

# Data warehouse data modeling

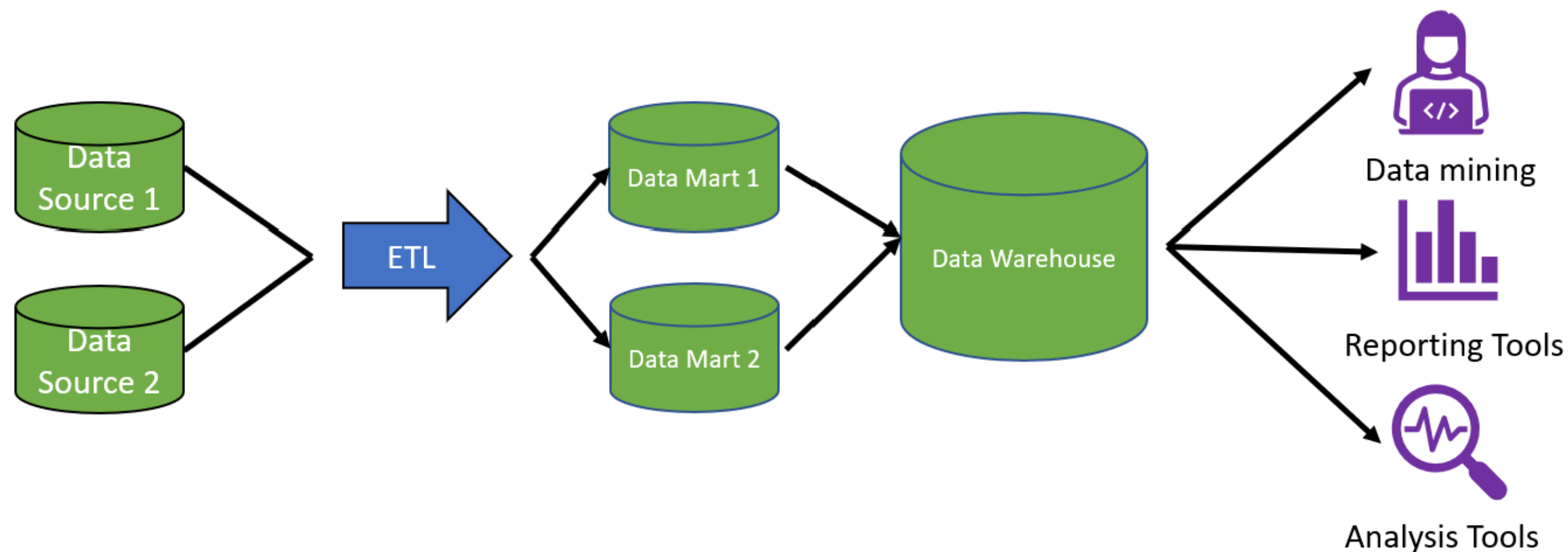
DATA WAREHOUSING CONCEPTS



**Aaren Stubberfield**  
Data Scientist

# Data models

- Bottom-up, Kimball model = star & snowflake schemas
- Denormalized data models



# It's Bravo again!

- Hypothetical publicly traded company
  - Sells home office furniture



<sup>1</sup> Photo from Pexels by Pixabay

# Fact table

- Measurements, metrics, or facts about an organization
- Links to dimension tables for more details

**Table Name:** Sales\_Order\_Fact

Keys	ColumnName
FK	CustomerID
FK	DateID
FK	ProductID
	UnitSold
	SalesAmount
	Tax

**Legend:** FK = Foreign Key

# Dimension table

- Dimensions/attributes about a process
- Holds reference data
- Dimension tables add more detail to fact table

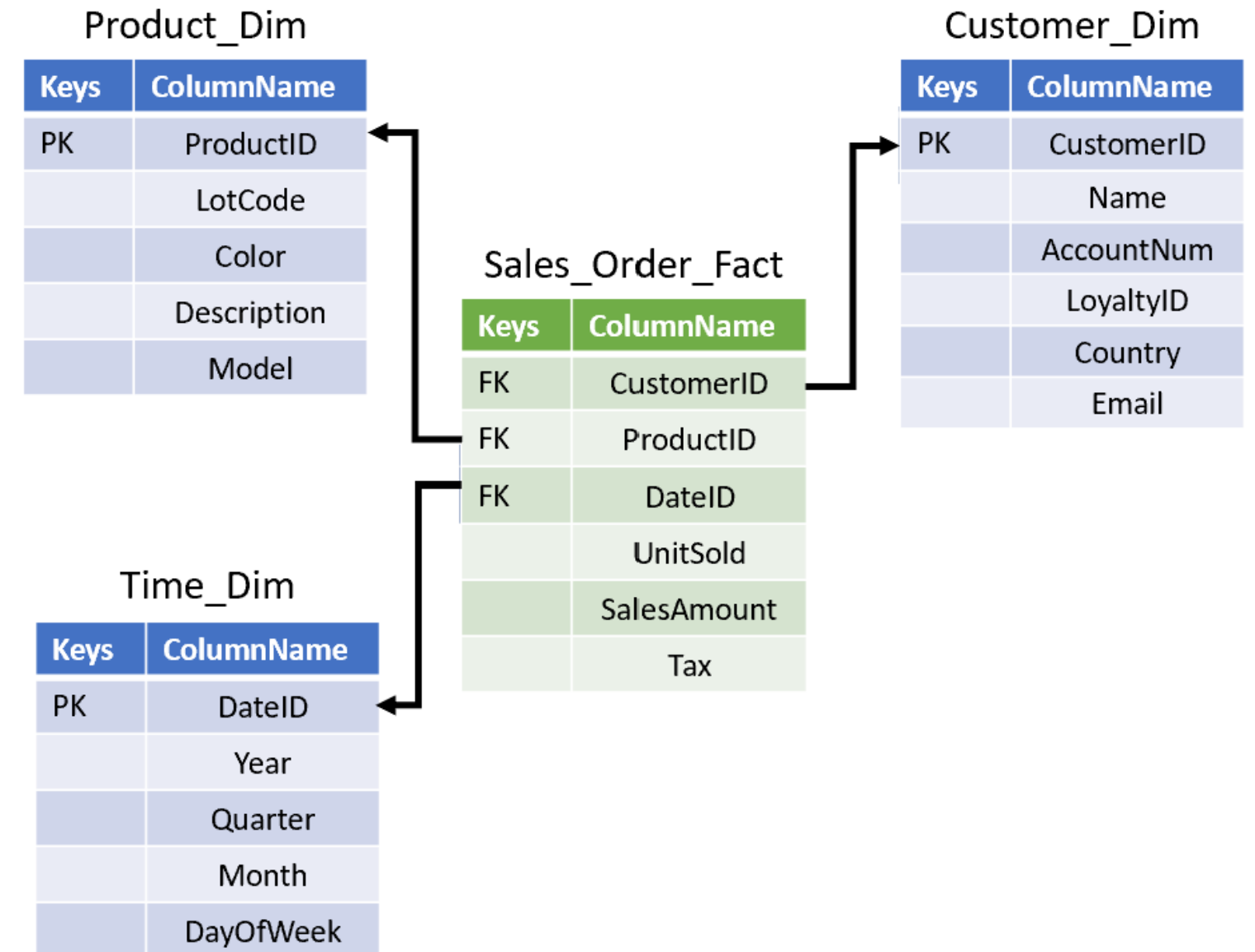
**Table Name:** Customer\_Dim

Keys	ColumnName
PK	CustomerID
	Name
	AccountNum
	LoyaltyID
	Country
	Email

**Legend:** PK = Primary Key

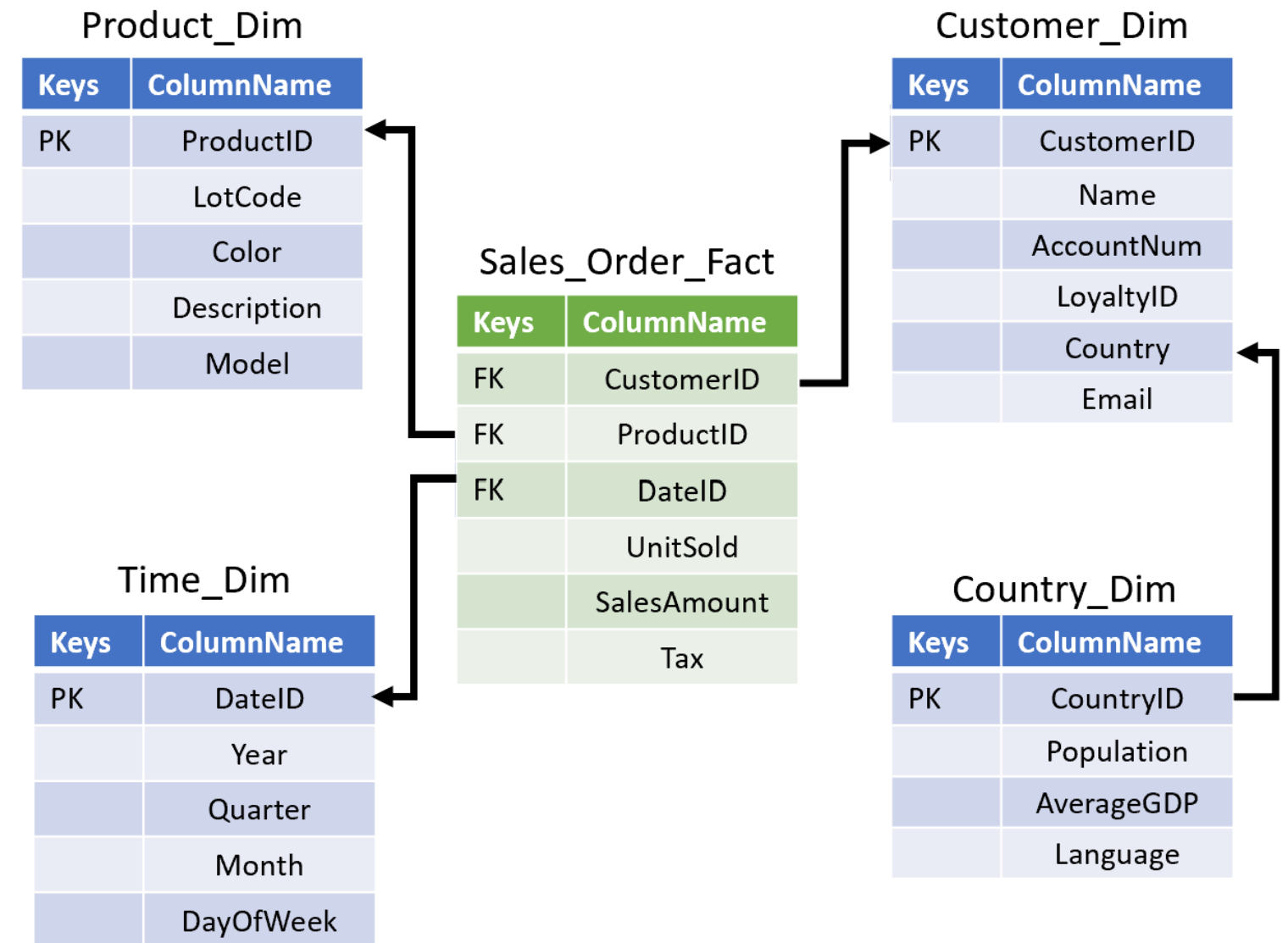
# Star schema

- A central fact table, with one or more dimensional tables
- Easy for business users



# Snowflake schema

- Dimensional table connected through another dimensional table



**Let's practice!**  
DATA WAREHOUSING CONCEPTS



# Kimball's four step process

DATA WAREHOUSING CONCEPTS



**Aaren Stubberfield**  
Data Scientist

# Step 1 - Select the organizational process

- Ask questions about a process
- Kimball bottom-up approach starts with a business process



## Examples of organizational processes:

- Invoice and billing
- Product quality monitoring
- Marketing

# Step 2 - Declare the grain

- Grain = level to store fact table
- A level of data that cannot be split further

## Examples of business processes:

- Music service -> Song grain
- Shipping service -> Line item grain

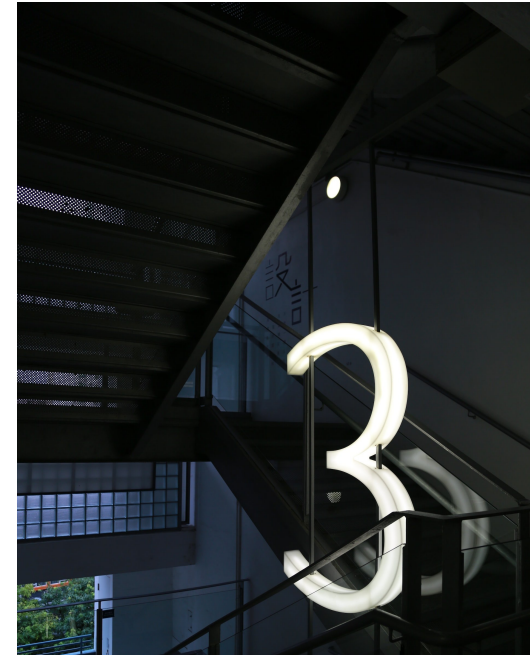


# Step 3 - Identify the dimensions

- Choose dimensions that apply to each row
- How to describe the data?
- Business users and analysts = valuable feedback

## Examples of common dimensions:

- **Time:** year, quarter, and month
- **Location:** address, state, and country
- **Users:** names and email address



<sup>1</sup> Photo by Alison Pang on Unsplash

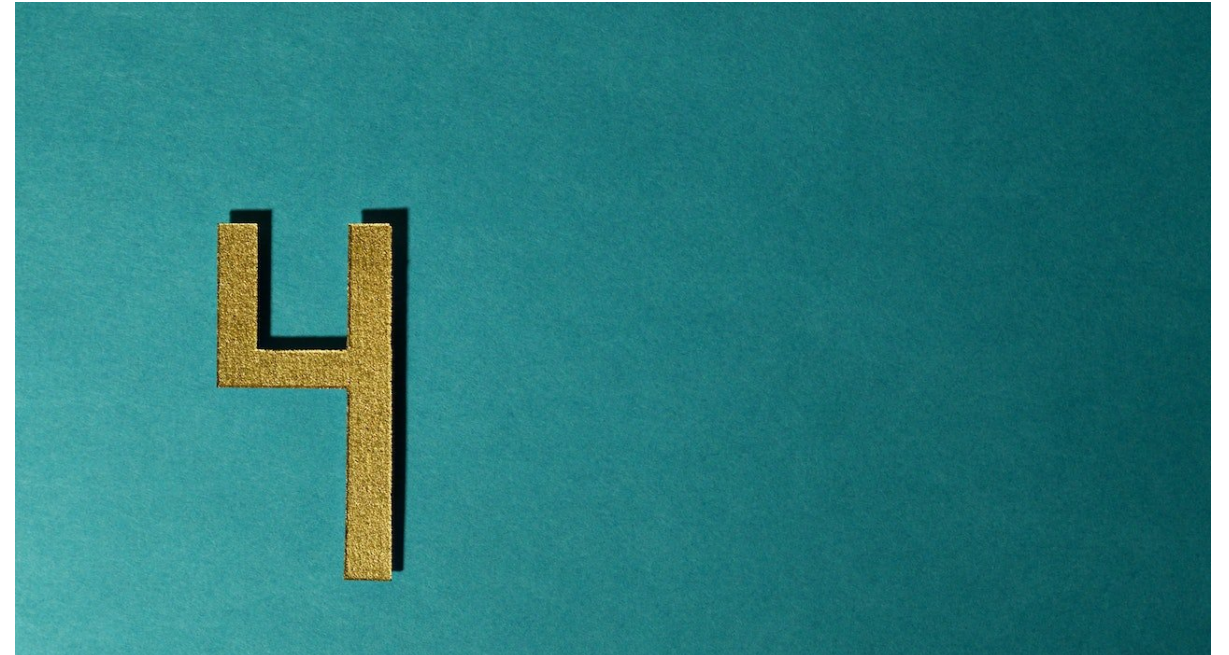


# Step 4 - Identify the facts

- Numerical facts for each fact table row
- What are we answering?
- Metrics should be true at selected grain

## Examples of facts:

- Music service: total number of plays, sales revenue of a song
- Ride-sharing: travel distance, time needed



<sup>1</sup> Photo by Miguel Á. Padriñán

# Summary

## Steps:

1. Select the organizational process.
2. Declare the grain.
3. Identify the dimensions.
4. Identify the facts.

**Let's practice!**  
DATA WAREHOUSING CONCEPTS

# Slowly changing dimensions

DATA WAREHOUSING CONCEPTS



**Aaren Stubberfield**  
Data Scientist



# The challenge

Original

ProductID	Description	Category
12345	Tesla-ModelY	electric-veh.



Update Category:

- **Current:** electric-veh.
- **New:** electric-crossover

# Type I

- Update value in table
- Will lose any history

## Original

ProductID	Description	Category
12345	Tesla-ModelY	<del>electric-veh.</del>

## New

ProductID	Description	Category
12345	Tesla-ModelY	electric-crossover

# Type II

- Add a row with the updated value
- The history is retained

## Original

ProductID	Description	Category
12345	Tesla-ModelY	electric-veh.

## New

ProductID	Description	Category	StartDate	EndDate
12345	Tesla-ModelY	electric-veh.	1970-01-01	2022-03-10
20053	Tesla-ModelY	electric-crossover	2022-03-11	2050-12-31

# Type III

- Add column to dimension table to track changes
- Can view past and current data together
- Can require reporting changes and limited tracking

## Original

ProductID	Description	Category
12345	Tesla-ModelY	electric-veh.

## New

ProductID	Description	Category	PastCategory
12345	Tesla-ModelY	electric-crossover	electric-veh.

# Modern approach

- Snapshot the whole dimension table
- Use historical snapshots for historical reports

**Let's practice!**  
DATA WAREHOUSING CONCEPTS

# Row vs. column data store

DATA WAREHOUSING CONCEPTS



**Aaren Stubberfield**  
Data Scientist

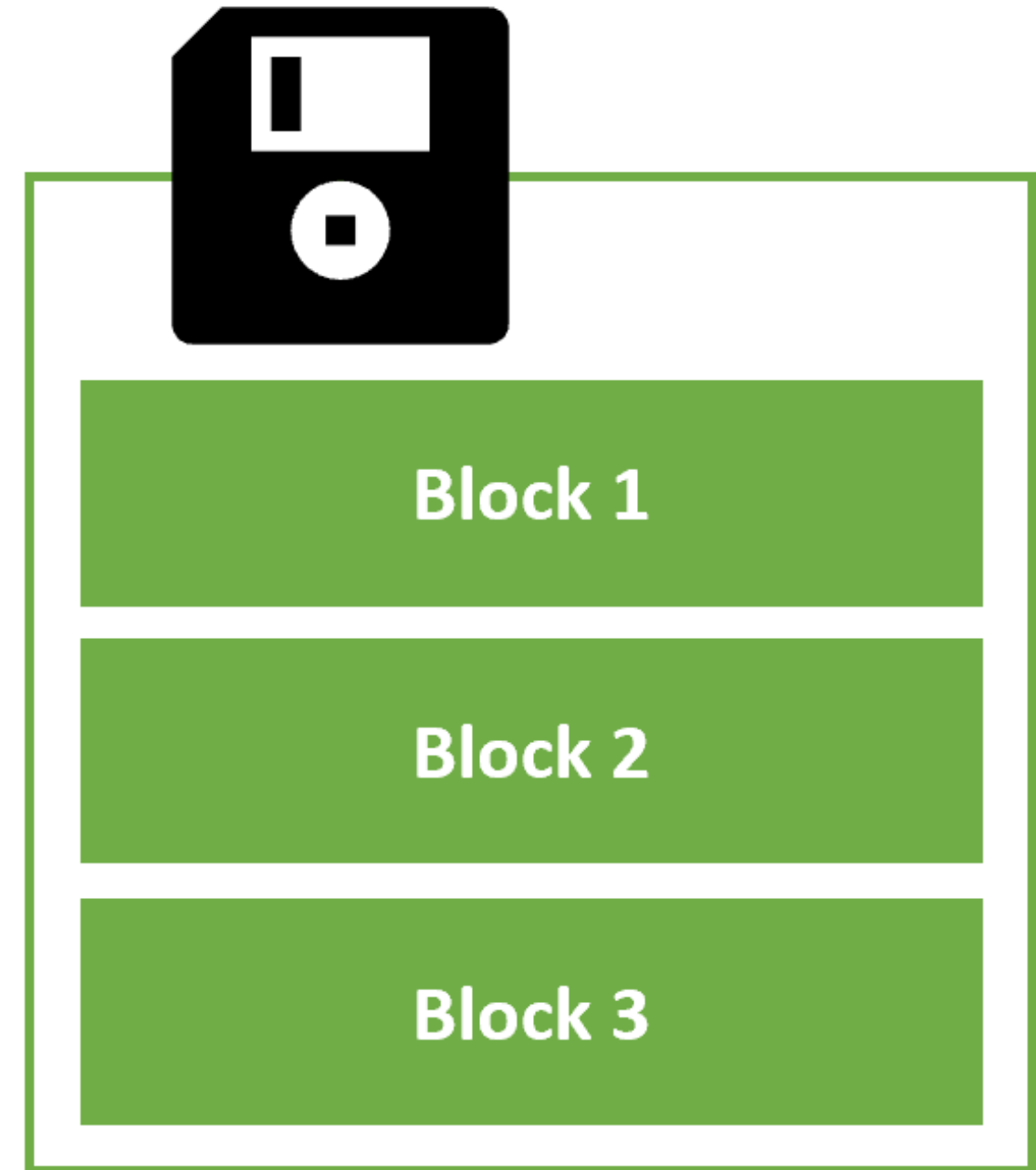
# Why is it important?

- Optimizing queries for speed
- Column store format for data warehouse tables is best for analytic workloads



# Basics of computer storage

- Computers store data in blocks.
- Reads the required blocks when retrieving data.
- Reading fewer blocks increases the overall speed of the process.



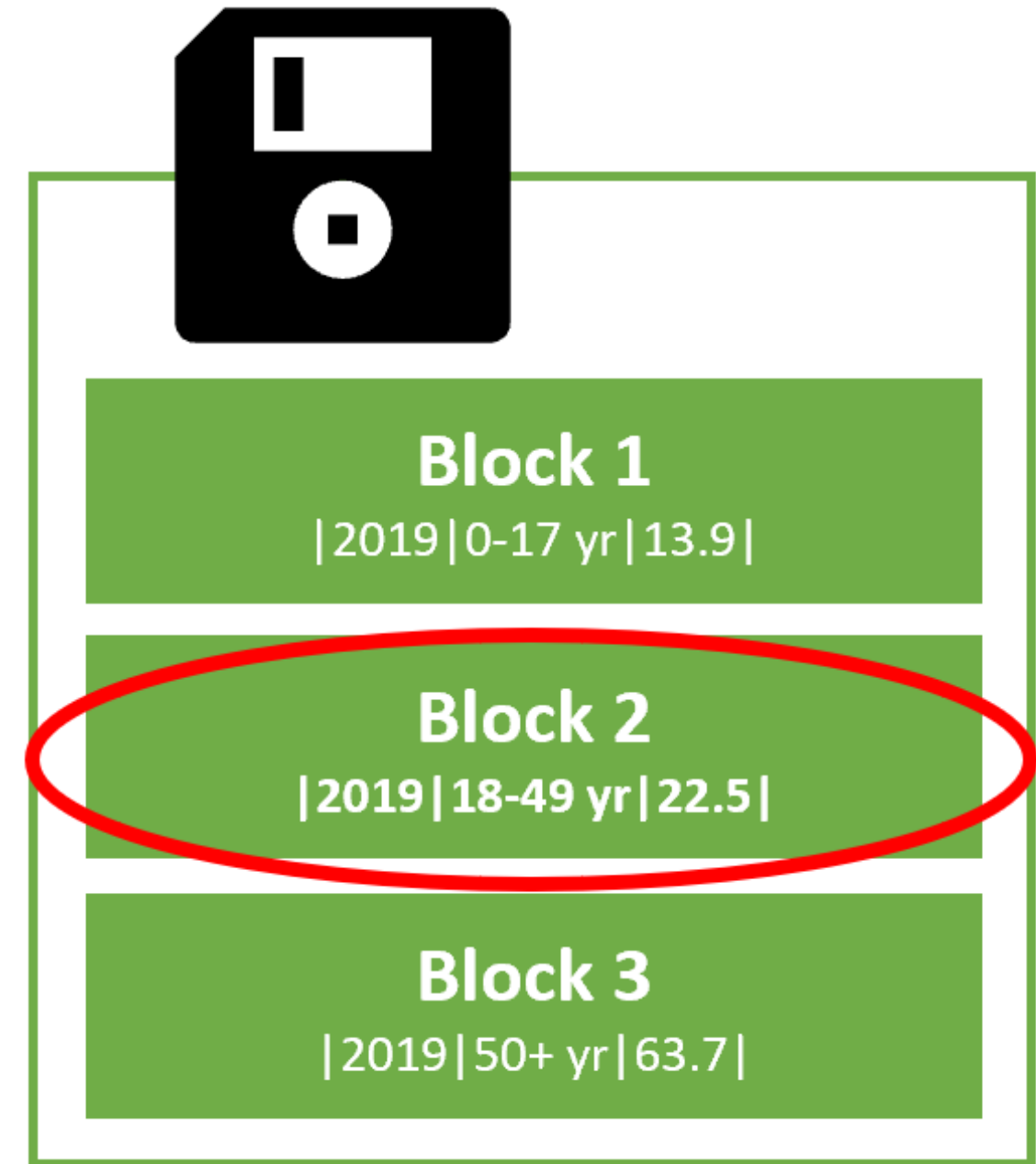
# Example of health table

- CDC (Centers for Disease Control and Prevention)
- Flu infection data by age groups over multiple seasons

SEASON	AGE GROUP	HOSPITALIZATION PERCENTAGE
2019	0-17 yr	13.9%
2019	18-49 yr	22.5%
2019	50+ yr	63.7%
2020	0-17 yr	3.9%
2020	18-49 yr	18.1%
2020	50+ yr	78%
2021	0-17 yr	15.6%
2021	18-49 yr	23.3%
2021	50+ yr	61.1%

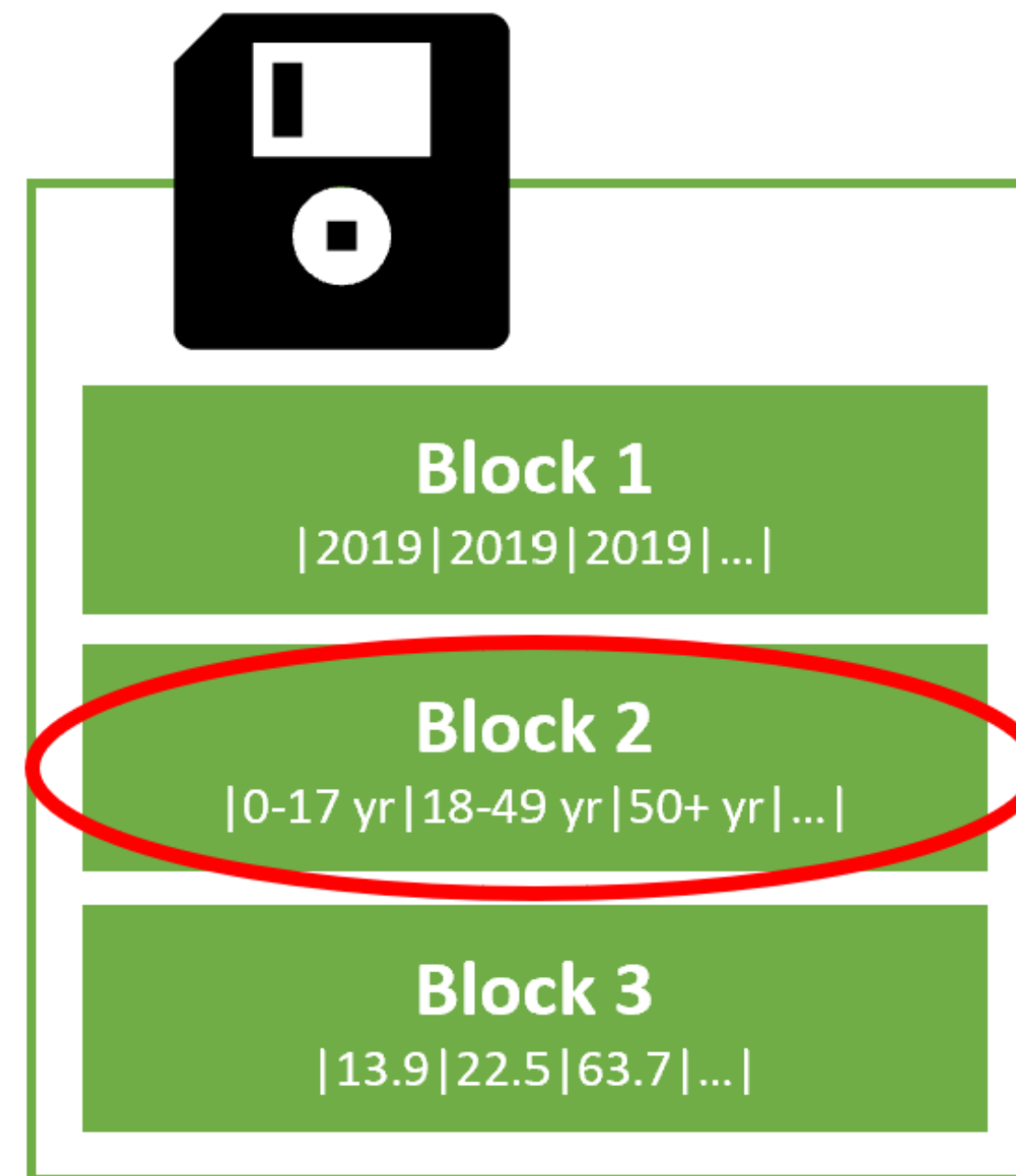
# Row store example

SEASON	AGE GROUP	HOSPITALIZATION PERCENTAGE
2019	0-17 yr	13.9%
<b>2019</b>	<b>18-49 yr</b>	<b>22.5%</b>
2019	50+ yr	63.7%
2020	0-17 yr	3.9%
2020	18-49 yr	18.1%
2020	50+ yr	78%
2021	0-17 yr	15.6%
2021	18-49 yr	23.3%
2021	50+ yr	61.1%



# Column store example

SEASON	AGE GROUP	HOSPITALIZATION PERCENTAGE
2019	<i>0-17 yr</i>	13.9%
2019	<i>18-49 yr</i>	22.5%
2019	<i>50+ yr</i>	63.7%
2020	<i>0-17 yr</i>	3.9%
2020	<i>18-49 yr</i>	18.1%
2020	<i>50+ yr</i>	78%
2021	<i>0-17 yr</i>	15.6%
2021	<i>18-49 yr</i>	23.3%
2021	<i>50+ yr</i>	61.1%



# Summary

## Row Store

- Row data is stored together in blocks
- Ideal for transactional workloads

## Column Store

- Column data is stored together in blocks
- Ideal for analytical workloads
- Better data compression

# It's practice time!

DATA WAREHOUSING CONCEPTS