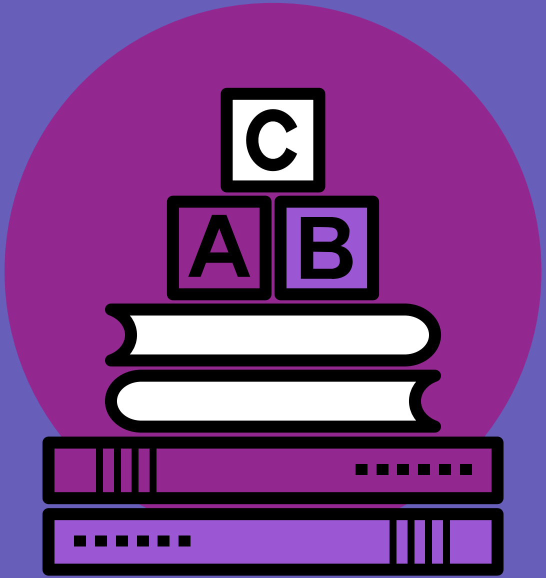


WIE3008 Business Analytics and Intelligence

INTRODUCTION

AP Dr. Nor Liyana Mohd Shuib

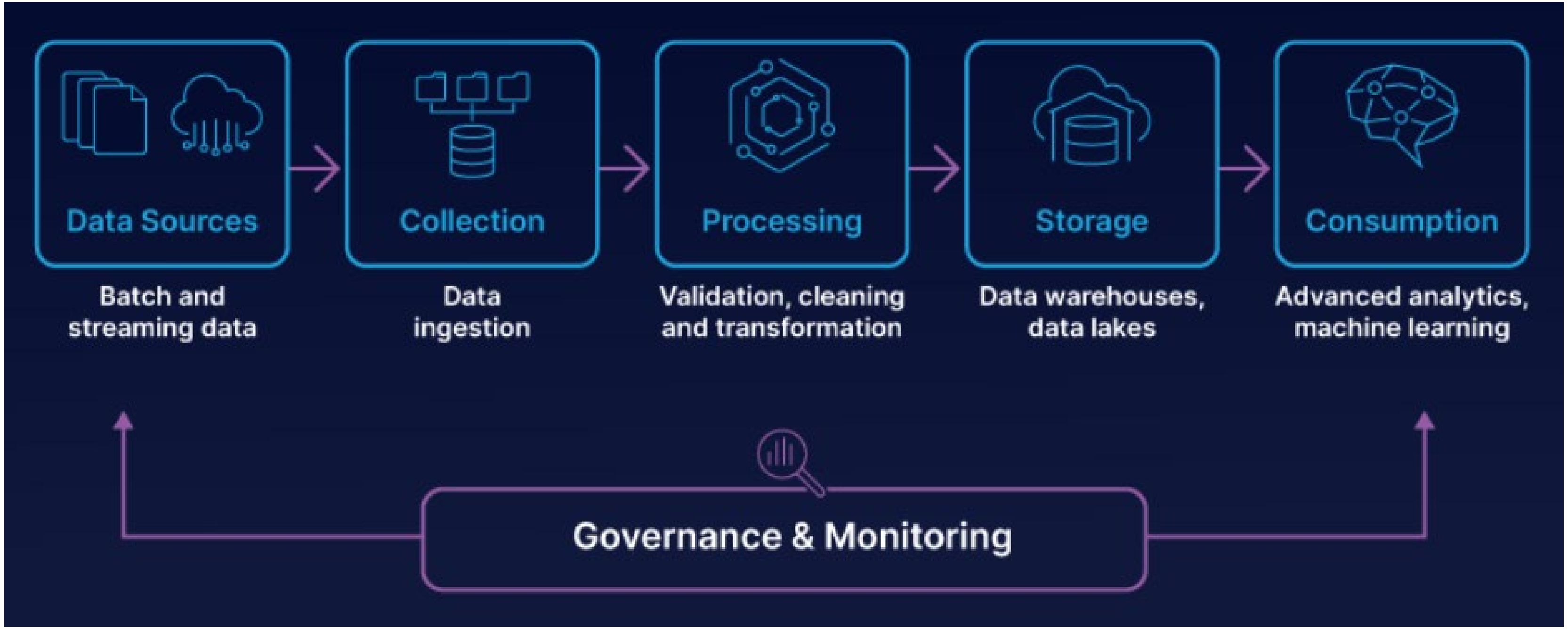
Data Pipeline

Data Pipeline

- A *pipeline definition* specifies the business logic of your data management.

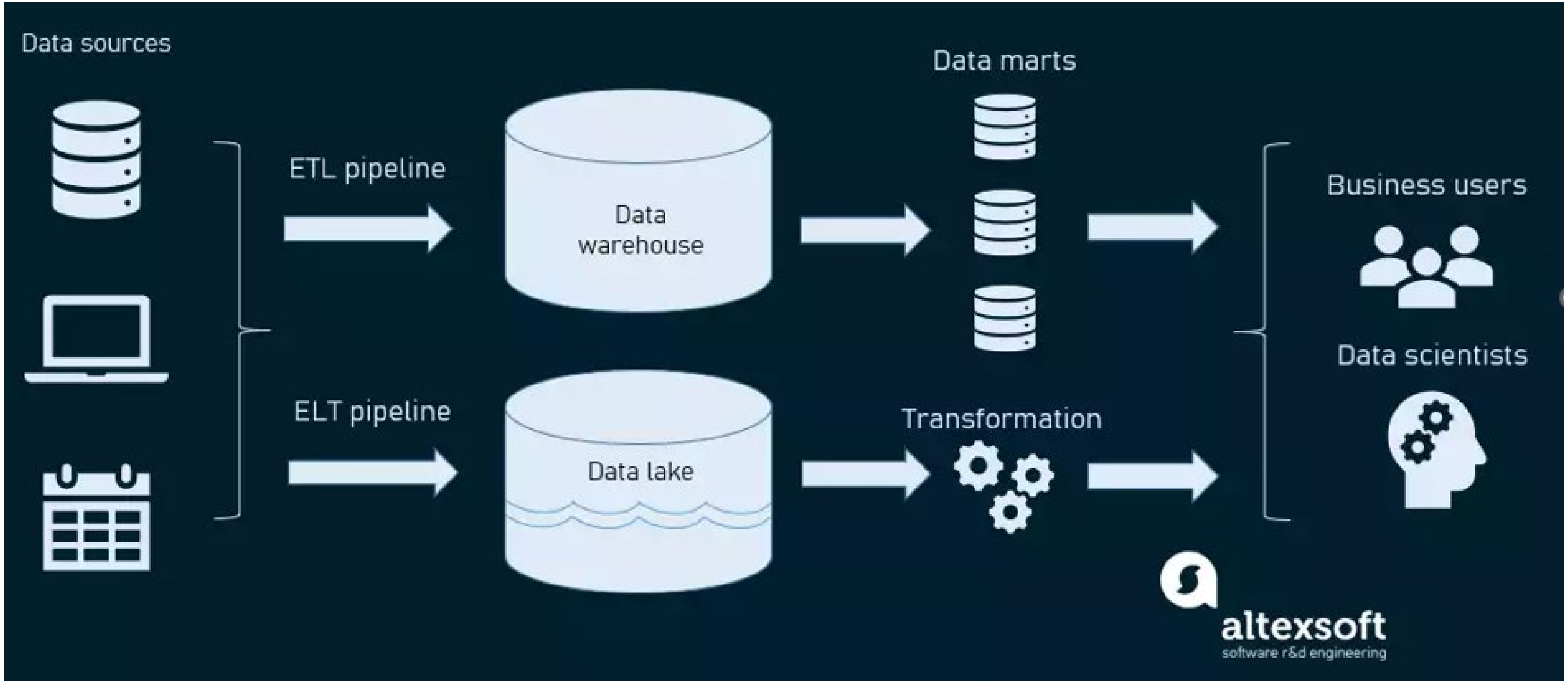
- Adata pipeline is a set of actions that ingest raw data from disparate sources and move the data to a destination for storage and analysis. Apipeline also may include filtering and features that provide resiliency against failure (Stichdata.com).

Data Pipeline



Source

Data Architecture



Source

Data Warehouse

vs Data Lake

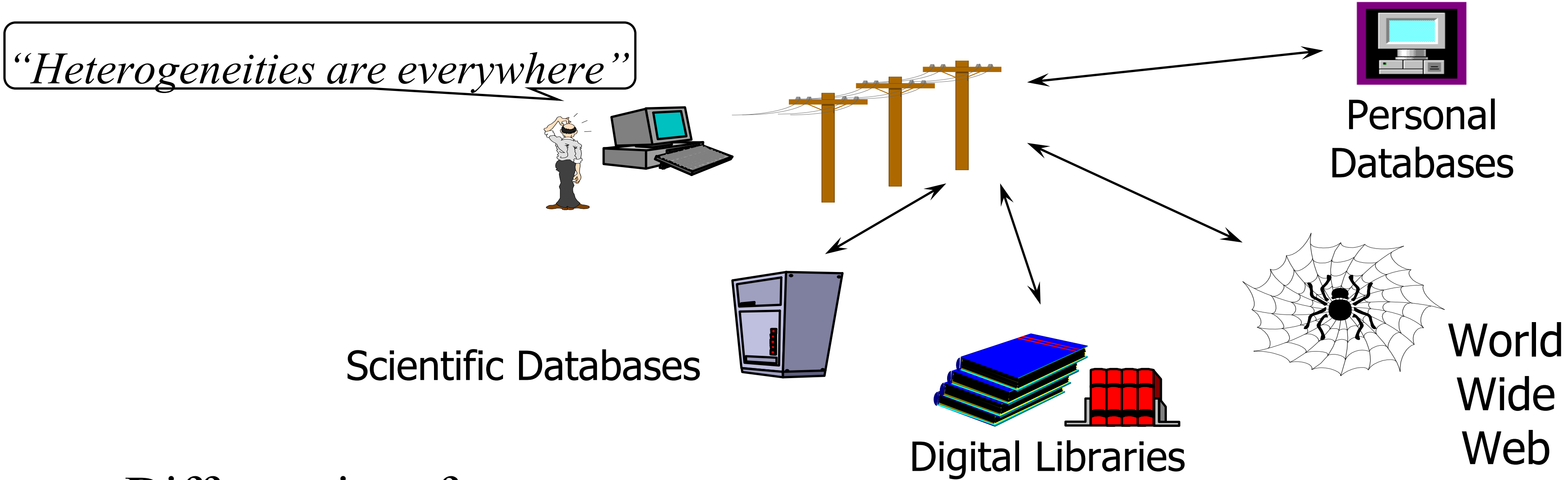
Data, Data everywhere

yet ...

- I can’t find the data I need
  - data is scattered over the network
  - many versions, subtle differences

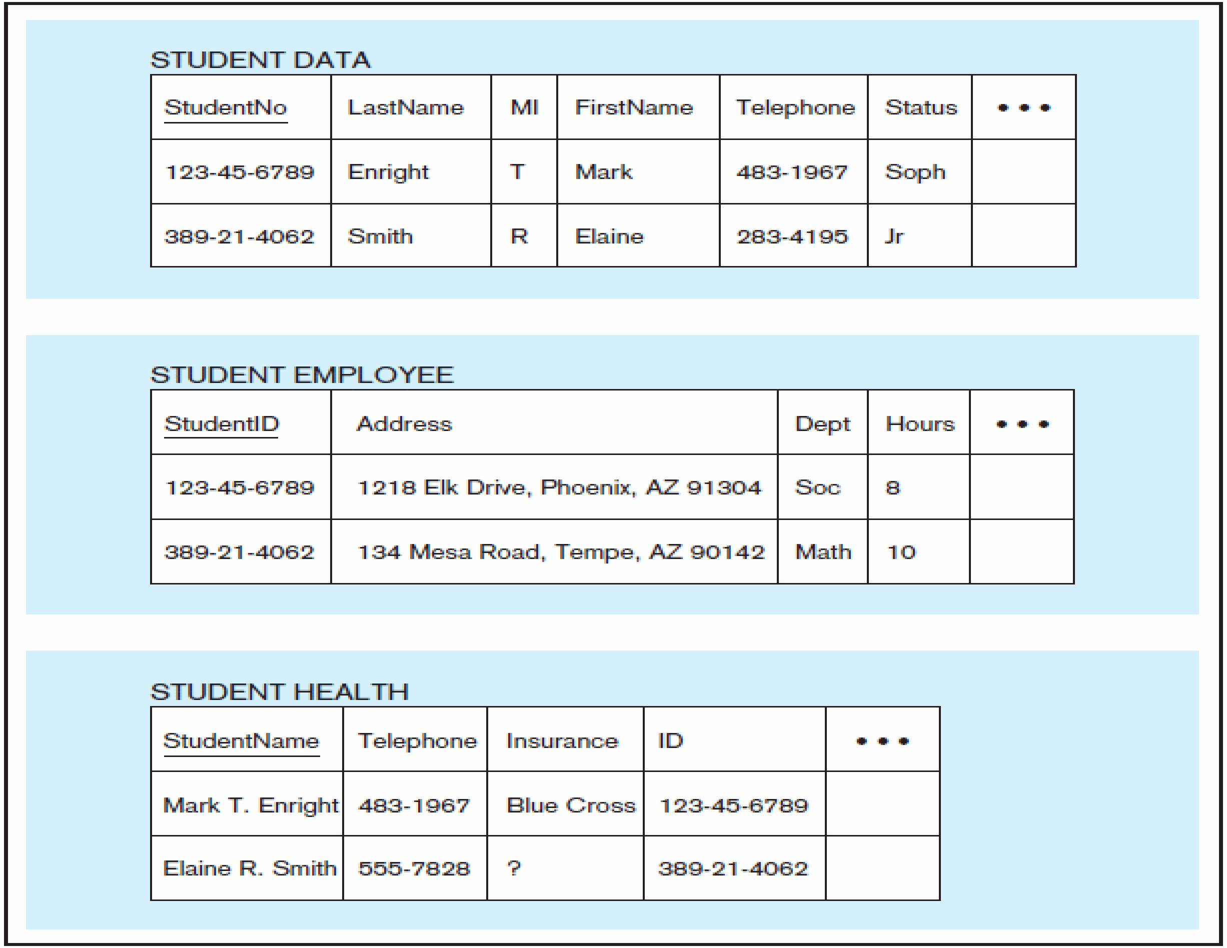
- ⌘ I can’t get the data I need
  - ☐ need an expert to get the data
- ⌘ I can’t understand the data I found
  - ☐ available data poorly documented
- ⌘ I can’t use the data I found
  - ☐ results are unexpected
  - ☐ data needs to be transformed from one form to other

Problem: Heterogeneous Information Sources



- Different interfaces
- Different data representations
- Duplicate and inconsistent information

Figure 1 Examples of heterogeneous data



STUDENT DATA

StudentNo	LastName	MI	FirstName	Telephone	Status	• • •
123-45-6789	Enright	T	Mark	483-1967	Soph	
389-21-4062	Smith	R	Elaine	283-4195	Jr	

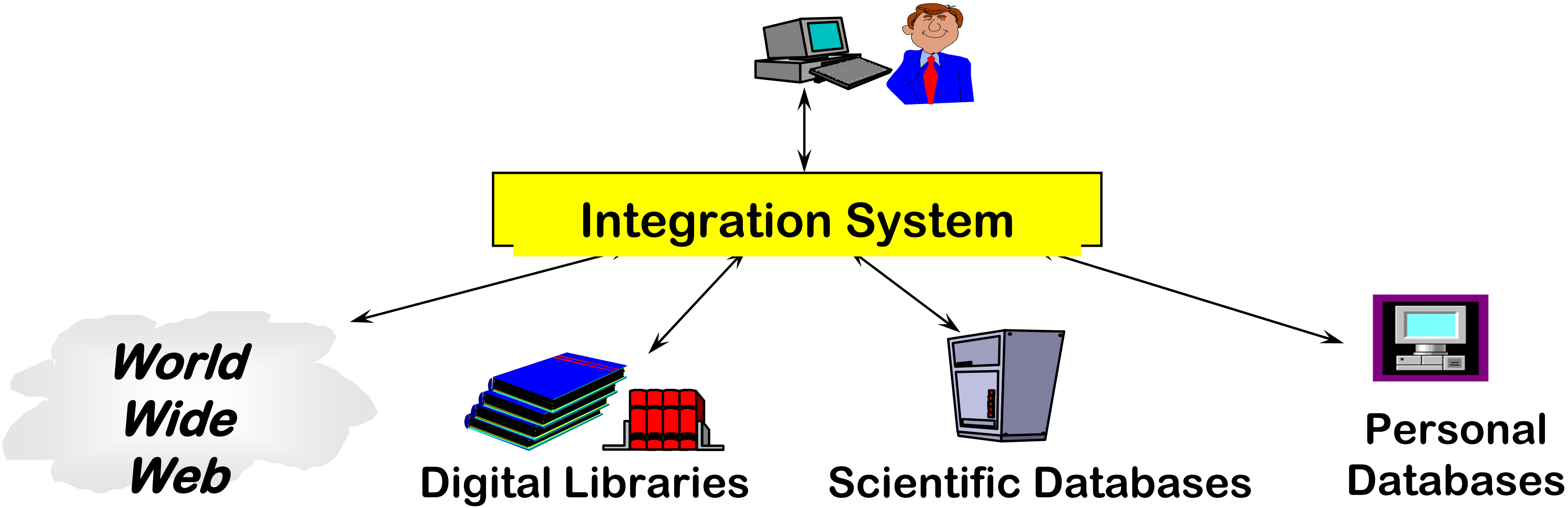
STUDENT EMPLOYEE

StudentID	Address	Dept	Hours	• • •
123-45-6789	1218 Elk Drive, Phoenix, AZ 91304	Soc	8	
389-21-4062	134 Mesa Road, Tempe, AZ 90142	Math	10	

STUDENT HEALTH

StudentName	Telephone	Insurance	ID	• • •
Mark T. Enright	483-1967	Blue Cross	123-45-6789	
Elaine R. Smith	555-7828	?	389-21-4062	

Goal: Unified Access to Data



- Collects and combines information
- Provides integrated view, uniform user interface
- Supports sharing

Data Warehouse vs Data Lake

Dimension	Data Warehouse	Data Lake
Definition	A data warehouse is a repository for data collected and generated by business applications for a predetermined purpose.	A data lake is a vast repository that stores raw data in its native format.
The nature of data	Structured, processed, predefined schema	Any data in raw/native format, no predefined schema
Purpose of Data	Currently in use	Not yet determined
Processing	Schema-on-write (SQL)	Schema-on-read (No SQL)
Retrieval speed	Very fast	Slow
Cost	Expensive for large data volumes	Designed for low-cost storage
Agility	Less agile, fixed configuration	Highly agile, flexible configuration
Novelty/newness	Not new/matured	Very new/maturing
Security	Security	Not yet well-secured
Accessibility	More complicated and costly to make changes	Highly accessible and quick to update
Process	ETL	ELT
Users	Business Analyst, Manager	Data scientists
Vendors	AWS, Cloudera, IBM, Google, Microsoft, Oracle, Teradata, SAP, Snowflake	AWS, Google, Informatica, Microsoft, Teradata and other data management providers

DATA LAKE vs DATA WAREHOUSE

DATA LAKE

Data

unstructured

Users

Data Scientists,
Data Analysts

Use cases

Stream Processing,
Machine Learning,
Real time analysis

Raw

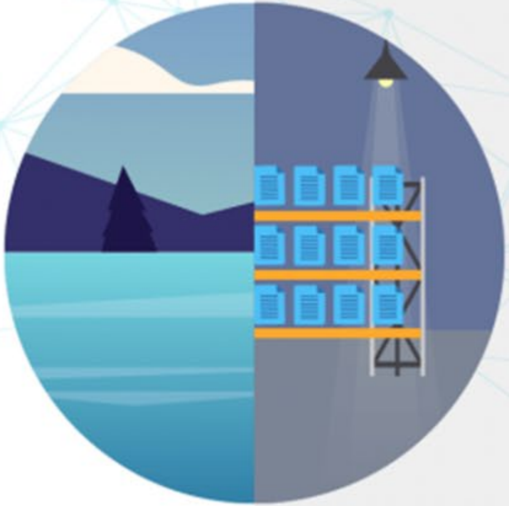
Data Lakes contain unstructured, semi structured and structured data with minimal processing. It can be used to contain unconventional data such as log and sensor data

Large

Data Lakes contain vast amounts of data in the order of petabytes. Since the data can be in any form or size, large amounts of unstructured data can be stored indefinitely and can be transformed when in use only

Undefined

Data in data lakes can be used for a wide variety of applications, such as Machine Learning, Streaming analytics, and AI



DATA WAREHOUSE

Data

Structured

Users

Business Analysts

Use cases

Batch Processing,
BI, Reporting

Refined

Data Warehouses contain highly structured data that is cleaned, pre-processed and refined. This data is stored for very specific use cases such as BI.

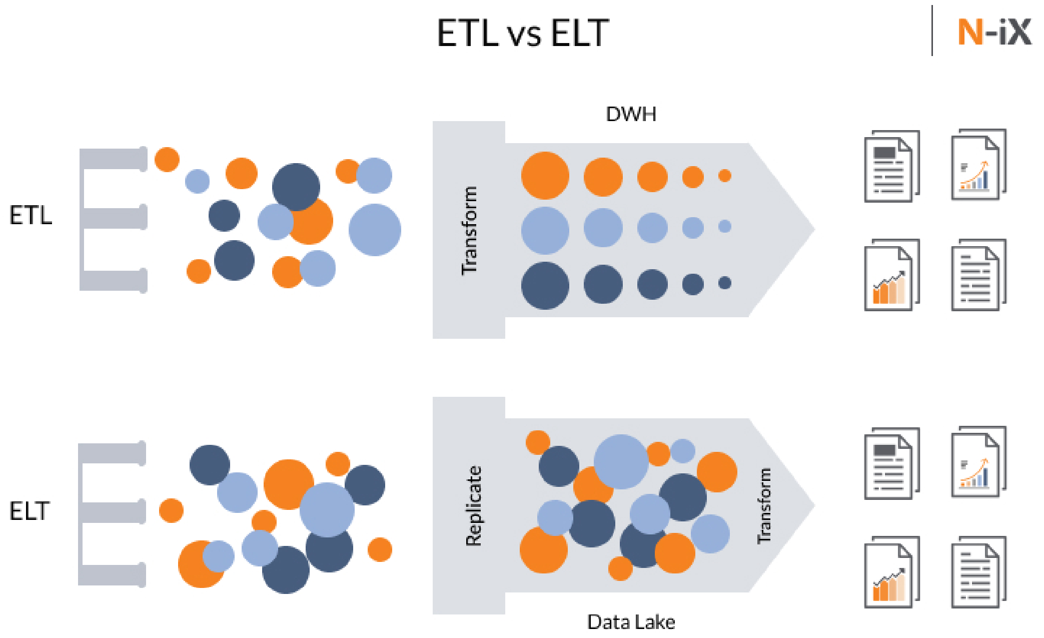
Smaller

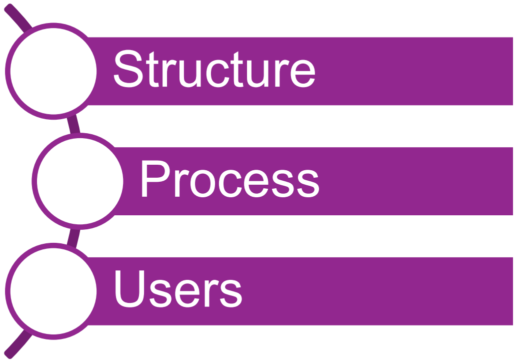
Data Warehouses contain less data in the order of terabytes. In order to maintain data cleanliness and health of the warehouse, Data must be processed before ingestion and periodic purging of data is necessary

Relational

Data Warehouses contain historic and relational data, such as transaction systems, operations etc

Data Warehouse vs Data Lake





Structure

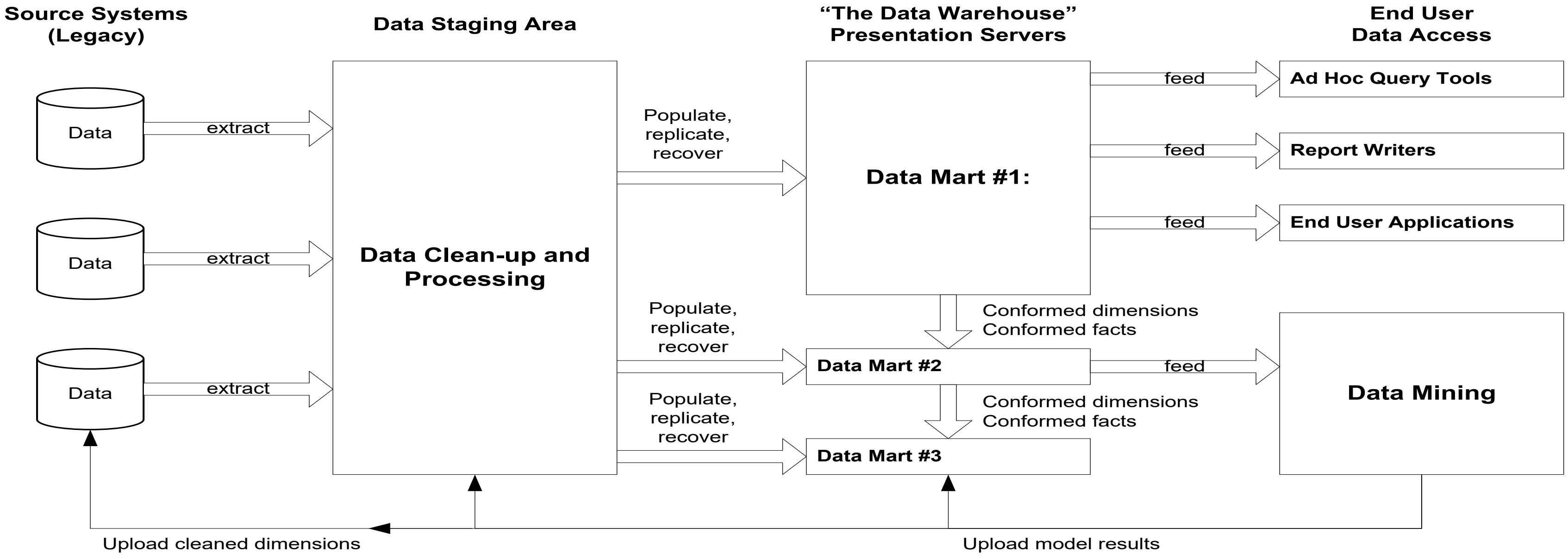
Process

Users

https://www.n-ix.com/data-lake-vs-data-warehouse/

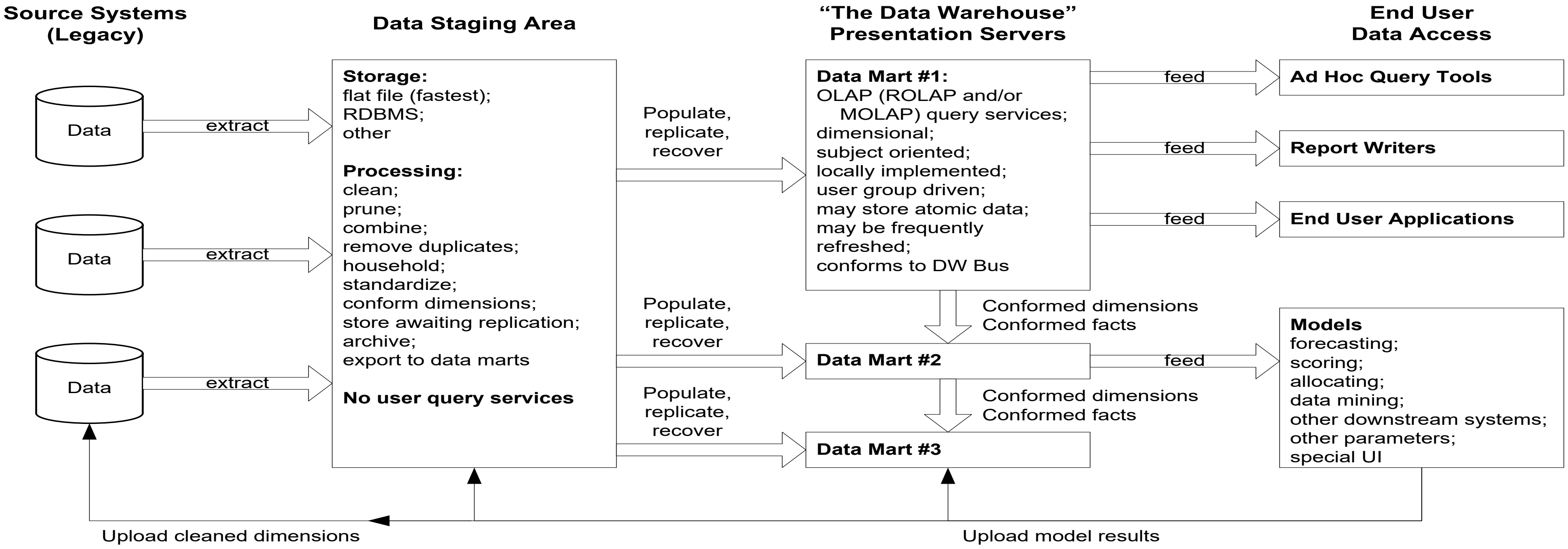
DW Components

Data Warehouse Components



SOURCE: Ralph Kimball

Data Warehouse Components – Detailed



Source Systems (Legacy)

Data Staging Area

"The Data Warehouse" Presentation Servers

End User Data Access

Storage:
flat file (fastest);
RDBMS;
other

Processing:
clean;
prune;
combine;
remove duplicates;
household;
standardize;
conform dimensions;
store awaiting replication;
archive;
export to data marts

No user query services

Data Mart #1:
OLAP (ROLAP and/or MOLAP) query services;
dimensional;
subject oriented;
locally implemented;
user group driven;
may store atomic data;
may be frequently refreshed;
conforms to DW Bus

Data Mart #2

Data Mart #3

Ad Hoc Query Tools

Report Writers

End User Applications

Models
forecasting;
scoring;
allocating;
data mining;
other downstream systems;
other parameters;
special UI

DW Component

- **Source Systems (Data sources)**
  - Operational databases
  - Other internal or external sources of information (e.g. files)
- **Data Staging**
  - **Extraction - Transformation - Loading (ETL)** tools for manipulating data from sources
  - Data staging area: Intermediate database where manipulation is done
- **OLAP**
  - OLAP Server: Supports multidimensional data and operations
- **End User:** Deals with data analysis and visualization
  - Composed of OLAP tools, reporting tools, **statistical** tools, **data-mining** tools, …

DW Component: Data Staging

- Extraction: Gathers data from multiple heterogeneous data sources
  - May be operational databases or files in various formats
  - May be internal or external to the organization
  - Uses APIs such as ODBC, JDBC, …for achieving interoperability
- Transformation: Modifies data to conform to the data warehouse format
  - Cleaning: Removes errors, inconsistencies, format transformation
  - Integration: Reconciles data from different sources
  - Aggregation: Summarizes data according to the granularity (level of detail) of the DW
- Loading: Feeds the DW with transformed data
  - Also includes refreshing the data warehouse at a specified frequency

DW Component: Data Warehouse

- Enterprise data warehouse: Centralized DW that encompasses all areas in an organization
- Data mart: Specialized DW targeted to a particular functional area or user group
  - Their data can be derived from the enterprise DW or collected from data sources
- Metadata repository: Describes the content of the DW
  - Business metadata: Meaning (semantics) of data, organization rules, policies, constraints, …
  - Technical metadata: How data is structured/ stored in the computer
    - Data sources, data warehouse, and data marts: logical and physical schemas, security information, monitoring information …
    - ETL process: Data lineage (trace to sources), rules, defaults, refresh and purging rules, algorithms for summarization, …

DW Component: Data Mart

A departmental small-scale “DW” that stores only limited/relevant data

- **Dependent data mart**
  A subset that is created directly from a data warehouse
- **Independent data mart**
  A small data warehouse designed for a strategic business unit or a department

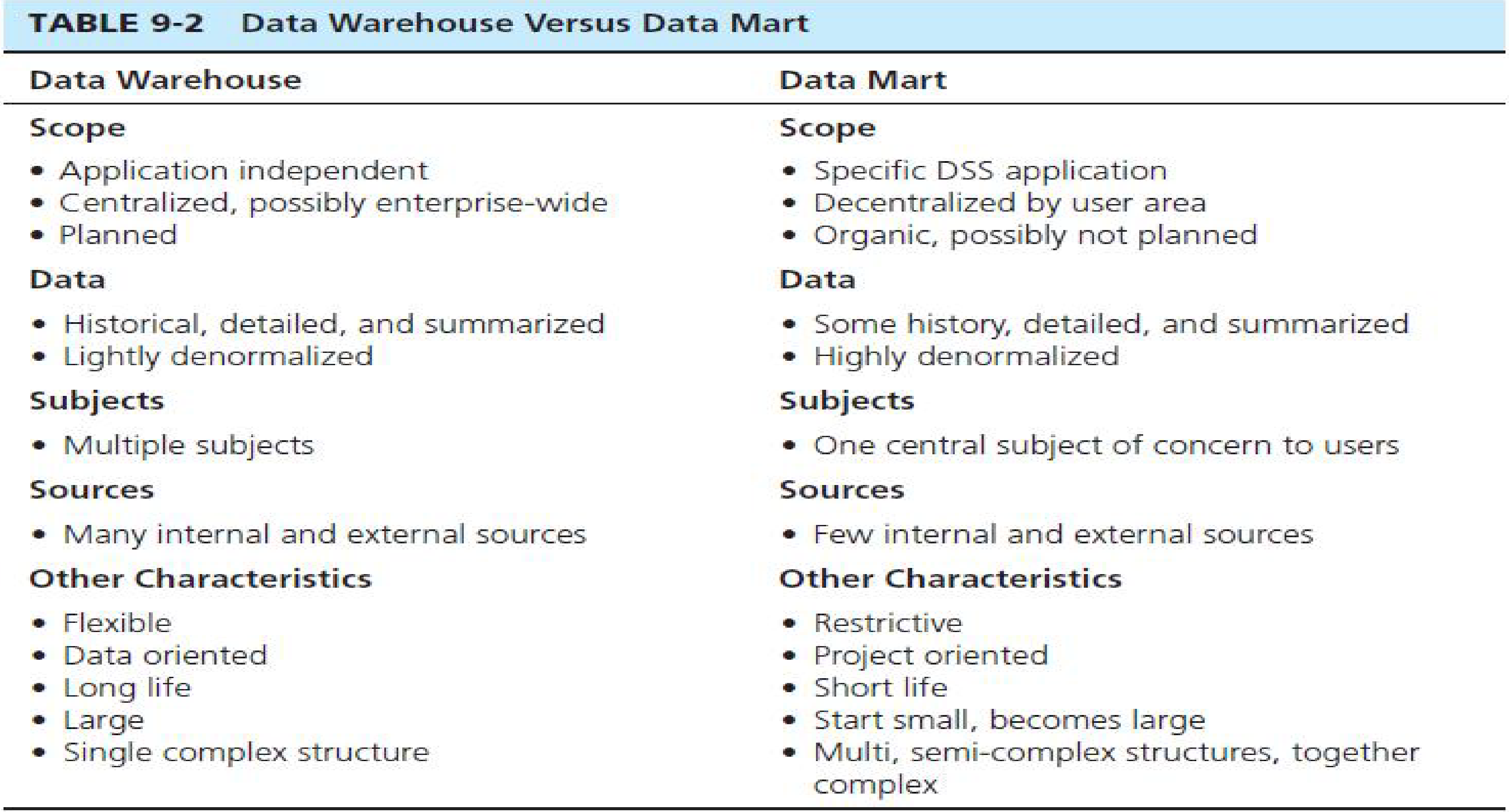
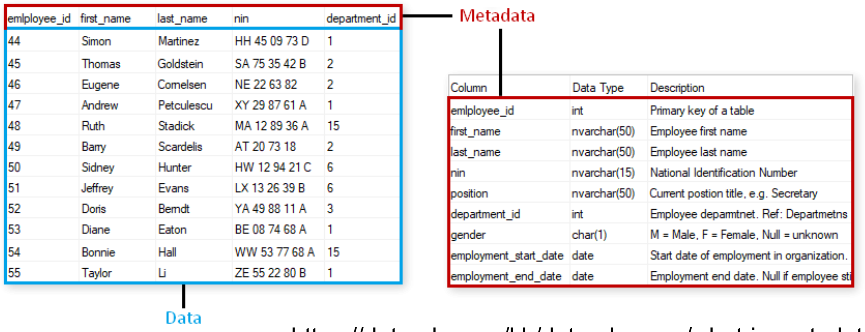


TABLE 9-2 Data Warehouse Versus Data Mart

Data Warehouse	Data Mart
Scope	**Scope**
• Application independent • Centralized, possibly enterprise-wide • Planned	• Specific DSS application • Decentralized by user area • Organic, possibly not planned
Data	**Data**
• Historical, detailed, and summarized • Lightly denormalized	• Some history, detailed, and summarized • Highly denormalized
Subjects	**Subjects**
• Multiple subjects	• One central subject of concern to users
Sources	**Sources**
• Many internal and external sources	• Few internal and external sources
Other Characteristics	**Other Characteristics**
• Flexible • Data oriented • Long life • Large • Single complex structure	• Restrictive • Project oriented • Short life • Start small, becomes large • Multi, semi-complex structures, together complex

DW Component: Metadata

- Metadata is Data about Data.
- Metadata describe the contents and its acquisition and use
- To ease indexing and search
- Information can include:
  - Source System(s) of the Data, contact information
  - Related tables or subject areas
  - Programs or Processes which use the data
  - Population rules (Update or Insert and how often)
  - Status of the Data Warehouse's processing and condition



emlployee_id	first_name	last_name	nin	department_id
44	Simon	Martinez	HH 45 09 73 D	1
45	Thomas	Goldstein	SA 75 35 42 B	2
46	Eugene	Comelsen	NE 22 63 82	2
47	Andrew	Petculescu	XY 29 87 61 A	1
48	Ruth	Stadick	MA 12 89 36 A	15
49	Barry	Scardelis	AT 20 73 18	2
50	Sidney	Hunter	HW 12 94 21 C	6
51	Jeffrey	Evans	LX 13 26 39 B	6
52	Doris	Bemdt	YA 49 88 11 A	3
53	Diane	Eaton	BE 08 74 68 A	1
54	Bonnie	Hall	WW 53 77 68 A	15
55	Taylor	Li	ZE 55 22 80 B	1

Column	Data Type	Description
emlployee_id	int	Primary key of a table
first_name	nvarchar(50)	Employee first name
last_name	nvarchar(50)	Employee last name
nin	nvarchar(15)	National Identification Number
position	nvarchar(50)	Current postion title, e.g. Secretary
department_id	int	Employee depamtnet. Ref: Departmetns
gender	char(1)	M = Male, F = Female, Null = unknown
employment_start_date	date	Start date of employment in organization.
employment_end_date	date	Employment end date. Null if employee sti

https://dataedo.com/kb/data-glossary/what-is-metadata

DW Component: OLAP

- OLAP servers that provides multidimensional view from DWs and data marts
  - Can be ROLAP, MOLAP, or HOLAP
- Most database products provide OLAP extensions and related tools for manipulating cubes
- However, no standardized language for querying data cubes
  - Oracle uses Java and query language OLAP DML
  - SQL Server uses .NET and query language MDX
- XMLA (XML for Analysis) aims at providing a common language for exchanging multidimensional data

DW Component: End User Data Access

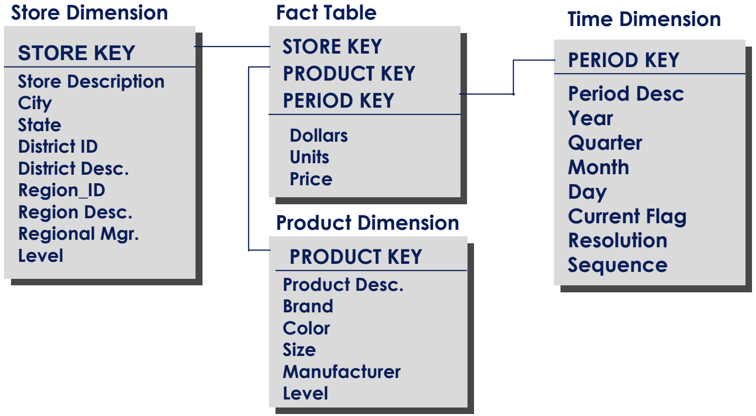
- OLAP tools: Allow interactive exploration and manipulation of the warehouse data
    - Facilitate formulation of ad hoc queries (no prior knowledge of them)
- Reporting tools: Enable production, delivery and management of reports (paper and web-based)
    - Use predefined queries
- Statistical tools: Used to analyze and visualize the cube data using statistical methods
- Data-mining tools: Allow users to analyze data to discover patterns, trends, enable predictions

SCHEMA

Schemas

- Facts, dimensions, and attributes can be organized in several ways, called schemas.
- The choice of schema depends on variables such as the type of reporting that the model needs to facilitate and the type of Business Intelligence tool being used.

The “Classic” Star Schema



Store Dimension

STORE KEY

Store Description
City
State
District ID
District Desc.
Region_ID
Region Desc.
Regional Mgr.
Level

Fact Table

STORE KEY
PRODUCT KEY
PERIOD KEY

Dollars
Units
Price

Product Dimension

PRODUCT KEY

Product Desc.
Brand
Color
Size
Manufacturer
Level

Time Dimension

PERIOD KEY

Period Desc
Year
Quarter
Month
Day
Current Flag
Resolution
Sequence

- A single fact table, with detail and summary data
- Fact table primary key has only one key column per dimension
- Each key is generated
- Each dimension is a single table, highly denormalized

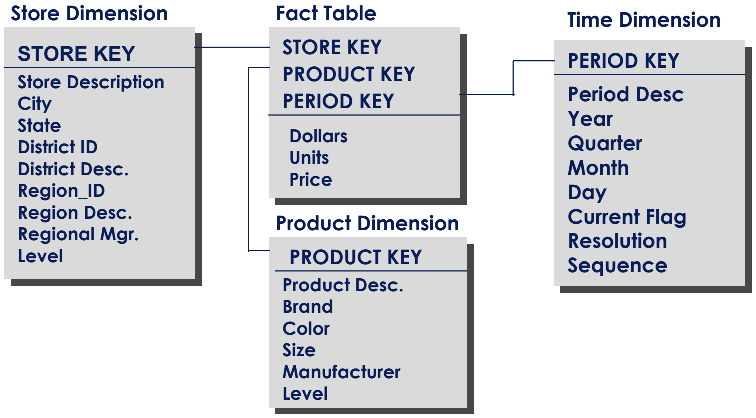
Benefits: Easy to understand, easy to define hierarchies, reduces # of physical joins, low maintenance, very simple metadata

Drawbacks: Summary data in the fact table yields poorer performance for summary levels, huge dimension tables a problem

Star schema

- In a ROLAP system, relations are often stored with star schemas
- A star schema consists of the fact table and one or more dimension tables.
    - The most commonly used and the simplest style of dimensional modeling
    - Contain a **fact table** surrounded by and connected to several **dimension tables**
- One fact table and a set of dimension tables
- Referential integrity constraints between fact table and dimension tables
- Dimension tables may contain redundancy in the presence of hierarchies
- Used to implement dimensional analysis using relational database technology
- Very common in data warehouse
    - Many variations
- Fact table
    - additive and non additive facts
- Dimension tables
    - become constraints (WHERE part of SQL)
    - A fact table in the middle connected to a set of dimension tables

The “Classic” Star Schema



Store Dimension

STORE KEY
Store Description
City
State
District ID
District Desc.
Region_ID
Region Desc.
Regional Mgr.
Level

Fact Table

STORE KEY
PRODUCT KEY
PERIOD KEY
Dollars
Units
Price

Product Dimension

PRODUCT KEY
Product Desc.
Brand
Color
Size
Manufacturer
Level

Time Dimension

PERIOD KEY
Period Desc
Year
Quarter
Month
Day
Current Flag
Resolution
Sequence

The biggest drawback: dimension tables must carry a *level* indicator for every record and every query must use it. In the example below, without the level constraint, keys for all stores in the NORTH region, including aggregates for region and district will be pulled from the fact table, resulting in error.

Example:

```
Select A.STORE_KEY, A.PERIOD_KEY, A.dollars from
Fact_Table A
          where A.STORE_KEY in (select STORE_KEY
                    from Store_Dimension B
                    where region = “North” and Level = 2)
```

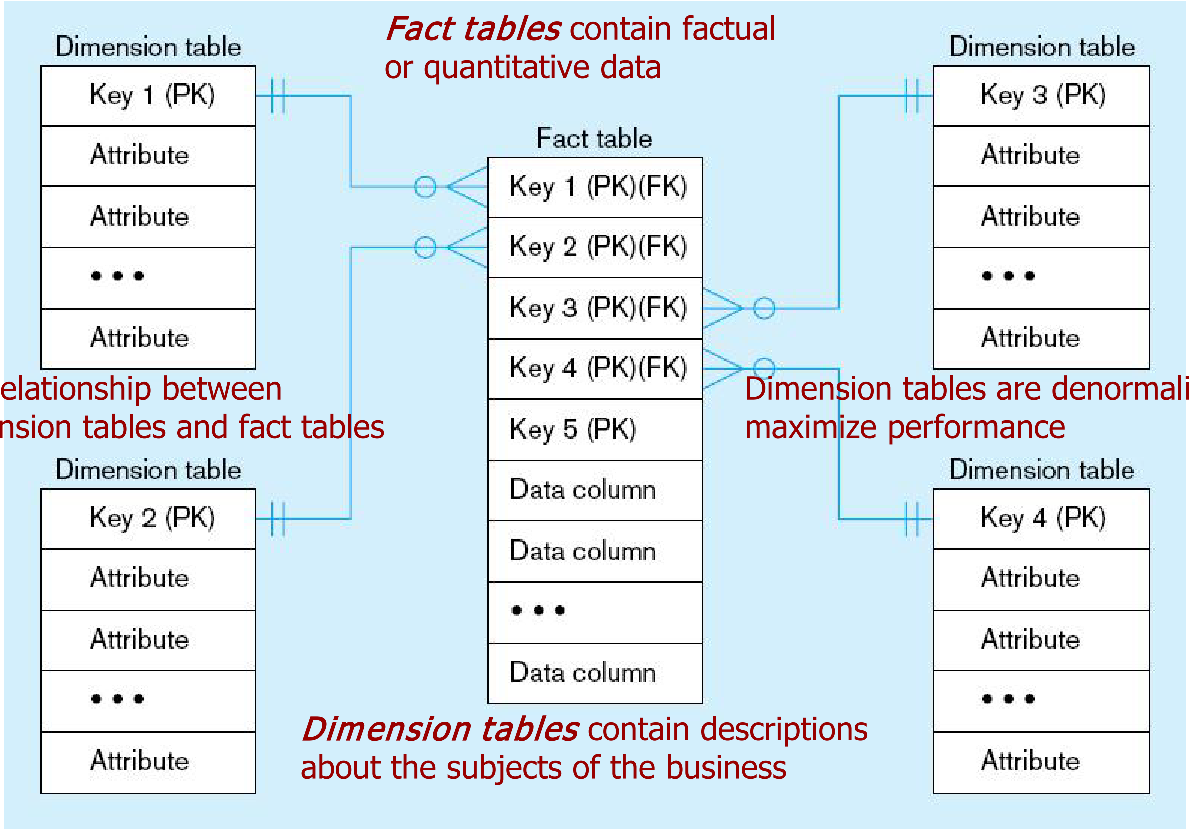
and *etc...*

Level is needed whenever aggregates are stored with detail facts.

Star schema

- The most common modeling paradigm is the star schema, in which the data warehouse contains
  - (1) a large central table (fact table) containing the bulk of the data, with no redundancy, and
  - (2) a set of smaller attendant tables (dimension tables), one for each dimension.
- The schema graph resembles a starburst, with the dimension tables displayed in a radial pattern around the central fact table.

Components of a **star schema**



Fact tables contain factual or quantitative data

Dimension table
- Key 1 (PK)
- Attribute
- Attribute
- • • •
- Attribute

Fact table
- Key 1 (PK)(FK)
- Key 2 (PK)(FK)
- Key 3 (PK)(FK)
- Key 4 (PK)(FK)
- Key 5 (PK)
- Data column
- Data column
- • • •
- Data column

Dimension table
- Key 3 (PK)
- Attribute
- Attribute
- • • •
- Attribute

1:N relationship between dimension tables and fact tables

Dimension tables are denormalized to maximize performance

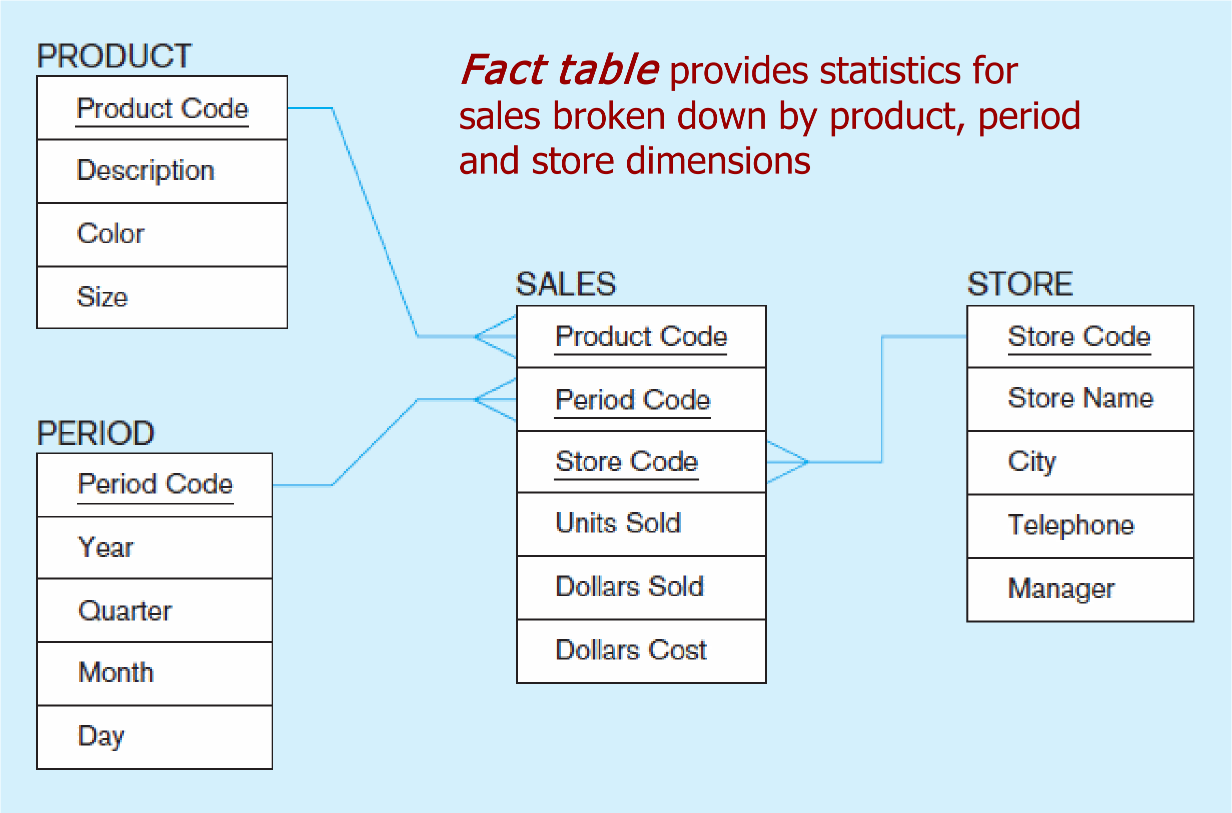
Dimension table
- Key 2 (PK)
- Attribute
- Attribute
- • • •
- Attribute

Dimension table
- Key 4 (PK)
- Attribute
- Attribute
- • • •
- Attribute

Dimension tables contain descriptions about the subjects of the business

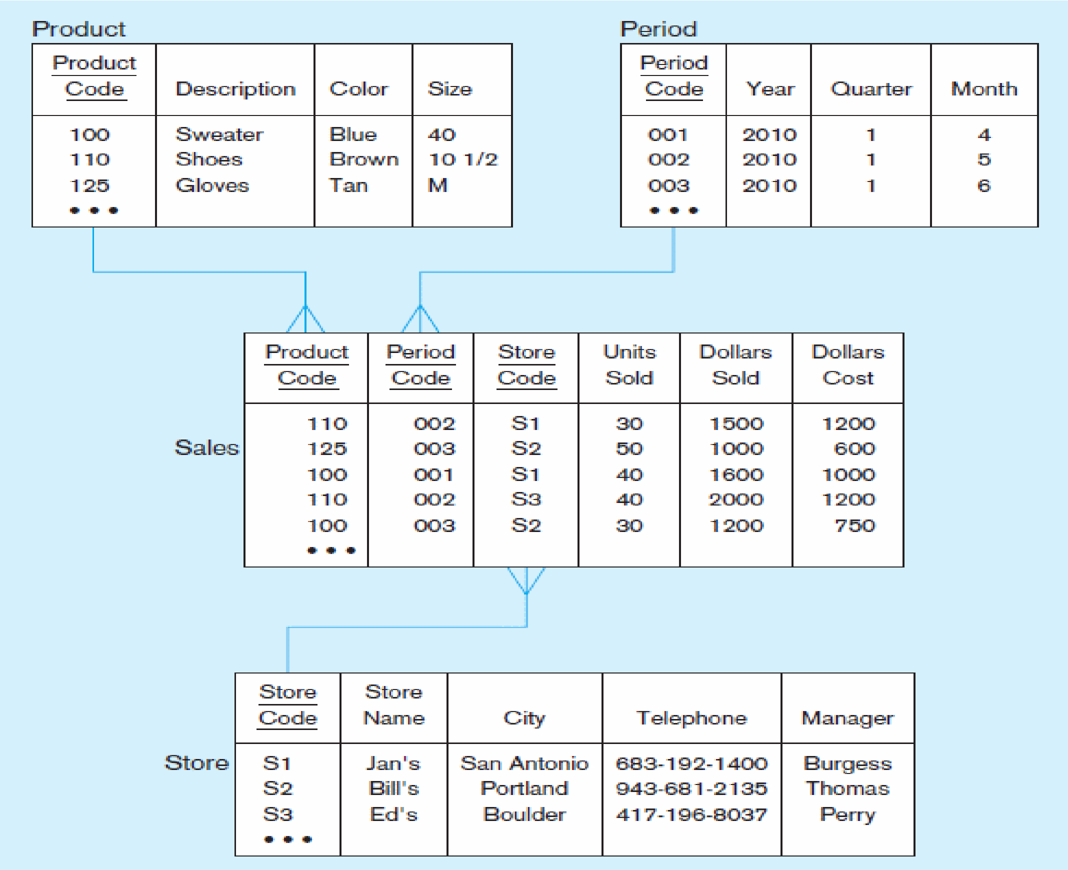
Excellent for ad-hoc queries, but bad for online transaction processing

Star schema example



Fact table provides statistics for sales broken down by product, period and store dimensions

Star schema with sample data



Product

Product Code	Description	Color	Size
100	Sweater	Blue	40
110	Shoes	Brown	10 1/2
125	Gloves	Tan	M
• • •			

Period

Period Code	Year	Quarter	Month
001	2010	1	4
002	2010	1	5
003	2010	1	6
• • •			

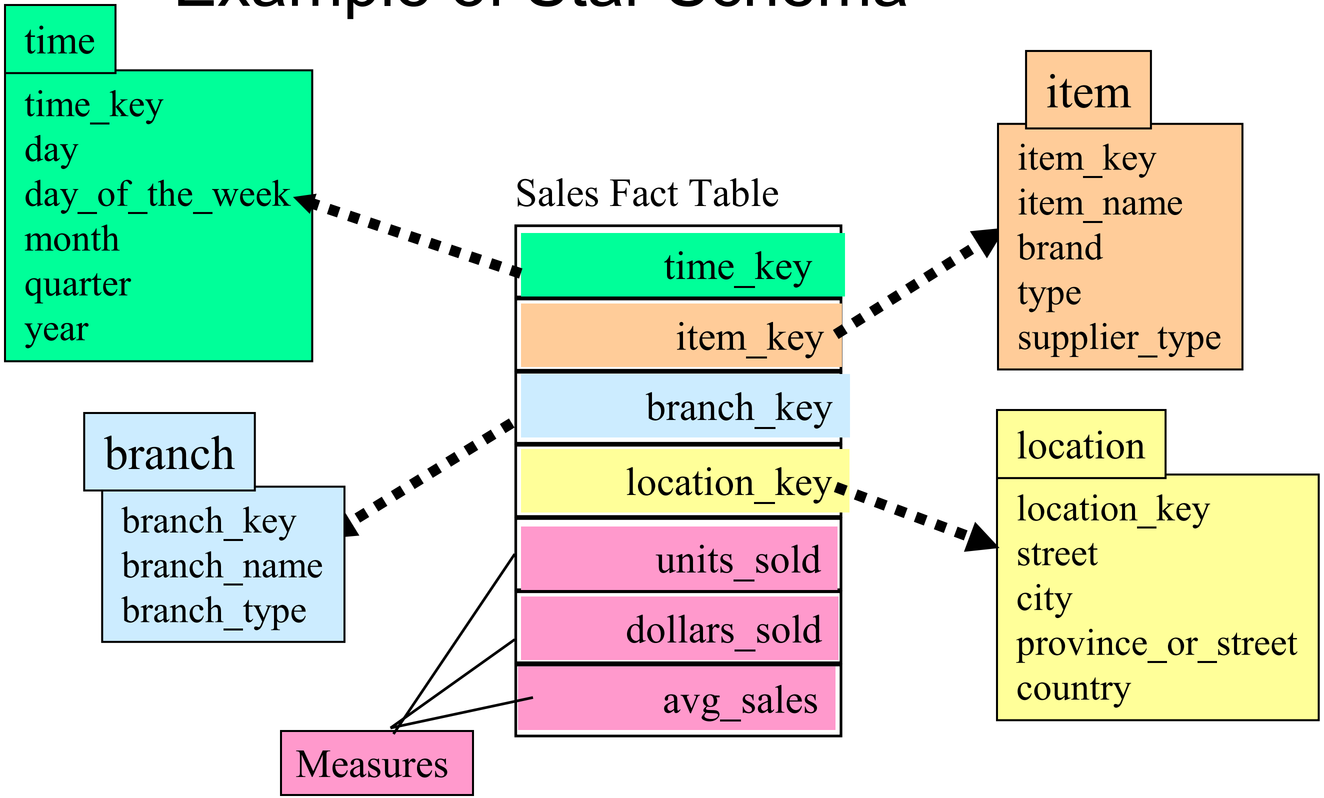
Sales

Product Code	Period Code	Store Code	Units Sold	Dollars Sold	Dollars Cost
110	002	S1	30	1500	1200
125	003	S2	50	1000	600
100	001	S1	40	1600	1000
110	002	S3	40	2000	1200
100	003	S2	30	1200	750
• • •					

Store

Store Code	Store Name	City	Telephone	Manager
S1	Jan's	San Antonio	683-192-1400	Burgess
S2	Bill's	Portland	943-681-2135	Thomas
S3	Ed's	Boulder	417-196-8037	Perry
• • •				

Example of Star Schema



time

time_key
day
day_of_the_week
month
quarter
year

Sales Fact Table

time_key
item_key
branch_key
location_key
units_sold
dollars_sold
avg_sales

item

item_key
item_name
brand
type
supplier_type

branch

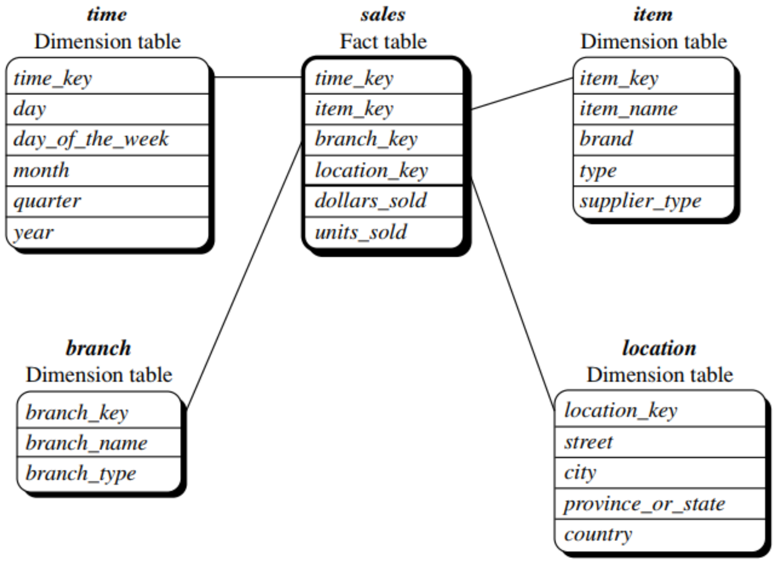
branch_key
branch_name
branch_type

location

location_key
street
city
province_or_street
country

Measures

Star schema



time
Dimension table

- *time_key*
- *day*
- *day_of_the_week*
- *month*
- *quarter*
- *year*

sales
Fact table

- *time_key*
- *item_key*
- *branch_key*
- *location_key*
- *dollars_sold*
- *units_sold*

item
Dimension table

- *item_key*
- *item_name*
- *brand*
- *type*
- *supplier_type*

branch
Dimension table

- *branch_key*
- *branch_name*
- *branch_type*

location
Dimension table

- *location_key*
- *street*
- *city*
- *province_or_state*
- *country*

Figure 4.6 Star schema of *sales* data warehouse.

Snowflake schema

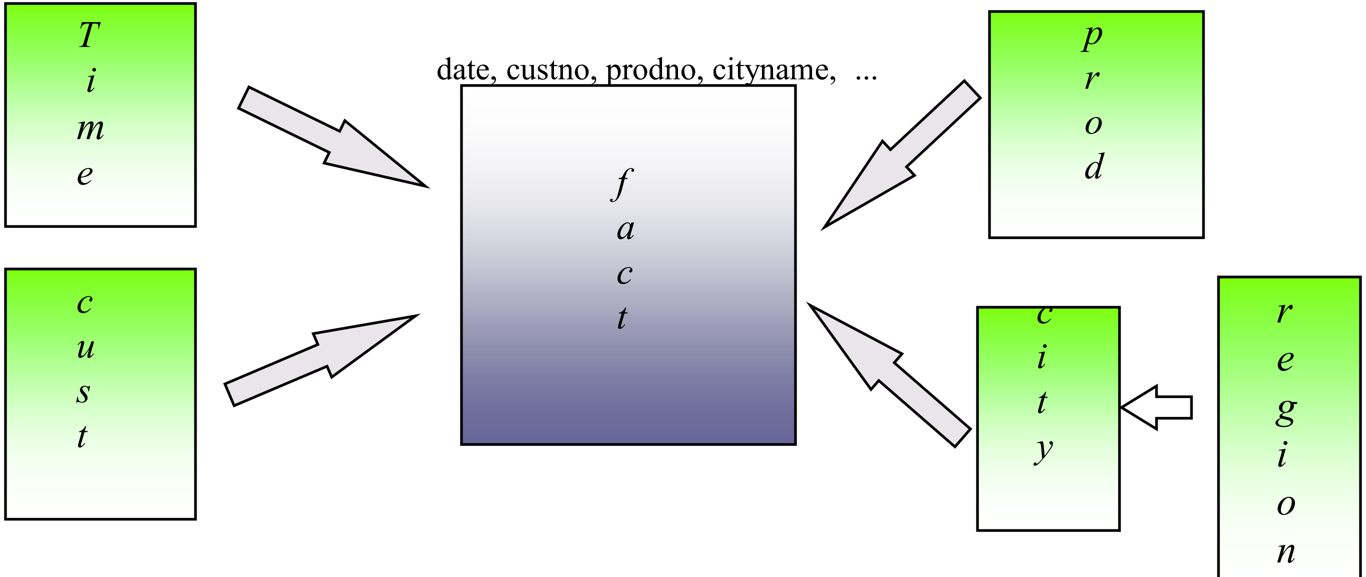
- The snowflake schema is a variant of the star schema model, where some dimension tables are normalized, thereby further splitting the data into additional tables. The resulting schema graph forms a shape similar to a snowflake.

Snowflakes schema

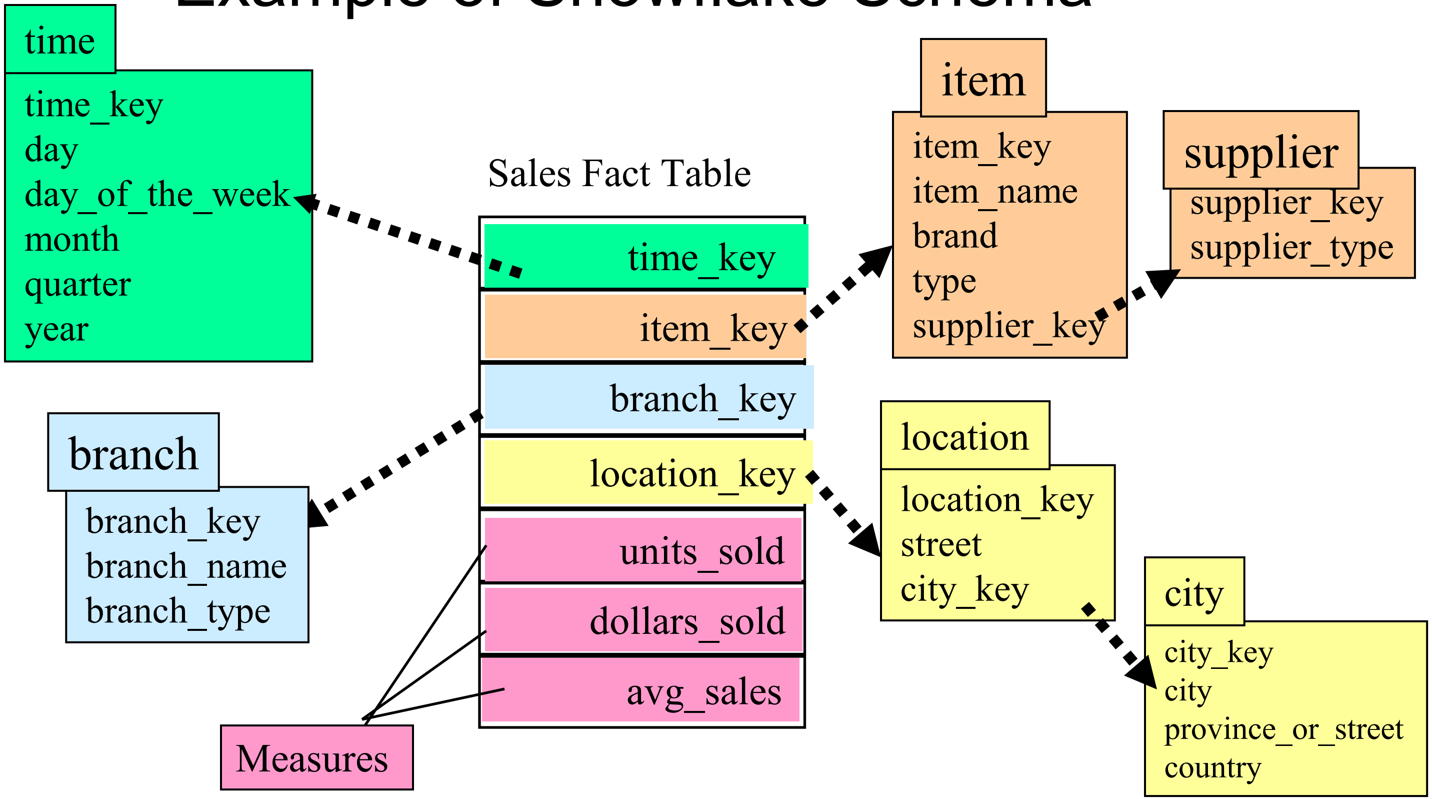
- **Snowflakes schema**
    - A refinement of star schema where some dimensional hierarchy is normalized into a set of smaller dimension tables, forming a shape similar to snowflake
    - An extension of star schema where the diagram resembles a snowflake in shape
- Snowflake schema: Avoids redundancy of star schemas by normalizing dimension tables
- Normalized tables optimize storage space, but decrease performance
- Starflake schema: Combination of the star and snowflake schemas, some dimensions normalized, other not

Snowflake schema

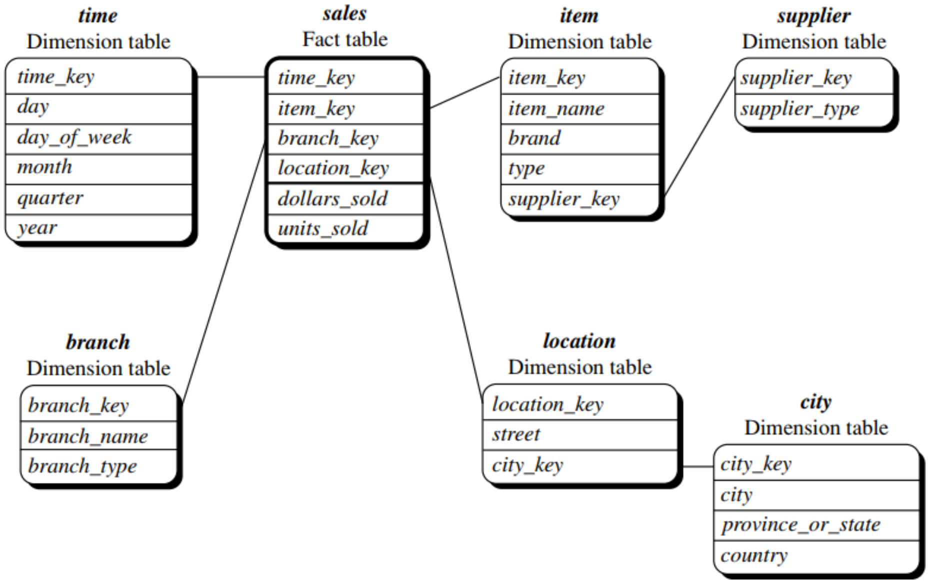
- Represent dimensional hierarchy directly by normalizing tables.
- Easy to maintain and saves storage



Example of Snowflake Schema



Snowflake schema



time
Dimension table
time_key
day
day_of_week
month
quarter
year

sales
Fact table
time_key
item_key
branch_key
location_key
dollars_sold
units_sold

item
Dimension table
item_key
item_name
brand
type
supplier_key

supplier
Dimension table
supplier_key
supplier_type

branch
Dimension table
branch_key
branch_name
branch_type

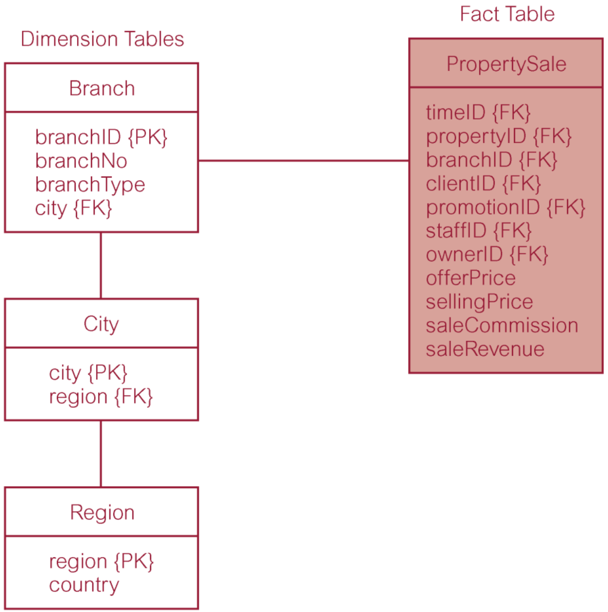
location
Dimension table
location_key
street
city_key

city
Dimension table
city_key
city
province_or_state
country

Figure 4.7 Snowflake schema of a *sales* data warehouse.

The snowflake schema

- A variation of the star schema where the dimension tables are normalized.



Fact Table

Dimension Tables

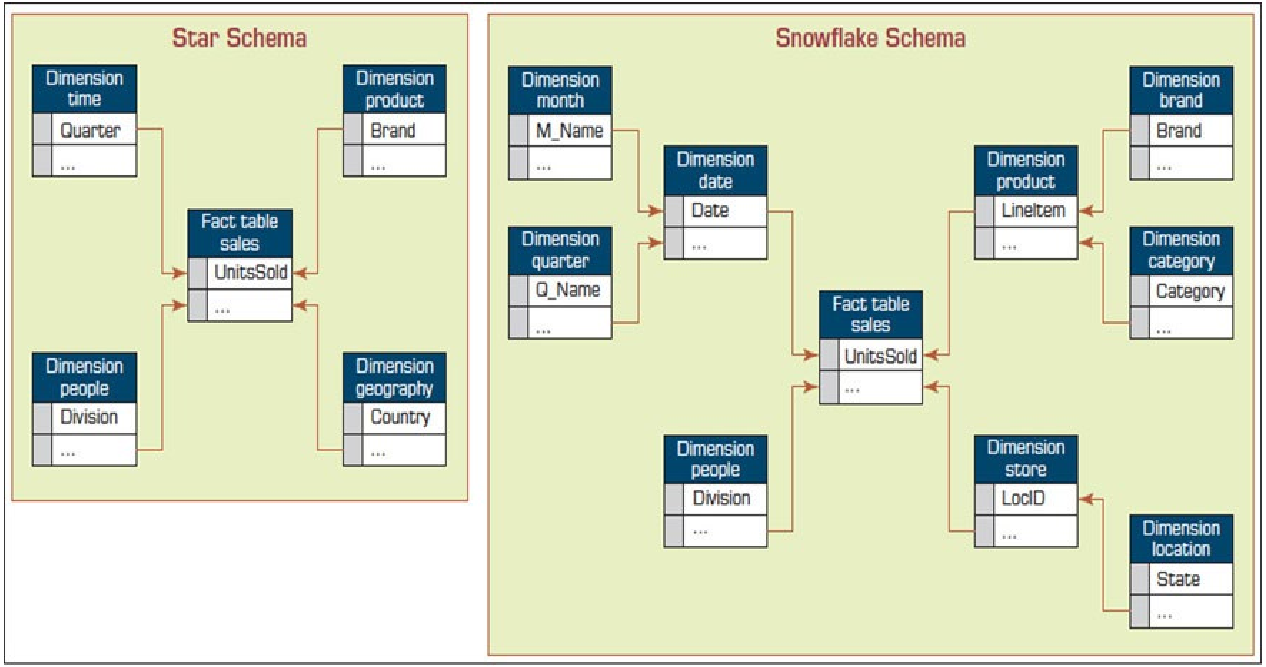
Branch
branchID {PK}
branchNo
branchType
city {FK}

City
city {PK}
region {FK}

Region
region {PK}
country

PropertySale
timeID {FK}
propertyID {FK}
branchID {FK}
clientID {FK}
promotionID {FK}
staffID {FK}
ownerID {FK}
offerPrice
sellingPrice
saleCommission
saleRevenue

Star Schema versus Snowflake Schema



The “Level” Problem

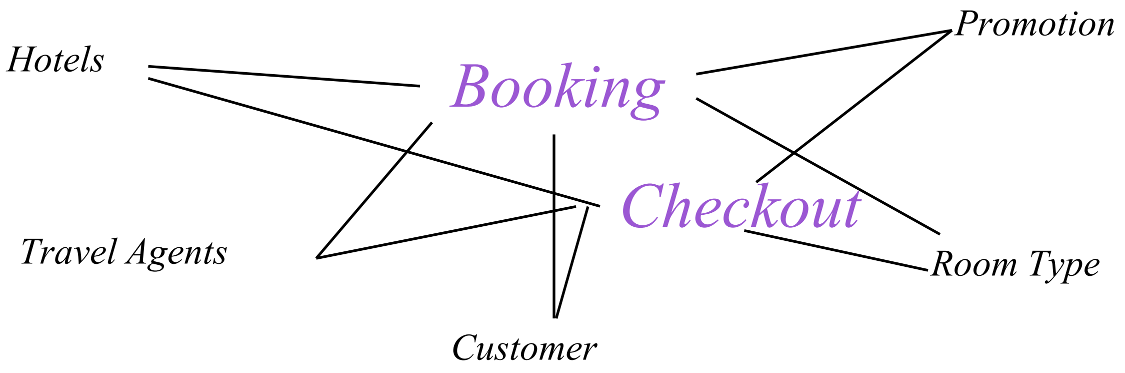
- Level is a problem because because it causes potential for error. If the query builder, human or program, forgets about it, perfectly reasonable looking WRONG answers can occur.
- One alternative: the FACT CONSTELLATION model...

Fact constellations schema

- Fact constellations schema: Multiple fact tables that share dimension tables
    - Multiple fact tables share dimension tables, viewed as a collection of stars, therefore called galaxy schema or fact constellation
    - Sophisticated applications may require multiple fact tables to share dimension tables. This kind of schema can be viewed as a collection of stars, and hence is called a galaxy schema or a fact constellation.

Fact Constellation

- Fact Constellation
  - Multiple fact tables that share many dimension tables
  - Booking and Checkout may share many dimension tables in the hotel industry



Fact constellation Schema

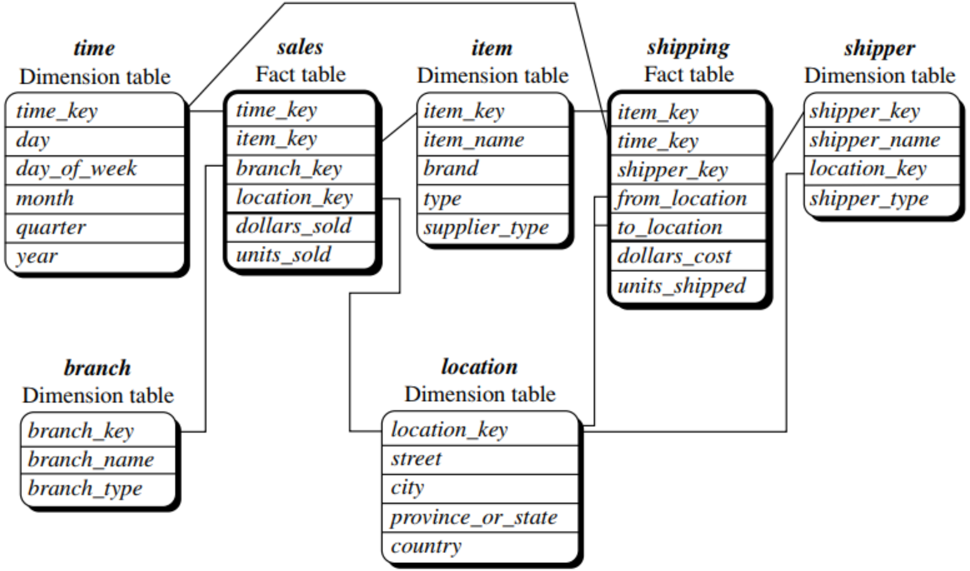
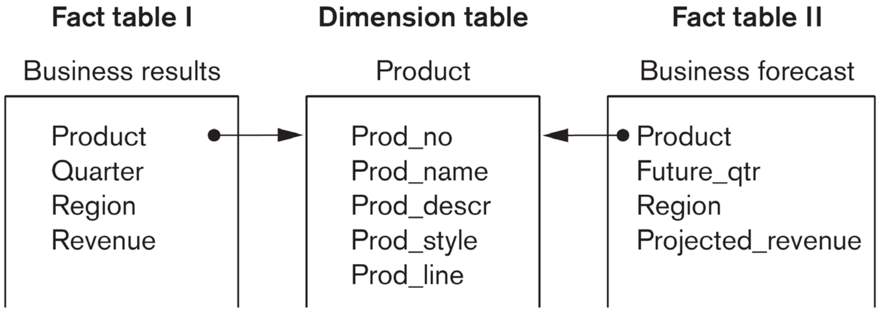


Figure 4.8 Fact constellation schema of a sales and shipping data warehouse.

Fact constellation

- A set of fact tables that share some dimension tables



Fact table I

Business results

Product
Quarter
Region
Revenue

Dimension table

Product

Prod_no
Prod_name
Prod_descr
Prod_style
Prod_line

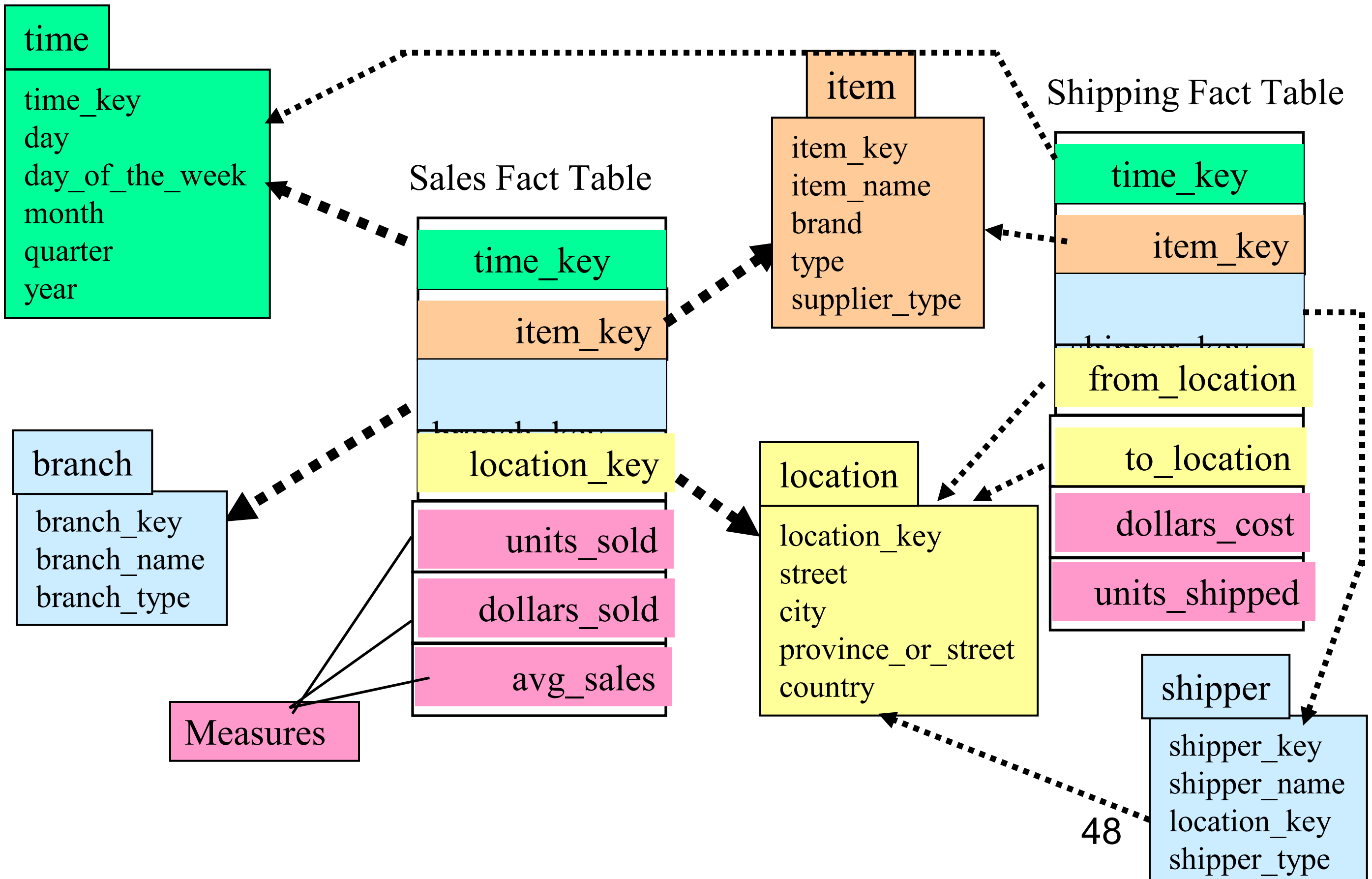
Fact table II

Business forecast

Product
Future_qtr
Region
Projected_revenue

Figure 29.9
A fact constellation.

Example of Fact Constellation



time

time_key
day
day_of_the_week
month
quarter
year

Sales Fact Table

time_key
item_key
branch_key
location_key
units_sold
dollars_sold
avg_sales

item

item_key
item_name
brand
type
supplier_type

Shipping Fact Table

time_key
item_key
shipper_key
from_location
to_location
dollars_cost
units_shipped

branch

branch_key
branch_name
branch_type

location

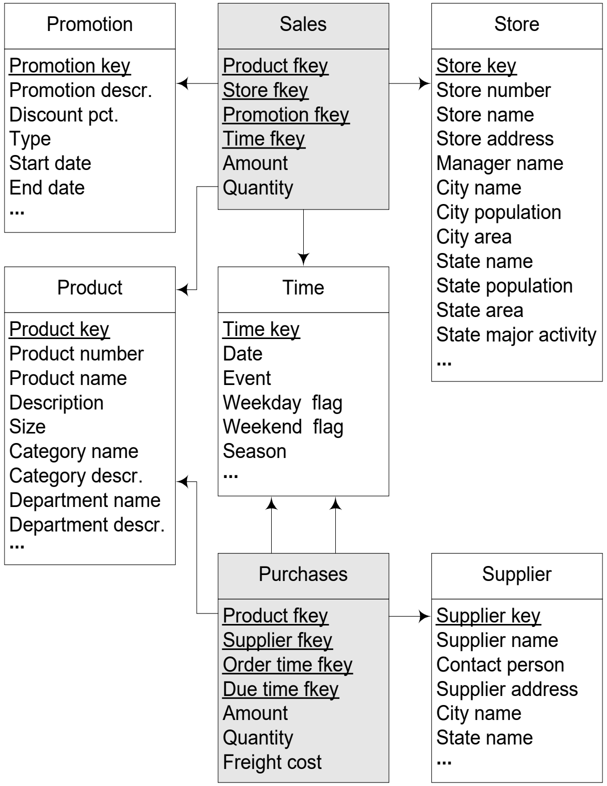
location_key
street
city
province_or_street
country

Measures

shipper

shipper_key
shipper_name
location_key
shipper_type

Logical DW Design: Constellation Schemas



Promotion

Promotion key
Promotion descr.
Discount pct.
Type
Start date
End date
...

Sales

Product fkey
Store fkey
Promotion fkey
Time fkey
Amount
Quantity

Store

Store key
Store number
Store name
Store address
Manager name
City name
City population
City area
State name
State population
State area
State major activity
...

Product

Product key
Product number
Product name
Description
Size
Category name
Category descr.
Department name
Department descr.
...

Time

Time key
Date
Event
Weekday flag
Weekend flag
Season
...

Purchases

Product fkey
Supplier fkey
Order time fkey
Due time fkey
Amount
Quantity
Freight cost

Supplier

Supplier key
Supplier name
Contact person
Supplier address
City name
State name
...