

Why do we need a
data warehouse?



Two purposes



Analytical
decision making



Operational
data keeping

- Receive orders
- React to complaints
- Fill up stock

⇒ Turn the wheel

Why do we need a
data warehouse?

Two purposes



Analytical
decision making



Operational
data keeping

- Receive orders
- React to complaints
- Fill up stock

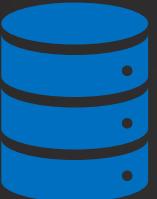
→ Turn the wheel

Why do we need a
data warehouse?

Two purposes

- What's the best category?
- How many sales compared to last month?
- What can be improved?

⇒ Evaluate performance
⇒ Decision-making



Analytical
decision making

OLAP = Online Analytical Processing



Operational
data keeping

- Receive orders
- React to complaints
- Fill up stock

⇒ Turn the wheel

OLTP = Online Transactional Processing

Why do we need a
data warehouse?

Two purposes

- o What's the best category?
- o How many sales compared to last month?
- o What can be improved?

⇒ Evaluate performance
⇒ Decision-making



Analytical
decision making



Operational
data keeping

- o Receive orders
- o React to complaints
- o Fill up stock

⇒ Turn the wheel

Why do we need a
data warehouse?

Two purposes

"Yes, we have a lot of data but we don't use it"

"Our data is very complicated and difficult to analyze"

"It's spread all over the different systems and difficult to access"

"I just want to see what is relevant!"

"We need to access data quick and easily"

"We want to make fact-based decisions!"

Two purposes

- o What's the best category?
- o How many sales compared to last month?
- o What can be improved?

⇒ Evaluate performance
⇒ Decision-making



Analytical
decision making



Operational
data keeping

- o Receive orders
- o React to complaints
- o Fill up stock

⇒ Turn the wheel

Why do we need a
data warehouse?

Two requirements

- Thousands of records at a time
- Fast query performance
- Historical context



Analytical
decision making



Operational
data keeping

- One record at a time
- Data input
- No long history

Why do we need a
data warehouse?

Two requirements

- Thousands of records at a time
- Fast query performance
- Historical context
- Usability



Analytical
decision making



Operational
data keeping

- One record at a time
- Data input
- No long history

DWH is there to address those analytical data needs!

DWH is Data Warehouse!

Why do we need a
data warehouse?

Two requirements

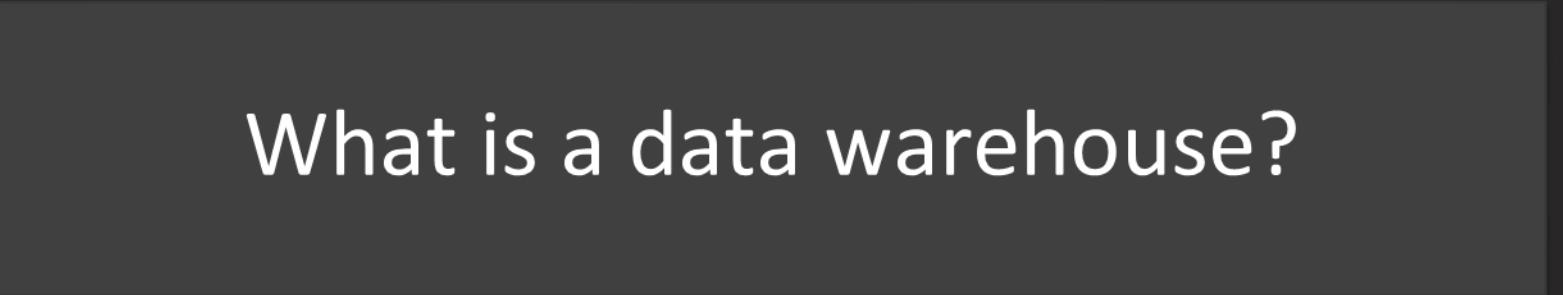
- o Thousands of records at a time
- o Fast
- o Historical
- o Usage

~ One record at a time

Used for reporting and data analysis

DWH is there to address those analytical data needs!

Why do we need a data warehouse?



What is a data warehouse?

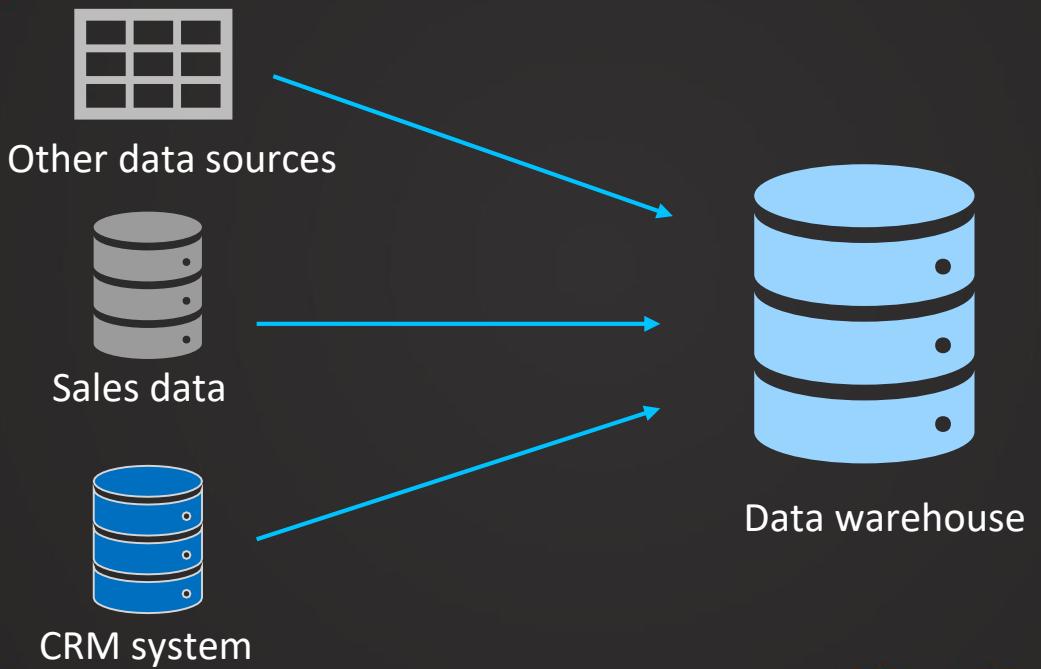
What is a data warehouse?

Data warehouse:

A database used and optimized for analytical purposes.

- ✓ **User friendly**
- ✓ **Fast query performance**
- ✓ **Enabling data analysis**

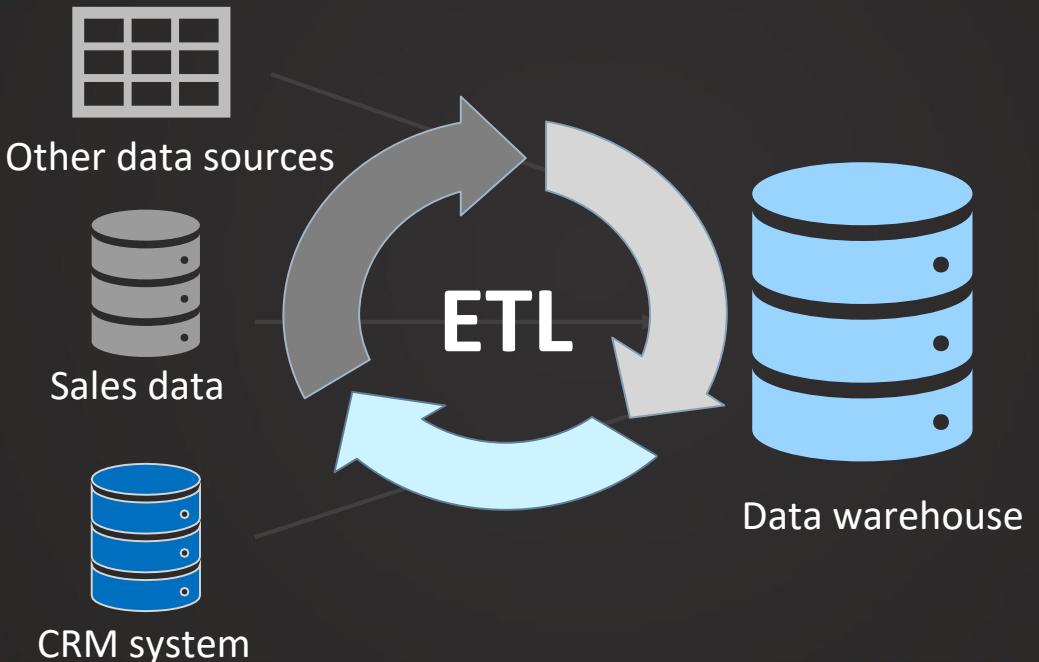
Understanding a data warehouse



Understanding a data warehouse

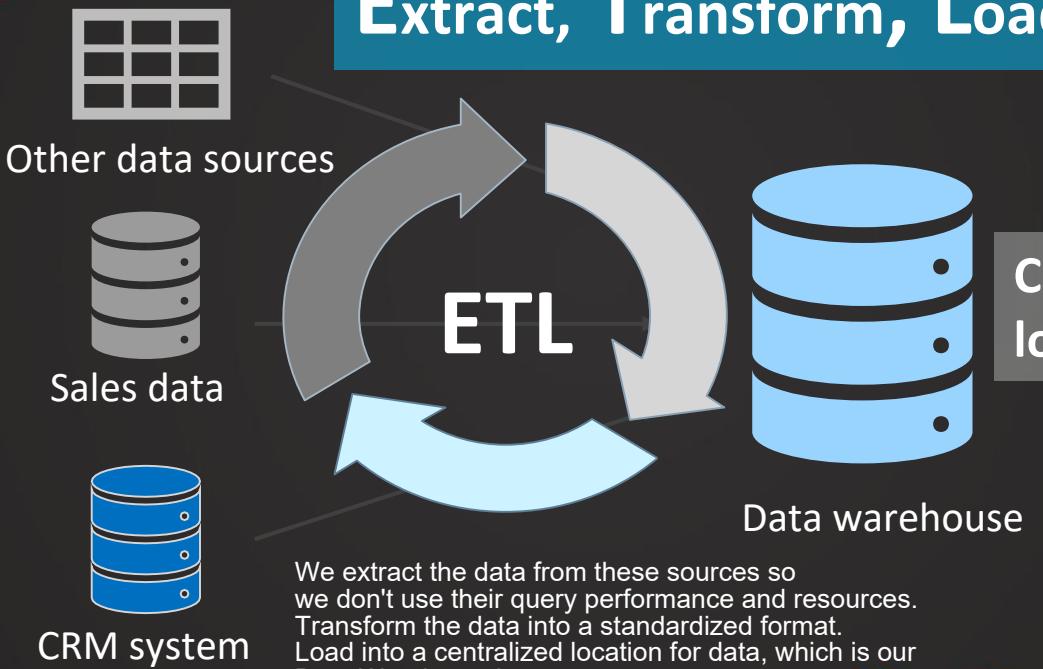
ETL Process is the process of data warehousing.

80 - 90% of our time goes into this when dealing with data warehousing.



Understanding a data warehouse

Extract, Transform, Load



Goals of a data warehouse

- ✓ **Centralized and consistent location for data**
- ✓ **Data must be accessible fast (query performance)**
- ✓ **User-friendly (easy to understand)**
- ✓ **Must load data consistently and repeatedly (ETL)**
- ✓ **Reporting and data visualization built on top**

Understanding a data warehouse



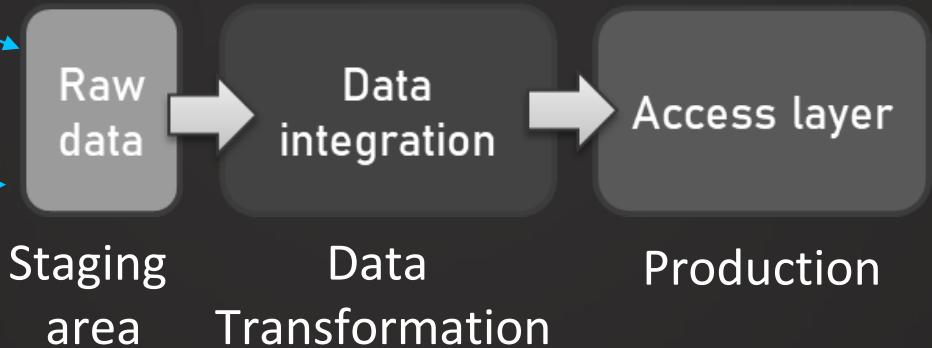
Other data sources



Sales data



CRM system





We create a data warehouse for
Business Intelligence...

What is Business Intelligence?

What is Business Intelligence?

Data analysis

- Data gathering
- Data storing
- Reporting
- Data visualization
- Data mining
- Predictive analytics

Strategies

Technologies

Infrastructures



Raw data

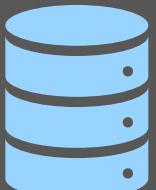


Transform



One of the most important components of Biz Intelligence

Data warehouse



Meaningful insights
Better decisions



Data Lake or Data Warehouse?

Data Lake or Data Warehouse?

**Data lake & data warehouse are
BOTH used as
centralized data storage**

Data Lake or Data Warehouse?

	Data Lake	Data Warehouse
Data	Raw	Processed
Technologies	Big data	Database
Structure	Unstructured	Structured
Usage	Not defined yet	Specific & ready to be used
Users	Data Scientists	Business users & IT

Data Lake or Data Warehouse?

Data Lake

Data Warehouse

Technologies	Raw Big data	Processed Database
Structure	Unstructured	Structured
Usage	Undefined yet	Specific & ready to be used
Users	Data Scientists	Business users & IT

Both!

Demos & Hands-on

- ✓ **Demonstrations & Hands-on assignments**
- ✓ **Assignment & Installations are optionally only**
- ✓ **Install ETL-Tool (Pentaho)**
- ✓ **Database Management System (PostgreSQL)**

The layers of a Data Warehouse

Data Warehouse Layers



Other data sources

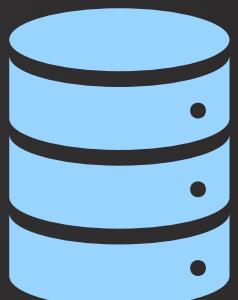
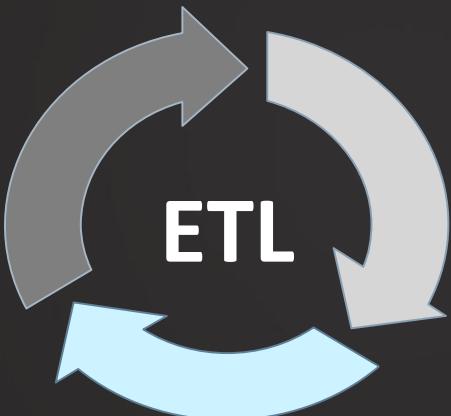


Sales data



CRM system

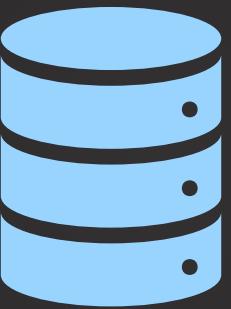
Extract, Transform, Load



Data warehouse

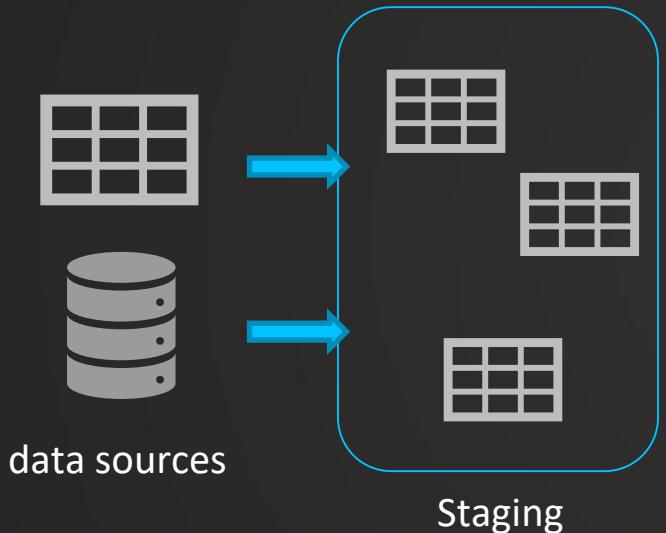
Centralized
location for data

Data Warehouse Layers



Data warehouse

Data Warehouse Layers



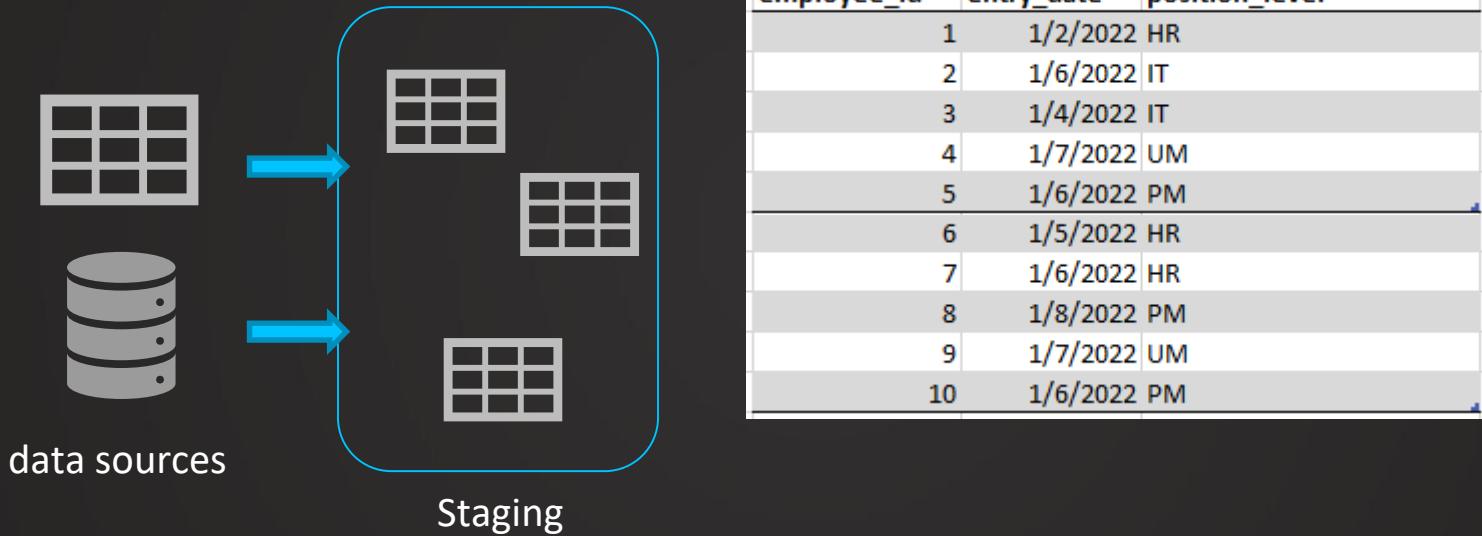
Department 1

employee_id	entry_date	position_level
1	1/2/2022	HR
2	1/6/2022	IT
3	1/4/2022	IT
4	1/7/2022	UM
5	1/6/2022	PM

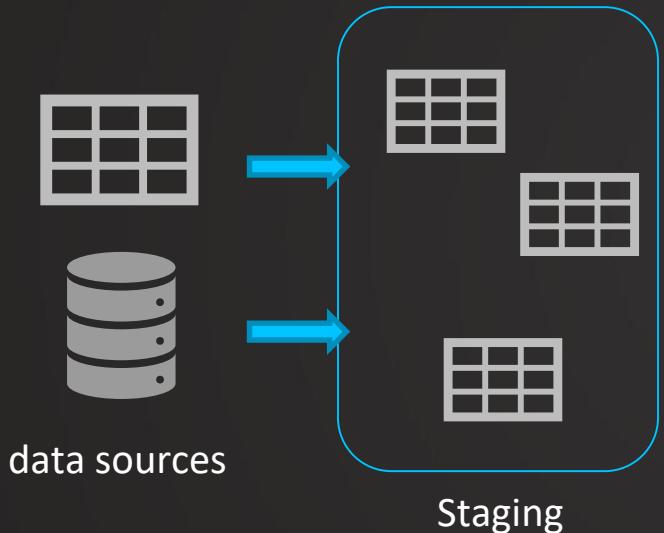
Department 2

employee_id	entry_date	position_level
6	1/5/2022	HR
7	1/6/2022	HR
8	1/8/2022	PM
9	1/7/2022	UM
10	1/6/2022	PM

Data Warehouse Layers



Data Warehouse Layers



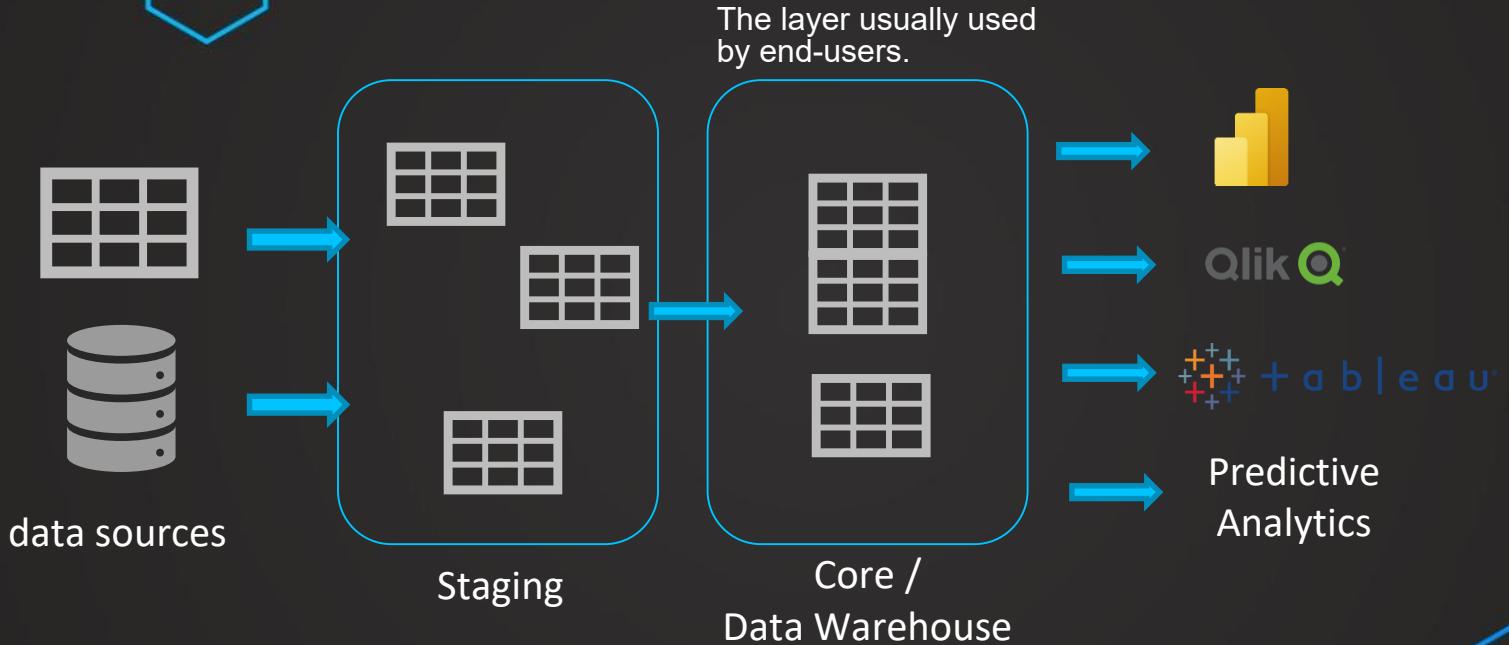
Department 1

employee_id	entry_date	position_level
1	1/2/2022	HR
2	1/6/2022	IT
3	1/4/2022	IT
4	1/7/2022	UM
5	1/6/2022	PM

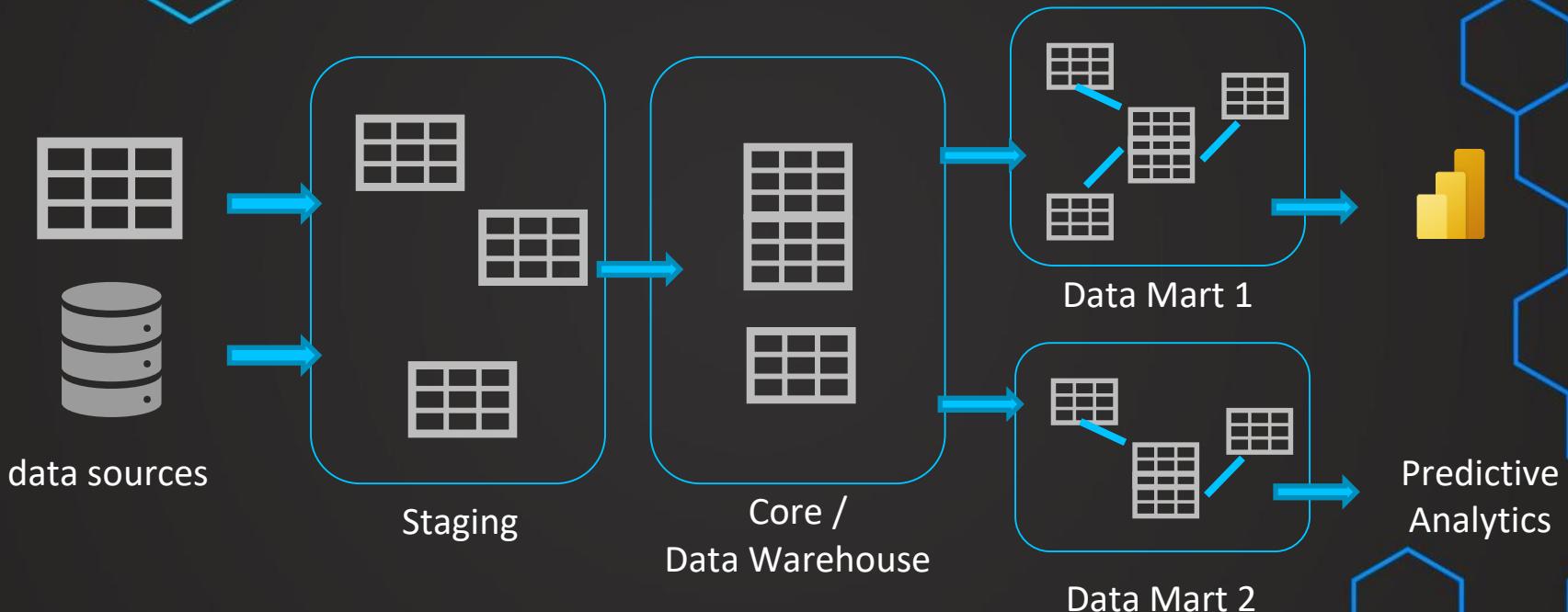
Department 2

employee_id	entry_date	position
1	1/2/2022	Human Ressources
2	1/6/2022	Information Technologies
3	1/4/2022	Information Technologies
4	1/7/2022	Upper Management
5	1/6/2022	Project Manager

Data Warehouse Layers



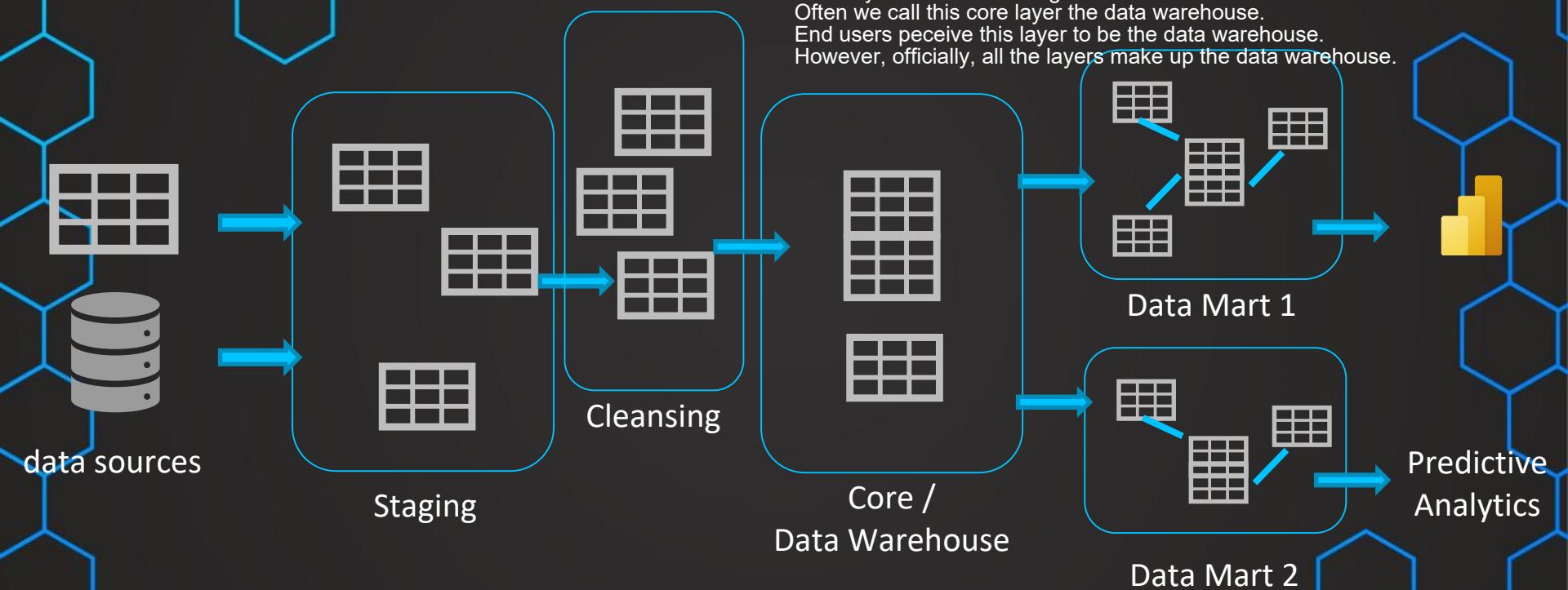
Data Warehouse Layers



Data Mart just takes the relevant tables.
This helps the query performance.
Data Mart is not always necessary.

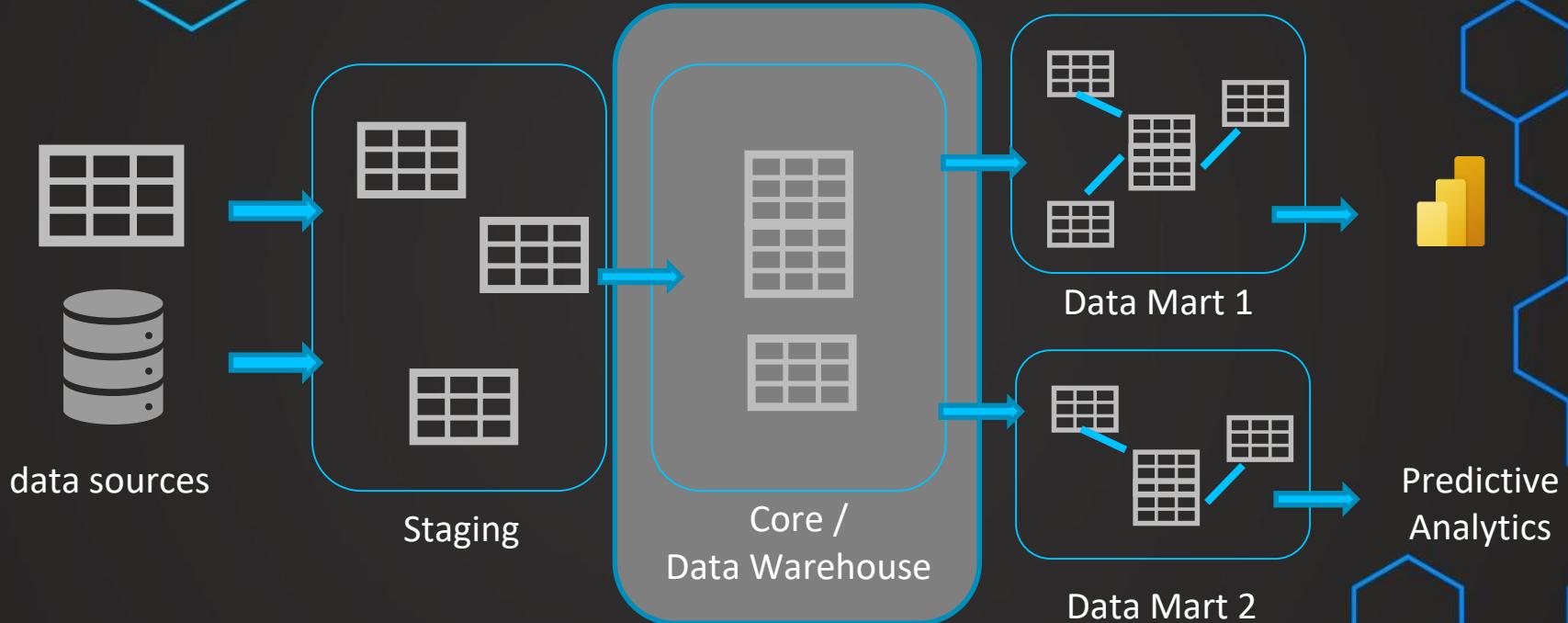
Data Warehouse Layers

Core layer is to be the "single source of truth". Often we call this core layer the data warehouse. End users perceive this layer to be the data warehouse. However, officially, all the layers make up the data warehouse.



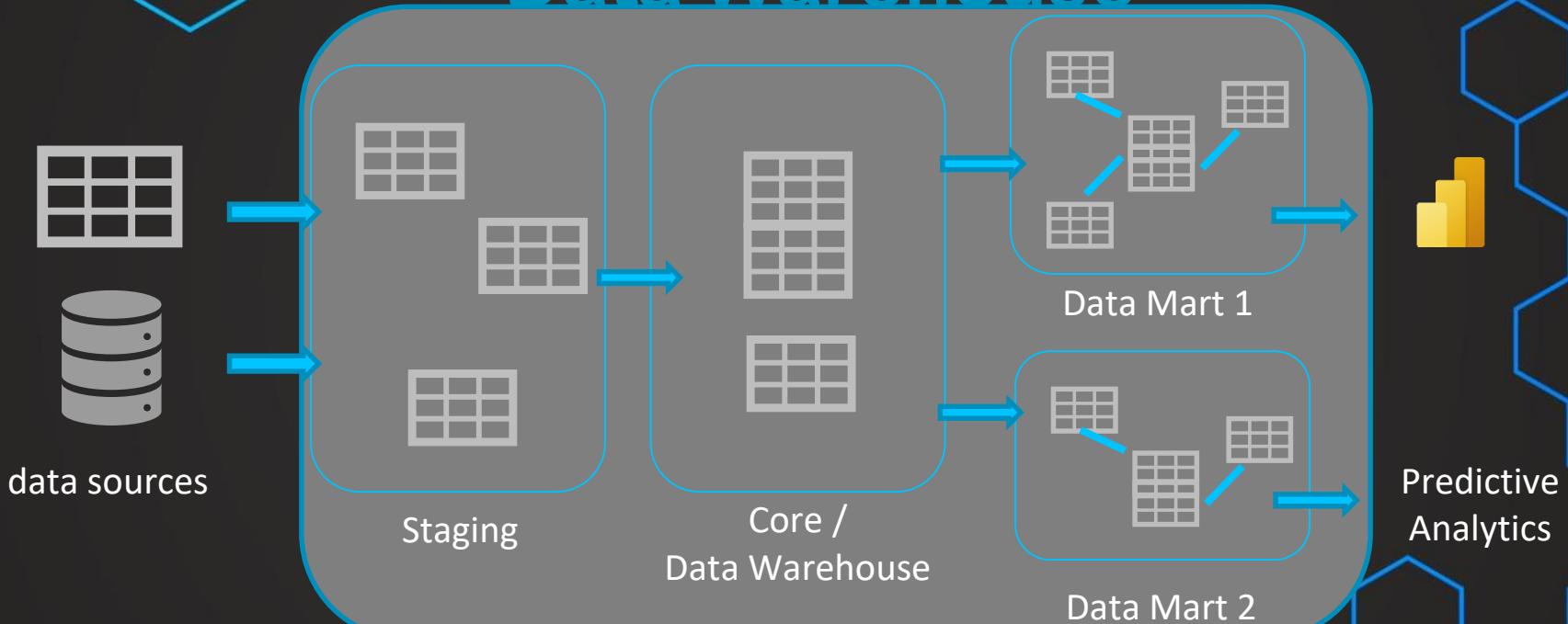
If the data is so messy, we may need a cleansing layer to clean up the data before we push it into the core warehouse.

Data Warehouse Layers



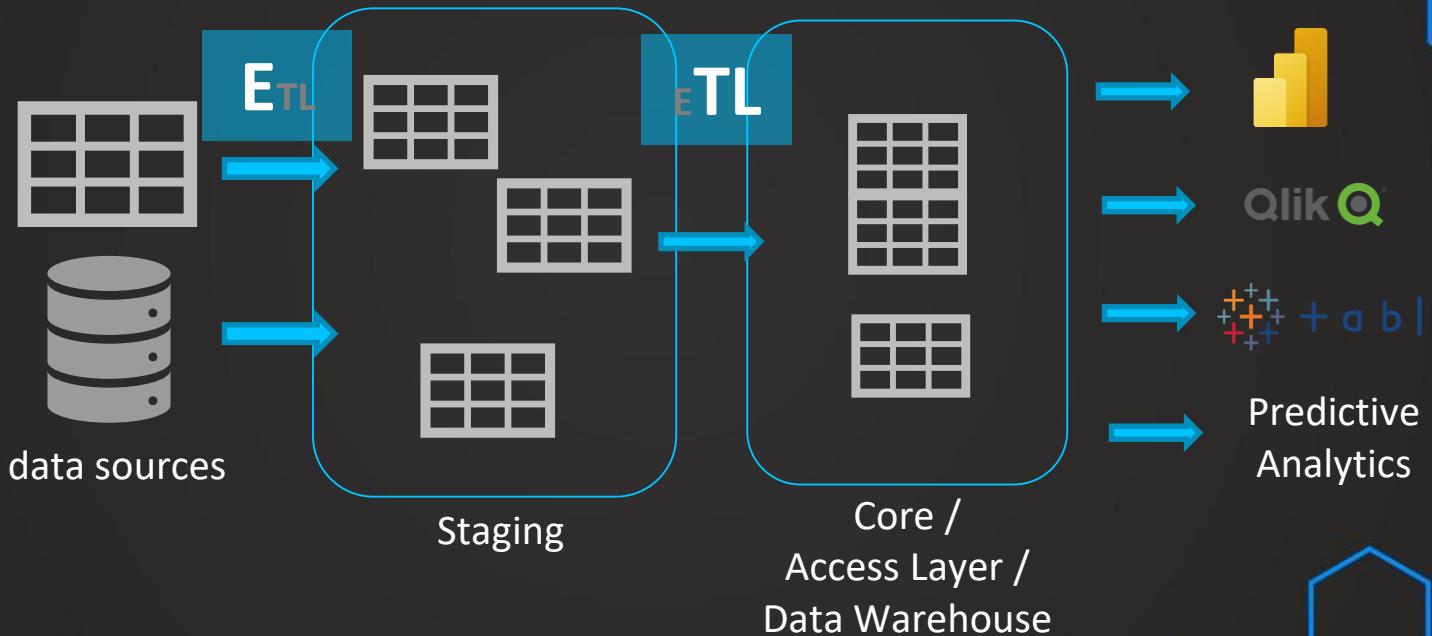
Data Warehouse Layers

Data Warehouse

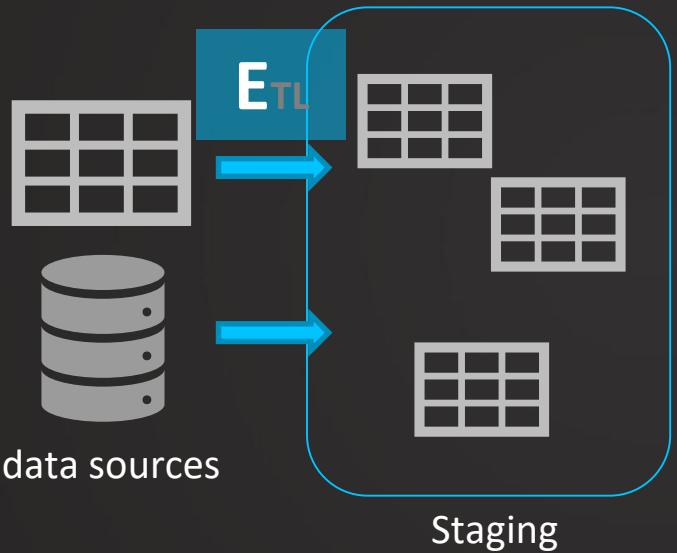


The Staging Area

Data Warehouse Layers

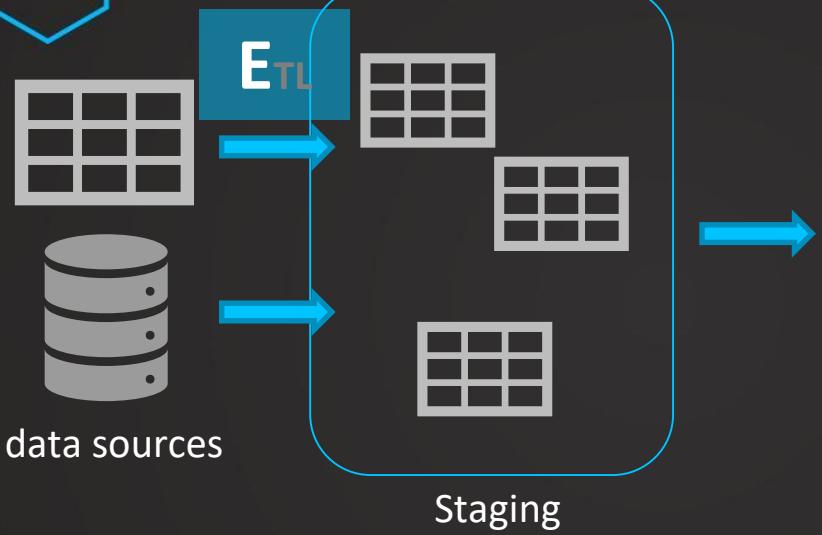


Data Warehouse Layers



- *"Short time on the source systems"*
- *"Quickly extract"*
- Move the data into relational database
- Start transformations from there

Data Warehouse Layers

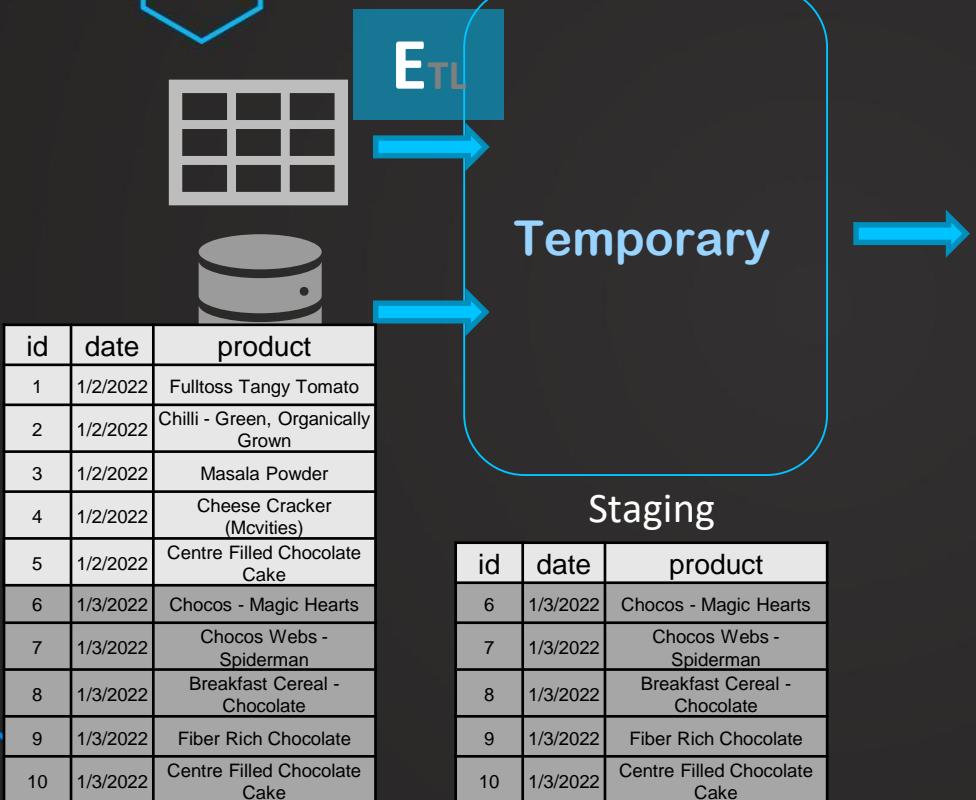


id	date	product
1	1/2/2022	Fulltoss Tangy Tomato
2	1/2/2022	Chilli - Green, Organically Grown
3	1/2/2022	Masala Powder
4	1/2/2022	Cheese Cracker (Mcvitie's)
5	1/2/2022	Centre Filled Chocolate Cake

id	date	product
1	1/2/2022	Fulltoss Tangy Tomato
2	1/2/2022	Chilli - Green, Organically Grown
3	1/2/2022	Masala Powder
4	1/2/2022	Cheese Cracker (Mcvitie's)
5	1/2/2022	Centre Filled Chocolate Cake

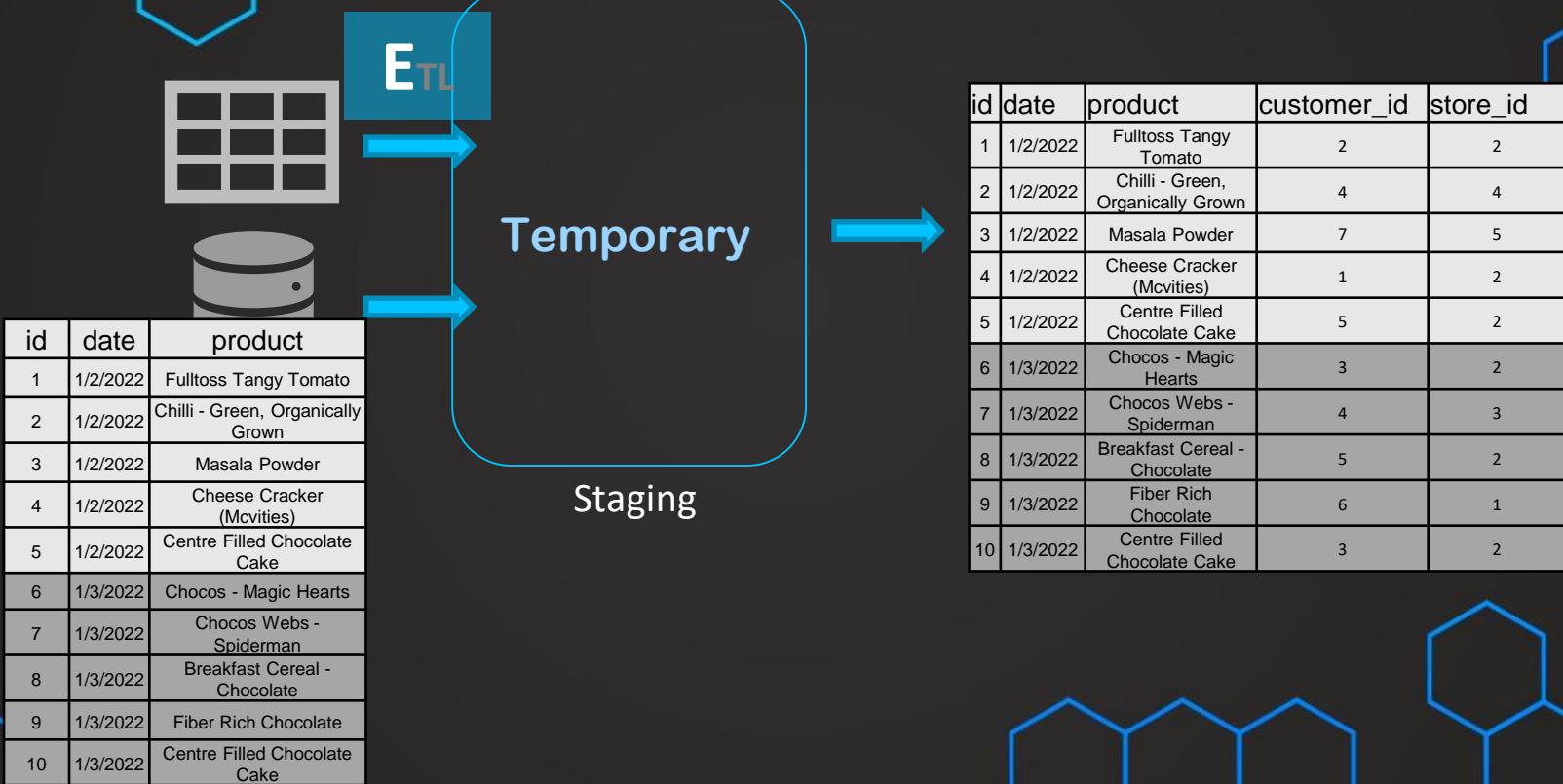
id	date	product	customer_id	store_id
1	1/2/2022	Fulltoss Tangy Tomato	2	2
2	1/2/2022	Chilli - Green, Organically Grown	4	4
3	1/2/2022	Masala Powder	7	5
4	1/2/2022	Cheese Cracker (Mcvitie's)	1	2
5	1/2/2022	Centre Filled Chocolate Cake	5	2

Data Warehouse Layers

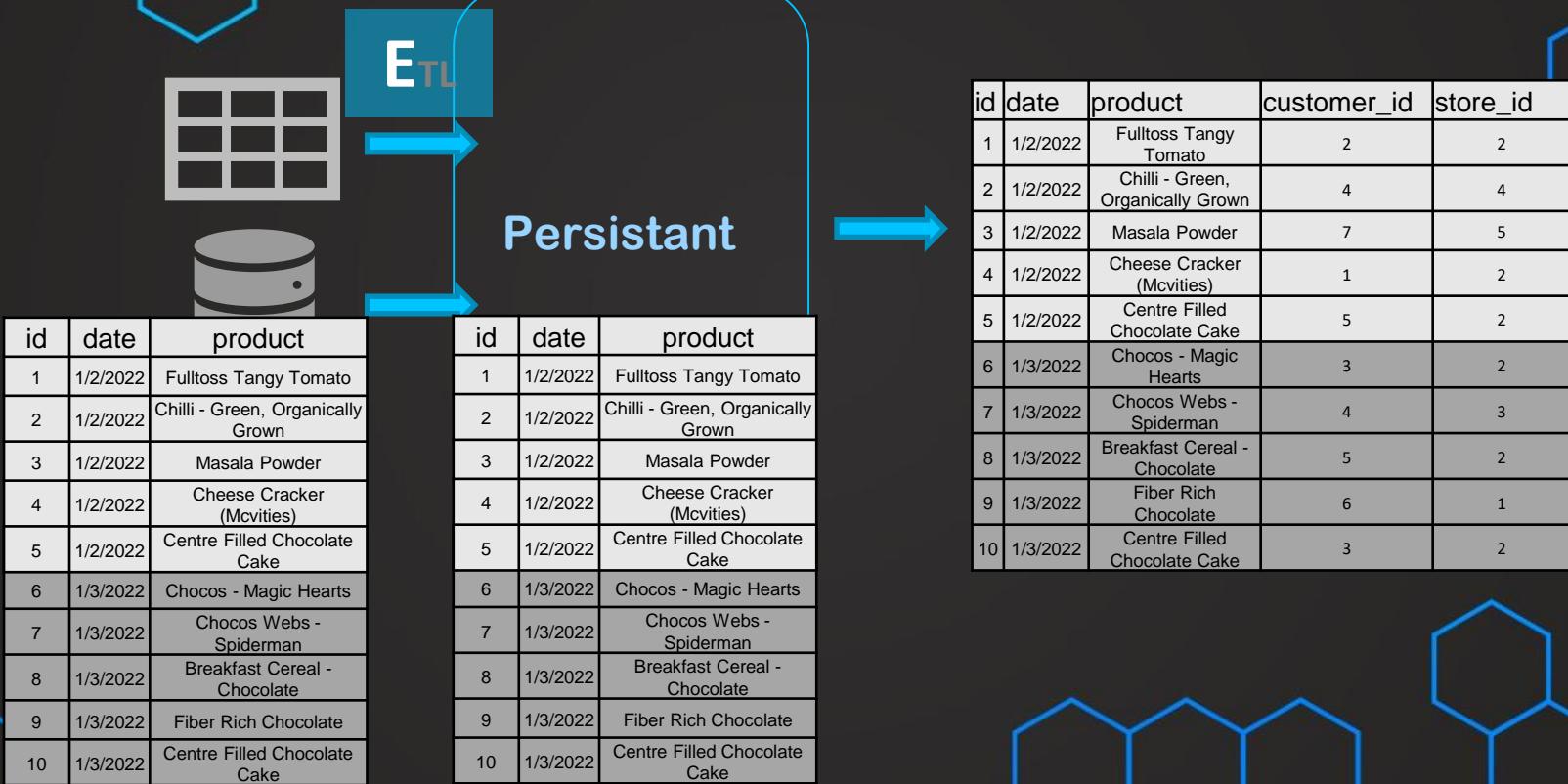


id	date	product	customer_id	store_id
1	1/2/2022	Fulltoss Tangy Tomato	2	2
2	1/2/2022	Chilli - Green, Organically Grown	4	4
3	1/2/2022	Masala Powder	7	5
4	1/2/2022	Cheese Cracker (Mcvities)	1	2
5	1/2/2022	Centre Filled Chocolate Cake	5	2
6	1/3/2022	Chocos - Magic Hearts	3	2
7	1/3/2022	Chocos Webs - Spiderman	4	3
8	1/3/2022	Breakfast Cereal - Chocolate	5	2
9	1/3/2022	Fiber Rich Chocolate	6	1
10	1/3/2022	Centre Filled Chocolate Cake	3	2

Data Warehouse Layers



Data Warehouse Layers

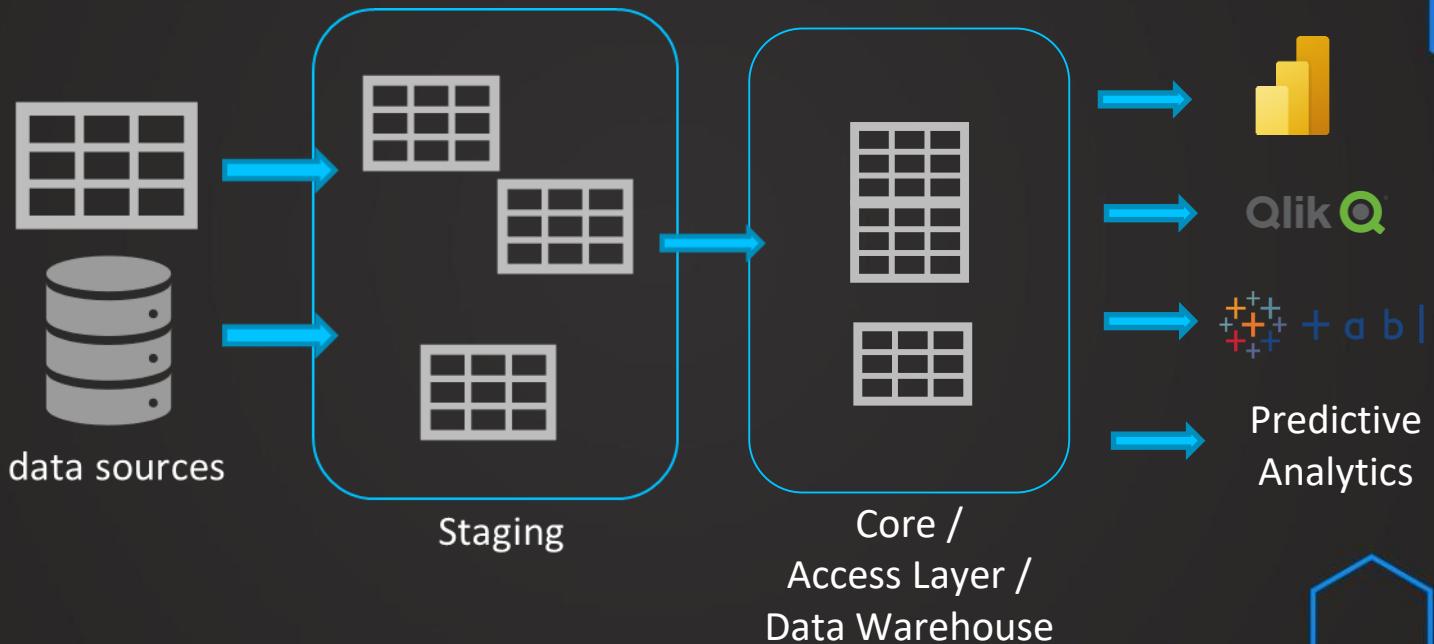


The Staging Layer

- ✓ Staging Layer is the landing zone extracted data
- ✓ Data in tables and on a separate database
- ✓ As little "touching" as possible
- ✓ We don't charge the source systems
- ✓ Temporary or Persistent Staging Layers

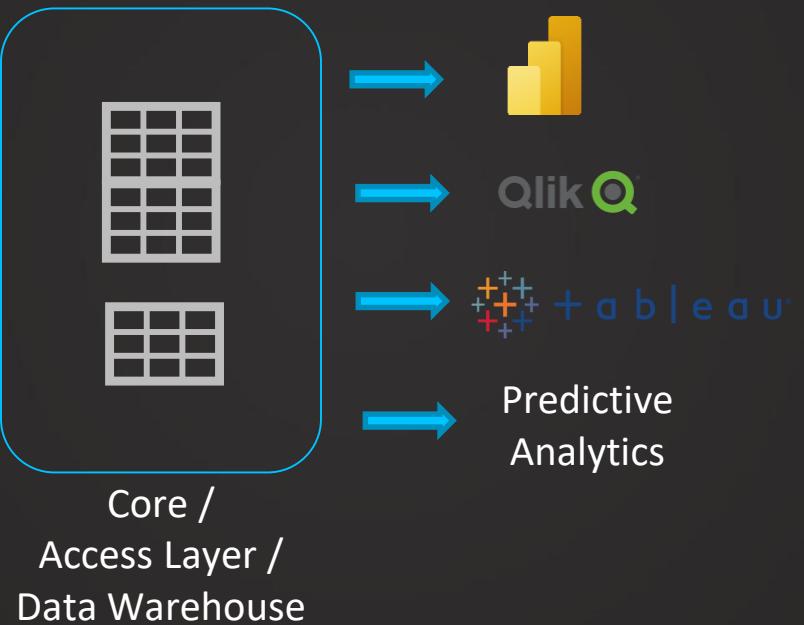
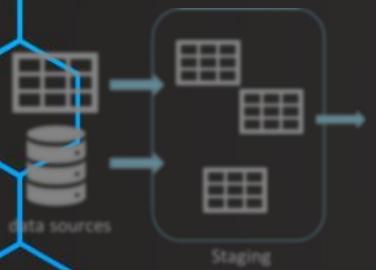
Data Marts

Data Warehouse Layers

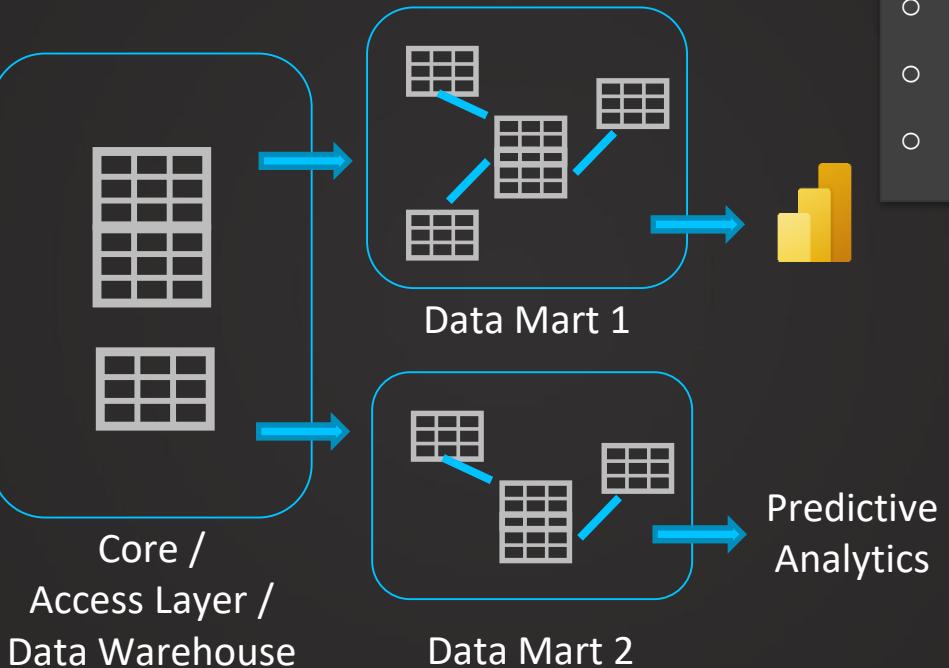


Type text here

Data Warehouse Layers



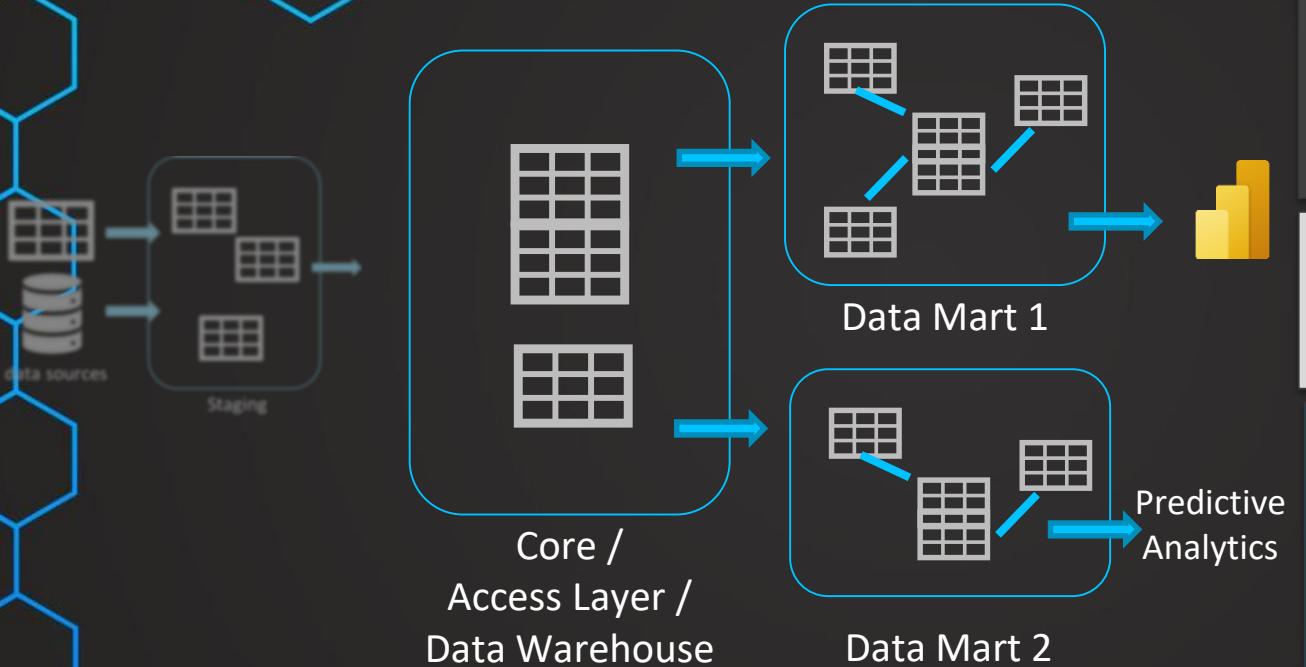
Data Warehouse Layers



Data Mart not always necessary.

- *Subset of a DWH*
- *Dimensional Model*
- Can be further aggregated

Data Warehouse Layers



- Subset of a DWH
 - Dimensional Model
 - Can be further aggregated
-
- *Usability + Acceptance*
 - *Performance*
-
- *Tools*
 - *Departments*
 - *Regions*
 - *Use-cases*

Data Marts

- ✓ Data Mart = Small scale DWH?
 - ⇒ Focus on the business problem
- ✓ Should you use a Data Mart or not?
 - ⇒ Focus on the business problem

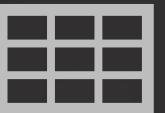
Relational Database

Relational Database

Relational Database



Relational database



Tables (relations)

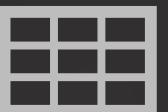
id	date	product	customer_id
1	1/2/2022	Fultoss Tangy Tomato	2
2	1/2/2022	Chilli - Green, Organically Grown	2
3	1/2/2022	Masala Powder	5
4	1/2/2022	Cheese Cracker (Mcvities)	1
5	1/2/2022	Centre Filled Chocolate Cake	5

```
SELECT      <column1>,  
            <column2>, ...  
FROM        <table_name>
```

Relational Database



Relational database



Tables (relations)

Primary key

id	date	product	customer_id
1	1/2/2022	Fultoss Tangy Tomato	2
2	1/2/2022	Chilli - Green, Organically Grown	2
3	1/2/2022	Masala Powder	5
4	1/2/2022	Cheese Cracker (Mcvities)	1
5	1/2/2022	Centre Filled Chocolate Cake	5

Foreign key

id	name	city
1	Frank	New York
2	Sarah	Chicago
3	Sabrina	New Orleans
4	Maya	Los Angeles
5	Marc	Delas

Relational Database

```
SELECT      sales.id,  
           product,  
           customer_id,  
           name  
FROM        sales  
  
LEFT JOIN   customer  
ON         customer_id = customer.id
```

Primary key

id	date	product	customer_id	name
1	1/2/2022	Fulltoss Tangy Tomato	2	Sarah
2	1/2/2022	Chilli - Green, Organically Grown	2	Sarah
3	1/2/2022	Masala Powder	5	Marc
4	1/2/2022	Cheese Cracker (Mcvities)	1	Frank
5	1/2/2022	Centre Filled Chocolate Cake	5	Marc

Foreign key

id	name	city
1	Frank	New York
2	Sarah	Chicago
3	Sabrina	New Orleans
4	Maya	Los Angeles
5	Marc	Delas

Relational Database

- 70s to 90s building logic & improving performance
- Operational systems: **1 table**
OLAP / Analysis: **multiple tables** (context)

Rise of RD => rise of OLAP / DWH

Primary key

id	date	product	customer_id	name
1	1/2/2022	Fultoss Tangy Tomato	2	Sarah
2	1/2/2022	Chilli - Green, Organically Grown	2	Sarah
3	1/2/2022	Masala Powder	5	Marc
4	1/2/2022	Cheese Cracker (Mcvities)	1	Frank
5	1/2/2022	Centre Filled Chocolate Cake	5	Marc

Foreign key

id	name	city
1	Frank	New York
2	Sarah	Chicago
3	Sabrina	New Orleans
4	Maya	Los Angeles
5	Marc	Delas

Relational Database

- Relational database management system (RDMS)

Oracle

Microsoft SQL Server

PostgreSQL

MySQL

Amazon Relational Database Service (RDS)

Azure SQL databases

(Snowflake)

Primary key

id	date	product	customer_id	name
1	1/2/2022	Fultoss Tangy Tomato	2	Sarah
2	1/2/2022	Chilli - Green, Organically Grown	2	Sarah
3	1/2/2022	Masala Powder	5	Marc
4	1/2/2022	Cheese Cracker (Mcvities)	1	Frank
5	1/2/2022	Centre Filled Chocolate Cake	5	Marc

Foreign key

id	name	city
1	Frank	New York
2	Sarah	Chicago
3	Sabrina	New Orleans
4	Maya	Los Angeles
5	Marc	Delas

In-memory databases

In-memory databases

- ✓ Highly optimized for query performance
- ✓ Good for Analytics / High query volume
- ✓ Usually used for data marts
- ✓ Relational and non-relational

In-memory databases



Traditional database

Disc

Response time

In-memory

In-memory databases



Traditional database

Disc

Response time

In-memory



In-memory database

Disc

→

In-memory

- columnar storage,
- parallel query plans,
- and other techniques

In-memory databases

- *Durability:* Lose all information when device loses power or is reset
- Durability added through snapshots / images
- Cost-factor
- Traditional DBs also trying reduce usage of disc



- *columnar storage,*
- *parallel query plans,*
- *and other techniques*

In-memory database



Disks

In-memory

Products in RDMS context

- ✓ SAP HANA
- ✓ MS SQL Server In-Memory Tables
- ✓ Oracle In-Memory
- ✓ Amazon MemoryDB

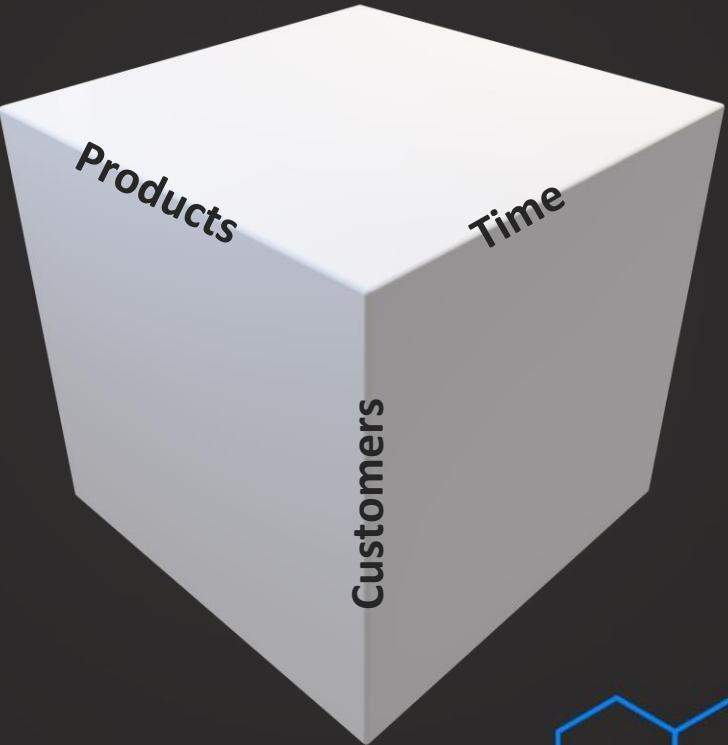
OLAP Cubes

OLAP Cubes

OLAP Cubes

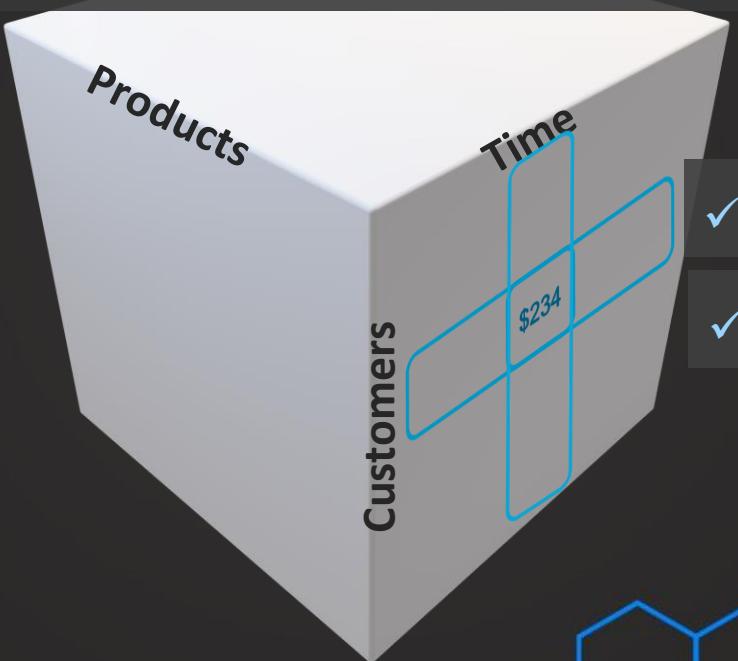
- ✓ Traditional DWH based on relational DBMS (ROLAP)
- ✓ Data is organized non-relational in Cube (MOLAP)
Cube = Multidimensional dataset
- ✓ Arrays instead of tables
- ✓ Main reason to use: Fast query performance
- ✓ Works well with many BI solutions

OLAP Cubes



OLAP Cubes

✓ Precalculated (aggregated values)



- ✓ High performance
- ✓ Interactive tools to drill / slice & dice

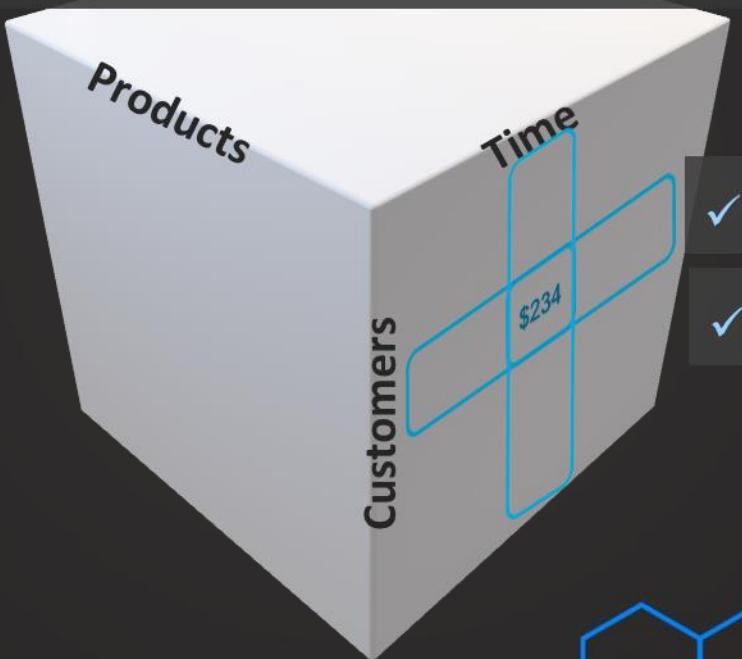
✓ MDX

✓ Multidimensional DBs

Benefits

- ✓ High performance
- ✓ Interactive tools to drill / slice & dice

✓ Precalculated (aggregated values)



✓ MDX

✓ Multidimensional DBs

Recommendation

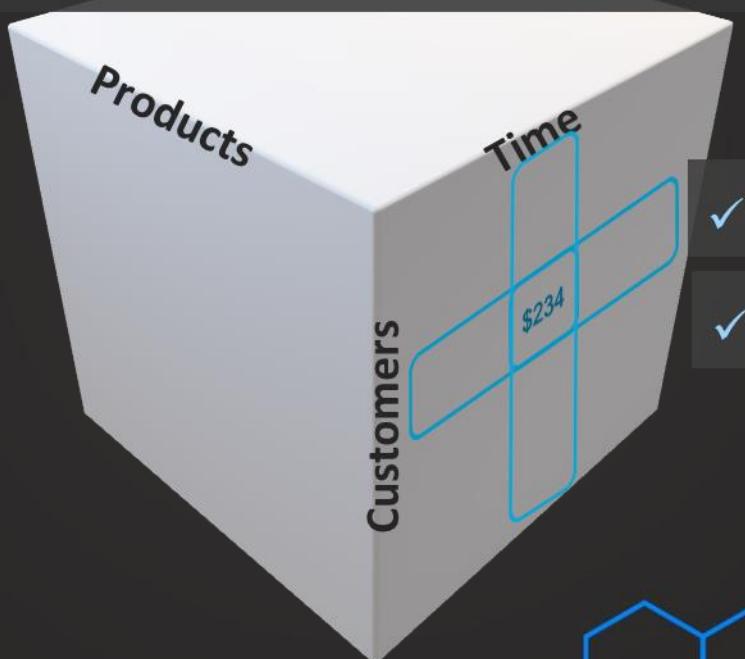
- ✓ Built for a specific use-case (as data marts in general)
- ✓ More efficient & less complex with separate data marts
- ✓ Good for interactive queries with hierarchies
- ✓ Optional after star schema is built in relational DB

Alternatives

- ✓ Less important today with advancement of hardware
- ✓ Alternatives:
 - Tabular models (SSAS)
 - ROLAP
 - columnar storage

OLAP Cubes

✓ Precalculated (aggregated values)



- ✓ High performance
- ✓ Interactive tools to drill / slice & dice

✓ MDX

✓ Multidimensional DBs

ODS (Operational Data Storage)

ODS

- ✓ Sometimes a little bit confusing
- ✓ Different understandings / definitions

ODS



Other data sources

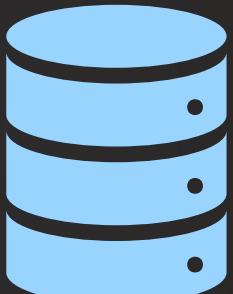
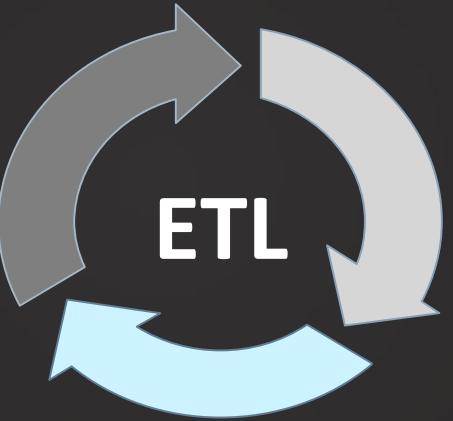


Sales data



CRM system

✓ **Operational decision making**

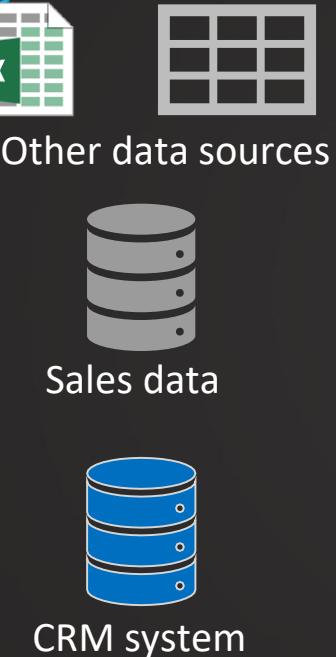


ODS

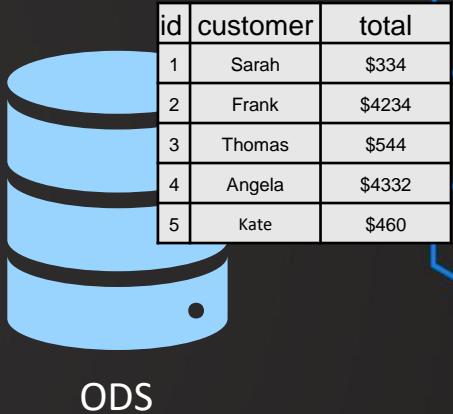
ODS

- ✓ No need for long history
- ✓ Needs to be very current or real-time

ODS

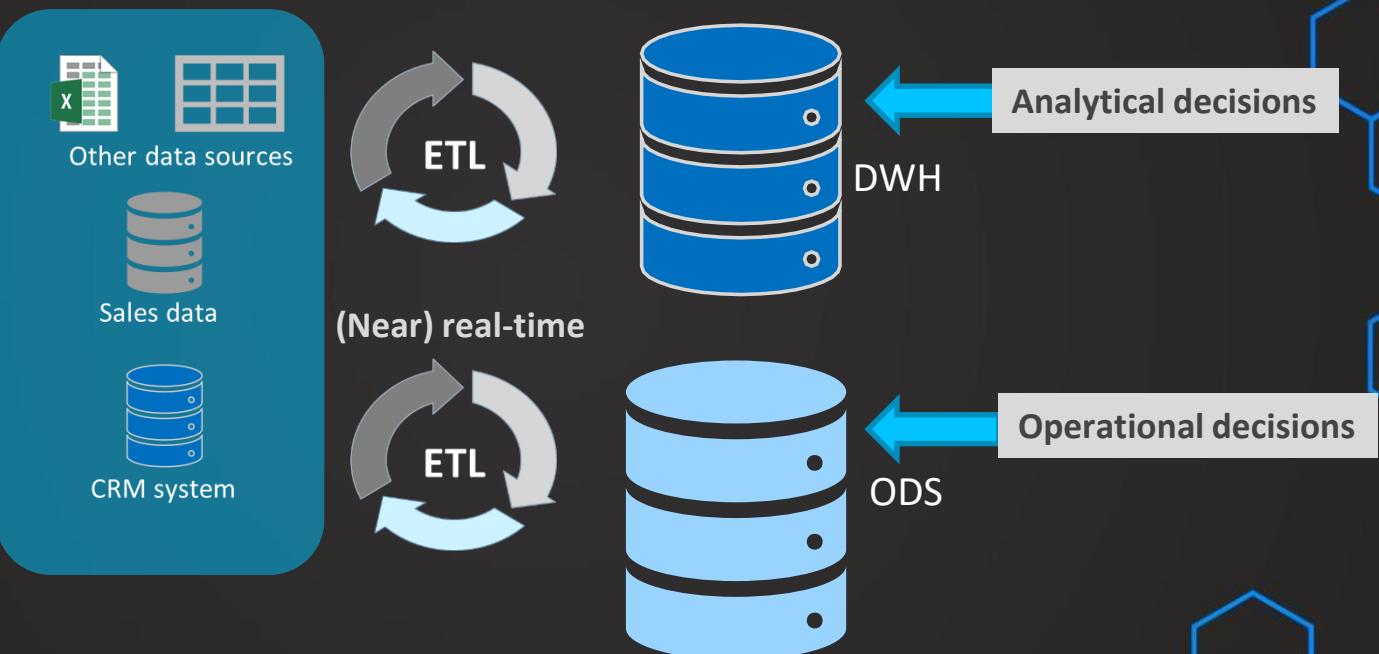


✓ (Near) real-time

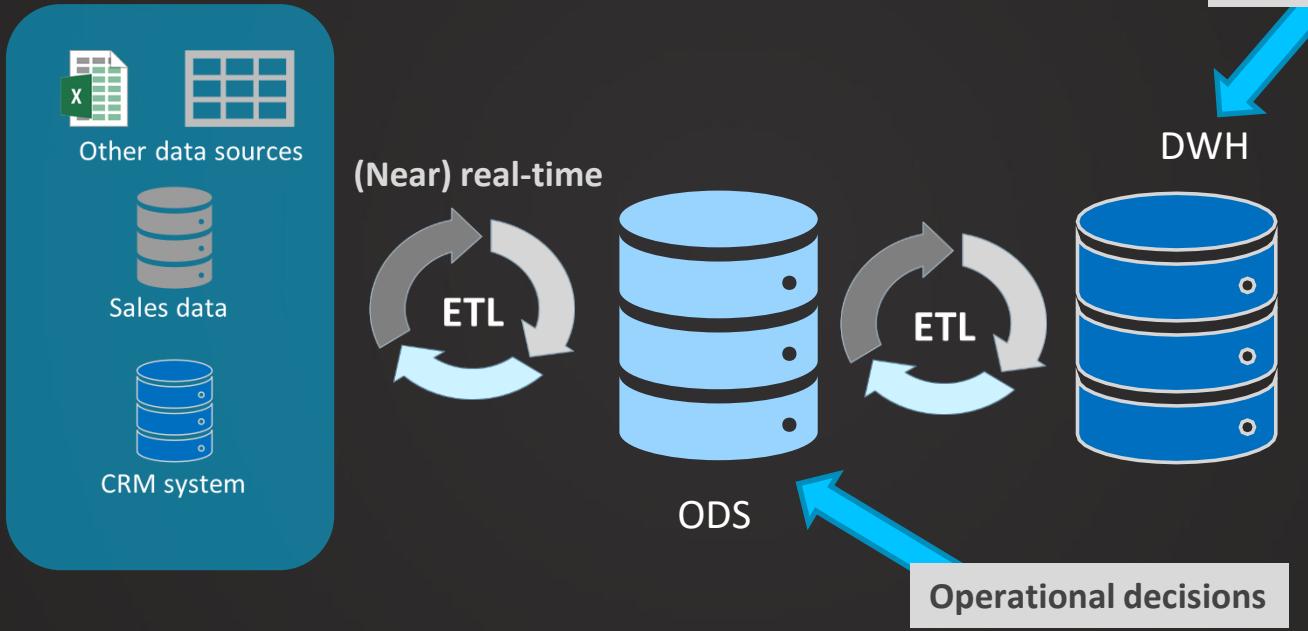


✓ Current state (Update logic)

ODS - Paralell



ODS - Sequential

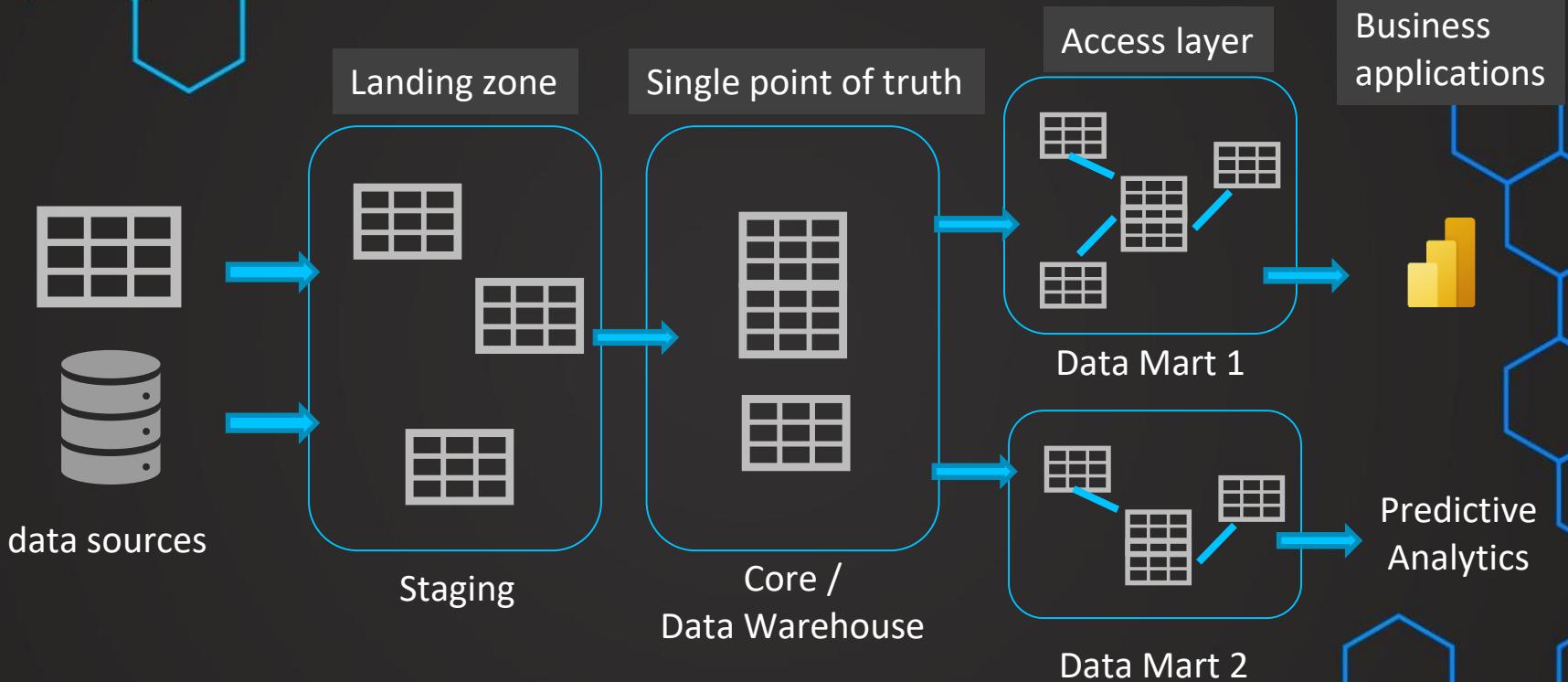


ODS

- ✓ **Getting less relevant**
 - ✓ Better performance (Faster ETL / DBs)
 - ✓ Big data technologies (very fast / real-time)
- ✓ **Don't get hung up with terminology!**

Summary

Data Warehouse Layers



The different layers

Staging

Core

Mart

- ✓ Landing zone
- ✓ Minimal transformation
- ✓ "Stage" the data in tables
- ✓ Always there
- ✓ Business Logic & Single Point of Truth
- ✓ Can be sometimes the access layer
- ✓ Access Layer
- ✓ Specific to one use-case
- ✓ Optimized for performance

In-memory databases



Traditional database

Disc

Response time

In-memory



In-memory database

Disc

→

In-memory

- columnar storage,
- parallel query plans,
- and other techniques



What is dimensional modeling?

Dimensional modeling

- ✓ Method of organizing data (in a data warehouse)

✓ Facts

- Measurement like profit

✓ Dimensions

- Context like category or period

Profit by year

Profit by category

Dimensional modeling

✓ Dimensions

✓ Facts

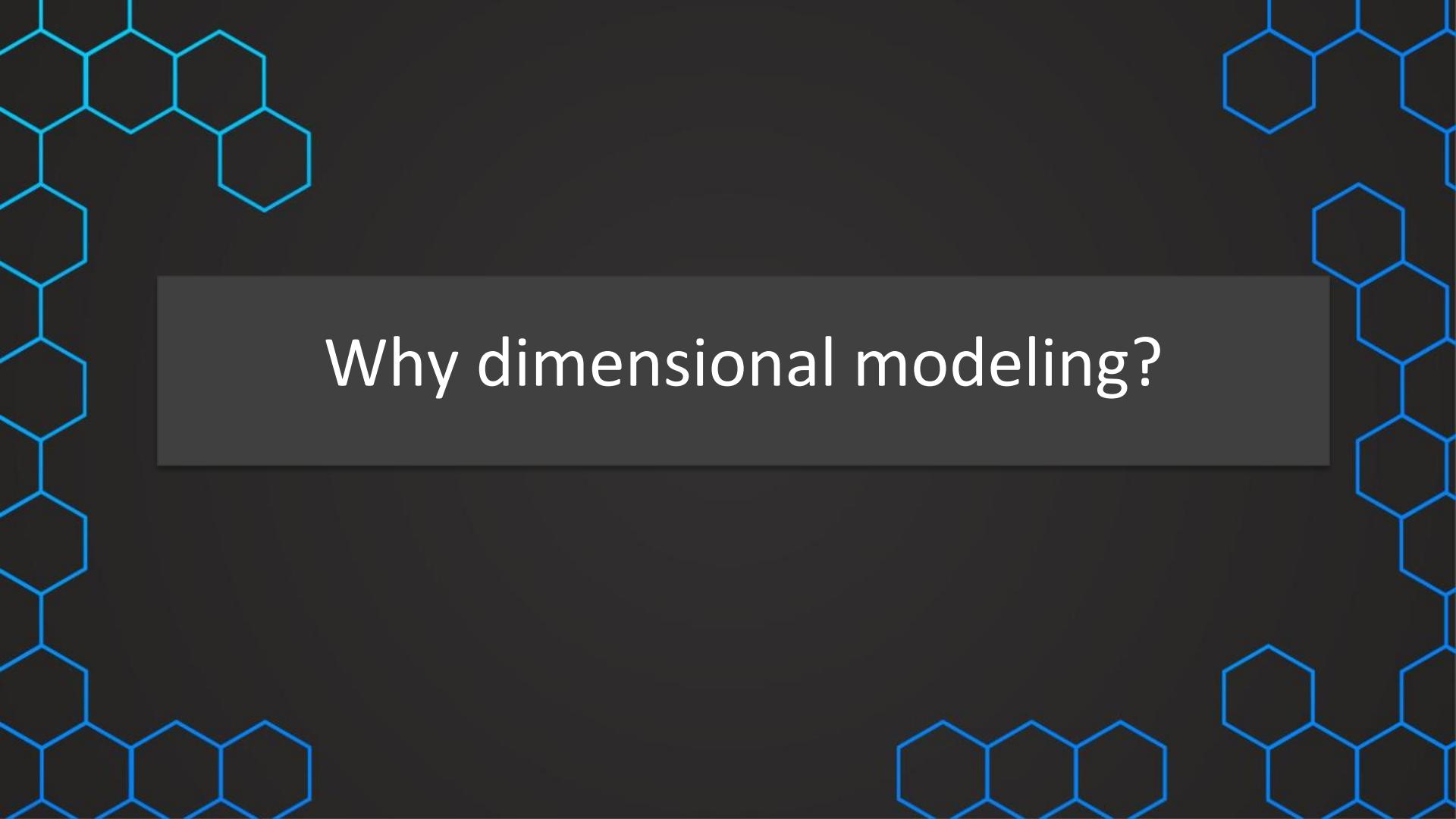
✓ Dimensions

✓ Dimensions

star schema

Dimensional modeling

- ✓ Unique technique of structuring data
- ✓ Commonly used in DWH
- ✓ Optimized for faster data retrieval
- ✓ Oriented around performance & usability
- ✓ Designed Reporting / OLAP



Why dimensional modeling?

Dimensional modeling

- ✓ Goal: Fast data retrieval
- ✓ Oriented around performance & usability

id	date	product	category	customer_id	name	profit
1	1/2/2022	Fulltoss Tangy Tomato	Vegetables	2	Sarah	\$23
2	1/2/2022	Chilli - Green, Organically Grown	Snacks	2	Sarah	\$12
3	1/2/2022	Masala Powder	Herbs	5	Marc	\$93
4	1/2/2022	Cheese Cracker (McVities)	Snacks	1	Frank	\$23
5	1/2/2022	Centre Filled Chocolate Cake	Snacks	5	Marc	\$21

Dimensional modeling

- ✓ Goal: Fast data retrieval
- ✓ Oriented around performance & usability

id	date	product	category	customer_id	name	profit
1	1/2/2022	Fulltoss Tangy Tomato	Vegetables	2	Sarah	\$23
2	1/2/2022	Chilli - Green, Organically Grown	Snacks	2	Sarah	\$12
3	1/2/2022	Masala Powder	Herbs	5	Marc	\$93
4	1/2/2022	Cheese Cracker (Mcvities)	Snacks	1	Frank	\$23
5	1/2/2022	Centre Filled Chocolate Cake	Snacks	5	Marc	\$21

Dimensional modeling

- ✓ Goal: Fast data retrieval
- ✓ Oriented around performance & usability

id	date	product	category	customer_id	profit
1	1/2/2022	Fulltoss Tangy Tomato	Vegetables	2	\$23
2	1/2/2022	Chilli - Green, Organically Grown	Snacks	2	\$12
3	1/2/2022	Masala Powder	Herbs	5	\$93
4	1/2/2022	Cheese Cracker (Mcvities)	Snacks	1	\$23
5	1/2/2022	Centre Filled Chocolate Cake	Snacks	5	\$21

Dimensional modeling

- ✓ Goal: Fast data retrieval
- ✓ Oriented around performance & usability

id	date	product	category	customer_id	profit
1	1/2/2022	Fulltoss Tangy Tomato	Vegetables	2	\$23
2	1/2/2022	Chilli - Green, Organically Grown	Snacks	2	\$12
3	1/2/2022	Masala Powder	Herbs	5	\$93
4	1/2/2022	Cheese Cracker (Mcvities)	Snacks	1	\$23
5	1/2/2022	Centre Filled Chocolate Cake	Snacks	5	\$21

Dimensional modeling

- ✓ Goal: Fast data retrieval
- ✓ Oriented around performance & usability

FK

id	date	product_id	customer_id	profit
1	1/2/2022	2	2	\$23
2	1/2/2022	5	2	\$12
3	1/2/2022	6	5	\$93
4	1/2/2022	23	1	\$23
5	1/2/2022	16	5	\$21

Profit Fact Table

PK

product_id	product	category
1	product 1	Vegetables
2	product 2	Snacks
3	product 3	Herbs
4	product 4	Snacks
5	product 5	Snacks

Product Dim

Dimensional modeling

- ✓ Goal: Fast data retrieval
- ✓ Oriented around performance & usability

id	date_id	product_id	customer_id	profit
1	20220102	2	2	\$23
2	20220102	5	2	\$12
3	20220102	6	5	\$93
4	20220102	23	1	\$23
5	20220102	16	5	\$21

Dimensional modeling

- ✓ Goal: Fast data retrieval
- ✓ Oriented around performance & usability

id	date_id	product_id	customer_id	profit
1	20220102	2	2	\$23
2	20220102	5	2	\$12
3	20220102	6	5	\$93
4	20220102	23	1	\$23
5	20220102	16	5	\$21

Profit Fact Table

date_id	weekday	month
20220102	Monday	January
20220103	Tuesday	January
20220104	Wednesday	January
20220105	Friday	January
20220106	Saturday	January

Date Dim

Dimensional modeling

- ✓ Goal: Fast data retrieval
- ✓ Oriented around performance & usability

Performance

Usability

Preferred technique for data warehouse!

Facts

Facts

✓ Dimensions

✓ Facts

✓ Dimensions

✓ Dimensions

star schema

Facts

- Foundation of DWH
- Key measurements
- Aggregated and analyzed

Dim_Product
product_id
name
category
subcategory
dimensions



Dim_Customer
customer_id
first name
last name
sex
city



Sales
sales_id
product_id
customer_id
units
price



- Usually...
- Aggregatable (numerical values)
 - Measureable vs. descriptive
 - Event- or transactional data
 - Date/time in a fact table

Dim_Date
date_id
year
quarter
month
week
day
weekday
holiday_flag



Di

Facts

- ✓ Fact table: PK, FK & Facts
- ✓ Grain: Most atomic level facts are defined

id	date_id	region_id	profit
1	20220102	1	\$23
2	20220102	2	\$12
3	20220102	2	\$93
4	20220102	3	\$23
5	20220102	16	\$21

- ✓ Different types of facts

Dimensions

Dimensions

✓ Dimensions

✓ Facts

✓ Dimensions

✓ Dimensions

star schema

Dimensions

- *Categorizes facts*
- *Supportive & descriptive*
- *Filtering, Grouping & Labeling*

✓

Dim_Product
product_id
name
category
subcategory
dimensions

✓

Dim_Customer
customer_id
first name
last name
sex
city

✓

Sales
sales_id
product_id
customer_id
units
price

Usually...

- *Non-Aggregatable*
- *Measureable vs. descriptive*
- *(More) static*

✓ Di

Dim_Date
date_id
year
quarter
month
week
day
weekday
holiday_flag

Dimensions

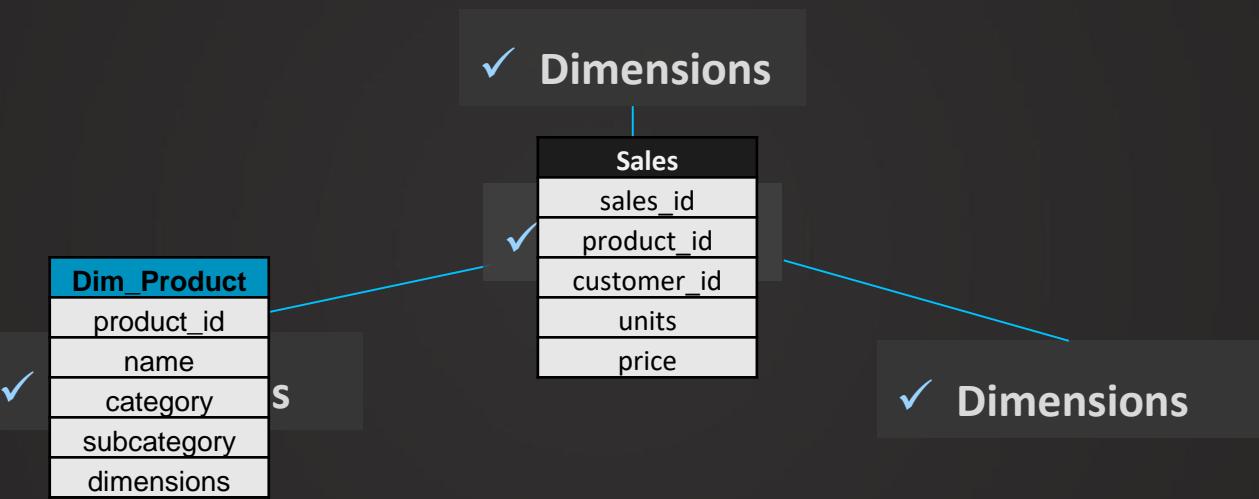
- ✓ Dimension table: PK, Dimension, (FK)
- ✓ People, products, places, time

customer_id	first_name	last_name	email
1	Mike	Miller	mike323@gmail.com
2	Sofia	Snider	snider_sof@gmail.com
2	Marco	Steadman	mstread23@gmail.com
3	Sarah	Griffith	sarah.griff@gmail.com
4	Jennifer	Lovell	jlovell@gmail.com

- ✓ Different types of dimension

Star schema

Star schema



Normalized

- Technique to avoid redundancy
- Minimizes storage
- Performance (write / update)
- Many tables
- Many joins necessary

PK

1:n

✓ Dimensions

product_id	name	category	sub_category
1	Chili	Herbs	Spices
2	Garlic	Fruits & Vegetables	Vegetable
3	Banana	Fruits & Vegetables	Fruits
4	Chocolate	Sweets & Snacks	Sweets
5	Chips	Sweets & Snacks	Snacks

Star schema

FK

sales_id	product_id	customer_id	units	price
1	3	23	1	2.99
2	5	13	1	1.99
3	2	7	2	3.49
4	3	16	1	2.29
5	3	13	5	1.49

✓ Facts

Denormalized

- There is data redundancy!
- Optimized to get data out
- Query performance (read)
- User experience

Normalized

- Technique to avoid redundancy
- Minimizes storage
- Performance (write / update)
- Many tables
- Many joins necessary

PK

product_id	name	category	sub_category
1	Chili	Herbs	Spices
2	Garlic	Fruits & Vegetables	Vegetable
3	Banana	Fruits & Vegetables	Fruits
4	Chocolate	Sweets & Snacks	Sweets
5	Chips	Sweets & Snacks	Snacks

1:n

✓ Dimensions

Star schema

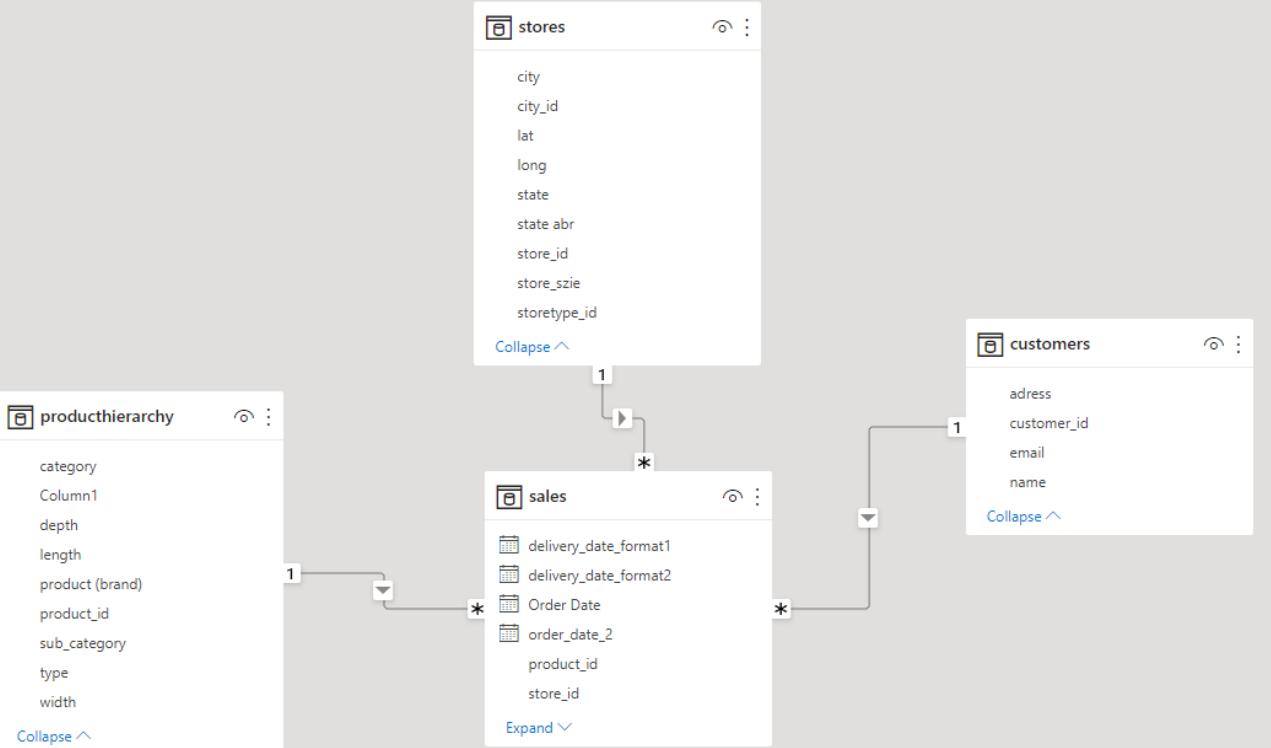
FK

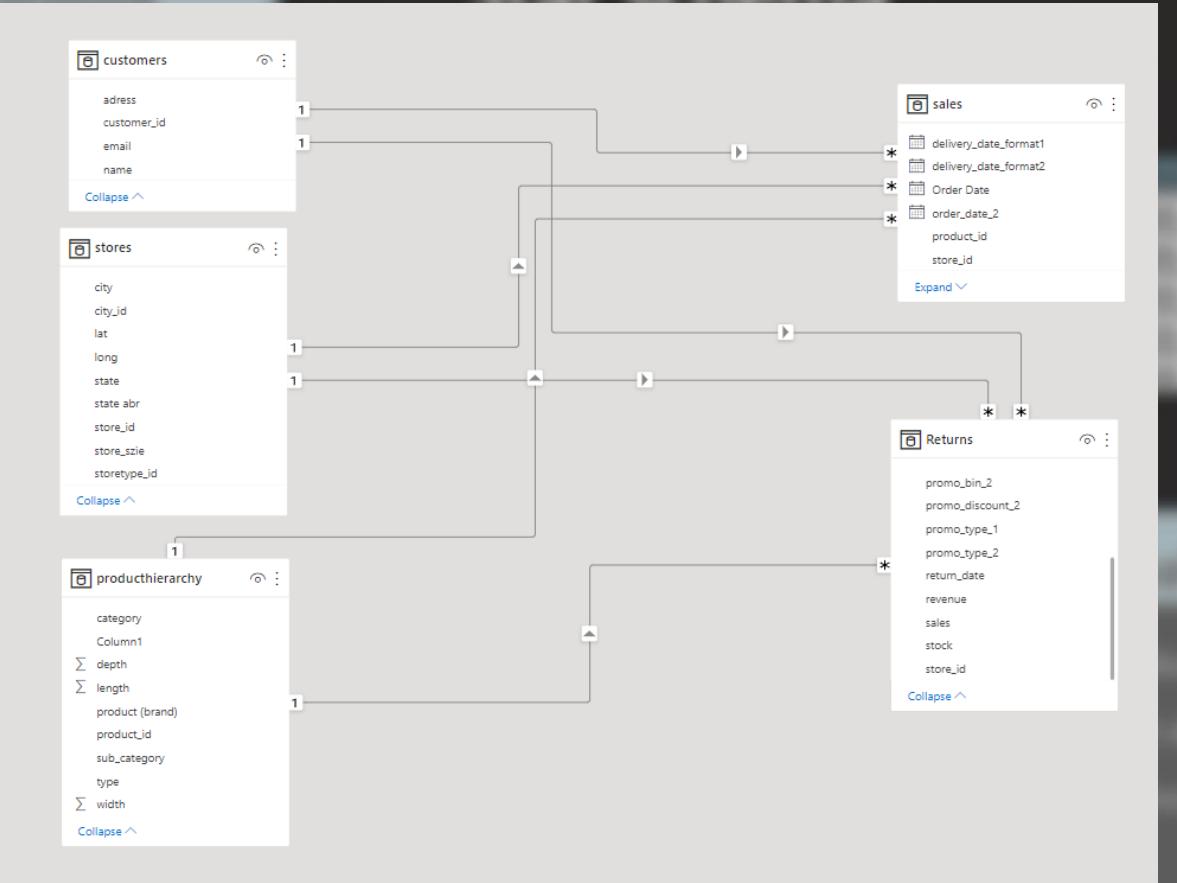
sales_id	product_id	customer_id	units	price
1	3	23	1	2.99
2	5	13	1	1.99
3	2	7	2	3.49
4	3	16	1	2.29
5	3	13	5	1.49

✓ Facts

Denormalized

- There is data redundancy!
- Optimized to get data out
- Query performance (read)
- User experience





Star schema

- ✓ **Most common schema in Data Mart**
- ✓ **Simplest form (vs. snowflake schema)**
- ✓ **Work best for specific needs**
(simple set of queries vs complex queries)
- ✓ **Usability + Performance for specific (read) use-case**

Snowflake schema

Star schema

sales_id	product_id	customer_id	units	price
1	3	23	1	2.99
2	5	13	1	1.99
3	2	7	2	3.49
4	3	16	1	2.29
5	3	13	5	1.49

✓ Facts

✓ Dimensions

product_id	name	category	sub_category
1	Chili	Herbs	Spices
2	Garlic	Fruits & Vegetables	Vegetable
3	Banana	Fruits & Vegetables	Fruits
4	Chocolate	Sweets & Snacks	Sweets
5	Chips	Sweets & Snacks	Snacks

Snowflake schema

✓ Facts

sales_id	product_id	customer_id	units	price
1	3	23	1	2.99
2	5	13	1	1.99
3	2	7	2	3.49
4	3	16	1	2.29
5	3	13	5	1.49

product_id	name	category_id	sub_category
1	Chili	1	Spices
2	Garlic	2	Vegetable
3	Banana	2	Fruits
4	Chocolate	3	Sweets
5	Chips	3	Snacks

Snowflake schema

(More) normalized

category_id	category
1	Herbs
2	Fruits & Vegetables
3	Sweets & Snacks

Snowflake schema

Advantage

- ✓ Less space (storage cost)
- ✓ No (less) redundant data
(easier to maintain/update,
less risk of corrupted data)
- ✓ Solves write slow downs

Disadvantage

- ✓ More complex
- ✓ More joins
(more complex SQL queries)
- ✓ Less performance Data Marts
/ Cubes

Snowflake schema

Data Mart

Core

✓ Star schema

✓ Star schema

✓ Maybe snowflake schema

Additivity in facts

Additivity

Additive

- ✓ Can be added across all dimensions
- ✓ Most flexible & useful

Semi-additive

- ✓ Can be added across a few dimensions

Non-additive

- ✓ Cannot be added across any dimension

Additive facts

sales_id	product_id	date_id	units	amount
1	3	20220101	1	2.99
2	5	20220102	1	1.99
3	2	20220102	2	3.49
4	3	20220103	1	2.29
5	3	20220104	5	1.49

product_id	name	category	sub_category
1	Chili	Herbs	Spices
2	Garlic	Fruits & Vegetables	Vegetable
3	Banana	Fruits & Vegetables	Fruits
4	Chocolate	Sweets & Snacks	Sweets
5	Chips	Sweets & Snacks	Snacks

category	units
Herbs	0
Fruits & Vegetables	9
Sweets & Snacks	1

name	Units
Chili	0
Garlic	2
Banana	7
Chocolate	0
Chips	1

Additive facts

date_id	Date	Day	Month
20220101	01/01/2022	1	1
20220102	02/01/2022	2	1
20220103	03/01/2022	3	1
20220104	04/01/2022	4	1
20220105	05/01/2022	5	1

Date	units
01/01/2022	2
02/01/2022	3
03/01/2022	1
04/01/2022	5

sales_id	product_id	date_id	units	price
1	3	20220101	1	2.99
2	5	20220102	1	1.99
3	2	20220102	2	3.49
4	3	20220103	1	2.29
5	3	20220104	5	1.49

product_id	name	category	sub_category
1	Chili	Herbs	Spices
2	Garlic	Fruits & Vegetables	Vegetable
3	Banana	Fruits & Vegetables	Fruits
4	Chocolate	Sweets & Snacks	Sweets
5	Chips	Sweets & Snacks	Snacks

category	units
Herbs	0
Fruits & Vegetables	9
Sweets & Snacks	1

name	Units
Chili	0
Garlic	2
Banana	7
Chocolate	0
Chips	1

Additivity

Additive

- ✓ Can be added across all dimensions
- ✓ Most flexible & useful
- ✓ Most facts are fully additive

Semi-additive

- ✓ Can be added across a few dimensions

Non-additive

- ✓ Cannot be added across any dimension

Semi-additive facts

Portfolio_id	Type
1	USD Cash
2	Stocks

Added across Types

Date_id	balance
20220101	\$170
20220102	\$270
20220103	\$160

Added across Date

Type	balance
USD Cash	\$250
Stocks	\$350

Semi-additive facts

Portfolio_id	Type
1	USD Cash
2	Stocks

Added across Types

Date_id	balance
20220101	\$170
20220102	\$270
20220103	\$160

Average across Date

Type	balance
USD Cash	\$83.33
Stocks	\$116.67

Additivity

Additive

- ✓ Can be added across all dimensions
- ✓ Most flexible & useful
- ✓ Most facts are fully additive

Semi-additive

- ✓ Can be added across a few dimensions
- ✓ Used carefully & less flexible
- ✓ Averaging might be an alternative

Non-additive

- ✓ Cannot be added across any dimension

Non-additive facts

- Price
- Percentages
- Ratios

sales_id	product_id	date_id	units	price
1	3	20220101	1	2.99
2	5	20220102	1	1.99
3	2	20220102	2	3.49
4	3	20220103	1	2.29
5	3	20220104	5	1.49

product_id	name	category	sub_category
1	Chili	Herbs	Spices
2	Garlic	Fruits & Vegetables	Vegetable
3	Banana	Fruits & Vegetables	Fruits
4	Chocolate	Sweets & Snacks	Sweets
5	Chips	Sweets & Snacks	Snacks

category	price
Herbs	\$0
Fruits & Vegetables	\$10.26
Sweets & Snacks	\$1.99

Additivity

Additive

- ✓ Can be added across all dimensions
- ✓ Most flexible & useful
- ✓ Most facts are fully additive

Semi-additive

- ✓ Can be added across a few dimensions
- ✓ Used carefully & less flexible
- ✓ Averaging might be an alternative
- ✓ Example: Balance

Non-additive

- ✓ Cannot be added across any dimension
- ✓ Limited analytical value
- ✓ Store underlying value
- ✓ Ratio, price etc.

Nulls in facts

Nulls in facts

balance_id	portfolio_id	balance	Incoming	Outgoing
1	1	\$50	null	null
2	1	\$100	\$50	null
3	1	\$100	null	null
4	2	\$120	null	null
5	2	\$170	\$50	null
6	2	\$60	null	\$110

```
SELECT  
    AVG(Incoming),  
    MIN(Incoming),  
    SUM(Incoming)  
FROM balance_table
```

AVG	MIN	SUM
\$50	\$50	\$100

Nulls in facts

balance_id	portfolio_id	balance	Incoming	Outgoing
1	1	\$50	\$0	\$0
2	1	\$100	\$50	\$0
3	1	\$100	\$0	\$0
4	2	\$120	\$0	\$0
5	2	\$170	\$50	\$0
6	2	\$60	\$0	\$110

```
SELECT  
    AVG(Incoming),  
    MIN(Incoming),  
    SUM(Incoming)  
FROM balance_table
```

AVG	MIN	SUM
\$16.67	\$0	\$100

Nulls in facts

balance_id	portfolio_id	balance	Incoming	Outgoing
1	1	\$50	\$0	\$0
2	null	\$100	\$50	\$0
3	1	\$100	\$0	\$0
4	2	\$120	\$0	\$0
5	null	\$170	\$50	\$0
6	2	\$60	\$0	\$110

Portfolio_id	Type
1	USD Cash
2	Stocks

AVG	MIN	SUM
\$16.67	\$0	\$100

Nulls in facts

balance_id	portfolio_id	balance	Incoming	Outgoing
1	1	\$50	\$0	\$0
2	999	\$100	\$50	\$0
3	1	\$100	\$0	\$0
4	2	\$120	\$0	\$0
5	999	\$170	\$50	\$0
6	2	\$60	\$0	\$110

Portfolio_id	Type
1	USD Cash
2	Stocks
999	Old types

AVG	MIN	SUM
\$16.67	\$0	\$100

Year-to-Date facts

Year-to-Date facts

- ✓ Often requested by business users
- ✓ Tempted to store them in columns
- ✓ Month-to-Date, Quarter-to-Date, Fiscal-Year-to-Date etc.
- ✓ Better store the underlying values in defined grain (!)
- ✓ Instead calculate all the to-Date variations in BI tool

Transactional fact table

Transactional fact table

- ✓ 1 row = measurement of 1 event / transaction
- ✓ Taken place at a specific time
- ✓ One transaction defines the lowest grain

	FK	FK	Measure	
	sales_id	product_id	date_id	units
1	3	20220101	1	
2	5	20220102	1	
3	2	20220102	2	
4	3	20220103	1	
5	3	20220104	5	

Sales transactions

	FK	FK	FK	Measure	
	call_id	emp_id	date_id	customer_id	duration
1	3	20220101	1	43	
2	5	20220102	1	12	
3	2	20220102	2	134	
4	3	20220103	1	62	
5	3	20220104	5	22	

Calls

Characteristics

- ✓ Most common and very flexible
- ✓ Typically additive
- ✓ Tend to have a lot of dimensions associated
- ✓ Can be enormous in size

FK FK Measure

sales_id	product_id	date_id	units
1	3	20220101	1
2	5	20220102	1
3	2	20220102	2
4	3	20220103	1
5	3	20220104	5

Sales transactions

FK FK FK Measure

call_id	emp_id	date_id	customer_id	duration
1	3	20220101	1	43
2	5	20220102	1	12
3	2	20220102	2	134
4	3	20220103	1	62
5	3	20220104	5	22

Calls

Periodic snapshot fact table

Periodic snapshot fact table

- ✓ 1 row = summarizes measure of many events / transactions
- ✓ Summarized of standard period (e.g. 1 day, 1 week etc.)
- ✓ Lowest period defines the grain

Measure Measure Measure

week_id	revenue	sales	cost
1	323	123	12
2	541	322	31
3	242	108	12
4	352	212	51
5	312	198	25

Sales transactions

Measure Measure Measure

day_id	no. calls	missed calles	duration
1	31	3	432
2	25	4	142
3	52	2	134
4	23	6	562
5	53	4	122

Calls

Characteristics

- ✓ Tend to be not as enormous in size
- ✓ Typically additive
- ✓ Tend to have a lot of facts and fewer dimensions associated
- ✓ No events = null or 0

Measure Measure Measure

week_id	revenue	sales	cost
1	323	123	12
2	541	322	31
3	242	108	12
4	352	212	51
5	312	198	25

Sales transactions

Measure Measure Measure

day_id	no. calls	missed calls	duration
1	31	3	432
2	25	4	142
3	52	2	134
4	23	6	562
5	53	4	122

Calls

Accumulation snapshot fact table

Accumulation snapshot fact table

- ✓ 1 row = summarizes measure of many events / transactions
- ✓ Summarized of lifespan of 1 process (e.g. order fulfillment)
- ✓ Definite beginning & definite ending (& steps in between)

	Date FK	Measure	Date FK	Date FK	Date FK	Date FK	Date FK	Date FK	Measure
order_id	Order Date FK	No. Products	Product_FK	Production Start FK	Production End FK	Inspection Date FK	Shipping Date FK	Damaged products	
1	20220102	100	32	20220103	20220110	20220112	20220113	3	
2	20220103	100	32	20220104	20220112	20220113	20220113	4	
3	20220103	100	32	20220103	20220112	20220113	20220114	1	
4	20220104	100	32	20220106	20220110	20220112	20220113	0	
5	20220104	100	32	20220108	20220117	20220119	20220120	6	

Order production

Characteristics

- ✓ Least common
- ✓ Workflow or process analysis
- ✓ Multiple Date/Time foreign keys (for each process step)
- ✓ Date/Time keys associated with role-playing dimension

	Date FK	Measure	Date FK	Date FK	Date FK	Date FK	Date FK	Date FK	Measure
order_id	Order Date FK	No. Products	Product_FK	Production Start FK	Production End FK	Inspection Date FK	Shipping Date FK	Damaged products	
1	20220102	100	32	20220103	20220110	20220112	20220113	3	
2	20220103	100	32	20220104	20220112	20220113	20220113	4	
3	20220103	100	32	20220103	20220112	20220113	20220114	1	
4	20220104	100	32	20220106	20220110	20220112	20220113	0	
5	20220104	100	32	20220108	20220117	20220119	20220120	6	

Order production

Types of fact tables

Types of fact tables

Type	Transactional	Periodic Snapshot	Accumulating Snapshot
Grain			
Date Dimensions			
No. Of dimensions			
Facts			
Size			
Performance			

Types of fact tables

Type	Transactional	Periodic Snapshot	Accumulating Snapshot
Grain	1 row = 1 transaction	1 row = 1 defined period (plus other dimensions)	1 row = lifetime of process /event
Date Dimensions			
No. Of dimensions			
Facts			
Size			
Performance			

Types of fact tables

Type	Transactional	Periodic Snapshot	Accumulating Snapshot
Grain	1 row = 1 transaction	1 row = 1 defined period (plus other dimensions)	1 row = lifetime of process /event
Date Dimensions	1 Transaction date	Snapshot date (end of period)	Multiple snapshot dates
No. Of dimensions			
Facts			
Size			
Performance			

Types of fact tables

Type	Transactional	Periodic Snapshot	Accumulating Snapshot
Grain	1 row = 1 transaction	1 row = 1 defined period (plus other dimensions)	1 row = lifetime of process /event
Date Dimensions	1 Transaction date	Snapshot date (end of period)	Multiple snapshot dates
No. Of dimensions	High	Lower	Very high
Facts			
Size			
Performance			

Types of fact tables

Type	Transactional	Periodic Snapshot	Accumulating Snapshot
Grain	1 row = 1 transaction	1 row = 1 defined period (plus other dimensions)	1 row = lifetime of process /event
Date Dimensions	1 Transaction date	Snapshot date (end of period)	Multiple snapshot dates
No. Of dimensions	High	Lower	Very high
Facts	Measures of transactions	Cumulative measures of transactions in period	Measures of process in lifespan
Size			
Performance			

Types of fact tables

Type	Transactional	Periodic Snapshot	Accumulating Snapshot
Grain	1 row = 1 transaction	1 row = 1 defined period (plus other dimensions)	1 row = lifetime of process /event
Date Dimensions	1 Transaction date	Snapshot date (end of period)	Multiple snapshot dates
No. Of dimensions	High	Lower	Very high
Facts	Measures of transactions	Cumulative measures of transactions in period	Measures of process in lifespan
Size	Largest (most detailed grain)	Middle (less detailed grain)	Lowest (highest aggregation)
Performance			

Types of fact tables

Type	Transactional	Periodic Snapshot	Accumulating Snapshot
Grain	1 row = 1 transaction	1 row = 1 defined period (plus other dimensions)	1 row = lifetime of process /event
Date Dimensions	1 Transaction date	Snapshot date (end of period)	Multiple snapshot dates
No. Of dimensions	High	Lower	Very high
Facts	Measures of transactions	Cumulative measures of transactions in period	Measures of process in lifespan
Size	Largest (most detailed grain)	Middle (less detailed grain)	Lowest (highest aggregation)
Performance	Can be improved with aggregation	Better (less detailed)	Good performance

Steps to create a fact table

Steps to create a fact table

What are the key decisions we need to take during the design?

4 key decisions

Considering the business needs

Tables & columns

Steps to create a fact table

1) Identify business process for analysis

Example:

*Sales,
Order processing*

sales_id	date	Sales amount
1	2022-01-01	\$41
2	2022-01-02	\$15
3	2022-01-02	\$24
4	2022-01-03	\$13
5	2022-01-04	\$52

2) Declare the grain

Example: Transaction, Order, Order lines, Daily, Daily + location

3) Identify dimensions that are relevant

What, when, where, how and why

Example: Time, locations, products, customers,...

Filtering & grouping

"Soul" for analysis

4) Identify facts for measurement

Defined by the grain & not by specific use-case

Steps to create a fact table

1) Identify business process for analysis

2) Declare the grain

Example: Transaction

3) Identify dimensions that are relevant

Example: Time, locations, products

4) Identify facts for measurement

Example: Sales amount & order quantity

sales_id	date	Sales amount
1	2022-01-01	\$41
2	2022-01-02	\$15
3	2022-01-02	\$24
4	2022-01-03	\$13
5	2022-01-04	\$52

sales_id	date	Sales amount	prod_id	loc_id
1	20220101	\$41	3	1
2	20220102	\$15	4	5
3	20220102	\$24	6	4
4	20220103	\$13	1	3
5	20220104	\$52	23	4

Factless fact table

Fact Table



Fact

Fact

Factless fact table

- ✓ Facts are usually numeric
- ✓ Sometimes only dimensionals aspects of an event are recorded
- ✓ Example new employee is registered

reg_id	Entry Date FK	dep_id	region_id	manager_id	Pos_id
1	20220102	1	2	3	10
2	20220103	3	3	4	112
3	20220103	4	6	3	202
4	20220104	4	8	6	110
5	20220104	3	4	8	17

Employee registration

Events

No metrics

Factless fact table

- ✓ How many employees have been registered last month?
- ✓ How many employees have been registered in a certain region?
- ✓ Example new employee is registered

reg_id	Entry Date FK	dep_id	region_id	manager_id	Pos_id
1	20220102	1	2	3	10
2	20220103	3	3	4	112
3	20220103	4	6	3	202
4	20220104	4	8	6	110
5	20220104	3	4	8	17

Employee registration

Events

No metrics

Factless fact table

promo_id	Promo date_id	prod_id	channel_id	campaign_id
1	20220102	5	2	3
2	20220103	3	3	4
3	20220103	4	6	3
4	20220104	4	8	6
5	20220104	3	4	8

Employee registration

Events

No metrics

Occurrence of events

Natural vs. Surrogate key

Natural vs. Surrogate key

Natural keys

product_id	name	category
PX30	Chili	Herbs
PT32	Garlic	Fruits & Vegetables
AX42	Banana	Fruits & Vegetables
DA24	Chocolate	Sweets & Snacks
PO20	Chips	Sweets & Snacks

Products

sales_id	date	Sales amount
GXF-EFS	2022-01-01	\$41
DOS-FWA	2022-01-02	\$15
DSF-GWS	2022-01-02	\$24
PTG-DWD	2022-01-03	\$13
ERW-DWD	2022-01-04	\$52

Sales

Natural vs. Surrogate key

Natural keys

- ✓ Come out of the source system

Product_PK	product_id	name	category
1	PX30	Chili	Herbs
2	PT32	Garlic	Fruits & Vegetables
3	AX42	Banana	Fruits & Vegetables
4	DA24	Chocolate	Sweets & Snacks

Surrogate key

Artificial keys

- ✓ Integer number
- ✓ _PK or _FK suffix
- ✓ Created by the database / ETL tool

Benefits

Surrogate key

- ✓ Improve performance (less storage/better joins)
- ✓ Handle dummy values (nulls / missing values) e.g. 999 or -1
- ✓ Integrate multiple source systems
- ✓ Easier administrate / update
- ✓ Sometimes there are even no natural keys available

Practical guidelines

Surrogate key

- ✓ Always use surrogate keys in tables as main PK and FK
- ✓ Both for Facts & Dimensions (except date dimension)
- ✓ Optionally keep the natural keys

Case study: E-Commerce

Case study: E-Commerce

Corporate IT

E-Commerce company

- ✓ 3 websites 
- ✓ Each website operated independently by multiple departments
- ✓ ~ 1000 individual products 
- ✓ Groceries, kitchen products, household products etc.



Case study: E-Commerce

Data collection

- ✓ Shopping cart check out
- ✓ Warehouse data



Sales data
Customer_id
Customer_name
Order_id
Order_line_name
Order_line_id
Quantity
Unit_price
Discounted_price
sales amount
promo_id
product_cost
DateTime

Customer_id	Customer name	Order_id	Order_line_name	Order_line_id	Quantity	Unit_price	Discounted_price	promo_id	sales amount	product_cost	DateTime
312	Franklin Miller	2314	Sunglasses SU-6		34	2	22.99	22.99 null		45.98	14.84 23/4/2022 13:34
312	Franklin Miller	2314	Beach towel red	156	3	8.99	8.99 null		26.97	4.87 23/4/2022 13:35	
312	Franklin Miller	2314	Swimsuit blue	643	1	16.99	14.99	3	14.99	12.53 23/4/2022 13:36	

Case study: E-Commerce

Goals

- ✓ Logistics in warehouse

- ✓ Maximizing profits
 - ❖ Profit margin, sales volume, product cost, promotions, discounts



Sales data
Customer_id
Customer_name
Order_id
Order_line_name
Order_line_id
Quantity
Unit_price
Discounted_price
sales amount
promo_id
product_cost
DateTime

Customer_id	Customer name	Order_id	Order_line_name	Order_line_id	Quantity	Unit_price	Discounted_price	promo_id	sales amount	product_cost	DateTime
312	Franklin Miller	2314	Sunglasses SU-6		34	2	22.99	22.99 null		45.98	14.84 23/4/2022 13:34
312	Franklin Miller	2314	Beach towel red		156	3	8.99	8.99 null		26.97	4.87 23/4/2022 13:35
312	Franklin Miller	2314	Swimsuit blue		643	1	16.99	14.99	3	14.99	12.53 23/4/2022 13:36

Case study: E-Commerce

Step 1

Identify Business process

✓ Business process for first DWH?

- ❖ Most critical for business
- ❖ Data availability, data quality

Sales transactions

- ❖ Which products sold
- ❖ What is sales profit
- ❖ Sales of each website
- ❖ Performance on different days
- ❖ Sales over time

Case study: E-Commerce

Step 2

Declare the grain

- ✓ What level of detail?
 - ❖ Most analytical value with atomic grain
 - ❖ Highest dimensionality

Order + Order line

Customer_id	Customer name	Order_id	Order_line_name	Order_line_id	Quantity	Unit_price	Discounted_price	Promo_id	sales amount	product_cost	DateTime
312	Franklin Miller	2314	Sunglasses SU-6		34	2	22.99	22.99	null	45.98	14.84 23/4/2022 13:34
312	Franklin Miller	2314	Beach towel red	156	3	8.99	8.99	null	26.97	4.87 23/4/2022 13:35	
312	Franklin Miller	2314	Swimsuit blue	643	1	16.99	14.99		3	14.99	12.53 23/4/2022 13:36

Customer_id	Customer name	Order_id	No. order lines	Total quantity	sales amount	DateTime
312	Franklin Miller	2314	3	6	87.94	23/4/2022 13:34

Case study: E-Commerce

Step 3

Identify dimensions

- ✓ Descriptive aspects of measures
- ❖ Naturally derived after grain defined

Dimensions

Customer_id	Customer name	Order_id	Order_line_name	Order_line_id	Quantity	Unit_price	Discounted_price	Promo_id	sales amount	product_cost	DateTime
312	Franklin Miller	2314	Sunglasses SU-6		34	2	22.99	22.99 null	45.98	14.84	23/4/2022 13:34
312	Franklin Miller	2314	Beach towel red	156	3	8.99	8.99 null		26.97	4.87	23/4/2022 13:35
312	Franklin Miller	2314	Swimsuit blue	643	1	16.99	14.99	3	14.99	12.53	23/4/2022 13:36

- ❖ Customer
- ❖ Products
- ❖ Promotions
- ❖ Time/date
- ❖ Website

Case study: E-Commerce

Step 3

Identify dimensions

- ✓ Descriptive aspects of measures
- ❖ Naturally derived after grain defined

Dimensions

Sales_PK	Website_FK	Customer_FK	Order_id	Order_line_FK	Quantity	Unit_price	Discounted_p	promo_FK	sales	amc	product_cost	Datetime_FK
1001	2	312	2314		34	2	22.99		22.99	-1	45.98	14.84
1002	2	312	2314		156	3	8.99		8.99	-1	26.97	4.87
1003	2	312	2314		643	1	16.99		14.99	3	14.99	12.53

- ❖ Customer
- ❖ Products
- ❖ Promotions
- ❖ Time/date
- ❖ Website

Case study: E-Commerce

Step 4

Identify facts for measurement

- ✓ What facts are in the fact table?
- ❖ Must comply with the grain

Facts

Sales_PK	Website_FK	Customer_FK	Order_id	Order_line_FK	Quantity	Unit_price	Discounted_p	promo_FK	Discount	sales	amount	product_cost	profit	Date	Time	DateTime_FK
1001	2	312	2314	34	2	22.99	22.99		-1	0	45.98	14.84	31.14	202204231300		
1002	2	312	2314	156	3	8.99	8.99		-1	0	26.97	4.87	22.1	202204231300		
1003	2	312	2314	643	1	16.99	14.99		3	2.00	14.99	12.53	2.46	202204231300		

- ✓ Additive
- ❖ Discount absolut (yes?)
- ❖ Discount percentage (no?)
- ❖ Profit

Case study: E-Commerce

Result

Sales_PK	Website_FK	Customer_FK	Order_id	Order_line_FK	Quantity	Unit_price	Discounted_p	promo_FK	Discount	sales_amount	product_cost	Profit	Date/Time_FK
1001	2	312	2314	34	2	22.99	22.99	-1	0	45.98	14.84	31.14	202204231300
1002	2	312	2314	156	3	8.99	8.99	-1	0	26.97	4.87	22.1	202204231300
1003	2	312	2314	643	1	16.99	14.99	3	2.00	14.99	12.53	2.46	202204231300

Website

Customer

Products

Date/time

Dimension tables

Dimensions tables

- ✓ Always has a Primary Key (PK)

Sales_PK	Website_FK	Customer_FK	Order_id	Order_line_ID
1001	2	312	2314	P034
1002	2	312	2314	P156
1003	2	312	2314	P643

Product_ID	Name	Category
P001	Sunglasses TR-7	Accessories
P002	Chocolate bar 70% cacao	Sweets
P003	Oat meal biscuits	Sweets

- ✓ Use surrogate key

Product_PK	Name	Category
1	Sunglasses TR-7	Accessories
2	Chocolate bar 70% cacao	Sweets
3	Oat meal biscuits	Sweets

Product_PK	Product_ID
1	P001
2	P002
3	P003

- ✓ Lookup table

Dimensions tables

Product_ID	Name	Category
P001	Sunglasses TR-7	Assecoirs
P002	Chocolate bar 70% cacao	Sweets
P003	Oat meal biscuits	Sweets

Product_PK	Name	Category
1	Sunglasses TR-7	Assecoirs
2	Chocolate bar 70% cacao	Sweets
3	Oat meal biscuits	Sweets

Product_PK	Product_ID
1	P001
2	P002
3	P003

Sales_PK	Website_FK	Customer_FK	Order_id	Order_line_ID	Product_FK	Unit_price
1001	2	312	2314	P034	34	22.99
1002	2	312	2314	P156	156	8.99
1003	2	312	2314	P643	643	16.99

Sales_PK	Website_FK	Customer_FK	Order_id	Order_line_ID	Product_FK	Unit_price
1001	2	312	2314	P034	34	22.99
1002	2	312	2314	P156	156	8.99
1003	2	312	2314	P643	643	16.99

```
SELECT
    S.*,
    P.Product_PK
FROM Sales_Fact S
LEFT JOIN Product_Dim as P
ON P.Product_ID = S.Order_line_ID
```

Dimensions tables

- ✓ Always has a Primary Key (PK)

Product PK	Name	Category
1	Sunglasses TR-7	Accessories
2	Chocolate bar 70% cacao	Sweets
3	Oatmeal biscuits	Sweets

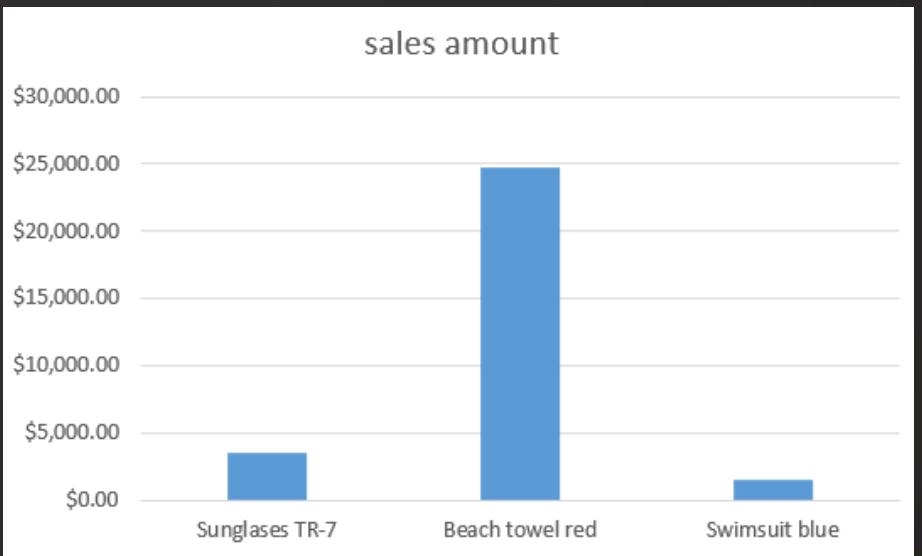
Customer_id	Customer name	Order_id	Order_line_name	Order_line_FK	Quantity	Unit_price	Discounted_price
312	Franklin Miller	2345	Sunglasses TR-7	1	2	23.99	23.99
312	Franklin Miller	2314	Beach towel red	156	3	8.99	8.99
312	Franklin Miller	2314	Swimsuit blue	643	1	16.99	14.99

- ✓ Relatively few rows / many columns with descriptive attributes

Dimensions tables

✓ Group & Filter ("slice & dice")

Order_line_name	sales amount
Sunglasses TR-7	\$3,543.00
Beach towel red	\$24,678.00
Swimsuit blue	\$1,456.00



Date Dimension

Date Dimension

- ✓ One of the most common & most important dimensions
- ✓ Contains date related features
 - ❖ Year, Month (name & number), Day, Quarter, Week, Weekday (name & number), ...
- ✓ Meaningful surrogate key YYYYMMDD
 - For example 2022-04-02 ⇔ 20220402
- ✓ Extra row for no date/null (source) ⇔ 1900-01-01 (dim)

Date Dimension

- ✓ Time is usually a separate dimension
- ✓ Can be populated in advance (e.g. for next 5 or 10 years)

Date features

- Numbers & Text (e.g. January, 1)
- Long & Abbreviated (Jan, January – Mon, Monday)
- Combinations of attributes (Q1, 2022-Q1)
- Fiscal dates (Fiscal Year etc.)
- Flags (Weekend, company holidays etc.)

Date PK	Date	Month	Short Month	Year-Quarter	Year	Weekday	Is Weekend
20220101	2022-01-01	January	Jan	2022-Q1	2022	Saturday	1
20220102	2022-01-02	January	Jan	2022-Q1	2022	Sunday	1
20220103	2022-01-03	January	Jan	2022-Q1	2022	Monday	0

Date Dimension

- ✓ Time is usually a separate dimension
- ✓ Can be populated in advance (e.g. for next 5 or 10 years)

Date features

- Numbers & Text (e.g. January, 1)
- Long & Abbreviated (Jan, January – Mon, Monday)
- Combinations of attributes (Q1, 2022-Q1)
- Fiscal dates (Fiscal Year etc.)
- Flags (Weekend, company holidays etc.)

Nulls in dimensions

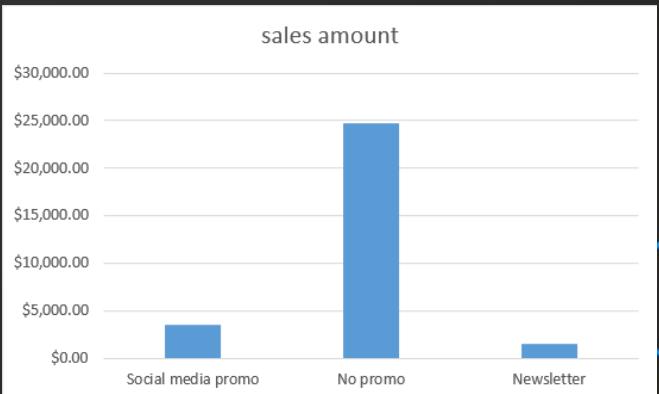
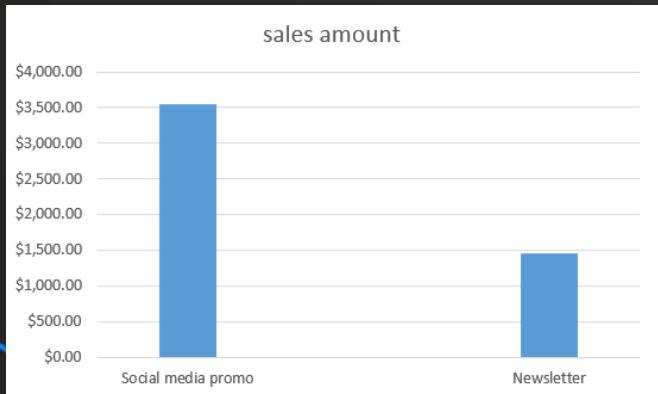
Nulls in dimensions

What we've learnt

- ✓ Nulls must be avoided in FKs

Customer_id	Customer name	Order_id	Order_line_name	Order_line_id	Quantity	Unit_price	Discounted_price	promo_id	sales amount	product_cost	DateTime
312	Franklin Miller	2314	Sunglasses SU-6		34	2	22.99	22.99	null	45.98	14.84 23/4/2022 13:34
312	Franklin Miller	2314	Beach towel red		156	3	8.99	8.99	null	26.97	4.87 23/4/2022 13:35
312	Franklin Miller	2314	Swimsuit blue		643	1	16.99	14.99		14.99	12.53 23/4/2022 13:36

- ❖ Nulls in FKs break referential integrity!
- ❖ They don't appear in Joins



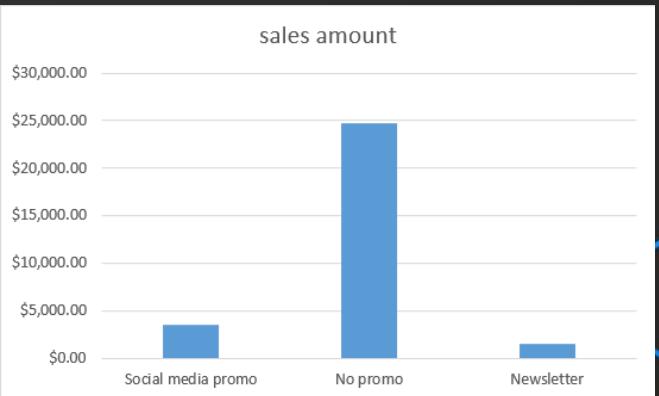
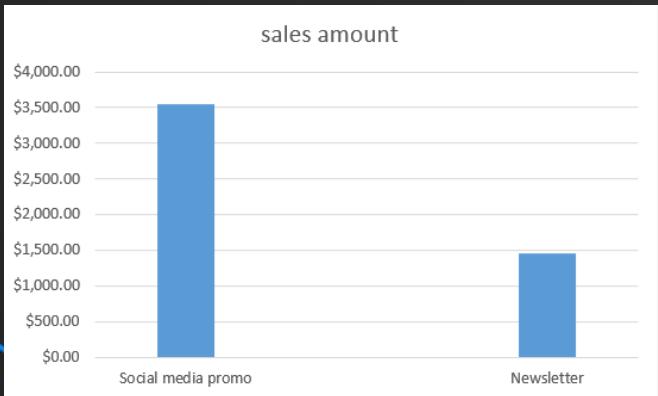
Nulls in dimensions

What we've learnt

- ✓ Nulls must be avoided in FKs

Sales_PK	Website_FK	Customer_FK	Order_id	Order_line_FK	Quantity	Unit_price	Discounted_pi	promo_FK	sales_amc	product_cost	DateTime_FK
1001	2	312	2314	34	2	22.99	22.99	-1	45.98	14.84	202204231300
1002	2	312	2314	156	3	8.99	8.99	-1	26.97	4.87	202204231300
1003	2	312	2314	643	1	16.99	14.99	3	14.99	12.53	202204231300

- ❖ Nulls in FKs break referential integrity!
- ❖ They don't appear in Joins



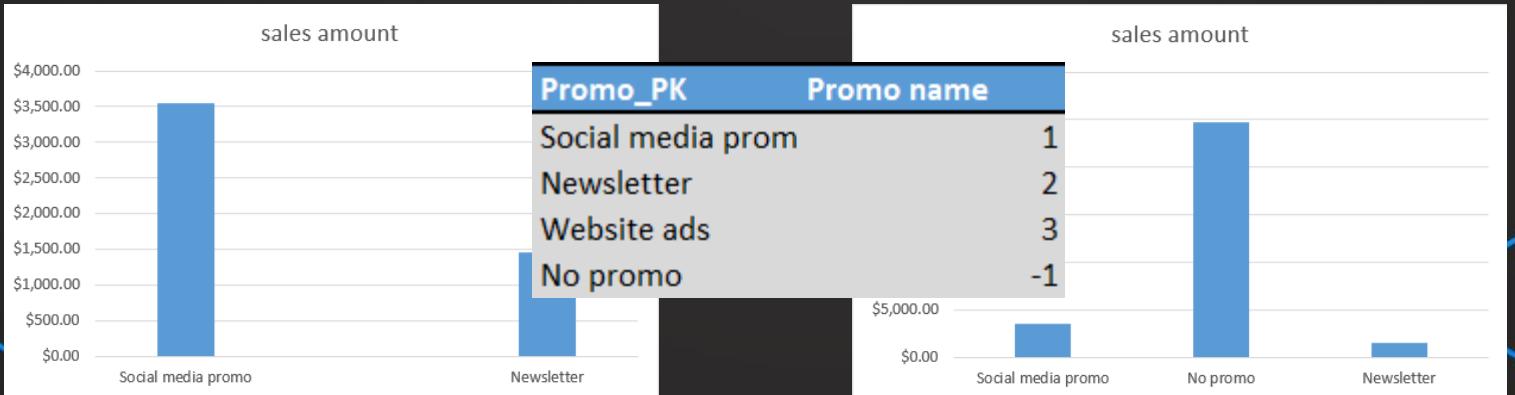
Nulls in dimensions

What we've learnt

- ✓ Nulls must be avoided in FKs

Sales_PK	Website_FK	Customer_FK	Order_id	Order_line_FK	Quantity	Unit_price	Discounted_pi	promo_FK	sales_amc	product_cost	DateTime_FK
1001	2	312	2314	34	2	22.99	22.99	-1	45.98	14.84	202204231300
1002	2	312	2314	156	3	8.99	8.99	-1	26.97	4.87	202204231300
1003	2	312	2314	643	1	16.99	14.99	3	14.99	12.53	202204231300

- ❖ Nulls in FKs break referential integrity!
- ❖ They don't appear in Joins



Nulls in dimensions

What we've learnt

- ✓ Nulls must be avoided in FKs

Sales_PK	Website_FK	Customer_FK	Order_id	Order_line_FK	Quantity	Unit_price	Discounted_pi	promo_FK	sales	amc	product_cost	DateTime_FK
1001	2	312	2314	34	2	22.99	22.99	-1	45.98	14.84	202204231300	
1002	2	312	2314	156	3	8.99	8.99	-1	26.97	4.87	202204231300	
1003	2	312	2314	643	1	16.99	14.99	3	14.99	12.53	202204231300	

- ❖ Nulls in FKs break referential integrity!
- ❖ They don't appear in Joins

Promo_PK	Promo name	
Social media prom		1
Newsletter		2
Website ads		3
No promo		-1

Date_PK	Date
20220101	1/1/2022
20220102	1/2/2022
20220103	1/3/2022
19000101	1/1/1900

Nulls in dimensions

What we've learnt

- ✓ Nulls must be avoided in FKs

Sales_PK	Website_FK	Customer_FK	Order_id	Order_line_FK	Quantity	Unit_price	Discounted_pi	promo_FK	sales	amc	product_cost	DateTime_FK
1001	2	312	2314	34	2	22.99	22.99	-1	45.98	14.84	202204231300	
1002	2	312	2314	156	3	8.99	8.99	-1	26.97	4.87	202204231300	
1003	2	312	2314	643	1	16.99	14.99	3	14.99	12.53	202204231300	

- ✓ Nulls can be present in Facts

Promo_PK	Promo name
Social media prom	1
Newsletter	2
Website ads	3
No promo	-1

Date_PK	Date
20220101	1/1/2022
20220102	1/2/2022
20220103	1/3/2022
19000101	1/1/1900

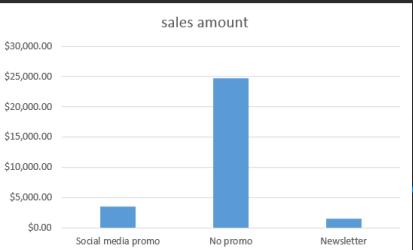
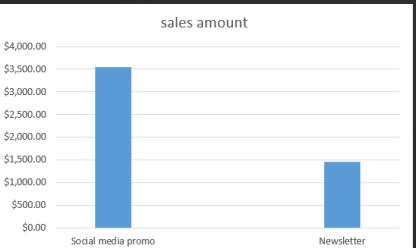
Dimensions

Nulls in dimensions

Promo_PK	Promo name	
Social media prom		1
Newsletter		2
Website ads		3
No promo		-1

Date_PK	Date
20220101	1/1/2022
20220102	1/2/2022
20220103	1/3/2022
19000101	1/1/1900

- ✓ Replace nulls with descriptive values
 - ✓ More understandable for business users
 - ✓ Values appear in aggregations in BI tools



Hierarchies in dimensions

Hierarchies in dimensions

Source data

✓ Often normalized

Product name	Category_id
Milk	1
Printer	2
Red Towel	3
Green Towel	3
Blue Towel	3

Category_id	Category
1	Groceries
2	Electronics
3	Houshold

Hierarchies in dimensions

Source data

✓ Often normalized

Product name	Category_id
Milk	1
Printer	2
Red Towel	3
Green Towel	3
Blue Towel	3

Category_id	Category
1	Groceries
2	Electronics
3	Houshold

Customer_id	Customer name	Order_id	Order_line_r	Product_FK	Category_FK
312	Franklin Miller	2314	Sunglasses SU	34	2

Hierarchies in dimensions

Source data

✓ Often normalized

Product name	Category_id
Milk	1
Printer	2
Red Towel	3
Green Towel	3
Blue Towel	3

Category_id	Category
1	Groceries
2	Electronics
3	Housheld

Customer_id	Customer name	Order_id	Order_line_r	Product_FK	Category_FK
312	Franklin Miller	2314	Sunglasses SU	34	2

✓ Snowflaked schema (should be avoided)

Hierarchies in dimensions

Source data

✓ Often normalized

Product name	Category_id
Milk	1
Printer	2
Red Towel	3
Green Towel	3
Blue Towel	3

Category_id	Category
1	Groceries
2	Electronics
3	Houshold

Some professionals have the habit to normalize data

⇒ Bad for usability & performance!

⇒ We should not do that!

Hierarchies in dimensions

What we should do

✓ Denormalize / flattened

Product name	Category_id
Milk	1
Printer	2
Red Towel	3
Green Towel	3
Blue Towel	3

Category_id	Category
1	Groceries
2	Electronics
3	Houshold

Product_ID	Product name	Category
1	Milk	Groceries
2	Printer	Electronics
3	Red Towel	Houshold
4	Green Towel	Houshold
5	Blue Towel	Houshold

Flattened dimension

Hierarchies in dimensions

What we should do

- ✓ Consider combinations if helpful

Year-Month	Year-Month	Year-Quarter
01-01-2022	Jan-2022	2022-Q1
02-01-2022	Jan-2022	2022-Q1
03-01-2022	Jan-2022	2022-Q1

Location_PK	City	State	City-State
1	Nashville	Tennessee	Nashville, Tennessee
2	Nashville	Indiana	Nashville, Indiana
3	Kansas City	Kansas	Kansas City, Kansas

Hierarchies in dimensions

Source data

✓ Often normalized

Product name	Category_id
Milk	1
Printer	2
Red Towel	3
Green Towel	3
Blue Towel	3

Category_id	Category
1	Groceries
2	Electronics
3	Housheld

Customer_id	Customer name	Order_id	Order_line_r	Product_FK	Category_FK
312	Franklin Miller	2314	Sunglasses SU	34	2

✓ Snowflaked schema (should be avoided)

Conformed dimensions

Conformed dimensions

Conformed dimension is a dimension that is shared by multiple fact tables / stars.

Used to compare facts across different fact tables.

Conformed dimension

✓ Dimension

Sales Fact

✓ Dimension

✓ Dimension

Conformed dimension

✓ Dimension

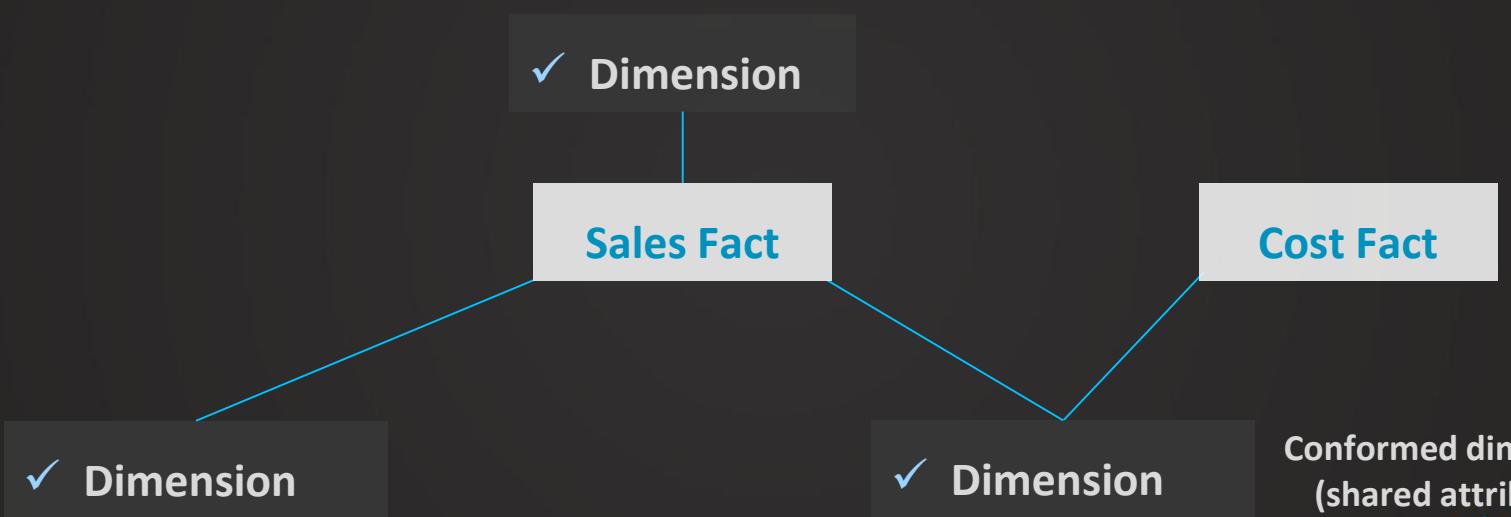
Sales Fact

Cost Fact

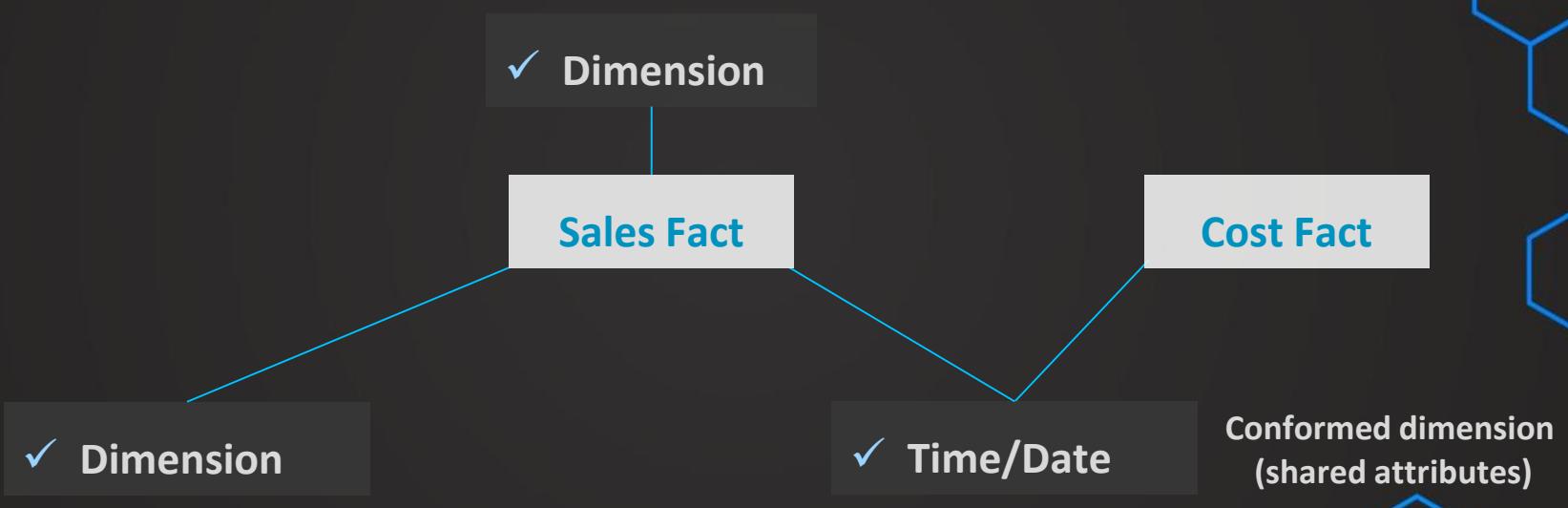
✓ Dimension

✓ Dimension

Conformed dimension



Conformed dimension



Conformed dimension
(shared attributes)

Conformed dimension

✓ Dimension

Sales Fact

Drill across

Cost Fact

✓ Dimension

✓ Time/Date

Conformed dimension
(shared attributes)

Conformed dimension

✓ Dimension

✓ Region

✓ Dimension

✓ Time/Date

Sales Fact

Drill across

Cost Fact

Conformed dimension
(shared attributes)

Conformed dimension

✓ Conformed Date dim

Month	Cost	Sales
January	\$50,300	\$67,300
February	\$55,300	\$71,400
March	\$65,100	\$79,400

✓ Conformed Region dim

Country	Cost	Sales
Spain	\$57,200	\$69,800
Belgium	\$15,300	\$21,900
wd	\$35,100	\$29,400

Conformed dimension

✓ Sales fact

Sales_PK	Sales	Date_FK
1	\$9,400	20220101
2	\$7,300	20220101
3	\$5,100	20220102

✓ Cost fact

Cost_PK	Cost	Date_FK
1	\$7,200	20220101
2	\$1,900	20220101
3	\$2,800	20220101

Conformed dimension

✓ Sales fact

Sales_PK	Sales	Date_FK
1	\$9,400	20220101
2	\$7,300	20220101
3	\$5,100	20220102

✓ Cost fact

Cost_PK	Cost	Date_FK
1	\$7,200	20220101
2	\$1,900	20220102
3	\$2,800	20220103

Same granularity not necessary!

Conformed dimension

✓ Sales fact

Sales_PK	Sales	Date_FK
1	\$9,400	20220101
2	\$7,300	20220101
3	\$5,100	20220102

✓ Cost fact

Cost_PK	Cost	DateMonth_FK
1	\$7,200	20220101
2	\$1,900	20220201
3	\$2,800	20220301

Different FK possible!

Conformed dimension

✓ Sales fact

Sales_PK	Sales	Date_FK
1	\$9,400	20220101
2	\$7,300	20220101
3	\$5,100	20220102

✓ Cost fact

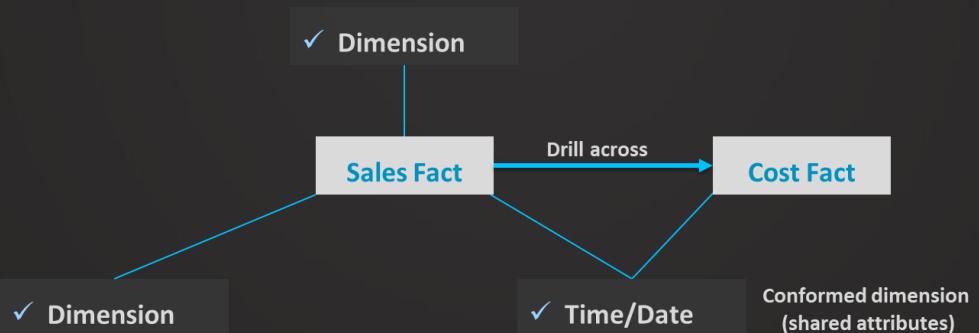
Cost_PK	Cost	DateMonth_FK
1	\$7,200	2022-01
2	\$1,900	2022-02
3	\$2,800	2022-03

Different FK possible!

Conformed dimension

✓ Conformed Date dim

Month	Cost	Sales
January	\$50,300	\$67,300
February	\$55,300	\$71,400
March	\$65,100	\$79,400



Conformed dimension
(shared attributes)

Degenerate dimension

Degenerate dimension

✓ **Transactional Sales fact**

Transaction PK	Amount	Payment FK
1	\$530	234-032
2	\$553	234-032
3	\$654	234-033

Payment PK	Header
234-032	Type A
234-033	Type A
234-034	Type B

Degenerate dimension

- ✓ **Transactional Sales fact**

Transaction PK	Amount	Payment FK
1	\$530	234-032
2	\$553	234-032
3	\$654	234-033

Payment PK	Header
234-032	Type A
234-033	Type A
234-034	Type B

- ✓ All relevant information have already
been extracted (to other dimensions)

Degenerate dimension

- ✓ **Transactional Sales fact**

Transaction PK	Amount	Payment FK	Payment PK
1	\$530	234-032	234-032
2	\$553	234-032	234-033
3	\$654	234-033	234-034

- ✓ All relevant information have already
been extracted (to other dimensions)
- ✓ Attribute can be still useful

Degenerate dimension

- ✓ **Transactional Sales fact**

Transaction PK	Amount	Payment DD
1	\$530	234-032
2	\$553	234-032
3	\$654	234-033

- ✓ All relevant information have already been extracted (to other dimensions)
- ✓ Attribute can be still useful
- ✓ Indicate that it is a deg. dim. (e.g. _DD)

Degenerate dimension

- ✓ Degenarate dimension the dimension key without an associated dimension

Transaction_PK	Amount	Payment_DD
1	\$530	234-032
2	\$553	234-032
3	\$654	234-033

Occuring mostly in Transactional facts

Invoice no., billing no. or order_id
typically are degenerate dimensions

Junk dimensions

Junk dimensions

Transaction_PK	Amount	Payment_Type	Incoming / Outbound	Is_Bonus
1	\$530	Wired	Incoming	Yes
2	\$553	Credit Card	Outbound	No
3	\$654	Cash	Incoming	No

1. Eliminate them if they are not relevant
2. Leave them as they are in the fact
3. One Flag => One dimension

What if they are relevant?

Long text values? Table size?

Very wide fact table?

Alternative: Junk dimension

Junk dimensions

What is a junk dimension?

Dimension with various flags / indicators
with low cardinality

Junk dimensions

What is a junk dimension?

Like a box were we store items we need but
have no separate storing location.

Junk dimensions

Note:

We call it "*junk dimension*" usually only internally.

Talking to business users we can refer to as

"transactional indicator dimension".

Junk dimensions

Transaction_PK	Amount	Payment_Type	Incoming / Outbound	Is_Bonus
1	\$530	Wired	Incoming	Yes
2	\$553	Credit Card	Outbound	No
3	\$654	Cash	Incoming	No

Transaction_PK	Amount	Transactional_Flag_FK
1	\$530	1
2	\$553	7
3	\$654	12

Flag_PK	Payment_Type	Incoming / Outbound	Is_Bonus
1	Wired	Incoming	Yes
2	Wired	Incoming	No
3	Wired	Outbound	Yes
4	Wired	Outbound	No

Junk dimensions

Payment_Type	Amount
Wired	\$5350
Credit Card	\$6553
Cash	\$6754

Is_Bonus	Amount
Yes	\$9350
No	\$11857

Junk dimensions

Number of combinations

$$3 \times 2 \times 2 = 12$$

Transaction_PK	Amount	Payment_Type	Incoming / Outbound	Is_Bonus
1	\$530	Wired	Incoming	Yes
2	\$553	Credit Card	Outbound	No
3	\$654	Cash	Incoming	No

Flag_PK	Payment_Type	Incoming / Outbound	Is_Bonus
1	Wired	Incoming	Yes
2	Credit Card	Outbound	No
3	Cash	Incoming	No
...			
12	Cash	Outbound	No

Junk dimensions

Many dimensions?

Many combinations!

9 indicators with 4 combinations

$$4^9 = 262144$$

1. Extract only available combinations of fact table
2. Two or more junk dimensions

$$4^5 = 1024$$

Role-playing dimension

Role-playing dimension

What is a role-playing dimension?

Dimension that is referenced multiple times
by a fact

Role-playing dimension

order_id	Date FK	Measure		Date FK
	Order Date FK	No. Products	Product_FK	Production Start FK
1	20220102	100	32	20220103
2	20220103	100	32	20220104
3	20220103	100	32	20220103
4	20220104	100	32	20220106
5	20220104	100	32	20220108

Role 1

Role 2

Date_PK	Date	Month	Short Month	Year-Quarter	Year	Weekday	Is_Weekend
20220101	2022-01-01	January	Jan	2022-Q1	2022	Saturday	1
20220102	2022-01-02	January	Jan	2022-Q1	2022	Sunday	1
20220103	2022-01-03	January	Jan	2022-Q1	2022	Monday	0

Role-playing dimension

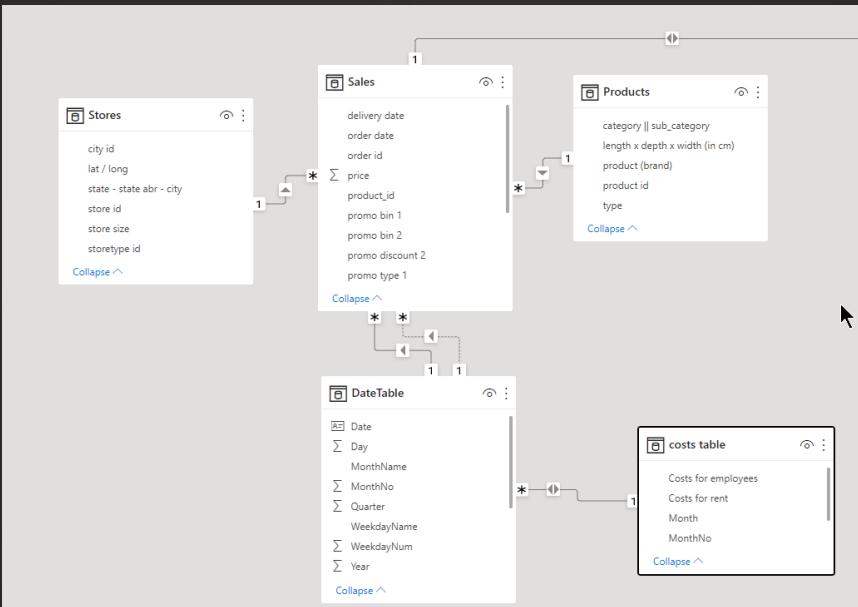
Month	Products (Orders received)
January	2500
February	2700
...	...

Month	Products (Production started)
January	2650
February	2450
...	...

order_id	Order Date FK	No. Products	Product_FK	Production Start FK
1	20220102	100	32	20220103
2	20220103	100	32	20220104
3	20220103	100	32	20220103
4	20220104	100	32	20220106
5	20220104	100	32	20220108

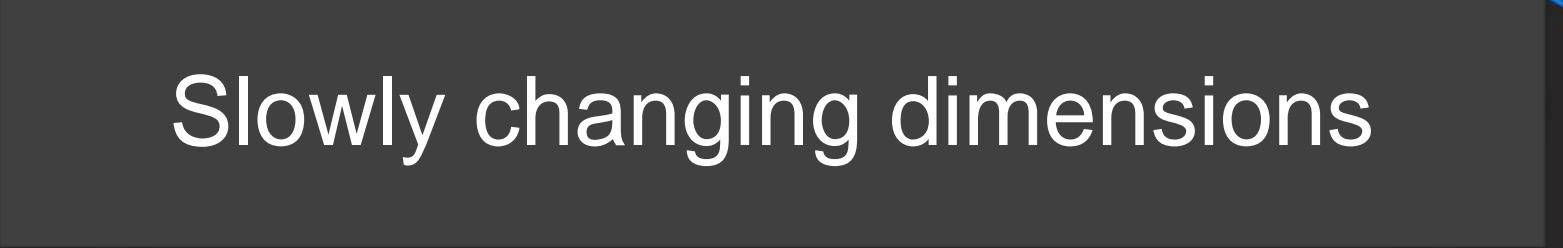
Role-playing dimension

- ✓ BI tools for example via active & inactive relationships



Role-playing dimension

- ✓ For analysis in SQL you can create *additional view for each role*
- ✓ No duplicated data but still we it appears like a separate dimension



Slowly changing dimensions

Slowly changing dimensions

Till now we have pretended dimensions never change...

... indeed they are rather static usually ...

... but surprise... they do change in the real world...

Develop a strategy to handle changes in dimensions...

Slowly changing dimensions

1. *Be proactive: Ask about potential changes*
2. *Business users + IT*
3. *Strategy for each changing attribute*

Kimball introduced SCD in 1995 and distinguished between different types (1, 2, 3, ...).

Type 0: Retain Original

Type 0: Retain Original

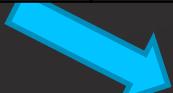
- ✓ *There won't be any changes*
- ✓ *Date Table (expect for holidays etc.)*
- ✓ *"Original"*
- ✓ *Very simple and easy to maintain*

Type 1: Overwrite

Type 1: Overwrite

- ✓ *Old attributes are just overwritten*
- ✓ *Only current state is reflected*

Product_Key	Name	Category
1	Sunglasses TR-7	Assecoirs
2	Chocolate bar 70% cacao	Sweets
3	Oat meal biscuits	Sweets



UPDATE

Product_Key	Name	Category
1	Sunglasses TR-7	Assecoirs
2	Chocolate bar 70% cacao	Sweets
3	<i>Delicious</i> Oat meal biscuits	<i>Buscuits</i>

Type 1: Overwrite

- ✓ *Very simple*
- ✓ *No Fact table needs to be modified*

Product_Key	Name	Category
1	Sunglasses TR-7	Assecoirs
2	Chocolate bar 70% cacao	Sweets
3	Oat meal biscuits	Sweets

UPDATE

Product_Key	Name	Category
1	Sunglasses TR-7	Assecoirs
2	Chocolate bar 70% cacao	Sweets
3	<i>Delicious</i> Oat meal biscuits	<i>Buscuits</i>

Type 1: Overwrite

- ✓ *Very simple*
- ✓ *No Fact table needs to be modified*

Product_Key	Name	Category
1	Sunglasses TR-7	Assecoirs
2	Chocolate bar 70% cacao	Sweets
3	Oat meal biscuits	Sweets

UPDATE

Product_Key	Name	Category
1	Sunglasses TR-7	Assecoirs
2	Chocolate bar 70% cacao	Sweets
3	<i>Delicious</i> Oat meal biscuits	<i>Buscuits</i>

Not so significant

Problem

- ❖ *History is lost!*
- ❖ *Insignificant changes*
- ❖ *Might affect / break existing queries*

More significant

Type 2: New row

Type 2: New row

- ✓ Problem with Type 1: *No history of dimensions!*
- ✓ Only current state is reflected

Product_Key	Name	Category
1	Sunglasses TR-7	Assecoirs
2	Chocolate bar 70% cacao	Sweets
3	Oat meal biscuits	Sweets

UPDATE

Product_Key	Name	Category
1	Sunglasses TR-7	Assecoirs
2	Chocolate bar 70% cacao	Sweets
3	<i>Delicious</i> Oat meal biscuits	<i>Buscuits</i>

Type 2: New row

- ✓ Problem with Type 1: *No history of dimensions!*
- ✓ Only current state is reflected

Sales_Key	Name	Amount
1	Sunglasses TR-7	\$25
2	Chocolate bar 70% cacao	\$3
3	Oat meal biscuits	\$4
4	Chocolate bar 70% cacao	\$3
5	Oat meal biscuits	\$4

Before



Category	Amount
Assecoirs	\$25
Sweets	\$14

After



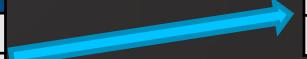
Category	Amount
Assecoirs	\$25
Sweets	\$6
Buscuits	\$8

Type 2: New row

- ✓ Problem with Type 1: *No history of dimensions!*
- ✓ Only current state is reflected

Sales_Key	Name	Amount
1	Sunglasses TR-7	\$25
2	Chocolate bar 70% cacao	\$3
3	Oat meal biscuits	\$4
4	Chocolate bar 70% cacao	\$3
5	Oat meal biscuits	\$4

Before



Category	Amount
Assecoirs	\$25
Sweets	\$14

After



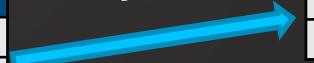
Category	Amount
Assecoirs	\$25
Sweets	\$6
Buscuits	\$8

Type 2: New row

- ✓ Problem with Type 1: *No history of dimensions!*
- ✓ Only current state is reflected

Sales_Key	Name	Amount
1	Sunglasses TR-7	\$25
2	Chocolate bar 70% cacao	\$3
3	Oat meal biscuits	\$4
4	Chocolate bar 70% cacao	\$3
5	Oat meal biscuits	\$4

Before



Category	Amount
Assecoirs	\$25
Sweets	\$14

After



Category	Amount
Assecoirs	\$25
Sweets	\$6
Buscuits	\$8

Type 2: New row

- ✓ Problem with Type 1: *No history of dimensions!*
- ✓ Only current state is reflected

Sales_Key	Name	Amount
1	Sunglasses TR-7	\$25
2	Chocolate bar 70% cacao	\$3
3	Oat meal biscuits	\$4
4	Chocolate bar 70% cacao	\$3
5	Oat meal biscuits	\$4

Before

Category	Amount
Assecoirs	\$25
Sweets	\$14

Correctly representing history

After

Category	Amount
Assecoirs	\$25
Sweets	\$6
Buscuits	\$8

Type 2: New row

- ✓ Type 2: Perfectly partitions history
- ✓ Changes are reflected with history

Sales_Key	Name	Amount
1	Sunglasses TR-7	\$25
2	Chocolate bar 70% cacao	\$3
3	Oat meal biscuits	\$4
4	Chocolate bar 70% cacao	\$3
5	Oat meal biscuits	\$4

Before

Correctly representing history

After

Category	Amount
Assecoirs	\$25
Sweets	\$14

Category	Amount
Assecoirs	\$25
Sweets	\$6
Buscuits	\$8

Default strategy

Category	Amount
Assecoirs	\$25
Sweets	\$7
Buscuits	\$4

Type 2: New row

Product Key	Name	Category
1	Sunglasses TR-7	Accessories
2	Chocolate bar 70% cacao	Sweets
3	Oat meal biscuits	Sweets

UPDATE

Product Key	Name	Category
1	Sunglasses TR-7	Accessories
2	Chocolate bar 70% cacao	Sweets
3	<i>Delicious</i> Oat meal biscuits	<i>Biscuits</i>

Type 2: New row

Product_Key	Name	Category
1	Sunglasses TR-7	Assecoirs
2	Chocolate bar 70% cacao	Sweets
3	Oat meal biscuits	Sweets

UPDATE

Product_Key	Name	Category
1	Sunglasses TR-7	Assecoirs
2	Chocolate bar 70% cacao	Sweets
3	<i>Delicious</i> Oat meal biscuits	<i>Buscuits</i>

Product_Key	Name	Category
1	Sunglasses TR-7	Assecoirs
2	Chocolate bar 70% cacao	Sweets
3	Oat meal biscuits	Sweets

Add Row

Product_Key	Name	Category
1	Sunglasses TR-7	Assecoirs
2	Chocolate bar 70% cacao	Sweets
3	Oat meal biscuits	Sweets
4	<i>Delicious</i> Oat meal biscuits	<i>Buscuits</i>

Type 2: New row

Sales_Key	Name	Product_FK	Amount
1	Sunglasses TR-7	1	\$25
2	Chocolate bar 70% cacao	2	\$3
3	Oat meal biscuits	3	\$4
4	Chocolate bar 70% cacao	2	\$3
5	Oat meal biscuits	4	\$4

Product_Key	Name	Category
1	Sunglasses TR-7	Assecoirs
2	Chocolate bar 70% cacao	Sweets
3	Oat meal biscuits	Sweets
4	Delicious Oat meal biscuits	Buscuits

Respecting history

Category	Amount
Assecoirs	\$25
Sweets	\$3
Buscuits	\$4

- No updates in fact
- From that moment new FK

Type 2: New row

No . of products?

Product_PK	Product_ID	Name	Category
1	SG-TR7	Sunglasses TR-7	Assecoirs
2	CH-B70	Chocolate bar 70% cacao	Sweets
3	OT-BSC	Oat meal biscuits	Sweets
4	OT-BSC	Delicious Oat meal biscuits	Buscuits

Count distinct Product_ID

All current products?

Administrate Type 2 SCD

Administrate Type 2 SCD

Product PK	Product ID	Name	Category	Ef Date	Ex Date
1	SG-TR7	Sunglasses TR-7	Assecoirs	2022-01-01	2100-01-01
2	CH-B70	Chocolate bar 70% cacao	Sweets	2022-01-01	2100-01-01
3	OT-BSC	Oat meal biscuits	Sweets	2022-01-01	2022-05-31
4	OT-BSC	Delicious Oat meal biscuits	Buscuits	2022-06-01	2100-01-01

*Period in which
values are valid*

*Instead of null better
date far in the future*

- ✓ Necessary also in ETL to use correct FK
- ✓ Requires Surrogate key instead of Natural key

Administrate Type 2 SCD

Correct FK?

Product PK	Product ID	Name	Category	Ef Date	Ex Date
1	SG-TR7	Sunglasses TR-7	Assecoirs	2022-01-01	2100-01-01
2	CH-B70	Chocolate bar 70% cacao	Sweets	2022-01-01	2100-01-01
3	OT-BSC	Oat meal biscuits	Sweets	2022-01-01	2022-05-31
4	OT-BSC	Delicious Oat meal biscuits	Buscuits	2022-06-01	2100-01-01

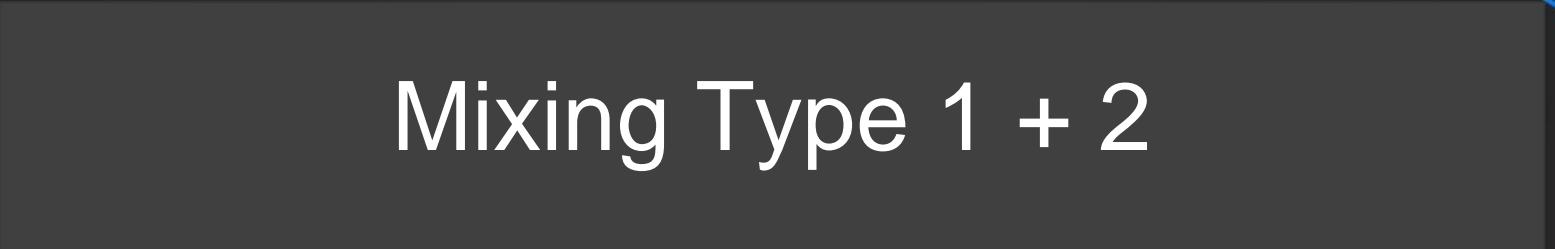
- ✓ Add row in the Dimension first
- ✓ Lookup in the Dimension with Natural key + Ef_/Ex_Date

Administrate Type 2 SCD

Correct FK?

Product_PK	Product_ID	Name	Category	Ef_Date	Ex_Date	Is_Current
1	SG-TR7	Sunglasses TR-7	Assecoirs	2022-01-01	2100-01-01	Yes
2	CH-B70	Chocolate bar 70% cacao	Sweets	2022-01-01	2100-01-01	Yes
3	OT-BSC	Oat meal biscuits	Sweets	2022-01-01	2022-05-31	No
4	OT-BSC	Delicious Oat meal biscuits	Buscuits	2022-06-01	2100-01-01	Yes

- ✓ Add row in the Dimension first
- ✓ Lookup in the Dimension with Natural key + Ef_/Ex_Date
- ✓ Additional for Is_Current



Mixing Type 1 + 2

Mixing Type 1 + 2

Product_PK	Product_ID	Name	Category	Ef_Date	Ex_Date
1	SG-TR7	Sunglasses TR-7	Assecoirs	2022-01-01	2100-01-01
2	CH-B70	Chocolate bar 70% cacao	Sweets	2022-01-01	2100-01-01
3	OT-BSC	Oat meal biscuits	Sweets	2022-01-01	2100-01-01
4	OT-BSC	Delicious Oat meal biscuits	Buscuits	2022-06-01	2100-01-01

✓ Some attributes can be Type 1 and some Type 2

Mixing Type 1 + 2

Product_PK	Product_ID	Name	Category	Ef_Date	Ex_Date
1	SG-TR7	Sunglasses TR-7	Assecoirs	2022-01-01	2100-01-01
2	CH-B70	Chocolate bar 70% cacao	Sweets	2022-01-01	2100-01-01
3	OT-BSC	Oat meal biscuits	Sweets	2022-01-01	2100-01-01
4	OT-BSC	Delicious Oat meal biscuits	Buscuits	2022-06-01	2100-01-01

- ✓ No set in stone rules but needs to be defined with business users
- ✓ *Not a technical decision*

Type 3: Additional Attributes

Type 3: Additonal Attributes

Product_PK	Product_ID	Name	Category	Ef_Date	Ex_Date
1	SG-TR7	Sunglasses TR-7	Assecoirs	2022-01-01	2100-01-01
2	CH-B70	Chocolate bar 70% cacao	Sweets	2022-01-01	2100-01-01
3	OT-BSC	Oat meal biscuits	Sweets	2022-01-01	2100-01-01
4	OT-BSC	Delicious Oat meal biscuits	Buscuits	2022-06-01	2100-01-01

- ✓ *Type 2 – Default strategy to maintain reflect history*
- ✓ *Type 1 – Static*
- ✓ *Type 3 – In-between: Switching back & forth between versions*

Type 3: Additonal Attributes

Product PK	Product ID	Name	Category	Prev Category
1	SG-TR7	Sunglasses TR-7	Assecoirs	Assecoirs
2	CH-B70	Chocolate bar 70% cacao	Sweets	Sweets
3	OT-BSC	Oat meal biscuits	Biscuit	Sweets

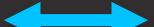
- ✓ *Instead of adding a row – we add a column*

Type 3: Additional Attributes

Product PK	Product ID	Name	Category	Prev Category
1	SG-TR7	Sunglasses TR-7	Assecoirs	Assecoirs
2	CH-B70	Chocolate bar 70% cacao	Sweets	Sweets
3	OT-BSC	Oat meal biscuits	Biscuit	Sweets

- ✓ *Instead of adding a row – we add a column*

Category	Amount
Assecoirs	\$25
Sweets	\$6
Buscuits	\$8



Prev Category	Amount
Assecoirs	\$25
Sweets	\$14

Type 3: Additonal Attributes

Product PK	Product ID	Name	Category	Prev Category
1	SG-TR7	Sunglasses TR-7	Assecoirs	Assecoirs
2	CH-B70	Chocolate bar 70% cacao	Sweets	Sweets
3	OT-BSC	Oat meal biscuits	Biscuit	Sweets

- ✓ *Instead of adding a row – we add a column*
- ✓ *Typically used for significant changes at a time
(e.g. restructurings in organizations)*

Type 3: Additonal Attributes

Sales_Key	Name	Region_FK	Amount
1	Sunglasses TR-7	1	\$25
2	Chocolate bar 70% cacao	2	\$3
3	Oat meal biscuits	3	\$4
4	Chocolate bar 70% cacao	2	\$3
5	Oat meal biscuits	3	\$4

Reg_PK	Region	Prev_Region
1	North	North
2	West	West
3	South	West

- ✓ *Instead of adding a row – we add a column*
- ✓ *Typically used for significant changes at a time
(e.g. restructurings in organizations)*
- ✓ *Enables switching between historic / current view*

Type 3: Additional Attributes

Sales_Key	Name	Region_FK	Amount
1	Sunglasses TR-7	1	\$25
2	Chocolate bar 70% cacao	2	\$3
3	Oat meal biscuits	3	\$4
4	Chocolate bar 70% cacao	2	\$3
5	Oat meal biscuits	3	\$4

Reg_PK	Region	Prev_Region
1	North	North
2	West	West
3	South	West
4	East	Not applicable

Limitations

- ❖ Not suitable for frequent or unpredictable changes => better Type 2
- ❖ Minor changes => better Type 1

- ✓ *New attributes => New rows*
- ✓ *It is possible to add multiple historic columns*

Least frequent type

Type 3: Additional Attributes

Sales_Key	Name	Region_FK	Amount
1	Sunglasses TR-7	1	\$25
2	Chocolate bar 70% cacao	2	\$3
3	Oat meal biscuits	3	\$4
	Chocolate bar 70% cacao	2	\$3
	Oat meal biscuits	3	\$4

Reg_PK	Region	Prev_Region
1	North	North
2	West	West
3	South	West

Limitations

- ✓ *Not suitable for frequent or unpredictable changes*
=> better Type 2
- ✓ *It is possible to add multiple historic columns*
- ✓ *Minor changes => better Type 1*

Least frequent type
(two versions needed)

What is an ETL?

What is an ETL?

- ✓ How to design dimensional model
- ✓ How to bring data from source to DWH

= ETL process

Data Warehouse Layers



Other data sources

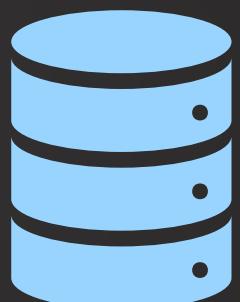


Sales data



CRM system

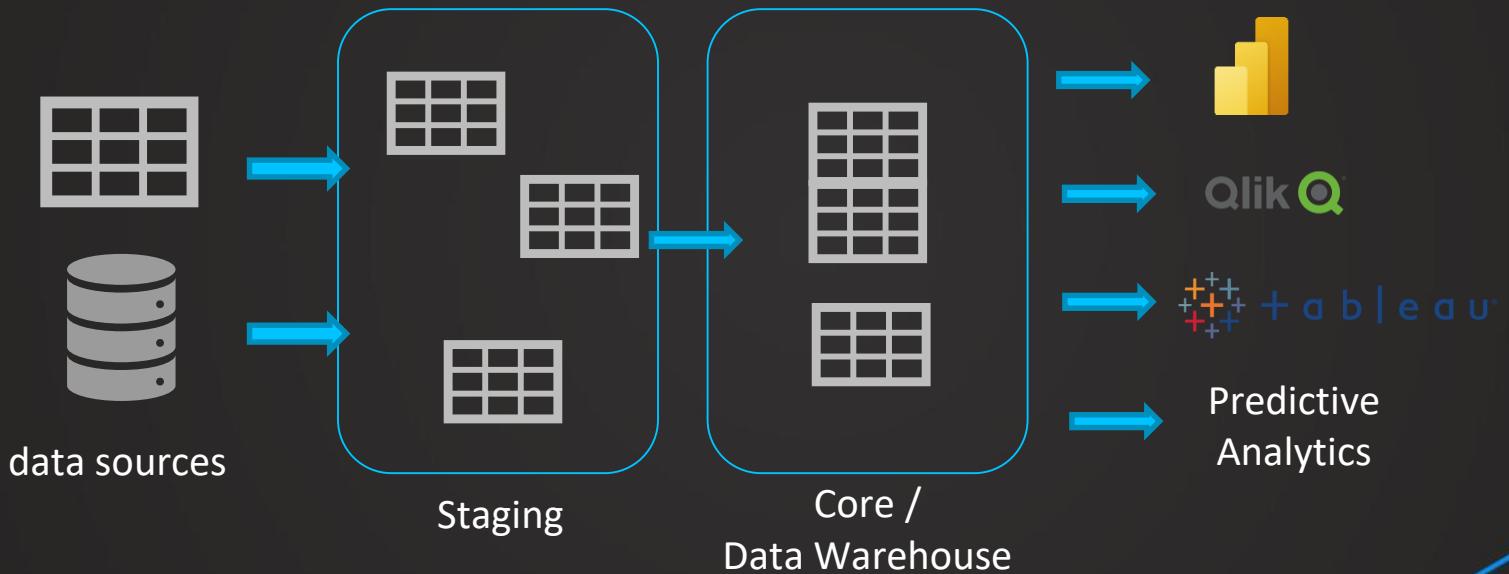
Extract, Transform, Load



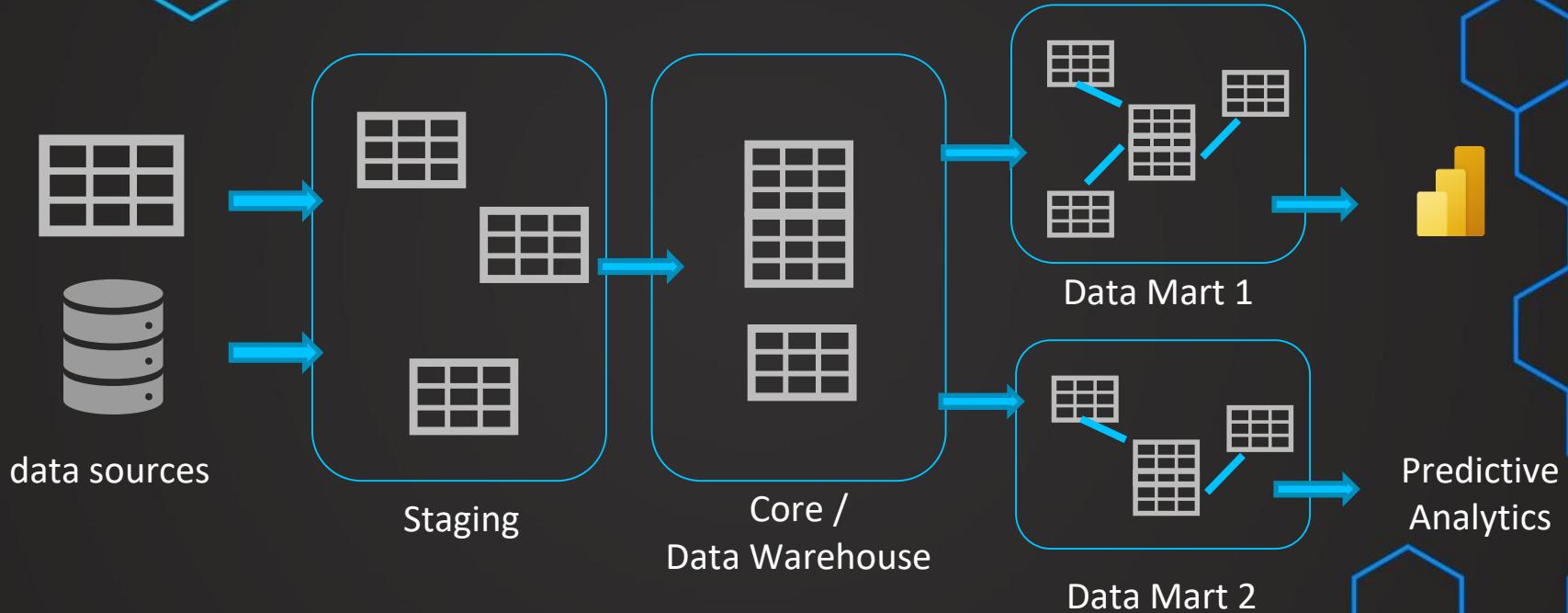
Data warehouse

Centralized
location for data

Data Warehouse Layers



Data Warehouse Layers



Extract, Transform, Load



ETL-Tool

Set of (built-in) tools to...

- ✓ Connect to different data sources
- ✓ Transform / Clean data
- ✓ Load data

Everything we need to build our DWH!

Extract, Transform, Load



ETL-Setup

Building workflows...

- ✓ **Staging workflow**
- ✓ **Core / Transformation workflow**
- ✓ **Data Mart workflow**

Extract, Transform, Load



ETL-Setup

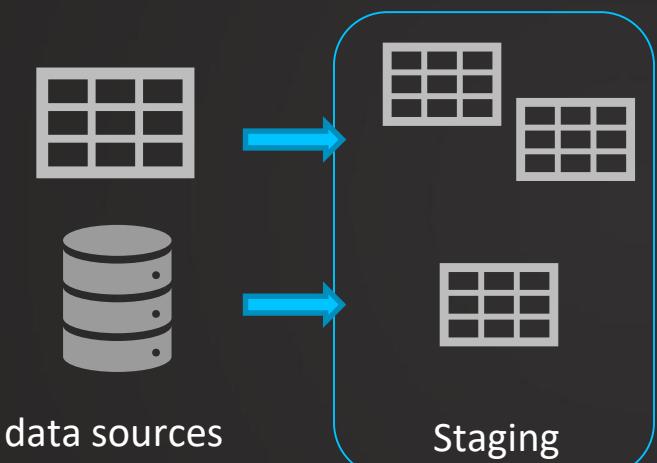
Jobs ...

- ✓ Run the workflows
- ✓ Are scheduled based on defined rules



Extract

Extracting



- ✓ **Data is part of DWH**
- ✓ **Understanding data**
- ✓ **From here data is transformed**
- ✓ **Transient (most commonly)**
- ✓ **All data copied and then deleted**

Extracting types

Initial Load

- ✓ First (real) run
- ✓ All data

Delta Load

- ✓ Subsequent runs
- ✓ Only additional data

Initial Load

Initial Load

- ✓ First initial extraction from source data
- ✓ After discussion with the business users + IT
 - What data is needed
 - When is a good time to load the data
(Night? Weekends?)
 - Smaller extractions to test

Initial Load

- ✓ **Initial Load to Core with Transformations**
- ✓ **After all the transformation steps have been designed**
- ✓ **Just done for all data from Staging (no filtering)**

Delta Load

Same structure

Delta Load

- ✓ Incremental periodic Extraction / Load
- ✓ Delta column for every table

Sales_Date	Name	Amount
2022-06-06	Sunglasses TR-7	\$25
2022-06-06	Chocolate bar 70% cacao	\$3
2022-06-07	Oat meal biscuits	\$4
2022-06-07	Chocolate bar 70% cacao	\$3
2022-06-08	Oat meal biscuits	\$4

- ✓ Transaction date, create_date, etc.

Delta Load

- ✓ Incremental periodic Extraction / Load
- ✓ Delta column for every table

Sales_Key	Name	Amount
1	Sunglasses TR-7	\$25
2	Chocolate bar 70% cacao	\$3
3	Oat meal biscuits	\$4
4	Chocolate bar 70% cacao	\$3
5	Oat meal biscuits	\$4

- ✓ Incrementing number (Suitable primary key)

Delta Load

- ✓ Incremental periodic Extraction / Load

Sales_Key	Name	Amount
1	Sunglasses TR-7	\$25
2	Chocolate bar 70% cacao	\$3
3	Oat meal biscuits	\$4
4	Chocolate bar 70% cacao	\$3
5	Oat meal biscuits	\$4

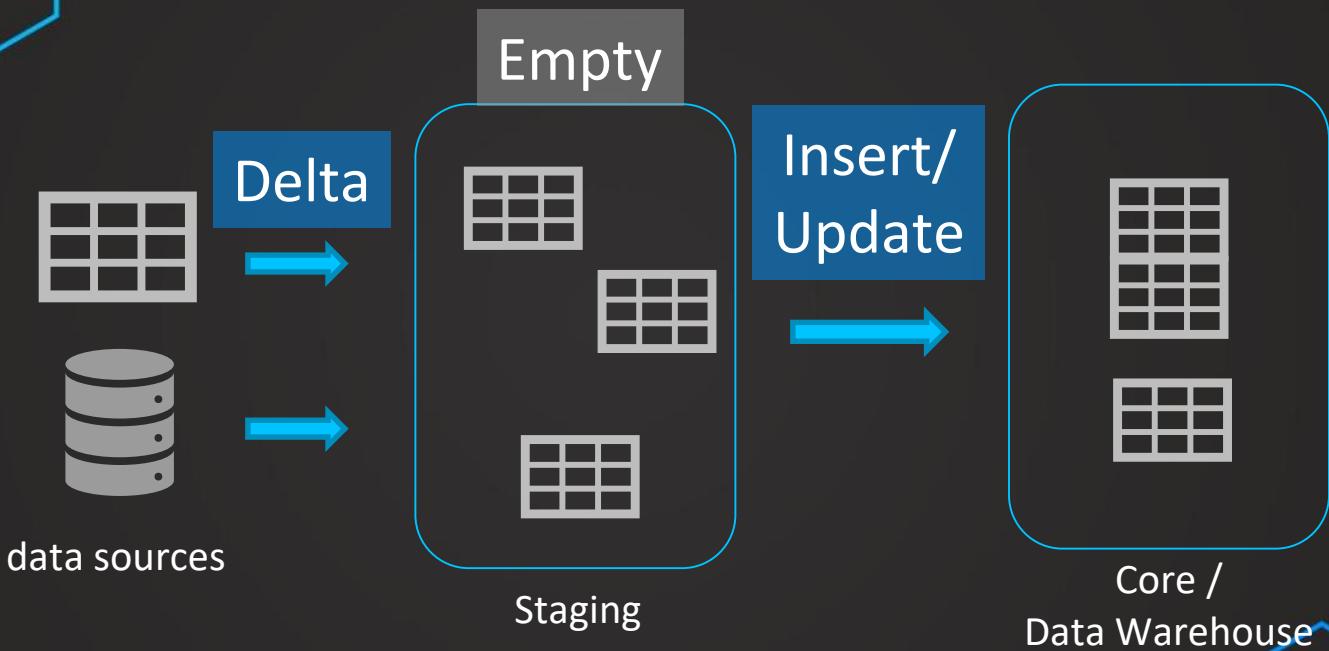
- ✓ Remember MAX(Sales_Key)
- ✓ MAX (Sales_Key) -> Variable X
- ✓ Next run: Sales_Key > X

What if there is no delta column?

- ✓ Some tools can capture automatically which data has been already loaded
- ✓ Just full load everytime and compare the data with data that is already loaded
- ✓ Depending on the data volumes -> performance

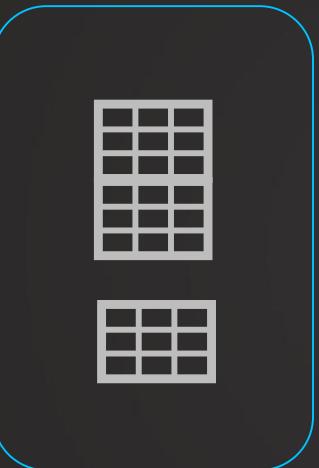
Load (Insert/Update)

Data Warehouse Layers



Data Warehouse Layers

Insert/
Update



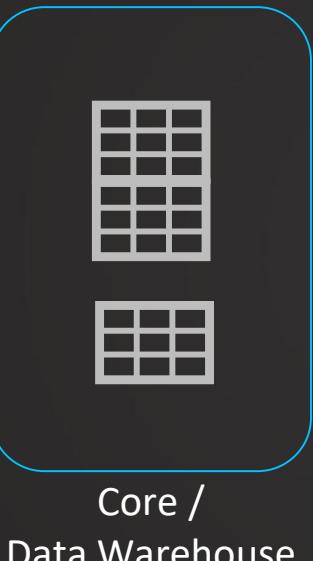
Core /
Data Warehouse

- **INSERT / APPEND**
- **UPDATE**

Product_PK	Name
1	Sunglasses TR-7
2	Chocolate bar 70% cacao
3	Oat meal biscuits
4	Chocolate bar 70% cacao
5	Oat meal biscuits

Data Warehouse Layers

Insert/
Update



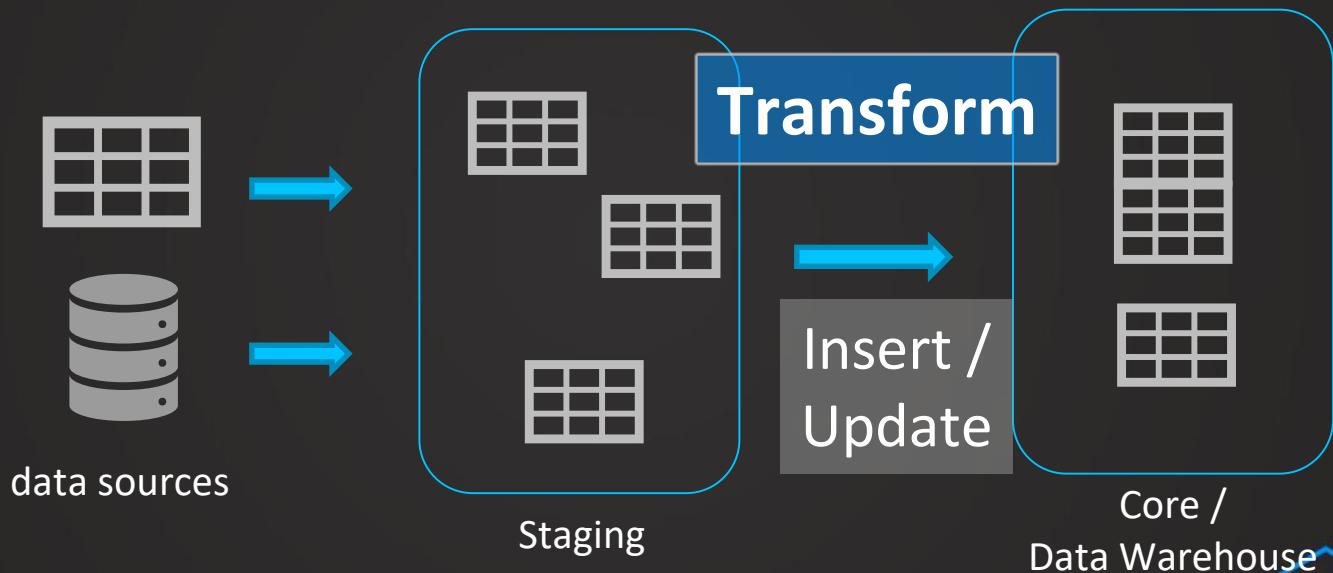
- **DELETE**

Product PK	Name	Deleted
1	Sunglasses TR-7	No
2	Chocolate bar 70% cacao	Yes
3	Oat meal biscuits	No
4	Chocolate bar 70% cacao	No
5	Oat meal biscuits	No

- **Typically we don't delete data**

Transform

Data Warehouse Layers



Main goals

Create a consolidated view of all data for
analysis purposes

1. **Consolidate** (from multiple systems)
2. **Reshape** (for analysis purposes)

Main goals

1. Consolidate (from multiple systems)

Transaction_ID	Amount	Date
T1	\$5030	10/1/2022
T2	\$5053	11/1/2022
T3	\$654	12/1/2022

Transaction_ID	Amount (in thousands)	Transaction_Date
T14	\$5.345	10-1-2022
T15	\$7.953	11-1-2022
T16	\$9.654	12-1-2022

Making the data compatible & consistent!

Main goals

2. Reshape according to business requirements

Transaction_ID	Amount	Date
T1	\$5030	10/1/2022
T2	\$5053	11/1/2022
T3	\$654	12/1/2022



Transaction_PK	Amount	Date_FK
1	\$5030	20220110
2	\$5053	20220111
3	\$654	20220112

Main goals

2. Reshape according to business requirements

Month	Januar-2022	February-2022	March-2022	Total
Amount	\$5030	\$6053	\$2455	\$13548



Month	Amount
Januar-2022	\$5030
February-2022	\$6053
March-2022	\$2455
Total	\$13548



Month	Amount
Januar-2022	\$5030
February-2022	\$6053
March-2022	\$2455

Clean & reshape data

Kinds of transformations

- **Deduplication**
- **Filtering (rows & columns)**
- **Cleaning & Mapping (Integration)**
- **Value Standardization (Integration)**
- **Key Generation**

Kinds of transformations

Basic

- Deduplication
- Filtering (rows & columns)
- Cleaning & Mapping (Integration)
- Value Standardization (Integration)
- Key Generation

Advanced

- Joining
- Splitting
- Aggregating
- Deriving new values

Basic

Kinds of transformations

- Deduplication

Store 1

product_id	name	category
P521	Almonds 150g	Nuts
P252	Garlic	Fruits & Vegetables
P533	Banana	Fruits & Vegetables
P684	Chocolate	Sweets & Snacks
P755	Spicy Chips	Sweets & Snacks

Store 2

product_id	name	category
P521	Almonds 150g	Nuts
P672	Orange Juice	Drinks
P423	Green Apples	Fruits & Vegetables
P564	Chocolate Cookies	Sweets & Snacks
P755	Spicy Chips	Sweets & Snacks

Basic

Kinds of transformations

- Deduplication

Product Dimension

product_id	name	category
P521	Almonds 150g	Nuts
P252	Garlic	Fruits & Vegetables
P533	Banana	Fruits & Vegetables
P684	Chocolate	Sweets & Snacks
P755	Spicy Chips	Sweets & Snacks
P521	Almonds 150g	Nuts
P672	Orange Juice	Drinks
P423	Green Apples	Fruits & Vegetables
P564	Chocolate Cookies	Sweets & Snacks
P755	Spicy Chips	Sweets & Snacks

Basic

Kinds of transformations

- Deduplication

Product Dimension

product_id	name	category
P521	Almonds 150g	Nuts
P252	Garlic	Fruits & Vegetables
P533	Banana	Fruits & Vegetables
P684	Chocolate	Sweets & Snacks
P755	Spicy Chips	Sweets & Snacks
P521	Almonds 150g	Nuts
P672	Orange Juice	Drinks
P423	Green Apples	Fruits & Vegetables
P564	Chocolate Cookies	Sweets & Snacks
P755	Spicy Chips	Sweets & Snacks

Basic

Kinds of transformations

- Deduplication

Product Dimension

product_id	name	category
P521	Almonds 150g	Nuts
P252	Garlic	Fruits & Vegetables
P533	Banana	Fruits & Vegetables
P684	Chocolate	Sweets & Snacks
P755	Spicy Chips	Sweets & Snacks
P672	Orange Juice	Drinks
P423	Green Apples	Fruits & Vegetables
P564	Chocolate Cookies	Sweets & Snacks

Basic

Kinds of transformations

- **Filtering rows**

Filter out irrelevant rows

Sales_Date	Name	Amount	Type
2022-06-06	Sunglasses TR-7	\$25	Sale
2022-06-06	Chocolate bar 70% cacao	\$3	Sale
2022-06-06	Sunglasses TR-7	\$-25	Refund
2022-06-07	Oat meal biscuits	\$4	Sale
2022-06-07	Chocolate bar 70% cacao	\$3	Sale
2022-06-08	Oat meal biscuits	\$4	Sale

Basic

Kinds of transformations

- **Filtering rows**

Filter out irrelevant rows

Sales_Date	Name	Amount	Type
2022-06-06	Sunglasses TR-7	\$25	Sale
2022-06-06	Chocolate bar 70% cacao	\$3	Sale
2022-06-06	Sunglasses TR-7	\$-25	Refund
2022-06-07	Oat meal biscuits	\$4	Sale
2022-06-07	Chocolate bar 70% cacao	\$3	Sale
2022-06-08	Oat meal biscuits	\$4	Sale

Basic

Kinds of transformations

- **Filtering rows**

Filter out irrelevant rows

Sales_Date	Name	Amount	Type
2022-06-06	Sunglasses TR-7	\$25	Sale
2022-06-06	Chocolate bar 70% cacao	\$3	Sale
2022-06-07	Oat meal biscuits	\$4	Sale
2022-06-07	Chocolate bar 70% cacao	\$3	Sale
2022-06-08	Oat meal biscuits	\$4	Sale

Basic

Kinds of transformations

- Filtering columns

Filter out irrelevant columns

Sales_Date	Name	Amount	Type
2022-06-06	Sunglasses TR-7	\$25	Sale
2022-06-06	Chocolate bar 70% cacao	\$3	Sale
2022-06-07	Oat meal biscuits	\$4	Sale
2022-06-07	Chocolate bar 70% cacao	\$3	Sale
2022-06-08	Oat meal biscuits	\$4	Sale

Basic

Kinds of transformations

- **Filtering columns**

Filter out irrelevant columns

Sales Date	Name	Amount
2022-06-06	Sunglasses TR-7	\$25
2022-06-06	Chocolate bar 70% cacao	\$3
2022-06-07	Oat meal biscuits	\$4
2022-06-07	Chocolate bar 70% cacao	\$3
2022-06-08	Oat meal biscuits	\$4

Basic

M => Male

F => Female

Kinds of transformations

- Cleaning & Mapping (Integration)

Mapping different values

Name	Gender
Taylor	M
Isabella	F
Sofia	F

Name	Gender
Lydia	Female
Naomi	Female
Leon	Male

Basic

M => Male

F => Female

Kinds of transformations

- Cleaning & Mapping (Integration)

Mapping different values

Name	Gender
Taylor	M
Isabella	Fe
Sofia	F

Name	Gender
Lydia	Female
Naomi	Female
Leon	Male

Basic

M => Male

F => Female

Kinds of transformations

- Cleaning & Mapping (Integration)

Mapping different values

Name	Gender
Taylor	Male
Isabella	Female
Sofia	Female

Name	Gender
Lydia	Female
Naomi	Female
Leon	Male

Basic

Kinds of transformations

- Cleaning & Mapping (Integration)

Mapping different values

Day	Sales
Monday	\$500
Tuesday	\$760
Wednesday	<i>null</i>

null => 0



Day	Sales
Monday	\$500
Tuesday	\$760
Wednesday	\$0

Basic

Kinds of transformations

- Cleaning & Mapping (Integration)

Mapping different values

Month	Sales
January '22	\$500
February '22	\$760
March '22	\$245

Month	Sales
January 2022	\$1500
February 2022	\$450
March 2022	\$321

Basic

Kinds of transformations

- **Value Standardization (Integration)**

Mapping different values

Month	Sales
January 2022	\$500
February 2022	\$760
March 2022	\$245

Month	Sales in thsd
January 2022	\$1.5
February 2022	\$4.550
March 2022	\$3.321



Month	Sales
January 2022	\$1500
February 2022	\$4550
March 2022	\$3321

Basic

Kinds of transformations

- Key Generation

Product Dimension

Product_PK	product_id	name	category
1	P521	Almonds 150g	Nuts
2	P252	Garlic	Fruits & Vegetables
3	P533	Banana	Fruits & Vegetables
4	P684	Chocolate	Sweets & Snacks
5	P755	Spicy Chips	Sweets & Snacks
6	P521	Almonds 150g	Nuts
7	P672	Orange Juice	Drinks
8	P423	Green Apples	Fruits & Vegetables
9	P564	Chocolate Cookies	Sweets & Snacks
10	P755	Spicy Chips	Sweets & Snacks

Advanced

Kinds of transformations

- Joining

Sales Fact

Sales_PK	product_id	Date
3	P533	2022-01-01
4	P252	2022-01-01
5	P755	2022-01-02
6	P684	2022-01-02
7	P755	2022-01-02

Product Dimension

Product_PK	product_id	name	category
1	P521	Almonds 150g	Nuts
2	P252	Garlic	Fruits & Vegetables
3	P533	Banana	Fruits & Vegetables
4	P684	Chocolate	Sweets & Snacks
5	P755	Spicy Chips	Sweets & Snacks

Advanced

Kinds of transformations

- Joining

Product Dimension

Product_PK	product_id	name	category
1	P521	Almonds 150g	Nuts
2	P252	Garlic	Fruits & Vegetables
3	P533	Banana	Fruits & Vegetables
4	P684	Chocolate	Sweets & Snacks
5	P755	Spicy Chips	Sweets & Snacks

Sales Fact

Sales_PK	product_id	Product_FK	Date
3	P533	3	2022-01-01
4	P252	2	2022-01-01
5	P755	5	2022-01-02
6	P684	4	2022-01-02
7	P755	5	2022-01-02

Advanced

Kinds of transformations

- Joining

Product Dimension

Product_PK	product_id	name	category	Eff_Date	Exp_Date
1	P521	Almonds 150g	Nuts	2021-01-01	2121-01-01
2	P252	Garlic	Fruits & Vegetables	2021-01-01	2121-01-01
3	P533	Banana	Fruits & Vegetables	2021-01-01	2121-01-01
4	P684	Chocolate	Sweets & Snacks	2021-01-01	2121-01-01
5	P755	Spicy Chips	Sweets & Snacks	2021-01-01	2121-01-01

Sales Fact

Sales_PK	product_id	Product_FK	Date
3	P533	3	2022-01-01
4	P252	2	2022-01-01
5	P755	5	2022-01-02
6	P684	4	2022-01-02
7	P755	5	2022-01-02

Advanced

Kinds of transformations

- Joining

Product Table

Product_PK	product_id	name	Category_id
1	P521	Almonds 150g	1
2	P252	Garlic	2
3	P533	Banana	2
4	P684	Chocolate	3
5	P755	Spicy Chips	3

Category table

Category_id	Category
1	Nuts
2	Fruits & Vegetables
3	Sweets

Advanced

Kinds of transformations

- Joining

Product Dimension

Product_PK	product_id	name	category
1	P521	Almonds 150g	Nuts
2	P252	Garlic	Fruits & Vegetables
3	P533	Banana	Fruits & Vegetables
4	P684	Chocolate	Sweets & Snacks
5	P755	Spicy Chips	Sweets & Snacks

Advanced

Kinds of transformations

- **Splitting**

- By length / position
- By Delimiter

Store Dimension

Store_id	Location
1	New York, NY 10011
2	Orland Park, IL 60462
3	Houston, TX 77002

Store_id	City	Location
1	New York	NY 10011
2	Orland Park	IL 60462
3	Houston	TX 77002

Store_id	City	State	ZIP
1	New York	NY	10011
2	Orland Park	IL	60462
3	Houston	TX	77002

Advanced

Kinds of transformations

▪ Aggregations

Sales_Date	Name	Amount
2022-06-06	Sunglasses TR-7	\$25
2022-06-06	Chocolate bar 70% cacao	\$3
2022-06-07	Oat meal biscuits	\$4
2022-06-07	Chocolate bar 70% cacao	\$3
2022-06-08	Oat meal biscuits	\$4



Sales_Date	No. of sales	Amount
2022-06-06	2	\$28
2022-06-07	2	\$7
2022-06-08	1	\$4

- SUM
- COUNT
- DISTINCT COUNT
- AVERAGE

Advanced

Kinds of transformations

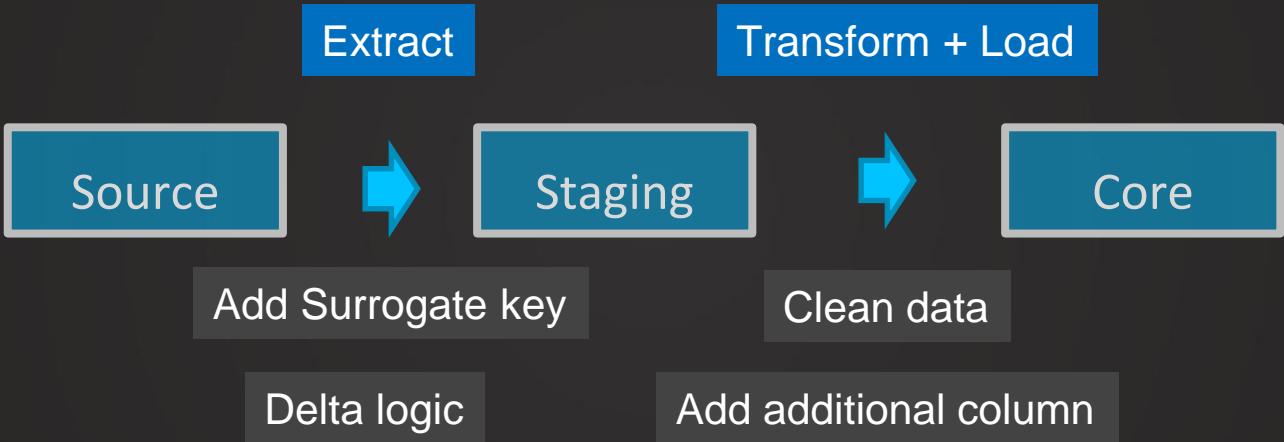
- Deriving Values

Sales Date	Name	Amount	Tax
2022-06-06	Sunglasses TR-7	\$25	17%
2022-06-06	Chocolate bar 70% cacao	\$3	6%
2022-06-07	Oat meal biscuits	\$4	6%
2022-06-07	Chocolate bar 70% cacao	\$3	6%
2022-06-08	Oat meal biscuits	\$4	6%



Sales Date	Name	Amount	Tax	Tax amount
2022-06-06	Sunglasses TR-7	\$25	17%	\$4.25
2022-06-06	Chocolate bar 70% cacao	\$3	6%	\$0.18
2022-06-07	Oat meal biscuits	\$4	6%	\$0.24
2022-06-07	Chocolate bar 70% cacao	\$3	6%	\$0.18
2022-06-08	Oat meal biscuits	\$4	6%	\$0.24

Demo: Plan of attack



Demo: Plan of attack

1. Look at the problem and plan
2. Set up tables & schema
3. Output staging table (+ truncate)
4. Transform + Load:
 - Read from staging
 - Transformation (Clean + Extract)
 - Update/Insert

Processing Order

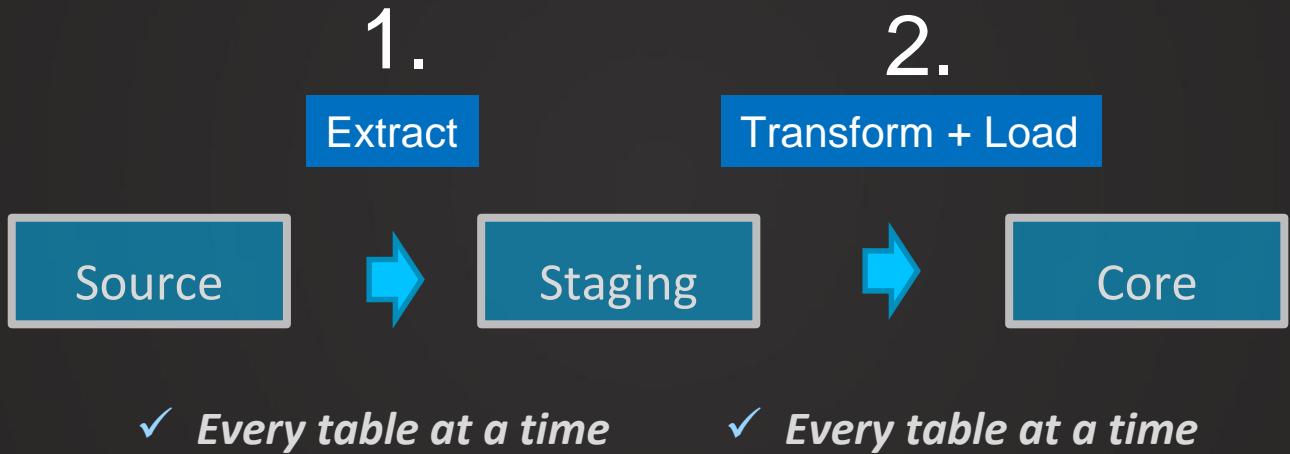
In which order should the steps be processed?

Facts -> Dimensions?

Dimensions -> Facts?

Consider dependencies and plan carefully!

Processing order



Processing order

Extract

Order doesn't really matter

Transform + Load

Dimensions

Facts

Product Dimension

Product_PK	product_id	name	category
1	P521	Almonds 150g	Nuts
2	P252	Garlic	Fruits & Vegetables
3	P533	Banana	Fruits & Vegetables
4	P684	Chocolate	Sweets & Snacks
5	P755	Spicy Chips	Sweets & Snacks

Sales Fact

Sales_PK	product_id	Product_FK	Date
3	P533	3	2022-01-01
4	P252	2	2022-01-01
5	P755	5	2022-01-02
6	P684	4	2022-01-02
7	P755	5	2022-01-02

Processing Order for dimensions

Transform + Load

Start



Dimension 1



Dimension 2

- ✓ *New values*
 - ✓ *Transformations*
 - ✓ *SCD Updates / Load*
- ✓ *New values*
 - ✓ *Transformations*
 - ✓ *SCD Updates / Load*

Processing Order for facts

Transform + Load

Start

Dimension 1

Dimension 2

Finish

Fact 2

Fact 1



Processing Order for facts

Extract

Start

Dimension 1

Dimension 2

Finish

Fact 2

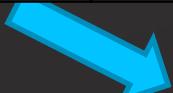
Fact 1



Type 1: Overwrite

- ✓ *Old attributes are just overwritten*
- ✓ *Only current state is reflected*

Product_Key	Name	Category
1	Sunglasses TR-7	Assecoirs
2	Chocolate bar 70% cacao	Sweets
3	Oat meal biscuits	Sweets



UPDATE

Product_Key	Name	Category
1	Sunglasses TR-7	Assecoirs
2	Chocolate bar 70% cacao	Sweets
3	<i>Delicious</i> Oat meal biscuits	<i>Buscuits</i>

Type 2: New row

- ✓ Problem with Type 1: *No history of dimensions!*
- ✓ Only current state is reflected

Product_Key	Name	Category
1	Sunglasses TR-7	Assecoirs
2	Chocolate bar 70% cacao	Sweets
3	Oat meal biscuits	Sweets

UPDATE

Product_Key	Name	Category
1	Sunglasses TR-7	Assecoirs
2	Chocolate bar 70% cacao	Sweets
3	<i>Delicious</i> Oat meal biscuits	<i>Buscuits</i>

Type 3: Additonal Attributes

Product_PK	Product_ID	Name	Category	Ef_Date	Ex_Date
1	SG-TR7	Sunglasses TR-7	Assecoirs	2022-01-01	2100-01-01
2	CH-B70	Chocolate bar 70% cacao	Sweets	2022-01-01	2100-01-01
3	OT-BSC	Oat meal biscuits	Sweets	2022-01-01	2100-01-01
4	OT-BSC	Delicious Oat meal biscuits	Buscuits	2022-06-01	2100-01-01

- ✓ *Type 2 – Default strategy to maintain reflect history*
- ✓ *Type 1 – Static*
- ✓ *Type 3 – In-between: Switching back & forth between versions*

Plan of attack

Case study:

Set up a complete ETL workflow

Plan of attack

1. Look at the problem and plan
2. Set up tables & schema
3. Staging
5. Core (dimension table)
6. Core (fact table)
7. Set up job & testing

Fact table design

1. Create DateKey
2. Include product_FK
3. Payment dimension
4. Additional columns:
 - total_cost
 - Add total_price
 - Add profit

Fact table design

1. Create DateKey
2. Include product_FK
3. Payment dimension
4. Additional columns:
 - total_cost
 - Add total_price
 - Add profit

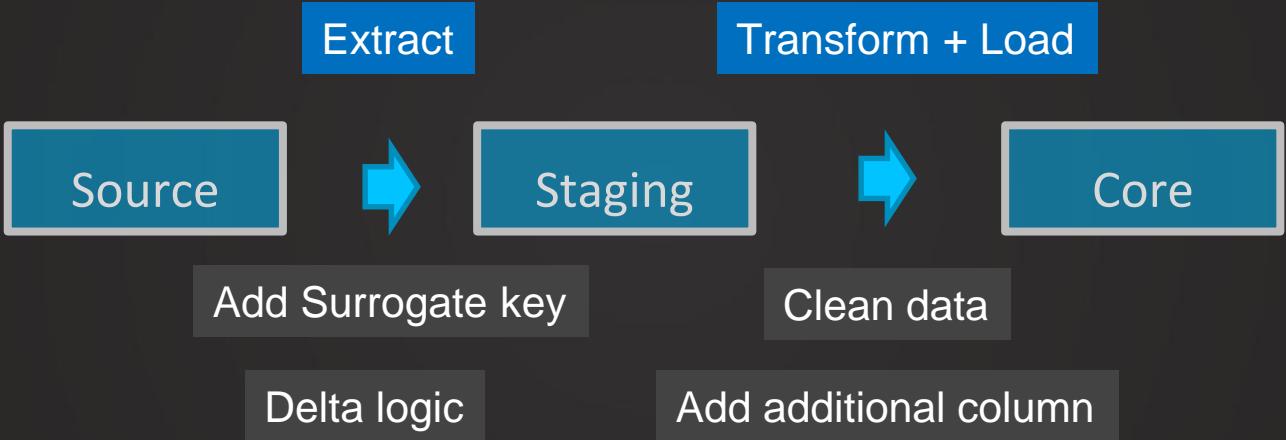
Fact table design

transaction_id	transactional_date	transactional_date_fk	product_id	product_fk	payment_fk	customer_id	credit_card	cost	quantity	price	total_price	total_cost	profit
	integer	timestamp without time zone	numeric	character varying	integer	integer	bigint	numeric	integer	numeric	numeric	numeric	numeric
1	2021-05-04 02:00:00	20210504	P0494	13519	1	4	4041593010498829	17.33	2	18.29	36.58	34.66	1.92
2	2021-05-04 03:04:00	20210504	P0221	13267	1	5	4041596151234556	0.59	1	1.49	1.49	0.59	0.90
3	2021-05-04 03:56:00	20210504	P0625	13643	1	5	4041594885335898	5.15	3	5.89	17.67	15.45	2.22
4	2021-05-04 05:20:00	20210504	P0431	13461	6	8	5108753677552345	10.67	2	11.59	23.18	21.34	1.84
5	2021-05-04 05:45:00	20210504	P0058	13113	4	5	5108752372298261	11.38	2	12.39	24.78	22.76	2.02
6	2021-05-04 06:58:00	20210504	P0385	13416	8	6	374288563442549	13.22	1	14.69	14.69	13.22	1.47
7	2021-05-04 07:03:00	20210504	P0575	13596	1	4	4041598869758	2.81	1	3.99	3.99	2.81	1.18

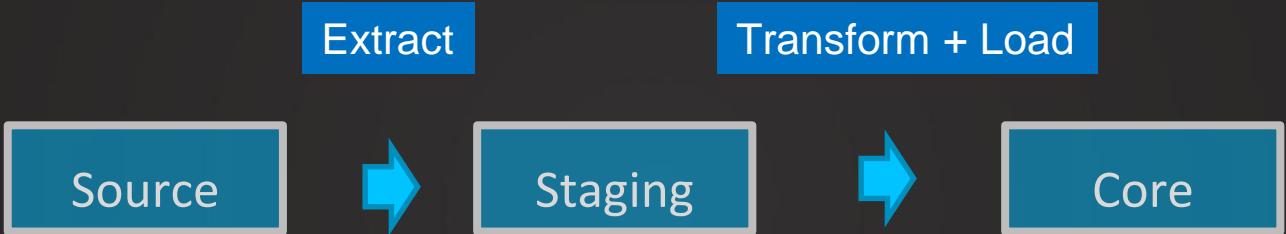
Plan of attack

1. Set up tables
2. Staging for sales fact
3. Create staging job
4. Core for dim_payment
5. Core for sales fact
6. Create core job

Processing order



Scheduling



Jobs or packages

Scheduling at specific times / frequencies

Scheduling

Can be done either...

In the ETL tool

External tool

(e.g. Windows Task scheduler or on server)

Guidelines

What are the requirements?

3 x / day?

1 x / day?

Every 30 min?

How long does it take?

5min?

1h?

What is a good time?

Initial Load vs. Delta Load

Effect on productive system

Short read access

Night? Morning?

ETL tools

Enterprise

Commercial

✓ Most mature

✓ Graphical interface

✓ Architectural needs

✓ Support

Open-source

Source code

✓ Often free

✓ Graphical interface

Support?

Ease of use?

Cloud-native

Cloud technology

Data already in cloud?

✓ Efficiency

Flexibility?

Custom

Own development

Customized

Internal resources

Maintainance?

Training?

ETL tools

Enterprise

Open-source

Cloud-native

Alteryx

Talend Open Studio

Azure Data Factory

Informatica

Pentaho Data Integration

AWS Glue

Oracle Data Integrator

Hadoop

Google Cloud Data Flow

Microsoft SSIS

Stitch

Choosing ETL tool

1. Evaluate current situation/needs

What do you want to improve?

Data sources & other tools?

Define your requirements!

Define responsibles

Who are the users?

Choosing ETL tool

2. Evaluate tools

	Text	Must have? K.O.?	Weight/ Importance	Rating
Cost			1-5	1-5
Connectors				
Capabilities				
Ease of use/work				
Reviews				
Support/Extras				

Total weighted score:

Choosing ETL tool

3. Test / Demo / Trial

Make a decision!

Choosing ETL tool

3. Test / Demo / Trial

Make a decision!

Choosing ETL tool

Enterprise

Informatica

Oracle Data Integrator

Microsoft SSIS

IBM DataStage

Open-source

Talend Open Studio

Pentaho Data Integration

Hadoop

Cloud-based

Azure Data Factory

AWS Glue

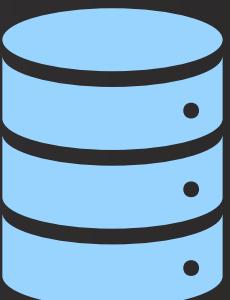
Google Cloud Data Flow

Stitch

What is ELT?

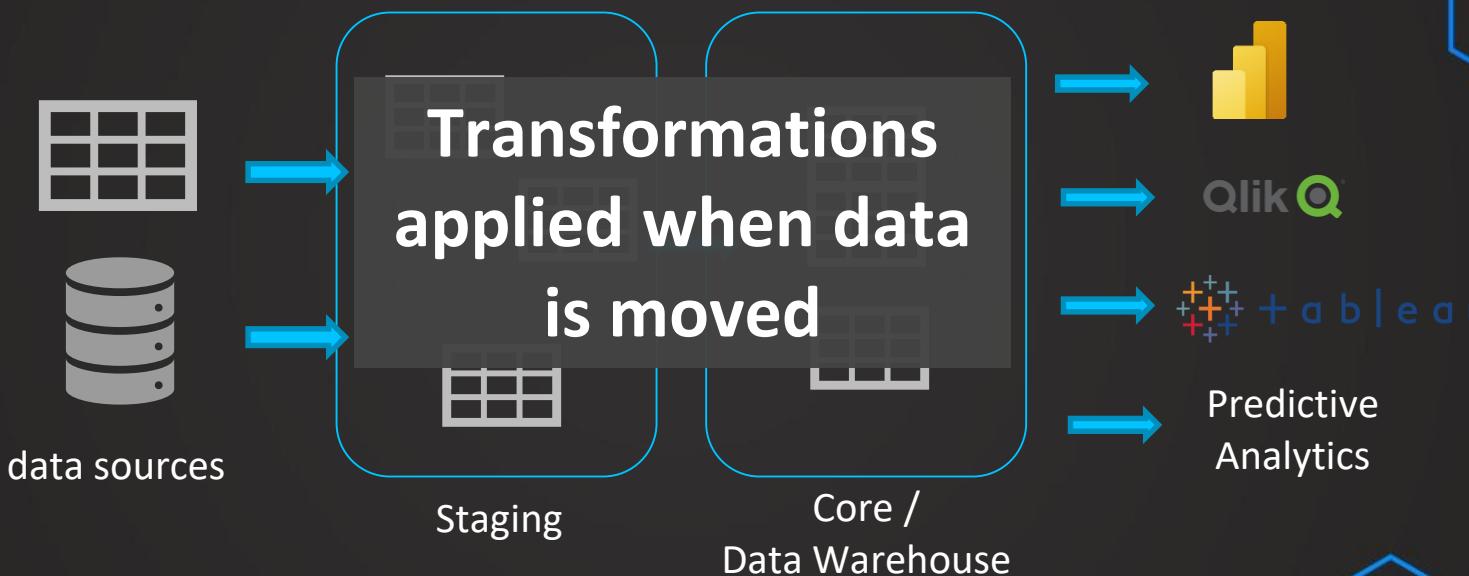
What is ELT?

Extract, Transform, Load

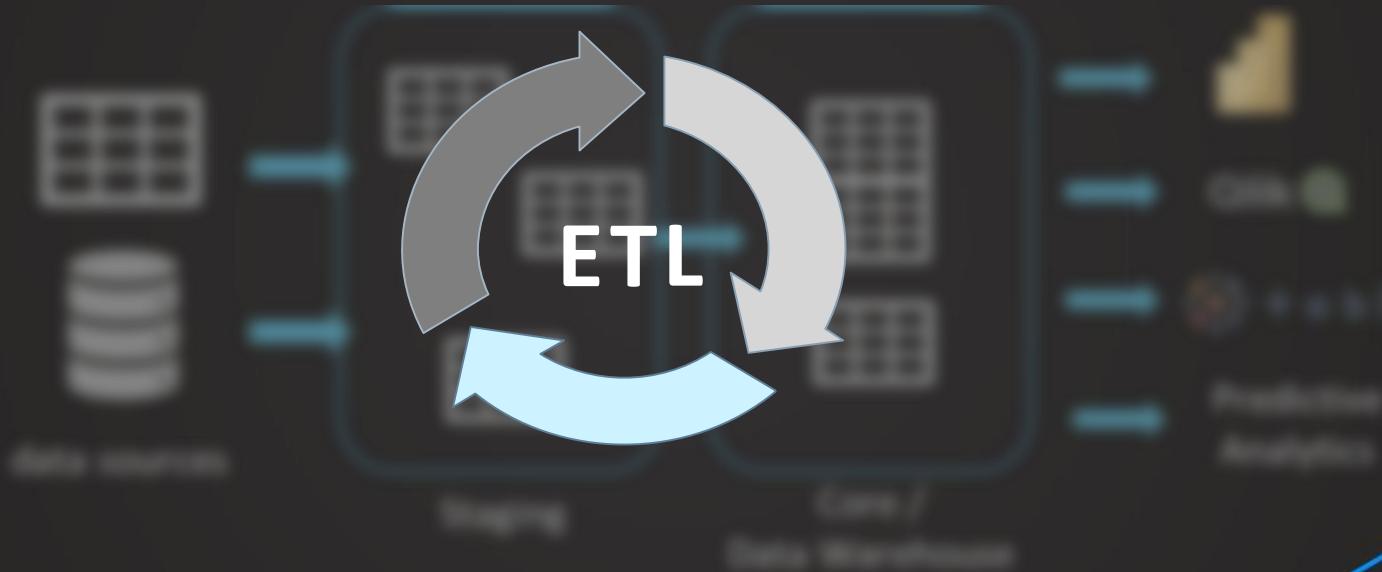


Data warehouse

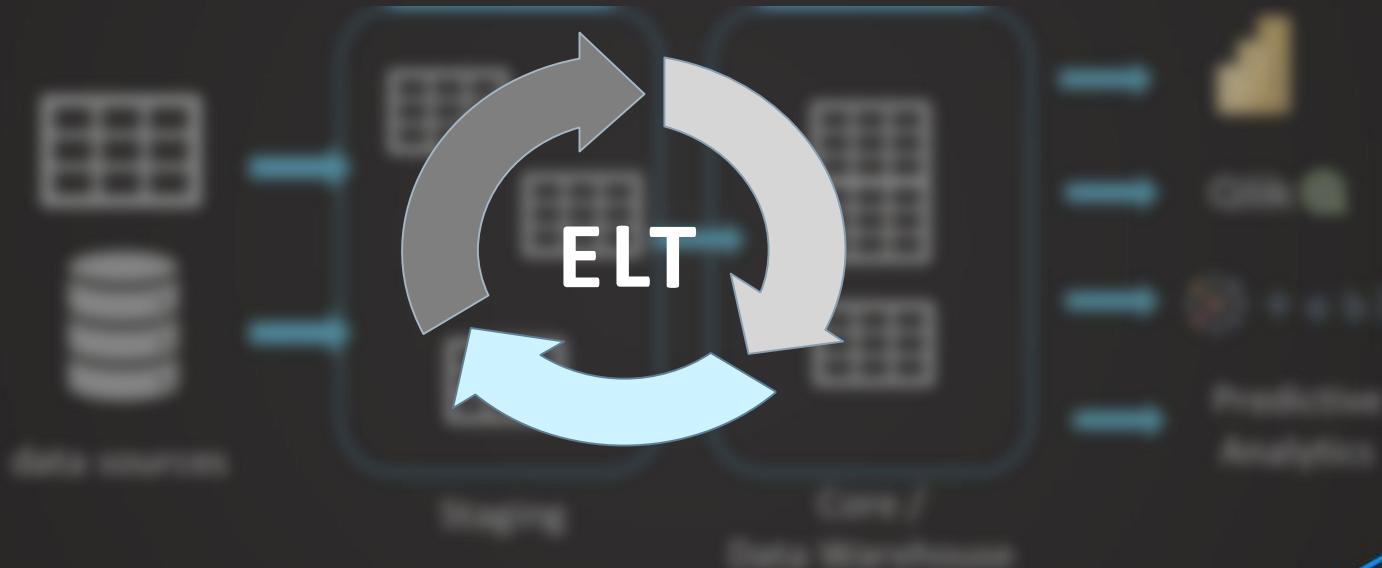
Data Warehouse Layers



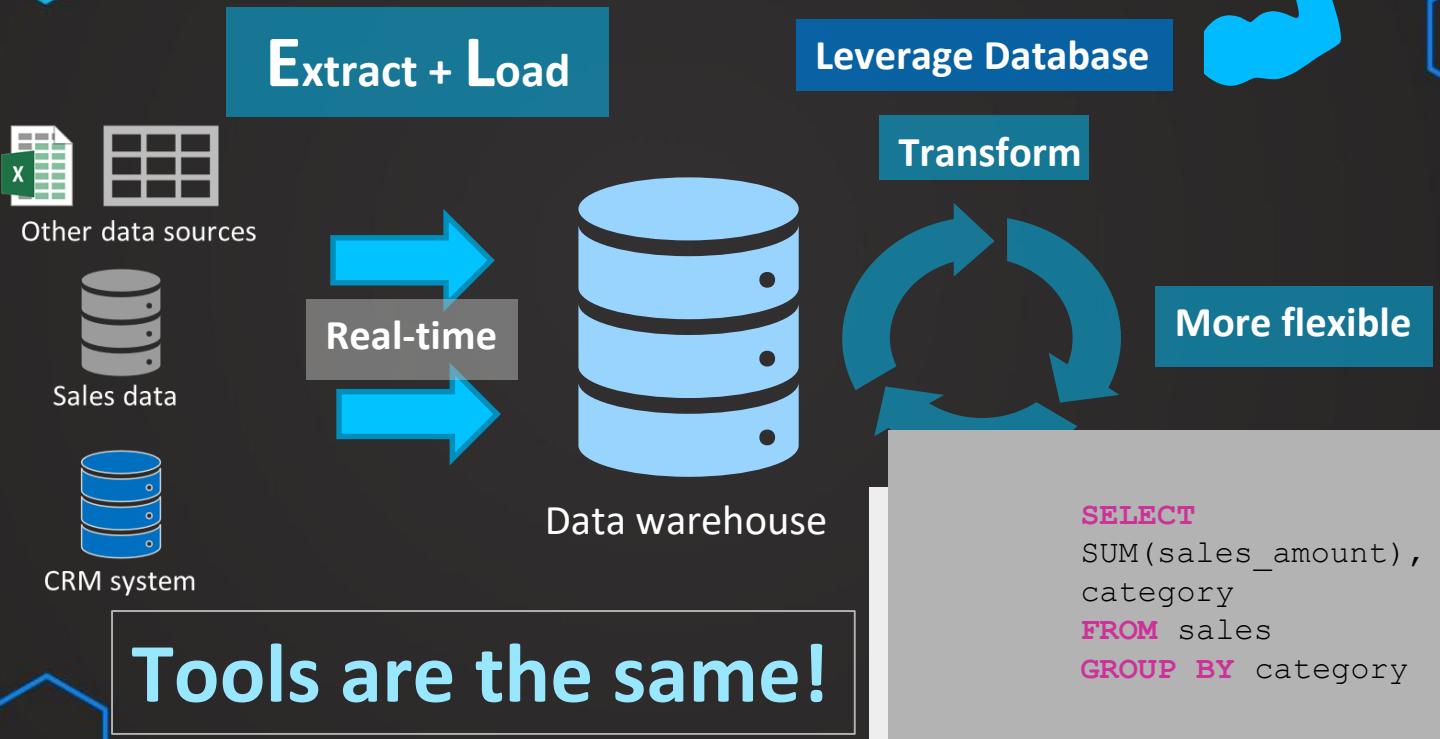
Data Warehouse Layers

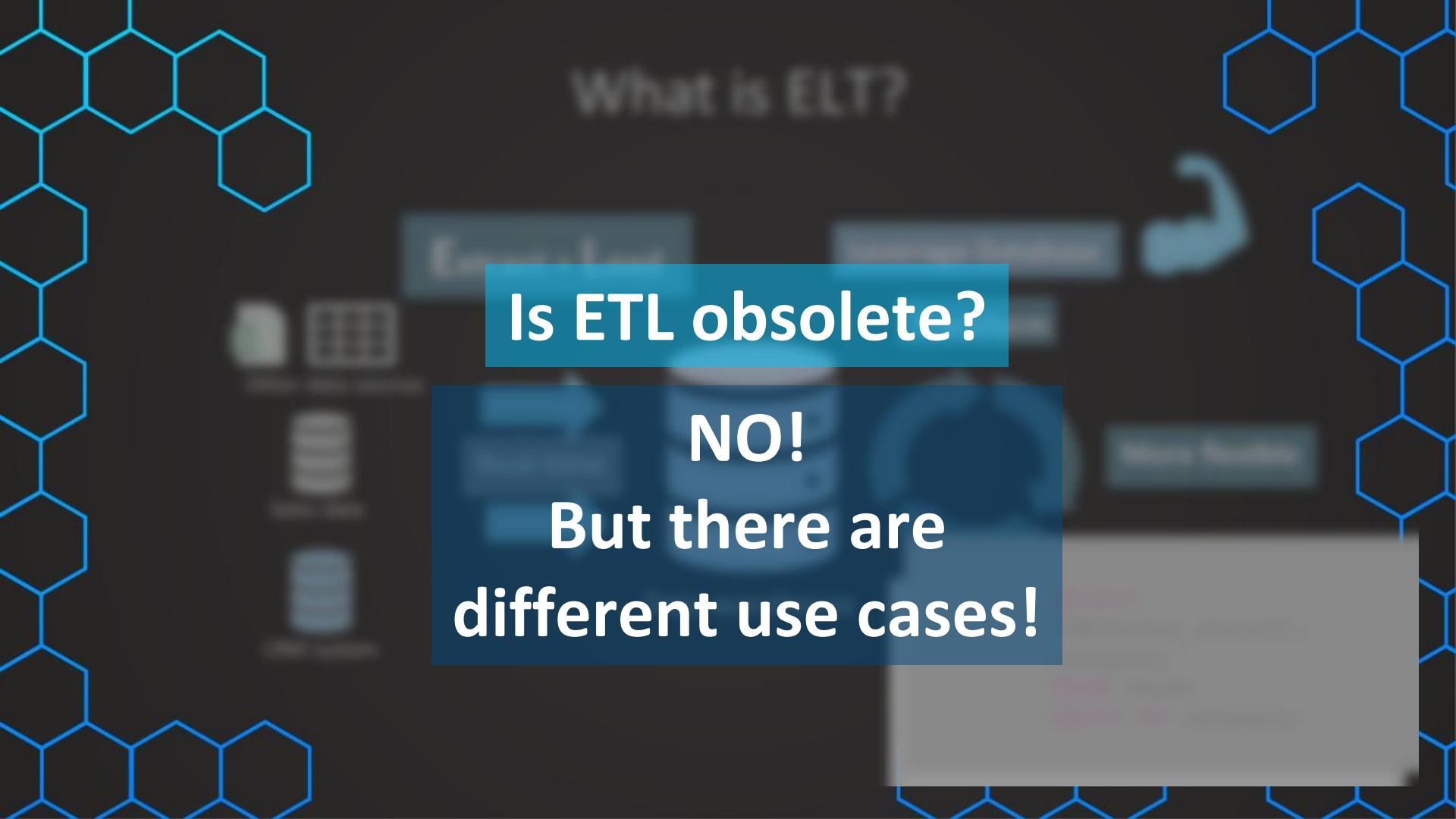


Data Warehouse Layers



What is ELT?





What is ELT?

Is ETL obsolete?

NO!

But there are
different use cases!

ETL vs. ELT

ETL

- ✓ More stable with defined transformations
- ✓ More generic use-cases
- ✓ Security

ELT

- ✓ Requires high performance DB
- ✓ More flexible
- ✓ Transformations can be changed quickly
- ✓ Real-time

ETL vs. ELT

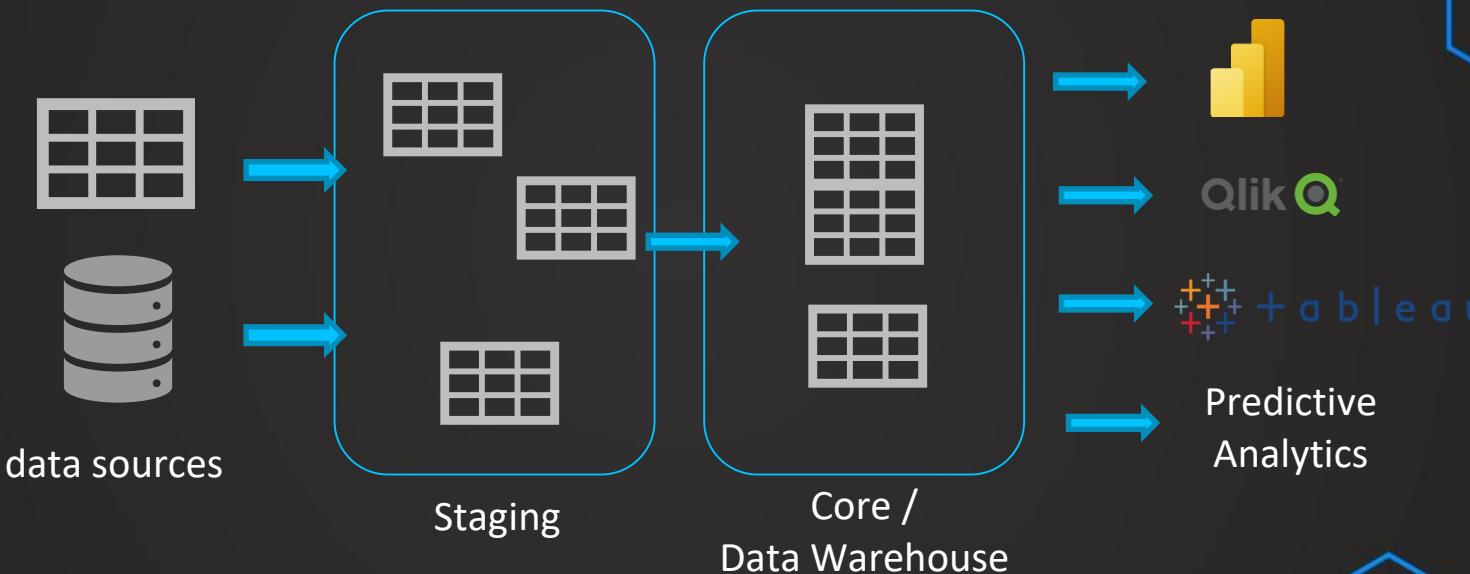
ETL

- ✓ Reporting
- ✓ Generic use cases
- ✓ Easy to use

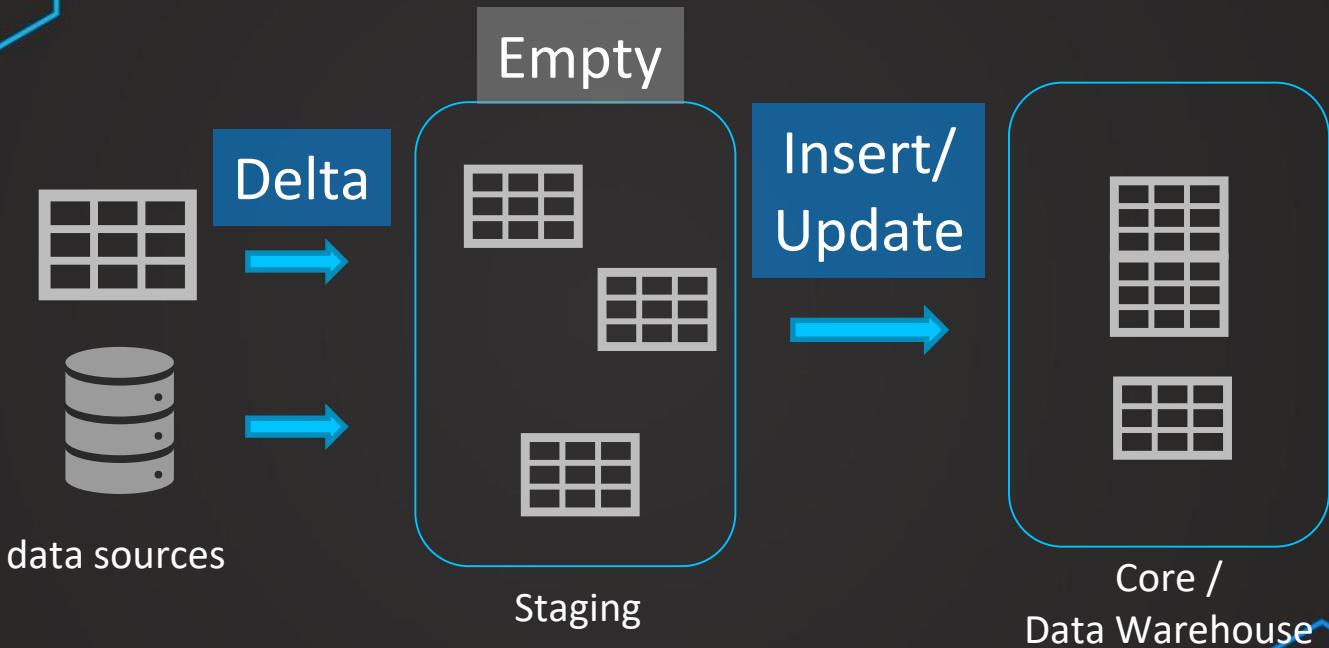
ELT

- ✓ Data Science, ML
- ✓ Real-time requirements
- ✓ Big data

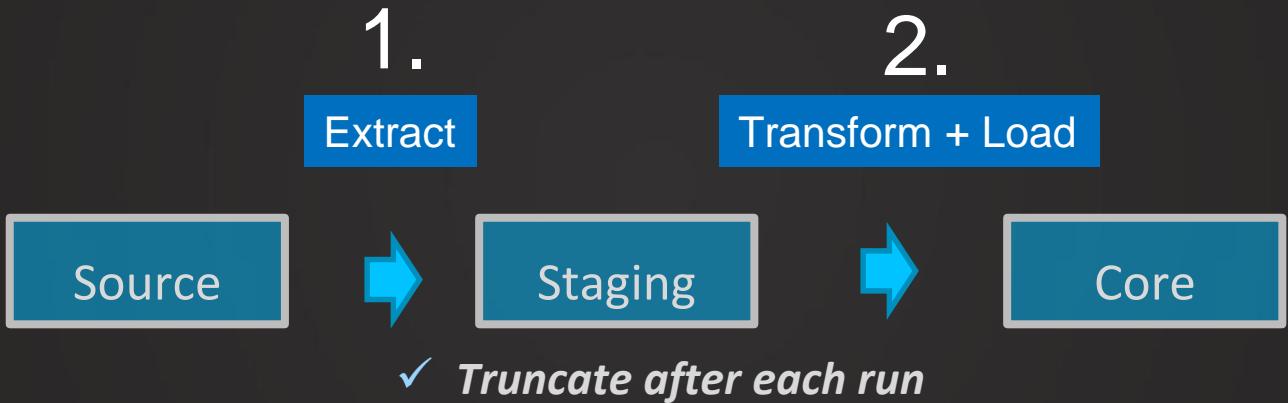
Data Warehouse Layers



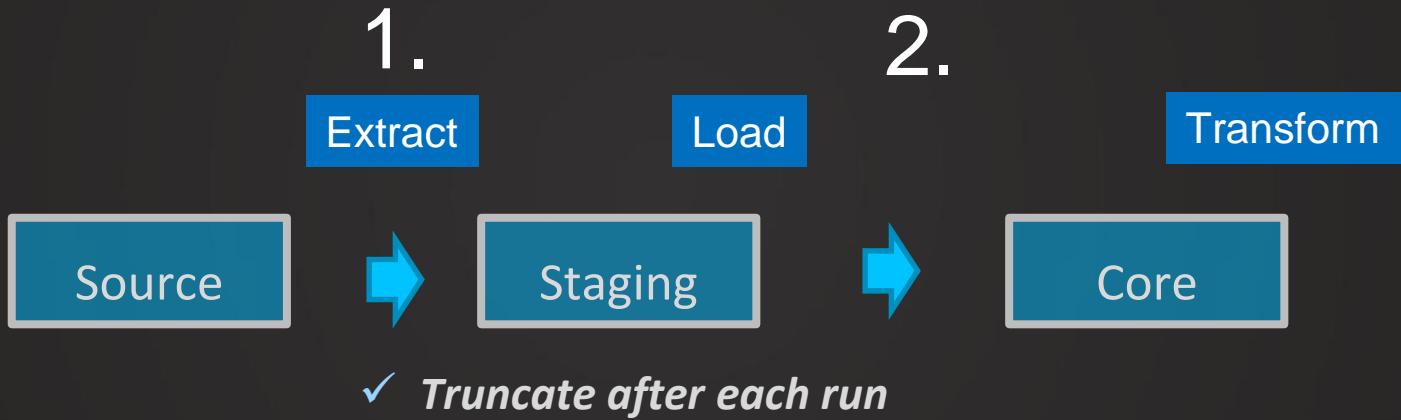
Data Warehouse Layers



Processing order



Processing order



Data Warehouse Use Cases

What's next?

What are the use cases?

Performance Integrated Strategic decisions

Easy to use Data Quality Accessible

Continuous Training of Machine Learning Models

Aggregate & Filter

Basis for reporting

Enables business users to analyze data

Predictive Analytics

Use Big Data

Using index

transaction_id [PK] integer	product_id character varying	customer_id integer	payment character varying	price numeric
1	P0494	4	visa	18.29
2	P0221	5	visa	1.49
3	P0625	5	visa	5.89
4	P0431	8	mastercard	11.59
5	P0058	5	mastercard	12.39

3, P0625, 5, visa

4, P0432, 8, mastercard

1, P0494, 4, visa

6, P0058, 5, mastercard

2, P0221, 5, visa

```
SELECT  
product_id  
FROM sales  
WHERE customer_id = 5
```

Table scan

Read-inefficient

Using index

transaction_id [PK] integer	product_id character varying	customer_id integer	payment character varying	price numeric
1	P0494	4	visa	18.29
2	P0221	5	visa	1.49
3	P0625	5	visa	5.89
4	P0431	8	mastercard	11.59
5	P0058	5	mastercard	12.39

1, P0494, 4, visa
6, P0058, 5, mastercard
3, P0625, 5, visa

2, P0221, 5, visa
4, P0432, 8, mastercard

```
SELECT  
product_id  
FROM sales  
WHERE customer_id = 5
```

Location	Value
1	4
2	5
5	8

Using Index

✓ Indexes help to make data reads faster!

❖ Slower data writes

❖ Additional storage

❖ B-tree Indexes

❖ Bitmap Indexes

Using index

transaction_id [PK] integer	product_id character varying	customer_id integer	payment character varying	price numeric
1	P0494	4	visa	18.29
2	P0221	5	visa	1.49
3	P0625	5	visa	5.89
4	P0431	8	mastercard	11.59
5	P0058	5	mastercard	12.39

1, P0494, 4, visa
6, P0058, 5, mastercard
3, P0625, 5, visa

2, P0221, 5, visa
4, P0432, 8, mastercard

```
SELECT  
product_id  
FROM sales  
WHERE customer_id = 5
```

Location	Value
1	4
2	5
5	8

Using index

Location	Value
1	4
2	5
5	8

- ✓ Different types of indexes
for different situations

❖ B-tree Indexes

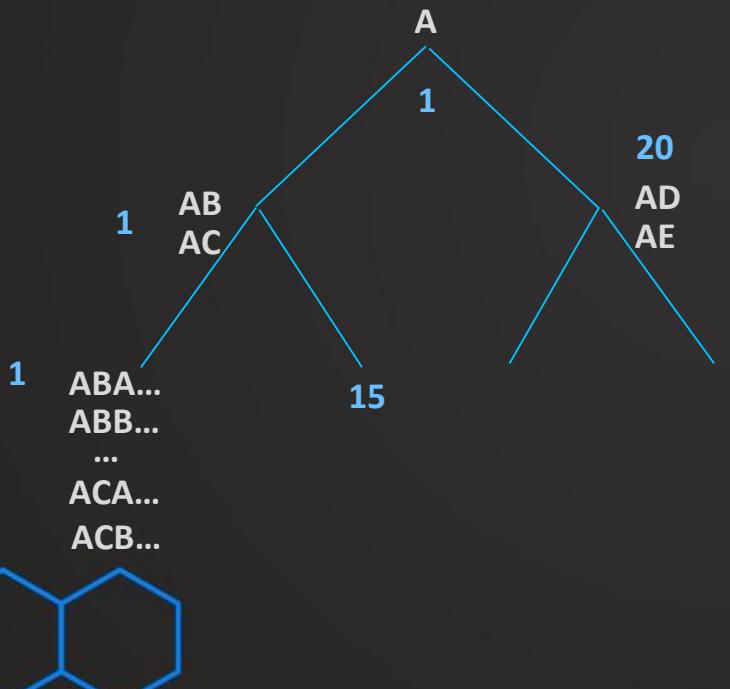
❖ Bitmap Indexes

1, P0494, 4, visa
6, P0058, 5, mastercard
3, P0625, 5, visa

2, P0221, 5, visa
4, P0432, 8, mastercard



B-tree Indexes

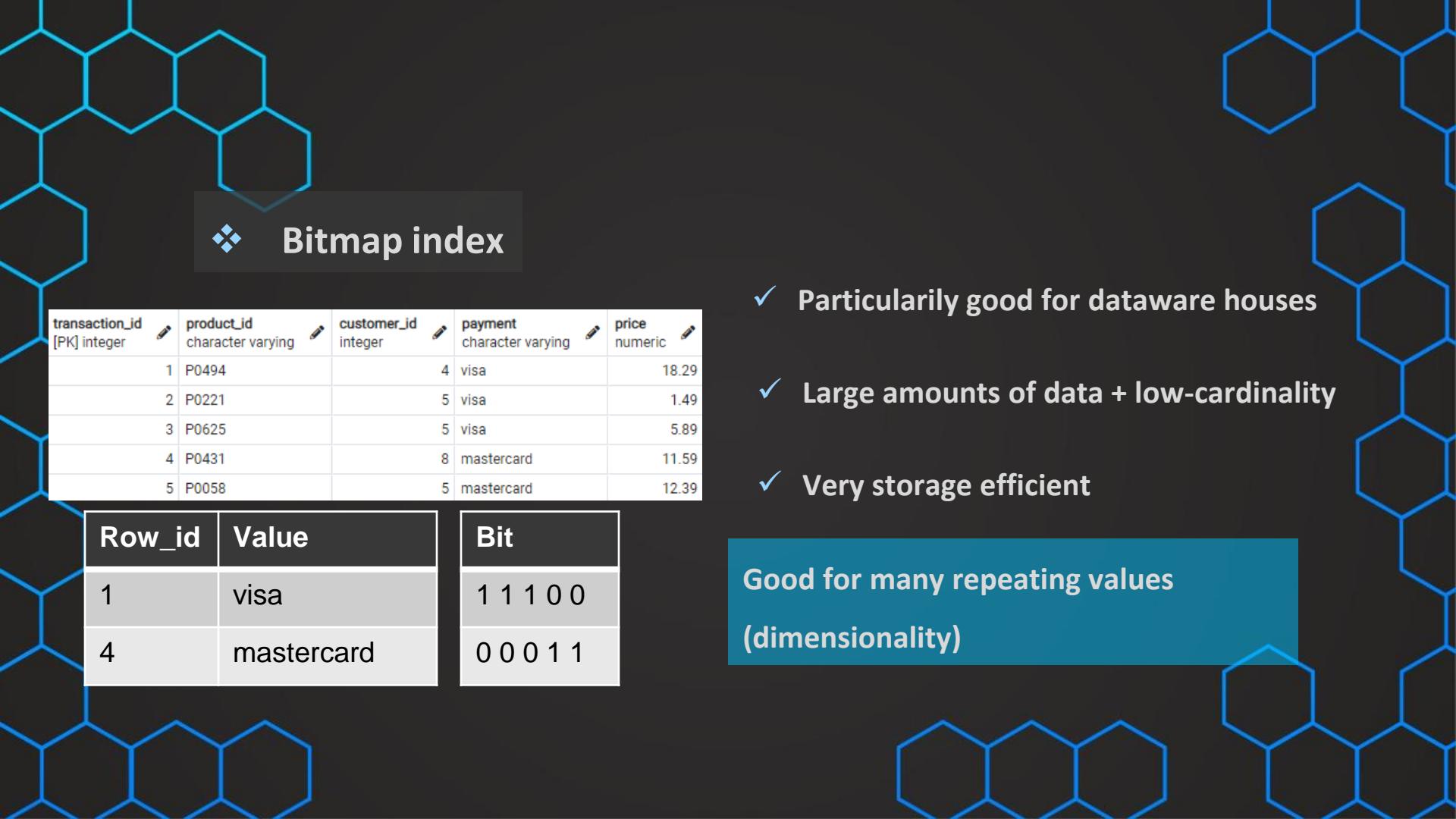


- ✓ Multi-level tree structure
- ✓ Breaks data down into pages or blocks
- ✓ Should be used for high-cardinality
(unique) columns
- ✓ Not entire table (costy in terms of storage)

Bitmap index

transaction_id [PK] integer	product_id character varying	customer_id integer	payment character varying	price numeric
1	P0494		4 visa	18.29
2	P0221		5 visa	1.49
3	P0625		5 visa	5.89
4	P0431		8 mastercard	11.59
5	P0058		5 mastercard	12.39

- ✓ Particularly good for dataware houses
- ✓ Large amounts of data + low-cardinality
- ✓ Very storage efficient
- ✓ More optimized for read & few DML-operations



transaction_id [PK] integer	product_id character varying	customer_id integer	payment character varying	price numeric
1	P0494		4	visa
2	P0221		5	visa
3	P0625		5	visa
4	P0431		8	mastercard
5	P0058		5	mastercard

Row_id	Value	Bit
1	visa	1 1 1 0 0
4	mastercard	0 0 0 1 1

- ✓ Particularly good for dataware houses
- ✓ Large amounts of data + low-cardinality
- ✓ Very storage efficient

Good for many repeating values
(dimensionality)

❖ Bitmap index

transaction_id [PK] integer	product_id character varying	customer_id integer	payment character varying	price numeric
1	P0494		4	visa
2	P0221		5	visa
3	P0625		5	visa
4	P0431		8	mastercard
5	P0058		5	mastercard

Value	1	2	3	4	5	6	7	8
mastercard				x	x			
visa	x	x	x					

- ✓ Particularly good for dataware houses
- ✓ Large amounts of data + low-cardinality
- ✓ Very storage efficient

Good for many repeating values
(dimensionality)

Guidelines

B-tree Index

Bitmap Index

Default index

Slow to update

Unique columns
(surrogate key, names)

Storage efficient

Great read performance

Guidelines

Should we put index on every column?

No! They come with a cost!

Storage + Create/Update time

Only when necessary!

Avoid full table reads

Small tables do not require indexes

On which columns?

Guidelines

1. Large tables

2. Columns that are used as filters

transaction_id [PK] integer	product_id character varying	customer_id integer	payment character varying	price numeric
1	P0494	4	visa	18.29
2	P0221	5	visa	1.49
3	P0625	5	visa	5.89
4	P0431	8	mastercard	11.59
5	P0058	5	mastercard	12.39

Guidelines

Fact tables

B-tree on surrogate key

Bitmap key on foreign keys

Dimension table

Size of table

Are they used in searches a lot?

Choose based on cardinality

Guidelines

```
CREATE INDEX index_name ON table_name [USING method]
(
    column_name [ASC | DESC] ,
    ...
);
```

Cloud vs. On-premises

On-premises

Cloud

What?

Own local hardware

- ✓ Storage layer
- ✓ Compute layer
- ✓ Software layer

Physical data center

What?

Software-as-a-service

- ✓ Pay for what you use

Managed service

- ✓ Optimized for scalable analytics

What is the right choice today?

Cloud vs. On-premises

On-premises

Benefits

- ✓ Full control
- ✓ Data governance & compliance

Problems

- ❖ Full responsibility
- ❖ High costs
- ❖ More internal resources
- ❖ Less flexible

Cloud

Benefits

- ✓ Fully managed
- ✓ Scalable
- ✓ Cost-efficient
- ✓ Managed security
- ✓ Availability

Problems

- ✓ Time to market
- ❖ Regulations
- ❖ Different providers?

Conclusion?

Which one to
choose?

- ✓ Cloud data warehouses are on the rise
- ✓ Most companies opt for cloud data warehouse

In most cases cloud data warehouse is the better choice nowadays!

Conclusion?

What are the options?

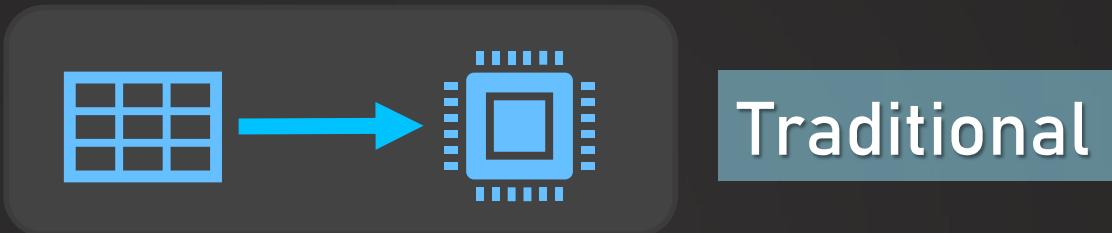
- ✓ Snowflake
- ✓ Amazon Redshift
- ✓ Azure Synapse
- ✓ Google Big Query

Conclusion?

What are the options?

- ✓ Snowflake
- ✓ Amazon Redshift
- ✓ Azure Synapse
- ✓ Google Big Query

Massive parallel processing (MPP)



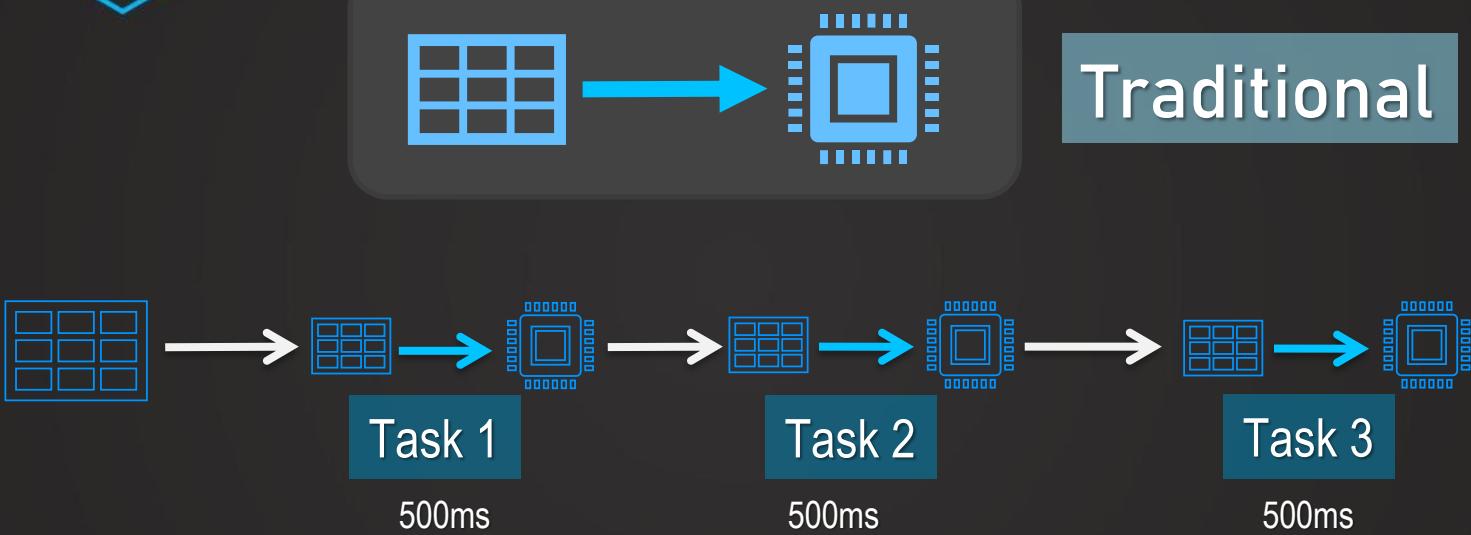
Massive parallel processing (MPP)

Example

transaction_id [PK] integer	product_id character varying	customer_id integer	payment character varying	price numeric
1	P0494	4	visa	18.29
2	P0221	5	visa	1.49
3	P0625	5	visa	5.89
4	P0431	8	mastercard	11.59
5	P0058	5	mastercard	12.39

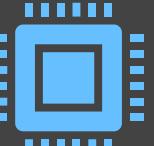
```
SELECT
*
FROM sales
WHERE customer_id = 5
```

Massive parallel processing (MPP)

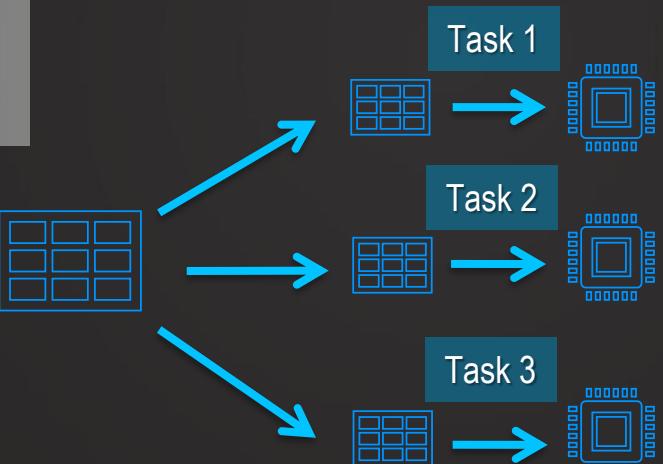


Massive parallel processing (MPP)

"Shared disk" architecture

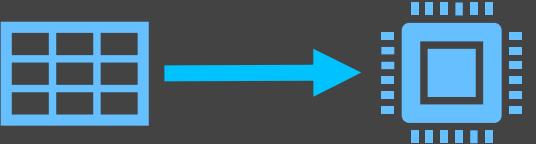


Traditional



MPP

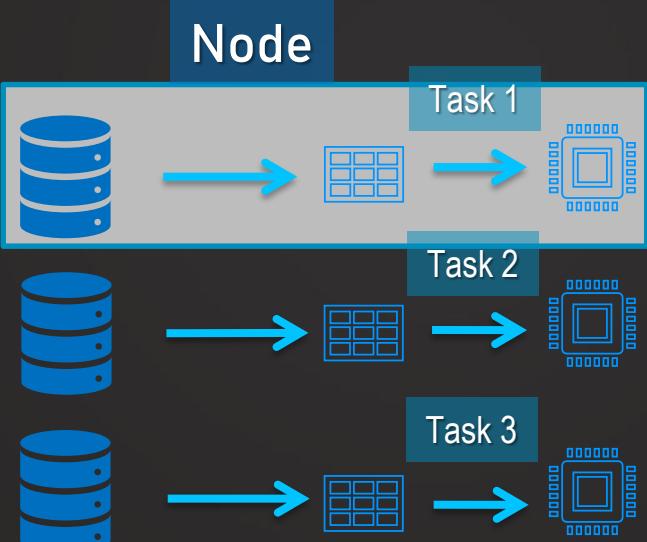
Massive parallel processing (MPP)



Traditional

"Shared nothing"
architecture

Independent
resources



Work load is split up
& processed individually

MPP

Massive parallel processing (MPP)

- ✓ Modern way of solving performance issues
- ✓ Millions of rows can be processed faster
- ✓ Many people can run queries at the same time with good performance
- ✓ Helpful with centralizing massive amounts of data

Columnar databases

transaction_id [PK] integer	product_id character varying	customer_id integer	payment character varying	price numeric
1	P0494	4	visa	18.29
2	P0221	5	visa	1.49
3	P0625	5	visa	5.89
4	P0431	8	mastercard	11.59
5	P0058	5	mastercard	12.39

3, P0625, 5, visa
4, P0432, 8, mastercard
1, P0494, 4, visa

6, P0058, 5, mastercard
2, P0221, 5, visa

```
SELECT  
product_id  
FROM sales
```

Traditional
Relational DB

All rows have to be scanned

Good for transactional DB

Bad for fast data retrieval!

Columnar databases

transaction_id [PK] integer	product_id character varying	customer_id integer	payment character varying	price numeric
1	P0494	4	visa	18.29
2	P0221	5	visa	1.49
3	P0625	5	visa	5.89
4	P0431	8	mastercard	11.59
5	P0058	5	mastercard	12.39

1, 2, 3, 4, 5

P0494, P0221, P0625, P0431, P0058

4, 5, 5, 8, 5

visa, visa, visa, mastercard, mastercard

18.29, 1.49, 5.89, 11.59, 12.39

```
SELECT  
product_id  
FROM sales
```

Less data needs to be processed!

Better compression, less storage

Columnar databases

100 columns

but only 5 columns are needed

5% of data needs to be
processed

- ✓ Important factor in improving analytical query performance

Guidelines

index if you frequently want to retrieve less than about 15% of the rows in a large table

Index columns used for joins to improve join performance

Small tables do not require indexes

PK automatically per default has an index

- There is a wide range of values (good for regular indexes).
- There is a small range of values (good for bitmap indexes).