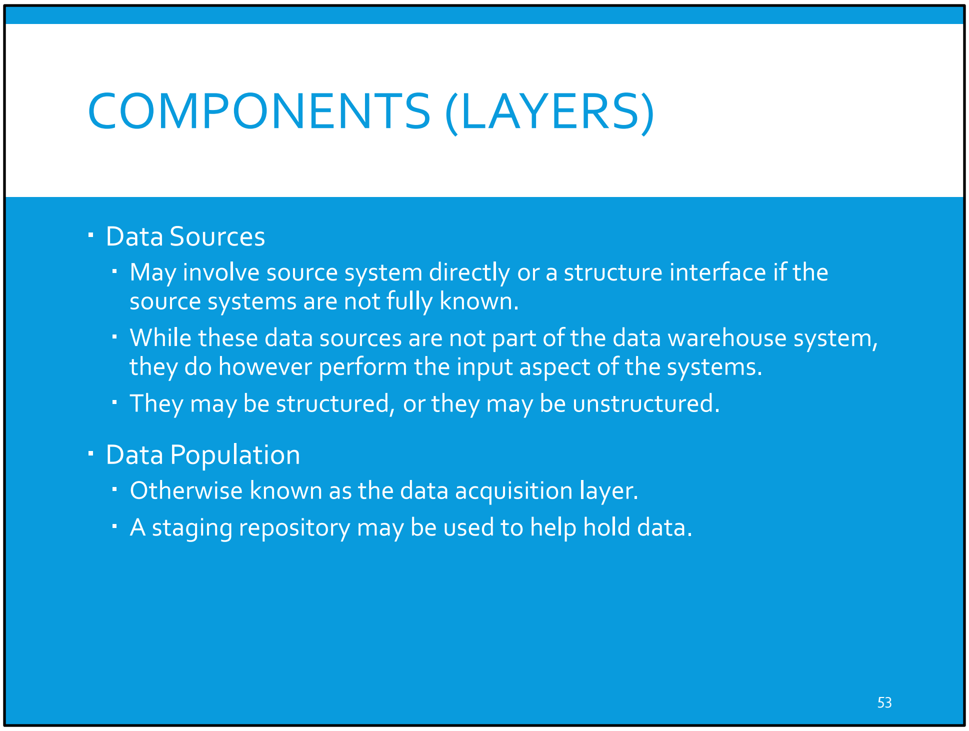
- The EDW is modeled in a normalized fashion which ensures the data if fully understood fundamentally, descriptively and associatively.

Data Distribution

- This is another ETL area with the DW system.

- This involves pulling the data from the data organization area and then distributing it to the various data marts.

- These data marts may be normalized (3NF) or denormalized (2NF, star or snowflaked) as required by the "information out" requirements.



COMPONENTS (LAYERS)

- Data Sources
  - May involve source system directly or a structure interface if the source systems are not fully known.
  - While these data sources are not part of the data warehouse system, they do however perform the input aspect of the systems.
  - They may be structured, or they may be unstructured.
- Data Population
  - Otherwise known as the data acquisition layer.
  - A staging repository may be used to help hold data.

Data Sources

- They may involve source system directly or a structure interface if the source systems are not fully known.

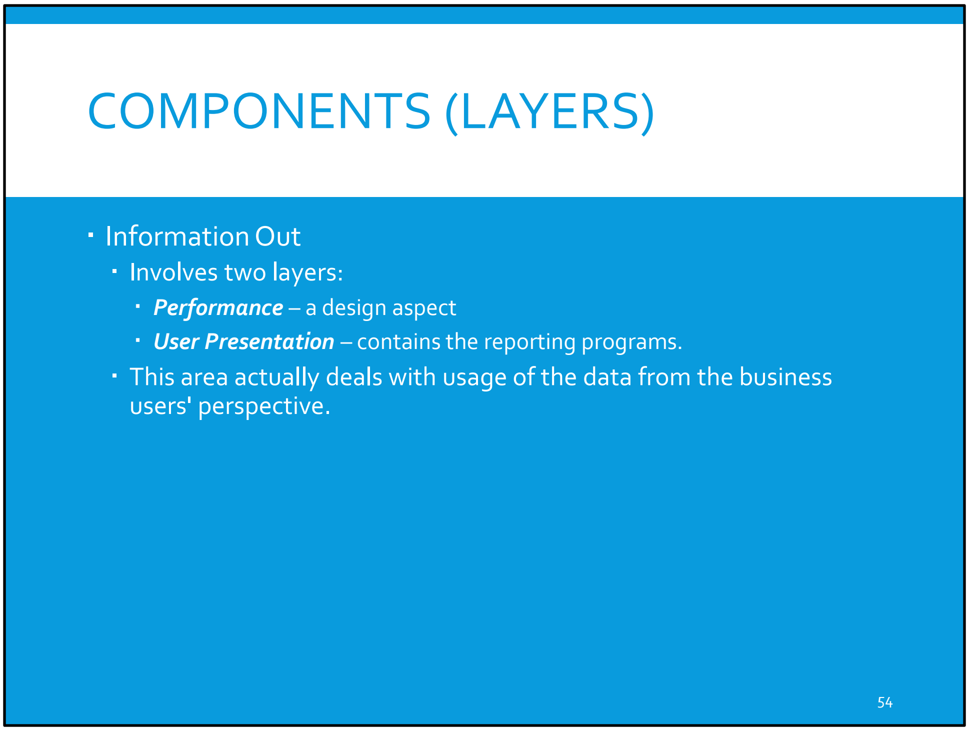
- While these data sources are not part of the data warehouse system, they do however perform the input aspect of the systems, which is why we discuss them there.

- They may be structured, meaning their fields are identifiable and usually consistent, or they may be unstructured, meaning the individual fields may contain more than one data item.

Data Population

- This is otherwise known as the data acquisition layer, data population is the ETL portion which is here to capture data from the data sources.

- A staging repository may be used to help hold data while waiting for other sources to capture to data for merging, processing and so on.

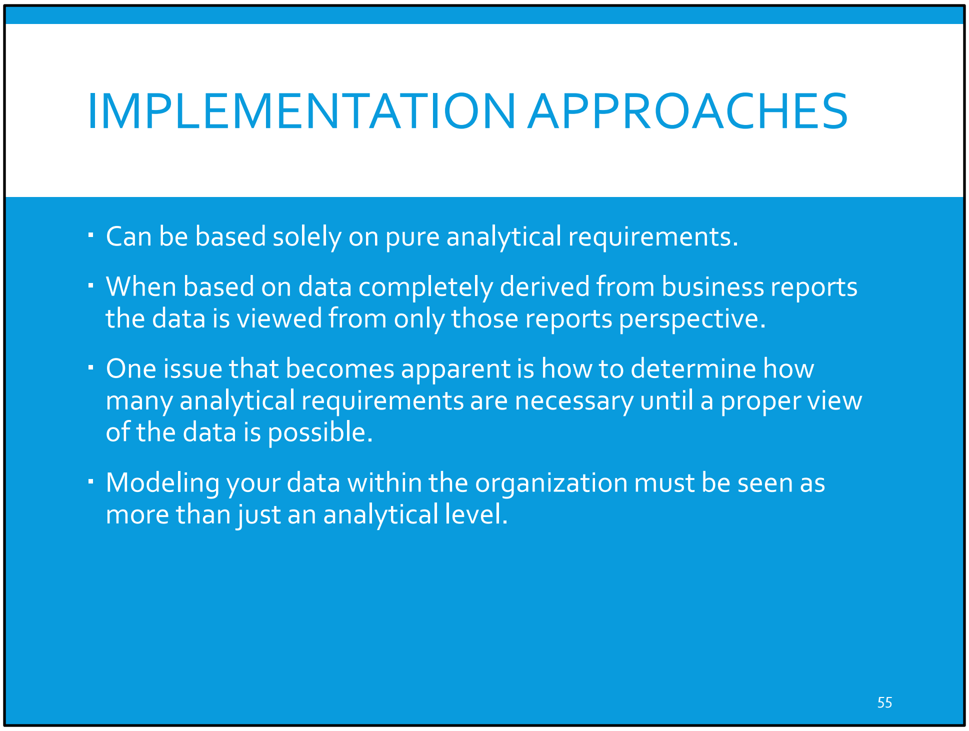


COMPONENTS (LAYERS)

- Information Out
  - Involves two layers:
    - ***Performance*** – a design aspect
    - ***User Presentation*** – contains the reporting programs.
  - This area actually deals with usage of the data from the business users' perspective.

Information Out

- To get the information out of the data warehouse it involves two layers:
  - ***Performance*** – a design aspect that is the underlying database structures, otherwise known as the data marts
  - ***User Presentation*** – this layer contains the reporting programs. Typically they are in the form of BI tools such as Cognos, Business Objects, etc. They require a programming effort to create the final reports which the business end users see.
- This area actually deals with usage of the data from the business users' perspective, it must then be designed for communicative purposes, which simply means it must be easy to use, and perform appropriately.



IMPLEMENTATION APPROACHES

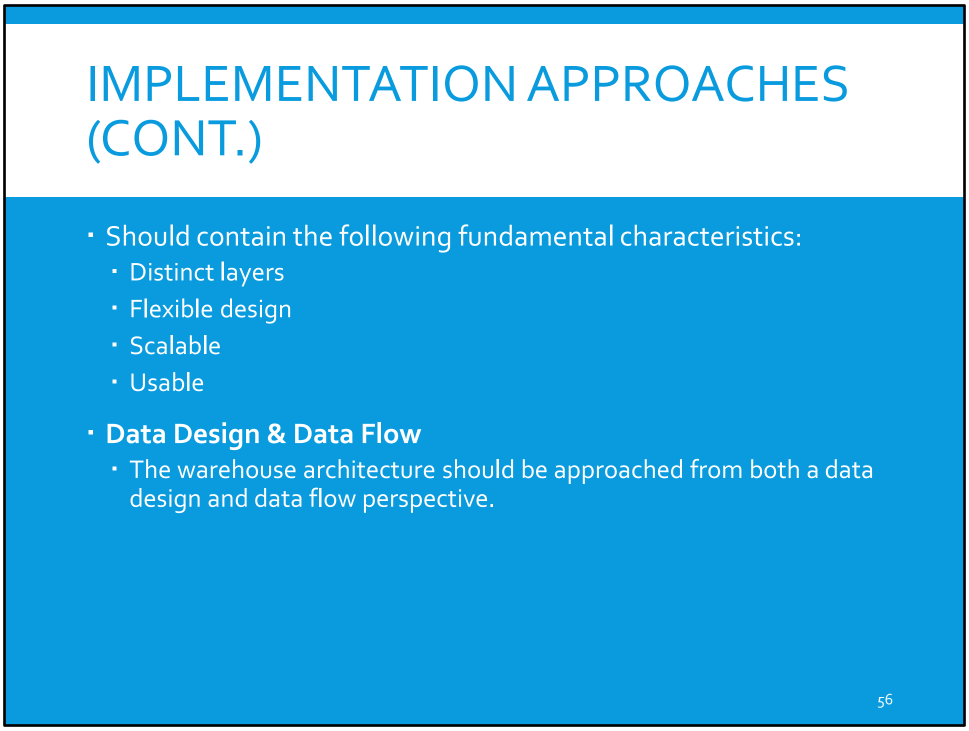
- Can be based solely on pure analytical requirements.
- When based on data completely derived from business reports the data is viewed from only those reports perspective.
- One issue that becomes apparent is how to determine how many analytical requirements are necessary until a proper view of the data is possible.
- Modeling your data within the organization must be seen as more than just an analytical level.

- A data warehouse can be based solely on pure analytical requirements, data as an enterprise assets or both.

- When your solution is based on data completely derived from business reports the data is then viewed from only those reports perspective.

- One issue that becomes apparent is how to determine how many analytical requirements are necessary until a proper view of the data is possible.

- Modeling your data within the organization must be seen as more than just an analytical level, meaning that the transactional and analytical aspects of the data should be separated.



IMPLEMENTATION APPROACHES (CONT.)

- Should contain the following fundamental characteristics:
  - Distinct layers
  - Flexible design
  - Scalable
  - Usable
- **Data Design & Data Flow**
  - The warehouse architecture should be approached from both a data design and data flow perspective.

The basic architecture of the enterprise data warehouse should contain the following fundamental characteristics:

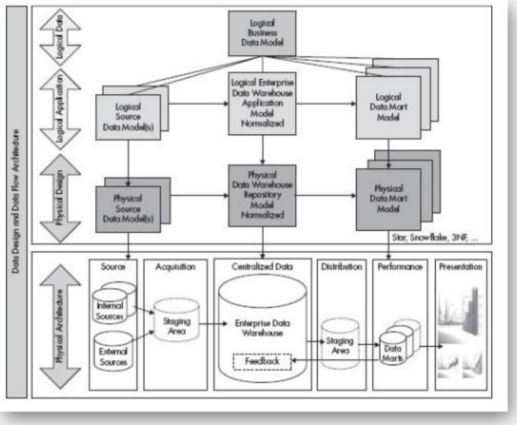
- Distinct layers
- Flexible design
- Scalable
- Usable

Data Design & Data Flow

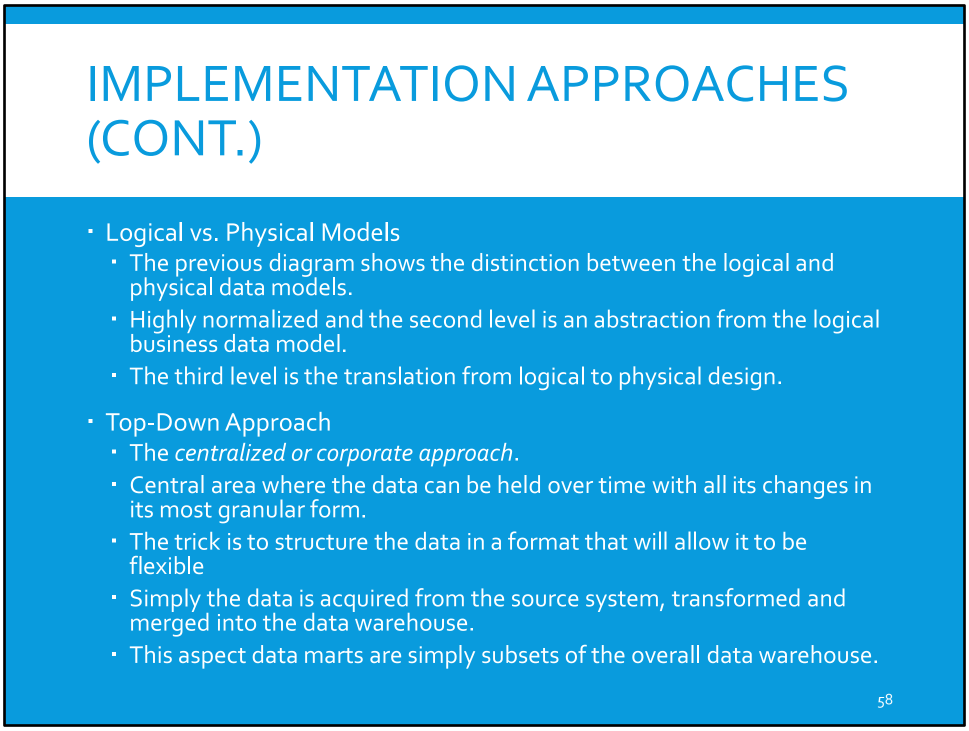
- The warehouse architecture should be approached from both a data design and data flow perspective. The diagram on the next page from out text illustrates how the different types of data models fit together along a simple DW architecture flow.

IMPLEMENTATION APPROACHES (CONT.)

Data design and data flow


Data Design and Data Flow Architecture
Logical Data
Logical Application
Physical Design
Physical Architecture
Logical Business Data Model
Logical Source Data Model(s)
Logical Enterprise Data Warehouse Application Model Normalized
Logical Data Mart Model
Physical Source Data Model(s)
Physical Data Warehouse Repository Model Normalized
Physical Data Mart Model
Star, Snowflake, 3NF, ...
Source
Acquisition
Centralized Data
Distribution
Performance
Presentation
Internal Sources
External Sources
Staging Area
Enterprise Data Warehouse
Feedback
Staging Area
Data Marts

(Laberge, 2011)



IMPLEMENTATION APPROACHES (CONT.)

- Logical vs. Physical Models
  - The previous diagram shows the distinction between the logical and physical data models.
  - Highly normalized and the second level is an abstraction from the logical business data model.
  - The third level is the translation from logical to physical design.
- Top-Down Approach
  - The *centralized or corporate approach*.
  - Central area where the data can be held over time with all its changes in its most granular form.
  - The trick is to structure the data in a format that will allow it to be flexible
  - Simply the data is acquired from the source system, transformed and merged into the data warehouse.
  - This aspect data marts are simply subsets of the overall data warehouse.

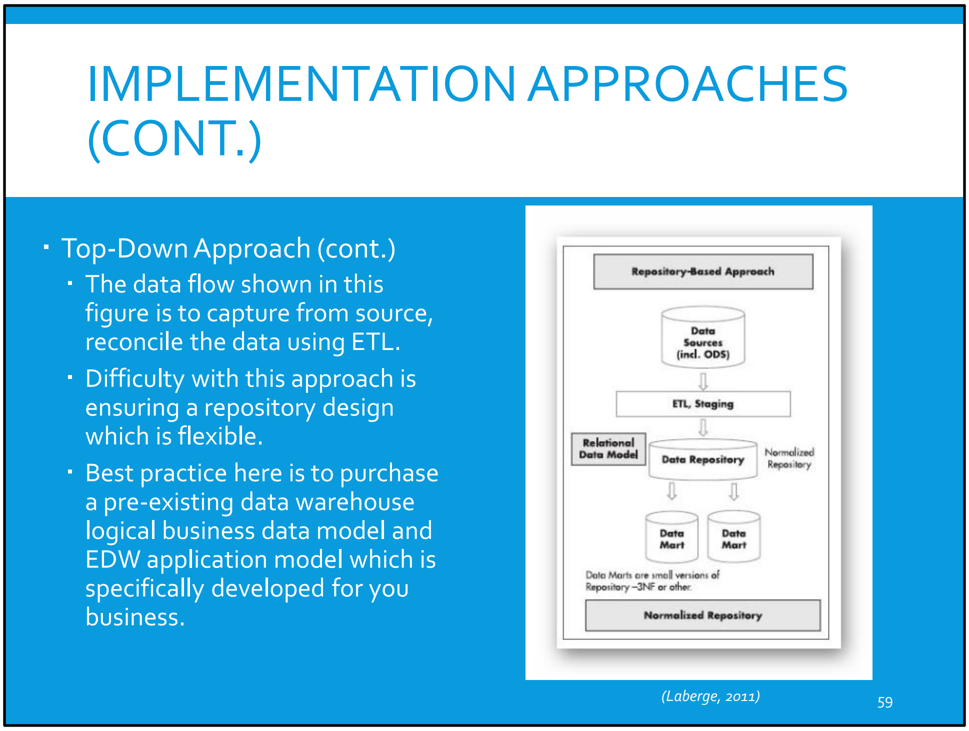
58

Logical vs. Physical Models

- The previous diagram shows the distinction between the logical and physical data models.
- This data model is highly normalized and the second level is an abstraction from the logical business data model for specific applications.
- The third level is the translation from logical to physical design.

Top-Down Approach

- Also known as the *centralized or corporate approach*. This approach and implementation was popularized by Bill Inmon and still has a large following.
- Here we build a central area where the data can be held over time with all its changes in its most granular form.
- The trick here is to structure the data in a format that will allow it to be flexible and grow as more projects add data to the warehouse.
- Quite simply the data is acquired from the source system, transformed and merged into the data warehouse.
- In this aspect data marts are simply subsets of the overall data warehouse and not necessarily in the same format.



IMPLEMENTATION APPROACHES (CONT.)

- Top-Down Approach (cont.)
  - The data flow shown in this figure is to capture from source, reconcile the data using ETL.
  - Difficulty with this approach is ensuring a repository design which is flexible.
  - Best practice here is to purchase a pre-existing data warehouse logical business data model and EDW application model which is specifically developed for you business.

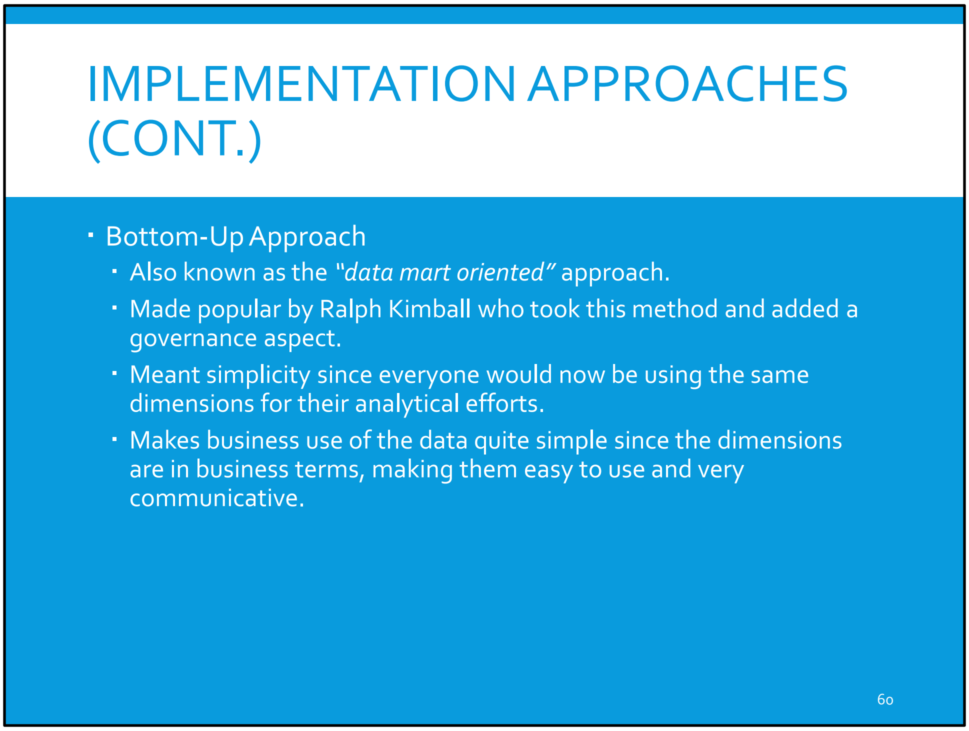
(Laberge, 2011)

Top-Down Approach (cont.)

- The data flow shown in this figure is to capture from source, reconcile the data using ETL routines and then populate the central repository which is designed in a normalized fashion.

- The difficulty with this approach is ensuring a repository design which is flexible and scalable as each project adds to its design.

- Best practice here is to purchase a pre-existing data warehouse logical business data model and EDW application model which is specifically developed for you business.



IMPLEMENTATION APPROACHES (CONT.)

- Bottom-Up Approach
  - Also known as the *"data mart oriented"* approach.
  - Made popular by Ralph Kimball who took this method and added a governance aspect.
  - Meant simplicity since everyone would now be using the same dimensions for their analytical efforts.
  - Makes business use of the data quite simple since the dimensions are in business terms, making them easy to use and very communicative.

60

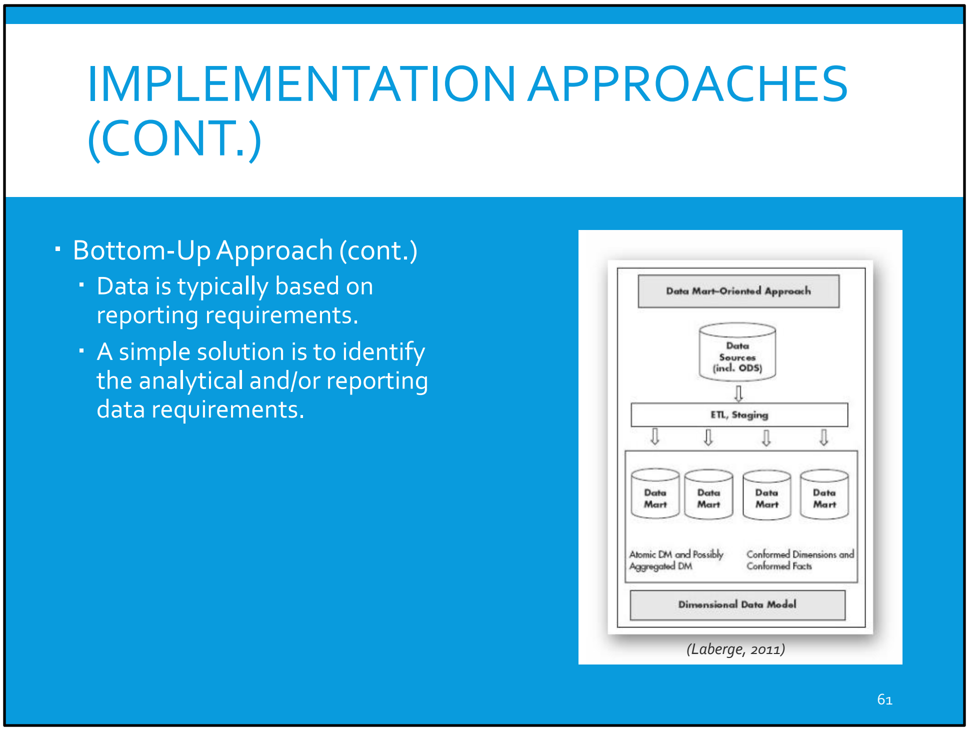
Bottom-Up Approach

- Also known as the *"data mart oriented"* approach, this implementation is focused on business analytics or oriented toward end results.

- This method was made popular by Ralph Kimball who took this method and added a governance aspect, which was to help preserve the definition and structures of specific tables across the enterprise.

- This meant simplicity since everyone would now be using the same dimensions for their analytical efforts.

- This also makes business use of the data quite simple since the dimensions are in business terms, making them easy to use and very communicative.



IMPLEMENTATION APPROACHES (CONT.)

- Bottom-Up Approach (cont.)
  - Data is typically based on reporting requirements.
  - A simple solution is to identify the analytical and/or reporting data requirements.

Data Mart-Oriented Approach
Data Sources (incl. ODS)
ETL, Staging
Data Mart
Data Mart
Data Mart
Data Mart
Atomic DM and Possibly Aggregated DM
Conformed Dimensions and Conformed Facts
Dimensional Data Model

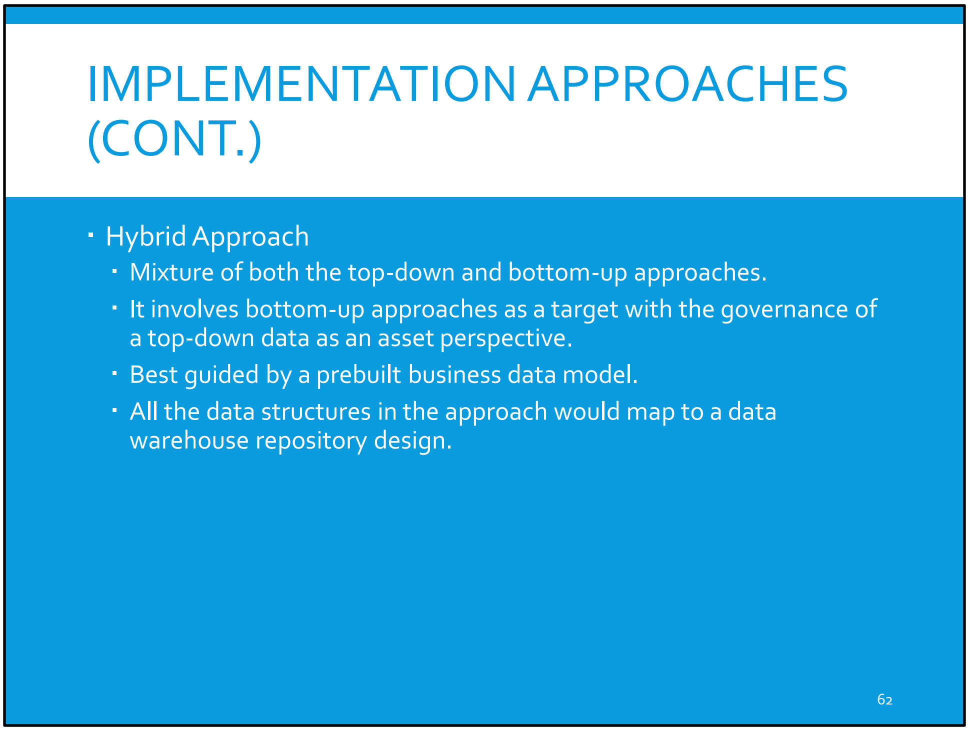
(Laberge, 2011)

61

Bottom-Up Approach (cont.)

- A negative aspect of this approach is that the data is typically based on reporting requirements.

- A simple solution is to identify the analytical and/or reporting data requirements and then have a business SME determine their associations.



IMPLEMENTATION APPROACHES (CONT.)

- Hybrid Approach
  - Mixture of both the top-down and bottom-up approaches.
  - It involves bottom-up approaches as a target with the governance of a top-down data as an asset perspective.
  - Best guided by a prebuilt business data model.
  - All the data structures in the approach would map to a data warehouse repository design.

62

Hybrid Approach

- This is a mixture of both the top-down and bottom-up approaches, and is the best of both worlds.

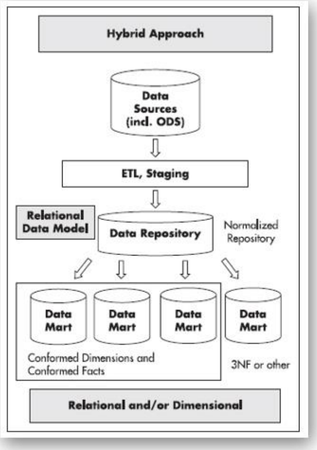
- It involves bottom-up approaches as a target with the governance of a top-down data as an asset perspective.

- It is best guided by a prebuilt business data model to set the enterprise data structure and associative foundation.

- All the data structures in the approach would map to a data warehouse repository design, and all would be governed by an enterprise data model. This approach would also use a staging environment as a holding area to reconcile any captured source data.

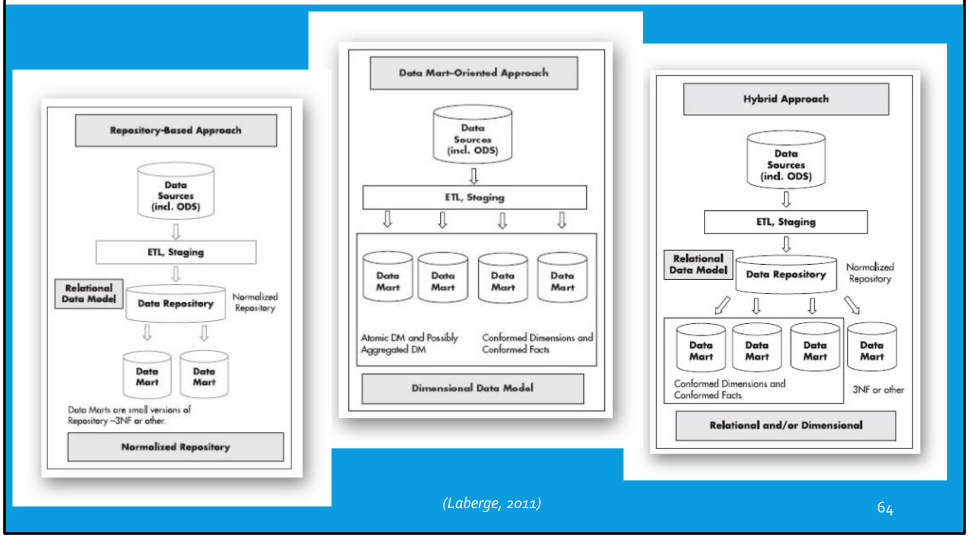
IMPLEMENTATION APPROACHES (CONT.)

Hybrid Approach



(Laberge, 2011)

TOP-DOWN, BOTTOM-UP, AND HYBRID APPROACHES

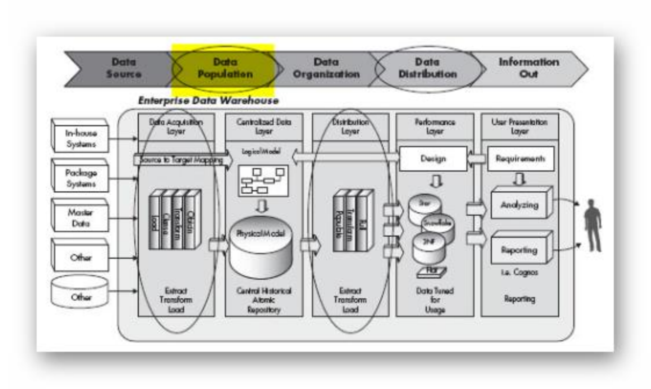


(Laberge, 2011)

ETL AND DATA QUALITY

OVERVIEW

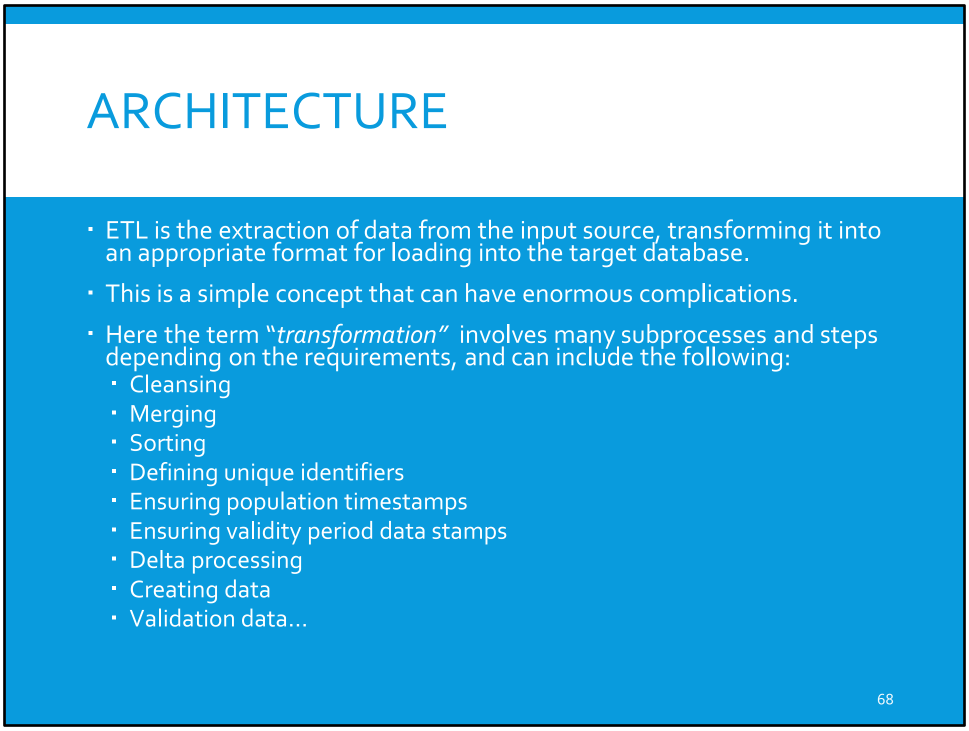
- The data warehouse is a system with input, processing and output.
- The first ETL process in the data warehouse is from source to preparation and holding area which is referred to as *data population* as shown below.



(Laberge, 2011)

OVERVIEW (CONT.)

- The high level ETL flow in one of the major determining factors of the overall data warehouse architecture.
- The data has to have value to the organization in many cases is fundamental to the operations of the business.
- It is vital that the data is managed with a high degree of transparency, quality and availability.
- Many organizations are building their data warehouses to support their business intelligence analysis.
- The ETL portion of the DW is typically the most challenging part of the project and takes approximately 70 percent of the overall effort.

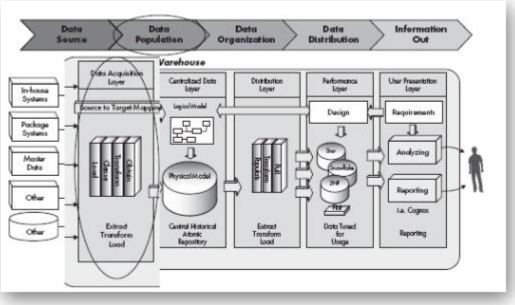


ARCHITECTURE

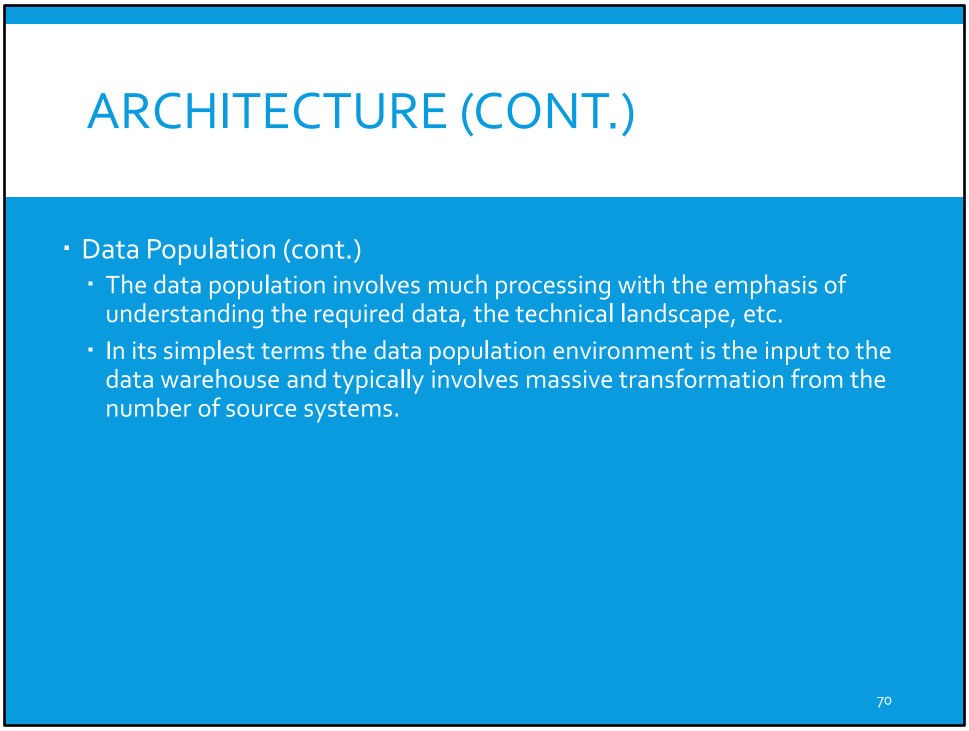
- ETL is the extraction of data from the input source, transforming it into an appropriate format for loading into the target database.
- This is a simple concept that can have enormous complications.
- Here the term "*transformation*" involves many subprocesses and steps depending on the requirements, and can include the following:
  - Cleansing
  - Merging
  - Sorting
  - Defining unique identifiers
  - Ensuring population timestamps
  - Ensuring validity period data stamps
  - Delta processing
  - Creating data
  - Validation data…

ARCHITECTURE (CONT.)

- Data Population
  - The figure below shows only one layer for the data population phase, but it can actually be split into two different perspective: data acquisition aspect of source to staging, and the data population aspect involving staging to the central data layer.



(Laberge, 2011)

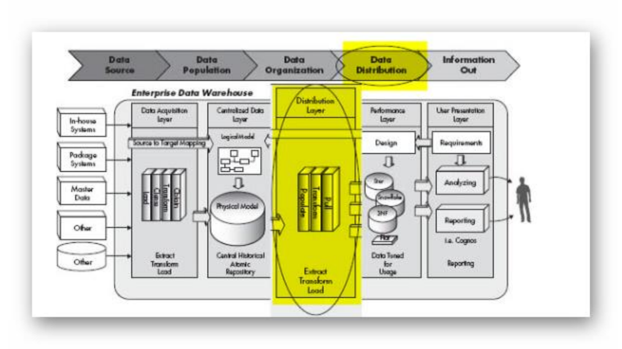


ARCHITECTURE (CONT.)

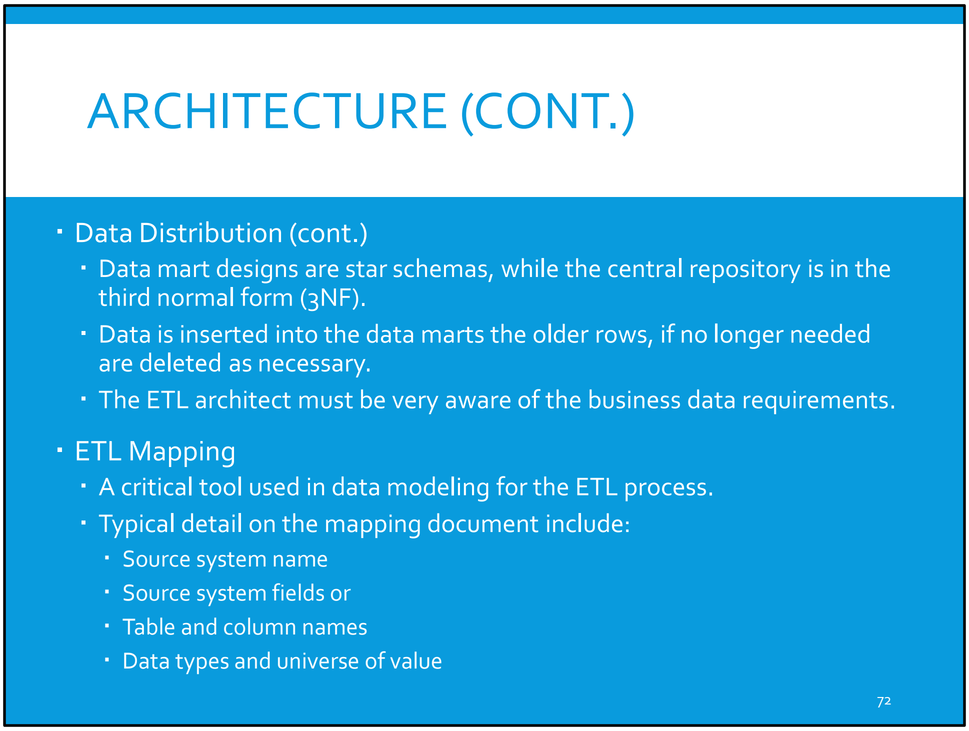
- Data Population (cont.)
  - The data population involves much processing with the emphasis of understanding the required data, the technical landscape, etc.
  - In its simplest terms the data population environment is the input to the data warehouse and typically involves massive transformation from the number of source systems.

ARCHITECTURE (CONT.)

- Data Distribution
  - The figure below shows the data distribution layer which is another ETL environment with focus on extracting from the holding area.



(Laberge, 2011)



ARCHITECTURE (CONT.)

- Data Distribution (cont.)
  - Data mart designs are star schemas, while the central repository is in the third normal form (3NF).
  - Data is inserted into the data marts the older rows, if no longer needed are deleted as necessary.
  - The ETL architect must be very aware of the business data requirements.
- ETL Mapping
  - A critical tool used in data modeling for the ETL process.
  - Typical detail on the mapping document include:
    - Source system name
    - Source system fields or
    - Table and column names
    - Data types and universe of value

Data Distribution (cont.)

- In many instances the data mart designs are star schemas, while the central repository is in the third normal form (3NF).

- Typically the data is inserted into the data marts the older rows, if no longer needed are deleted as necessary based on whatever service agreement is in place with the business users of the data mart.

- The ETL architect must be very aware of the business data requirements and also must ensure that the environment is scalable from many perspectives like: data volumes, DBMS capabilities, ETL routine flexibilities, runtime durations and tooling.

ETL Mapping

- The mapping document is a critical tool used in data modeling for the ETL process.

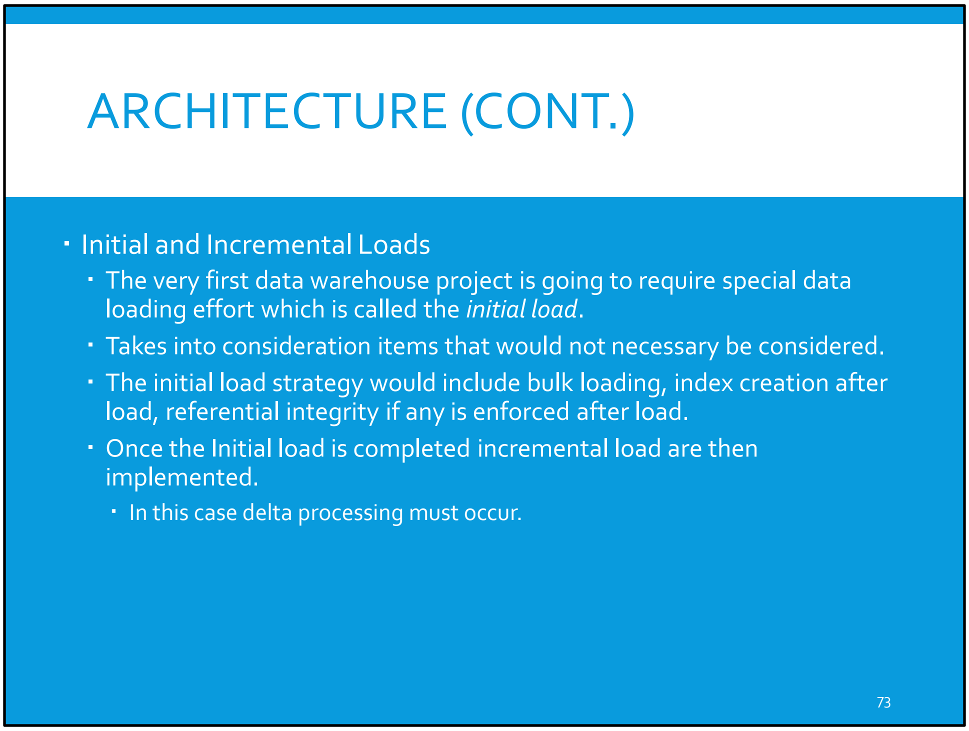
- Typical detail on the mapping document include:

  - Source system name

  - Source system fields or

  - Table and column names

  - Data types and universe of value



ARCHITECTURE (CONT.)

- Initial and Incremental Loads
  - The very first data warehouse project is going to require special data loading effort which is called the *initial load*.
  - Takes into consideration items that would not necessary be considered.
  - The initial load strategy would include bulk loading, index creation after load, referential integrity if any is enforced after load.
  - Once the Initial load is completed incremental load are then implemented.
    - In this case delta processing must occur.

Initial and Incremental Loads

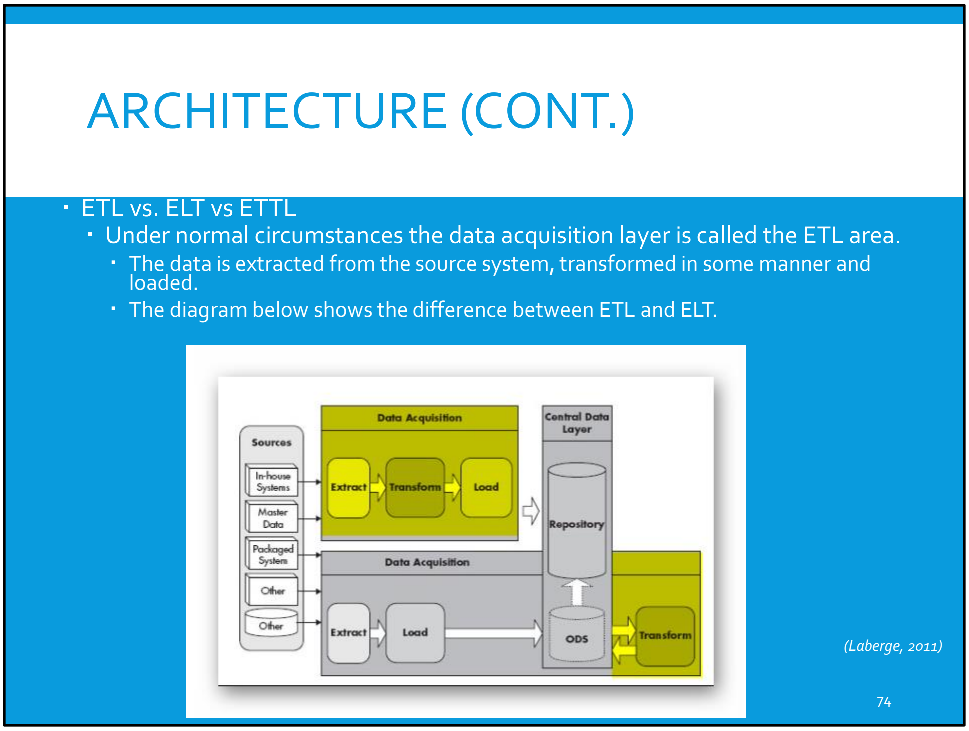
- The very first data warehouse project is going to require special data loading effort which is called the *initial load*.

- This process takes into consideration items that would not necessary be considered, such as initial code tables and date and time.

- The initial load strategy would include bulk loading, index creation after load, referential integrity if any is enforced after load.

- Once the Initial load is completed incremental load are then implemented. The difference between these two is that in incremental loads the data already exists in the database.

  - In this case delta processing must occur. This processing consist of routines to determine if an input row is new or has changed since the last load.



ARCHITECTURE (CONT.)

- ETL vs. ELT vs ETTL
  - Under normal circumstances the data acquisition layer is called the ETL area.
    - The data is extracted from the source system, transformed in some manner and loaded.
    - The diagram below shows the difference between ETL and ELT.

(Laberge, 2011)

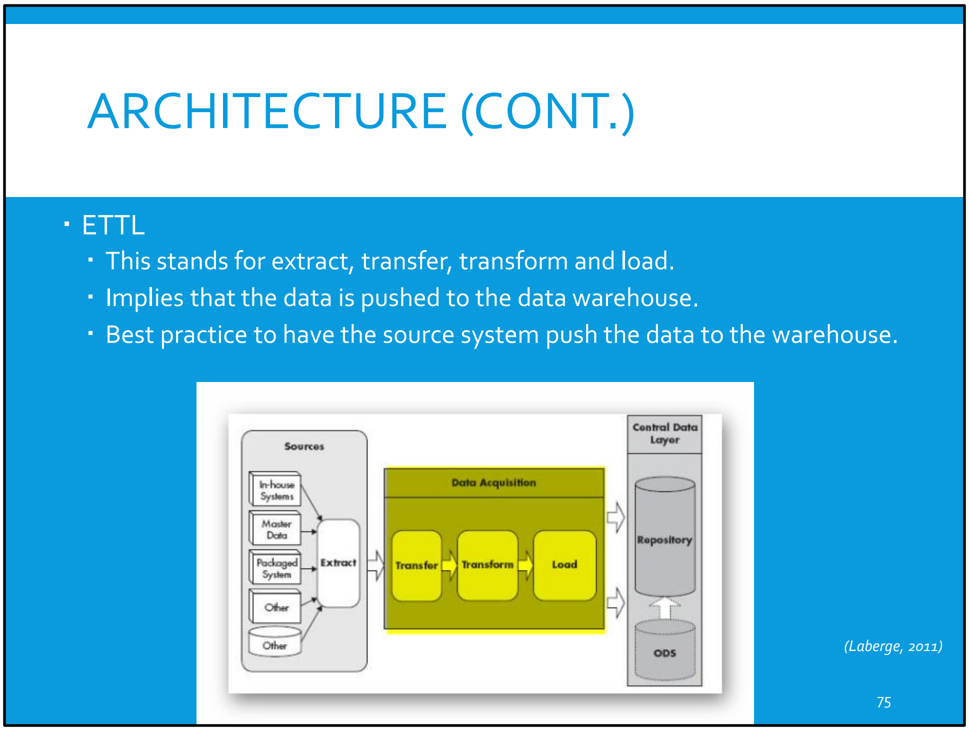
74

ETL vs. ELT vs ETTL

- Under normal circumstances the data acquisition layer is called the ETL area.

  - The data is extracted from the source system, transformed in some manner and loaded.

  - The diagram below shows the difference between ETL and ELT. In ETL the data is transformed at the operation system level, with ELT the data is loaded first then transformed.



ARCHITECTURE (CONT.)

- ETTL
  - This stands for extract, transfer, transform and load.
  - Implies that the data is pushed to the data warehouse.
  - Best practice to have the source system push the data to the warehouse.

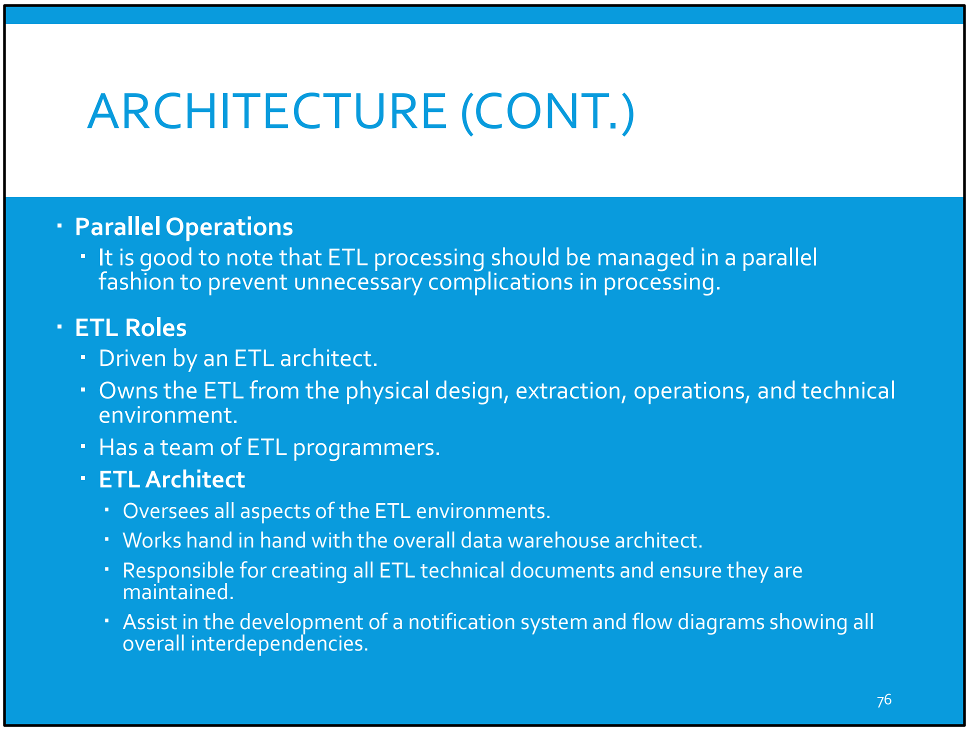
(Laberge, 2011)

ETTL

- This stands for extract, transfer, transform and load.

- This process implies that the data is pushed to the data warehouse instead of pulled as the ETL and ELT process show.

- It is best practice to have the source system push the data to the warehouse, which gives the source system the ability to manage the data transfer on its schedule



ARCHITECTURE (CONT.)

- **Parallel Operations**
  - It is good to note that ETL processing should be managed in a parallel fashion to prevent unnecessary complications in processing.
- **ETL Roles**
  - Driven by an ETL architect.
  - Owns the ETL from the physical design, extraction, operations, and technical environment.
  - Has a team of ETL programmers.
  - **ETL Architect**
    - Oversees all aspects of the ETL environments.
    - Works hand in hand with the overall data warehouse architect.
    - Responsible for creating all ETL technical documents and ensure they are maintained.
    - Assist in the development of a notification system and flow diagrams showing all overall interdependencies.

76

Parallel Operations

- It is good to note that ETL processing should be managed in a parallel fashion to prevent unnecessary complications in processing.

ETL Roles

- The ETL process is driven by an ETL architect.

- This architect owns the ETL from the physical design, extraction, operations, and technical environment and has the ability of forward thinking.

- This architect has a team of ETL programmers who a responsible for creating the actual ETL routines that manage the captured data and loading into the warehouse.

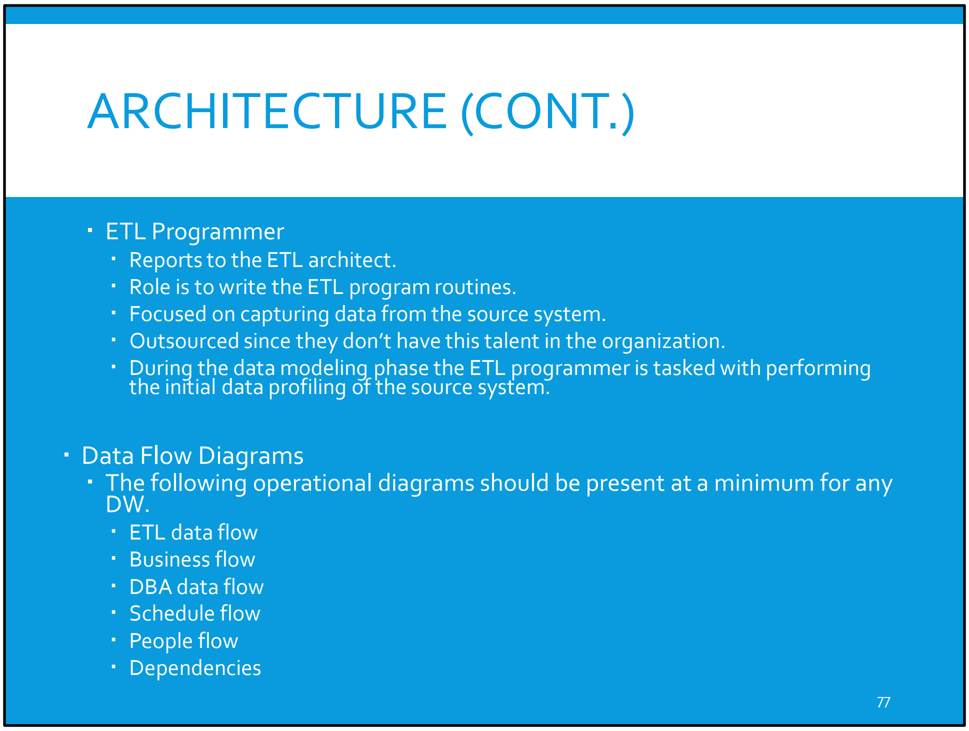
ETL Architect

- Oversees all aspects of the ETL environments and understands all the technical aspects of the data warehouse data.

- Works hand in hand with the overall data warehouse architect who has ultimate responsibility for ensuring the data warehouse is feasible and becomes tangible.

- This architect is initial responsible for creating all ETL technical documents and ensure they are maintained.

- This role also assist in the development of a notification system and flow diagrams showing all overall interdependencies.



ARCHITECTURE (CONT.)

- ETL Programmer
  - Reports to the ETL architect.
  - Role is to write the ETL program routines.
  - Focused on capturing data from the source system.
  - Outsourced since they don't have this talent in the organization.
  - During the data modeling phase the ETL programmer is tasked with performing the initial data profiling of the source system.
- Data Flow Diagrams
  - The following operational diagrams should be present at a minimum for any DW.
    - ETL data flow
    - Business flow
    - DBA data flow
    - Schedule flow
    - People flow
    - Dependencies

ETL Programmer

- Reports to the ETL architect.

- Their role is to write the ETL program routines and ensure the flow is seamless.

- They are focused on capturing data from the source system, transforming it whatever manner necessary to load it into the database.

- For many organizations this role is outsourced since they don't have this talent in the organization.

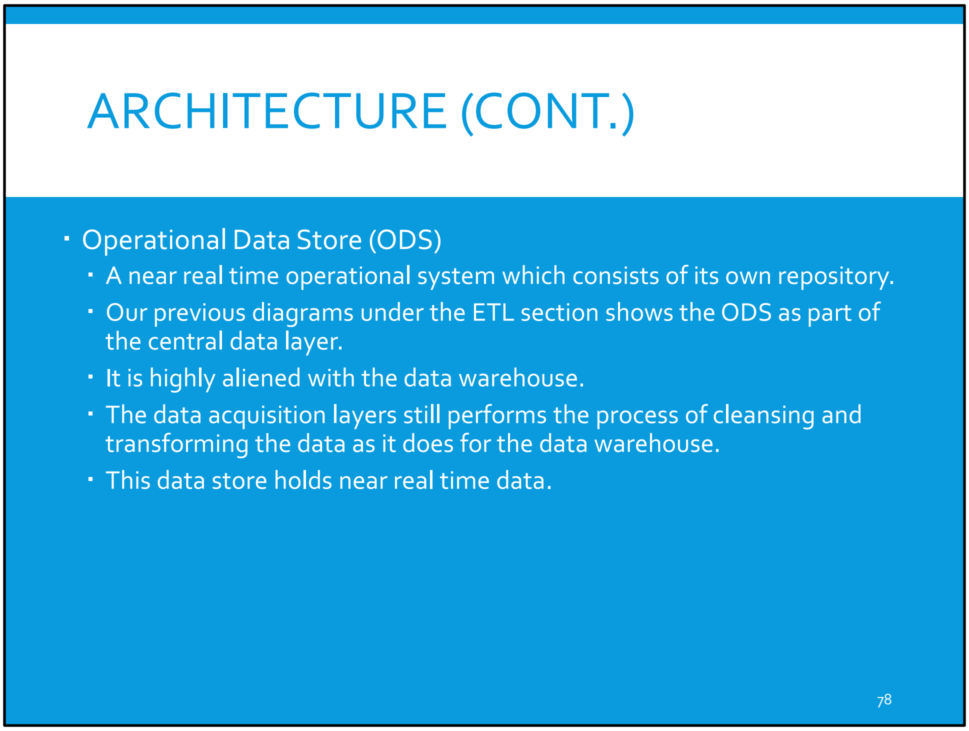
- During the data modeling phase the ETL programmer is tasked with performing the initial data profiling of the source system. This allows them to gain knowledge of what each table column holds and the expected data is present.

Data Flow Diagrams

- The following operational diagrams should be present at a minimum for any DW.

  - ETL data flow – shows overall data flow process.

- Business flow – pure business view of processes.
- DBA data flow – disk space usage, partitioning details, etc.
- Schedule flow – shows execution timings and durations.
- People flow – details on process owners.
- Dependencies – execution, table, timing dependencies.



ARCHITECTURE (CONT.)

- Operational Data Store (ODS)
  - A near real time operational system which consists of its own repository.
  - Our previous diagrams under the ETL section shows the ODS as part of the central data layer.
  - It is highly aliened with the data warehouse.
  - The data acquisition layers still performs the process of cleansing and transforming the data as it does for the data warehouse.
  - This data store holds near real time data.

78

Operational Data Store (ODS)

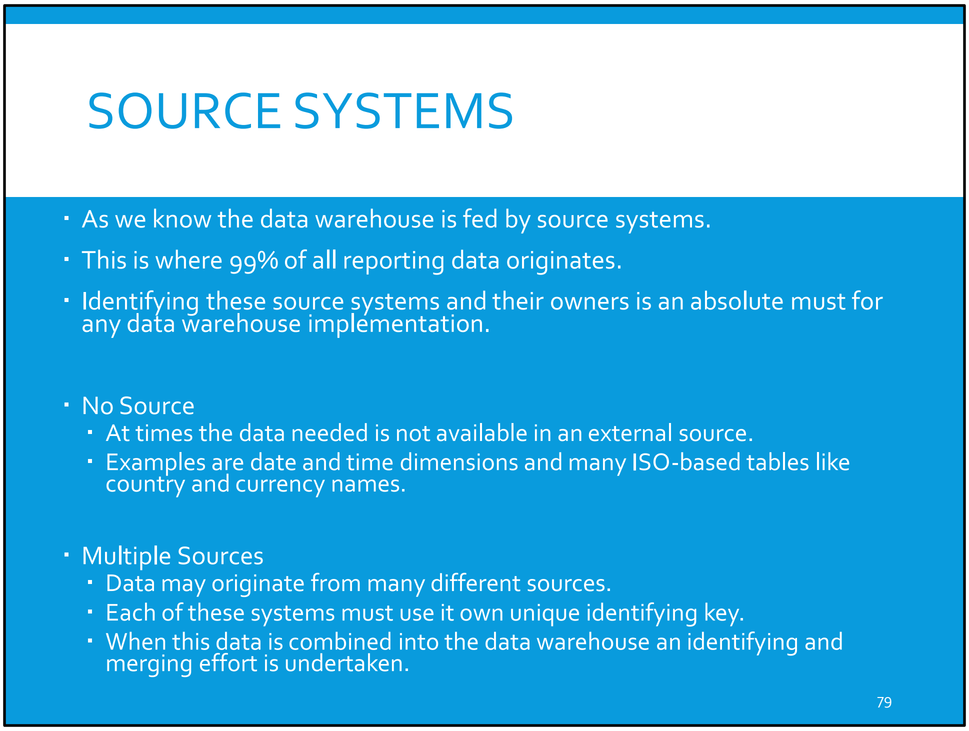
- This is a near real time operational system which consists of its own repository.

- Our previous diagrams under the ETL section shows the ODS as part of the central data layer, when in actuality it is own system and stands alone outside of the data warehouse.

- It is highly aliened with the data warehouse in such a manner that the data acquisition layer feeds data to the ODS.

- The data acquisition layers still performs the process of cleansing and transforming the data as it does for the data warehouse but would then feed the ODS from the source systems.

- This data store holds near real time data which means no history is captured.



SOURCE SYSTEMS

- As we know the data warehouse is fed by source systems.
- This is where 99% of all reporting data originates.
- Identifying these source systems and their owners is an absolute must for any data warehouse implementation.

- No Source
  - At times the data needed is not available in an external source.
  - Examples are date and time dimensions and many ISO-based tables like country and currency names.

- Multiple Sources
  - Data may originate from many different sources.
  - Each of these systems must use it own unique identifying key.
  - When this data is combined into the data warehouse an identifying and merging effort is undertaken.

- As we know the data warehouse is fed by source systems.

- This is where 99% of all reporting data originates.

- Identifying these source systems and their owners is an absolute must for any data warehouse implementation.

No Source

- At times the data needed is not available in an external source and must be created within the confines of the data warehouse itself.

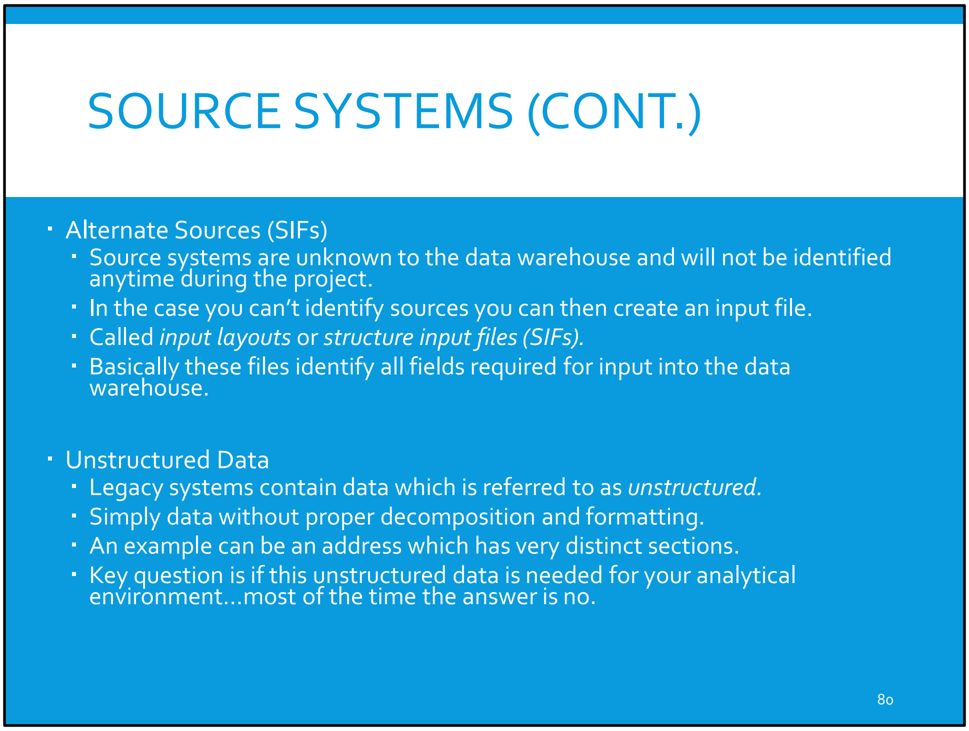
- Examples of this type of data are date and time dimensions and many ISO-based tables like country and currency names.

Multiple Sources

- At any given time data may originate from many different sources.

- Each of these systems must use it own unique identifying key to identify occurrence.

- When this data is combined into the data warehouse an identifying and merging effot is undertaken.



SOURCE SYSTEMS (CONT.)

- Alternate Sources (SIFs)
  - Source systems are unknown to the data warehouse and will not be identified anytime during the project.
  - In the case you can’t identify sources you can then create an input file.
  - Called *input layouts* or *structure input files (SIFs).*
  - Basically these files identify all fields required for input into the data warehouse.
- Unstructured Data
  - Legacy systems contain data which is referred to as *unstructured.*
  - Simply data without proper decomposition and formatting.
  - An example can be an address which has very distinct sections.
  - Key question is if this unstructured data is needed for your analytical environment…most of the time the answer is no.

Alternate Sources (SIFs)

- In many instances the source systems are unknown to the data warehouse and will not be identified anytime during the project.

- So in the case you can’t identify sources you can then create an input file to help identify what data the warehouse needs.

- These files are called *input layouts* or *structure input files (SIFs).*

- Basically these files identify all fields required for input into the data warehouse by the business with an “Action” field to help identify whether that row is an addition, update or expired.

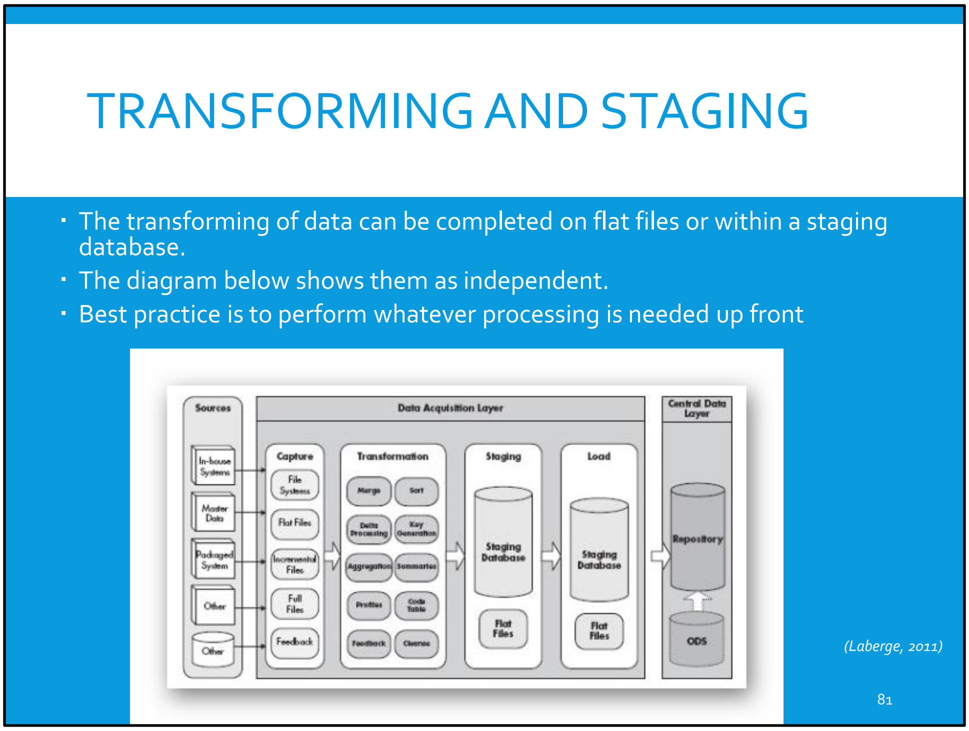
Unstructured Data

- In many organizations the legacy systems contain data which is referred to as *unstructured.*

- This is simply data without proper decomposition and formatting.

- An example can be an address which has very distinct sections like street number, name, city , state, and zip code. Being in an unstructured state all this data is in one column.

- So here the key question is if this unstructured data is needed for your analytical environment…most of the time the answer is no.



TRANSFORMING AND STAGING

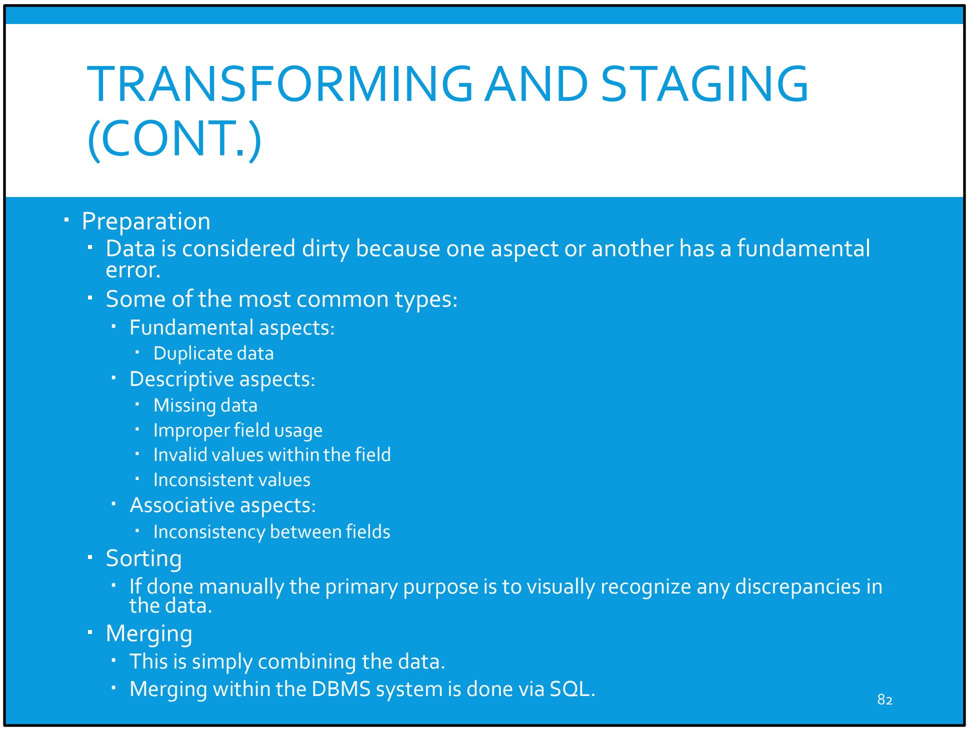
- The transforming of data can be completed on flat files or within a staging database.
- The diagram below shows them as independent.
- Best practice is to perform whatever processing is needed up front

(Laberge, 2011)

- The transforming of data can be completed on flat files or within a staging database, which is why we are discussion them together in this context.

- The diagram below shows them as independent to simply ensure the concepts are distinct.

- A best practice is to perform whatever processing is needed up front as soon as possible



TRANSFORMING AND STAGING (CONT.)

- Preparation
  - Data is considered dirty because one aspect or another has a fundamental error.
  - Some of the most common types:
    - Fundamental aspects:
      - Duplicate data
    - Descriptive aspects:
      - Missing data
      - Improper field usage
      - Invalid values within the field
      - Inconsistent values
    - Associative aspects:
      - Inconsistency between fields
  - Sorting
    - If done manually the primary purpose is to visually recognize any discrepancies in the data.
  - Merging
    - This is simply combining the data.
    - Merging within the DBMS system is done via SQL.

Preparation

- Data is considered dirty because one aspect or another has a fundamental error.
- This list details some of the most common types of dirty data scenarios.
  - Fundamental aspects:
    - Duplicate data
  - Descriptive aspects:
    - Missing data
    - Improper field usage
    - Invalid values within the field

        - Inconsistent values

    - Associative aspects:

        - Inconsistency between fields

- Sorting

    - If done manually the primary purpose is to visually recognize any discrepancies in the data.

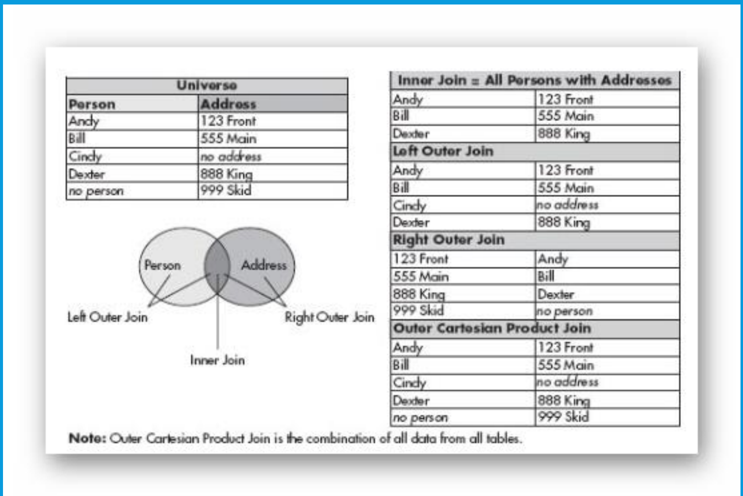
- Merging

    - This is simply combining the data.

    - Merging within the DBMS system is done via SQL. The chart on the next slide shows the different SQL joins.

TRANSFORMING AND STAGING (CONT.)

Merge using fundamental SQL joins

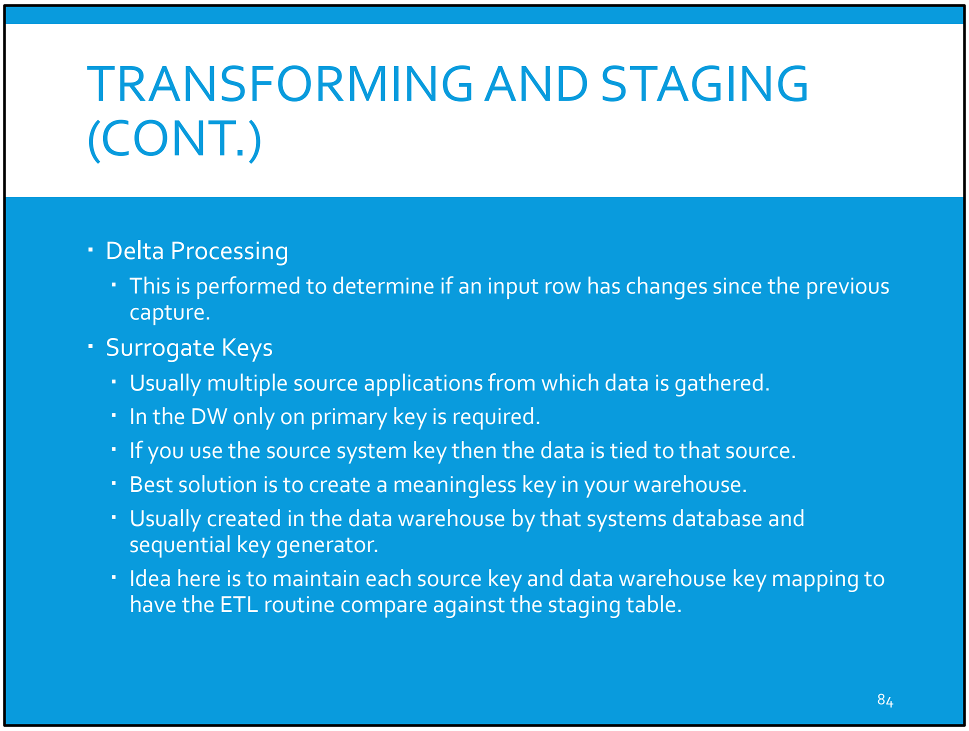


Universe	
Person	**Address**
Andy	123 Front
Bill	555 Main
Cindy	*no address*
Dexter	888 King
no person	999 Skid

Inner Join = All Persons with Addresses	
Andy	123 Front
Bill	555 Main
Dexter	888 King
Left Outer Join	
Andy	123 Front
Bill	555 Main
Cindy	*no address*
Dexter	888 King
Right Outer Join	
123 Front	Andy
555 Main	Bill
888 King	Dexter
999 Skid	*no person*
Outer Cartesian Product Join	
Andy	123 Front
Bill	555 Main
Cindy	*no address*
Dexter	888 King
no person	999 Skid

Note: Outer Cartesian Product Join is the combination of all data from all tables.

(Laberge, 2011)



TRANSFORMING AND STAGING (CONT.)

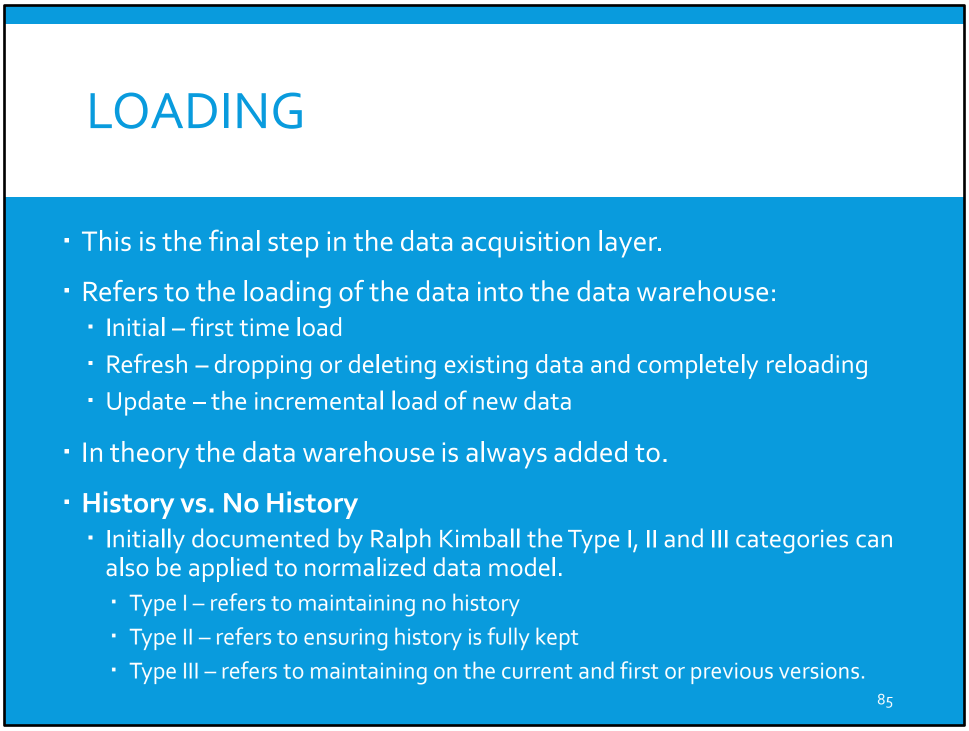
- Delta Processing
  - This is performed to determine if an input row has changes since the previous capture.
- Surrogate Keys
  - Usually multiple source applications from which data is gathered.
  - In the DW only on primary key is required.
  - If you use the source system key then the data is tied to that source.
  - Best solution is to create a meaningless key in your warehouse.
  - Usually created in the data warehouse by that systems database and sequential key generator.
  - Idea here is to maintain each source key and data warehouse key mapping to have the ETL routine compare against the staging table.

Delta Processing

- This is performed to determine if an input row has changes since the previous capture.

Surrogate Keys

- There are usually multiple source applications from which data is gathered.
- In the DW only on primary key is required, but which one do you use?
- If you use the source system key then the data is tied to that source, and if the source systems chooses to change its primary key then the data warehouse has an issue.
- The best solution is to create a meaningless key in your warehouse and always map this key to the source system key. This type of key is called a *surrogate key*.
- Usually created in the data warehouse by that systems database and sequential key generator.
- The idea here is to maintain each source key and data warehouse key mapping to have the ETL routine compare against the staging table.

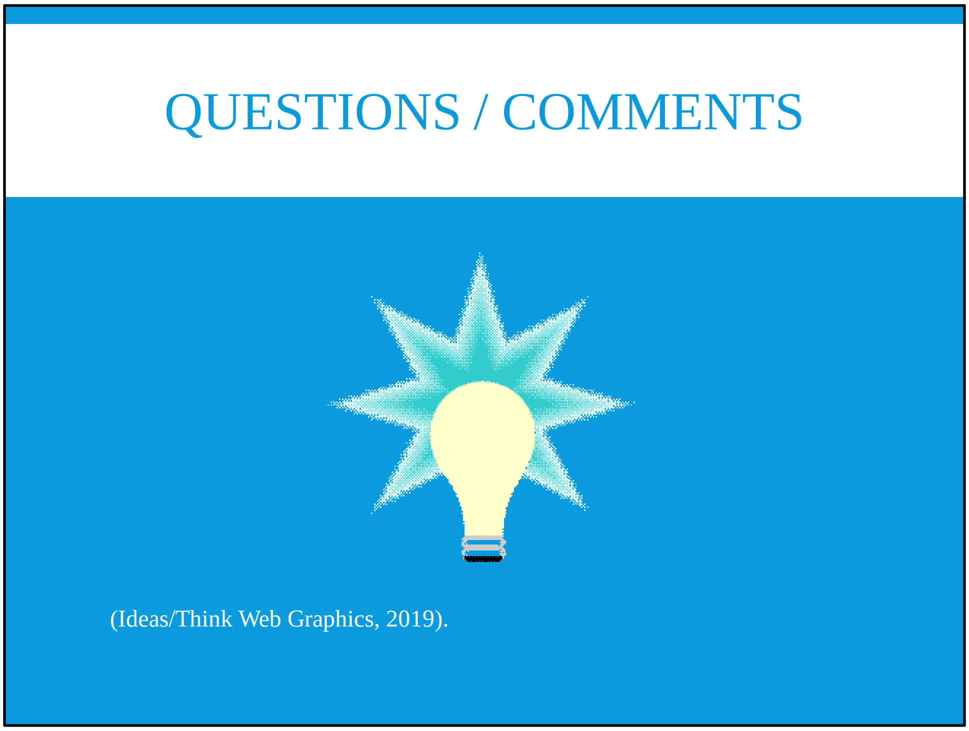


LOADING

- This is the final step in the data acquisition layer.
- Refers to the loading of the data into the data warehouse:
  - Initial – first time load
  - Refresh – dropping or deleting existing data and completely reloading
  - Update – the incremental load of new data
- In theory the data warehouse is always added to.
- **History vs. No History**
  - Initially documented by Ralph Kimball the Type I, II and III categories can also be applied to normalized data model.
    - Type I – refers to maintaining no history
    - Type II – refers to ensuring history is fully kept
    - Type III – refers to maintaining on the current and first or previous versions.

85

- This is the final step in the data acquisition layer.
- It refers to the loading of the data into the data warehouse repository, which includes three types of loading:
  - Initial – first time load
  - Refresh – dropping or deleting existing data and completely reloading
  - Update – the incremental load of new data
- In theory the data warehouse is always added to, in reality existing data can be modified which is know as Type 1 processing.
- History vs. No History
  - Initially documented by Ralph Kimball the Type I, II and III categories can also be applied to normalized data model.
    - Type I – refers to maintaining no history
    - Type II – refers to ensuring history is fully kept
    - Type III – refers to maintaining on the current and first or previous versions.



QUESTIONS / COMMENTS

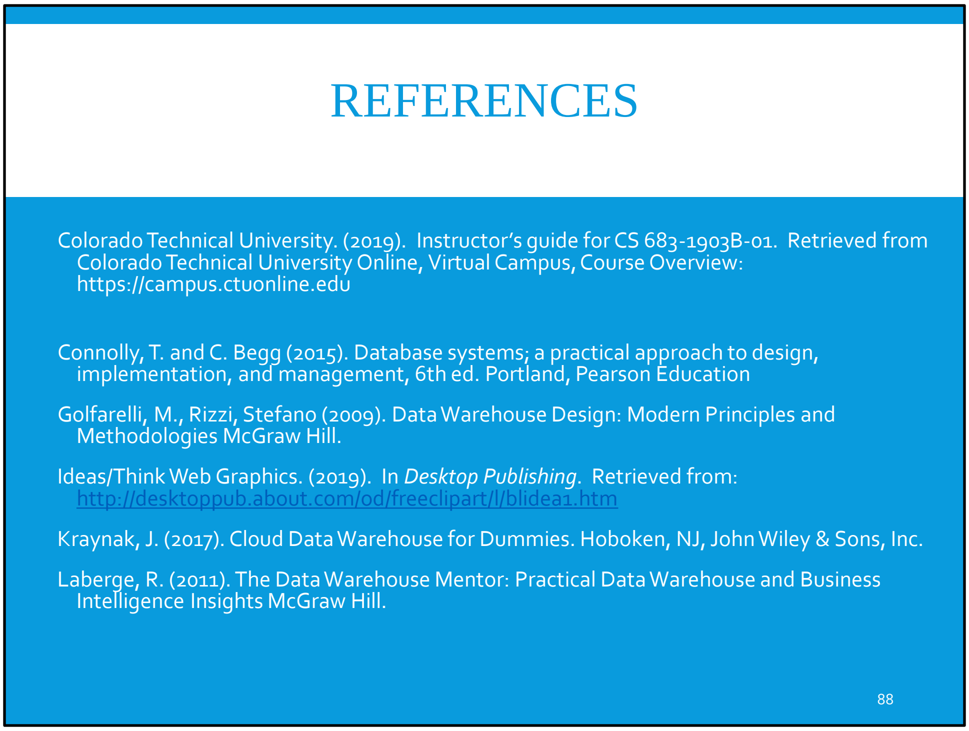
(Ideas/Think Web Graphics, 2019).

CONTACT INFORMATION

- My e-mail address- JConklin@coloradotech.edu
- Office Hours - Wednesday 6:00 P.M. – 7:00 P.M. CST

  Saturday 11:00 A.M. – 12:00 P.M. CST
- Live Chats - Thursday 7:00 P.M. – 8:00 P.M. CST

* Please note that only one live chat session per week is required for this course. However, optional live chat sessions may be held sporadically throughout the course.



REFERENCES

Colorado Technical University. (2019). Instructor’s guide for CS 683-1903B-01. Retrieved from Colorado Technical University Online, Virtual Campus, Course Overview: https://campus.ctuonline.edu

Connolly, T. and C. Begg (2015). Database systems; a practical approach to design, implementation, and management, 6th ed. Portland, Pearson Education

Golfarelli, M., Rizzi, Stefano (2009). Data Warehouse Design: Modern Principles and Methodologies McGraw Hill.

Ideas/Think Web Graphics. (2019). In *Desktop Publishing*. Retrieved from: http://desktoppub.about.com/od/freeclipart/l/blidea1.htm

Kraynak, J. (2017). Cloud Data Warehouse for Dummies. Hoboken, NJ, John Wiley & Sons, Inc.

Laberge, R. (2011). The Data Warehouse Mentor: Practical Data Warehouse and Business Intelligence Insights McGraw Hill.