# CS150A Database

Lu Sun

School of Information Science and Technology

ShanghaiTech University

Dec. 13, 2022

Today:

- Analytics and ML in Data Systems:
  - Part 1
  - Data warehouse & Data Lake

Readings:

- Database Management Systems (DBMS), Chapter 25

*Acknowledgement: Joseph E. Gonzalez@UCBerkeley's scourse notes*

# Transaction Processing vs Analytics

## Online Transaction Processing (OLTP)

- Many small queries:
  - Freq. use of indexes
  - Many writes
  - Concurrency and Logging
- Managing the "Now"
  - Source of truth
- Fairly simple queries with few predicates and relations

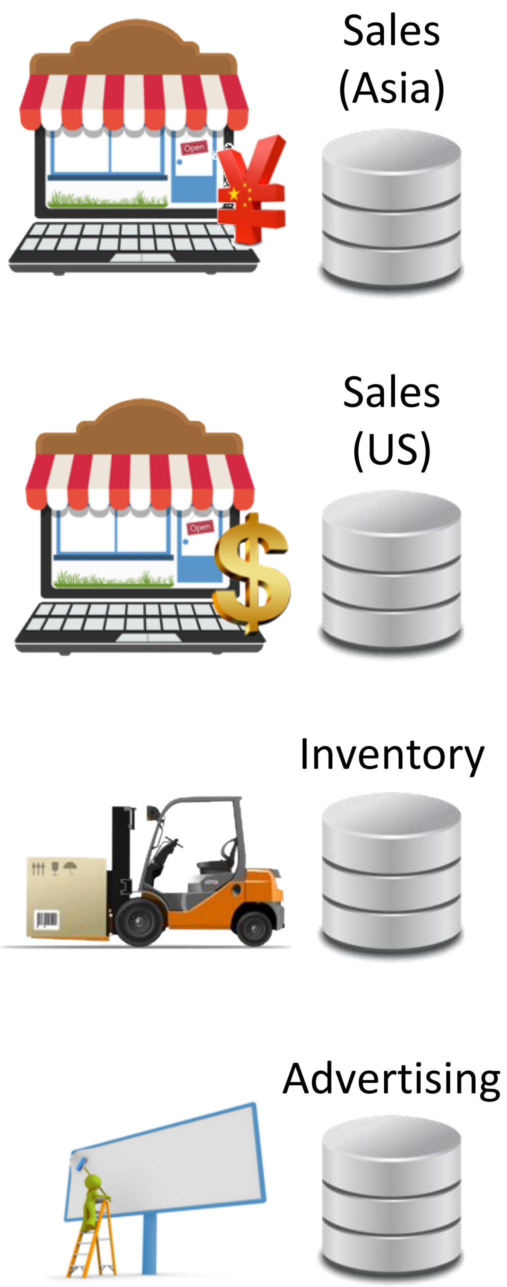
## Online Analytics Processing (OLAP) & Data Mining/ML

- Exploratory Full Table Queries
  - e.g., Agg. Sales Per Market
  - Infrequent (but bulk) writes
  - Limited transaction processing
- Recording the history
  - What was our inventory at the end of last two quarters
- Complex queries with many predicates and many relations

# Analytics & ML queries:

- What was our total sales by market last quarter?
  - Summarization
- What is our predicted sales for next quarter?
  - Forecasting
- Which users will likely leave our service?
  - Churn prediction
- If a user buys X what else are they likely to buy?
  - Collaborative filtering & Recommender Systems

You embark on the journey of a data scientist …




# Data Everywhere

- Stored Across Multiple Operational OLTP Systems
  - Different formats (e.g., currency)
    - Different schemas (acquisitions ...)
  - Mission critical
    - Serving live sales traffic
    - Managing inventory
    - ... Be careful!
- Often limited historical data

We would like a consolidated, cleaned, historical snapshot of the data.

# Data Warehouse


Collects and organizes historical data from multiple sources

Data is *periodically* **ETL**ed into the data warehouse:

- **Extracted** from remote sources
- **Transformed** to standard schemas
- **Loaded** into the (typically) relational system

# Extracting Data from Sources

- Need to collect data from multiples sources
  - Various RDBMS vendors
  - Structured files JSON, XML

- Often done using SQL interfaces

- Validate extracted data
  - Flag corrupted records …

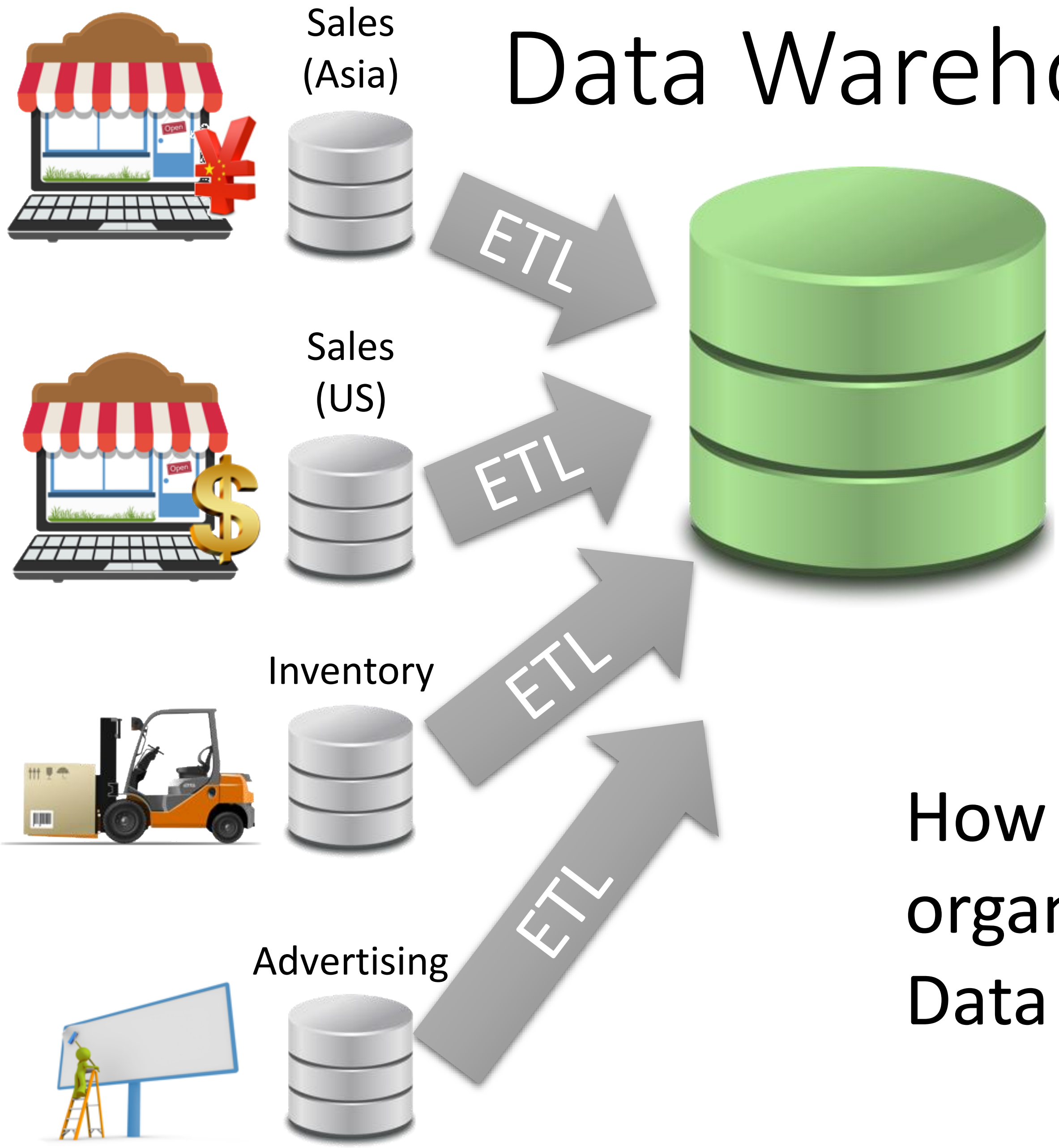
# Transforming “Cleaning” Data

- Additional data validation and filtering
- Schema manipulation
  - Extract key fields
  - Encoding text
  - Verifying and enforcing constraints
- Data normalization (time zones, currency)

# Loading Data

- Data is bulk loaded into large relations
  - Fact tables … (more on this later)
- Update:
  - Indexes
  - Metadata tables: Data about the data
    - When and how was it collected
    - Meaning of fields
  - Updating materialized views …
- Occasionally move older data to archival storage
  - Data aging

# Data Warehouse




How is data organized in the Data Warehouse?

# Example Sales Data:

| pname | category | price | qty | date | day | city | state | country |
|---|---|---|---|---|---|---|---|---|
| Corn | Food | 25 | 25 | 3/30/16 | Wed. | Omaha | NE | USA |
| Corn | Food | 25 | 8 | 3/31/16 | Thu. | Omaha | NE | USA |
| Corn | Food | 25 | 15 | 4/1/16 | Fri. | Omaha | NE | USA |
| Galaxy 1 | Phones | 18 | 30 | 1/30/16 | Wed. | Omaha | NE | USA |
| Galaxy 1 | Phones | 18 | 20 | 3/31/16 | Thu. | Omaha | NE | USA |
| Galaxy 1 | Phones | 18 | 50 | 4/1/16 | Fri. | Omaha | NE | USA |
| Galaxy 1 | Phones | 18 | 8 | 1/30/16 | Wed. | Omaha | NE | USA |
| Peanuts | Food | 2 | 45 | 3/31/16 | Thu. | Seoul | | Korea |
| Galaxy 1 | Phones | 18 | 100 | 4/1/16 | Fri. | Seoul | | Korea |

- **Big** table: many *columns* and *rows*
  - Substantial redundancy → expensive to store and access
- Could we organize the data a little better?

# Multidimensional Data Model

*Sales* **Fact Table**

| pid | timeid | locid | sales |
|---|---|---|---|
| 11 | 1 | 1 | 25 |
| 11 | 2 | 1 | 8 |
| 11 | 3 | 1 | 15 |
| 12 | 1 | 1 | 30 |
| 12 | 2 | 1 | 20 |
| 12 | 3 | 1 | 50 |
| 12 | 1 | 1 | 8 |
| 13 | 2 | 1 | 10 |
| 13 | 3 | 1 | 10 |
| 11 | 1 | 2 | 35 |
| 11 | 2 | 2 | 22 |
| 11 | 3 | 2 | 10 |
| 12 | 1 | 2 | 26 |

**Dimension Tables**

Locations

| locid | city | state | country |
|---|---|---|---|
| 1 | Omaha | Nebraska | USA |
| 2 | Seoul | | Korea |
| 5 | Richmond | Virginia | USA |

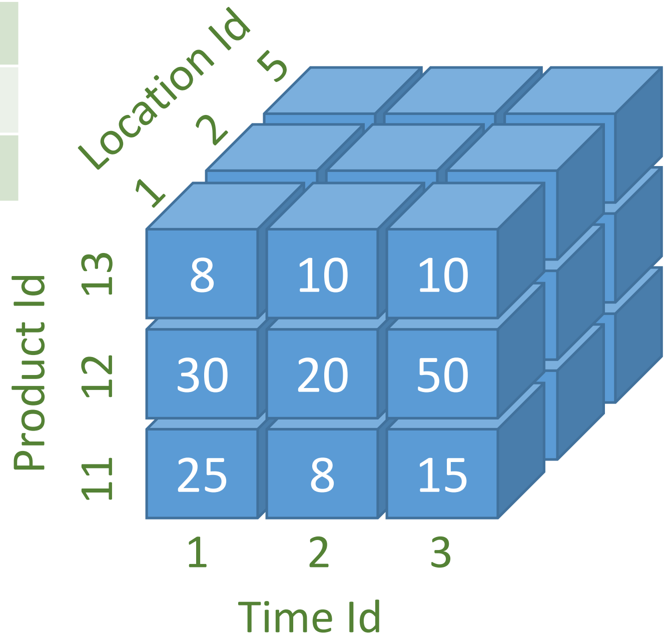
Products

| pid | pname | category | price |
|---|---|---|---|
| 11 | Corn | Food | 25 |
| 12 | Galaxy 1 | Phones | 18 |
| 13 | Peanuts | Food | 2 |

Time

| timeid | Date | Day |
|---|---|---|
| 1 | 3/30/16 | Wed. |
| 2 | 3/31/16 | Thu. |
| 3 | 4/1/16 | Fri. |

- Multidimensional “Cube” of data



# Multidimensional Data Model

*Sales* **Fact Table**

| pid | timeid | locid | sales |
|---|---|---|---|
| 11 | 1 | 1 | 25 |
| 11 | 2 | 1 | 8 |
| 11 | 3 | 1 | 15 |
| 12 | 1 | 1 | 30 |
| 12 | 2 | 1 | 20 |
| 12 | 3 | 1 | 50 |
| 12 | 1 | 1 | 8 |
| 13 | 2 | 1 | 10 |
| 13 | 3 | 1 | 10 |
| 11 | 1 | 2 | 35 |
| 11 | 2 | 2 | 22 |
| 11 | 3 | 2 | 10 |
| 12 | 1 | 2 | 26 |

Locations

| locid | city | state | country |
|---|---|---|---|
| 1 | Omaha | Nebraska | USA |
| 2 | Seoul | | Korea |
| 5 | Richmond | Virginia | USA |

**Dimension Tables**

Products

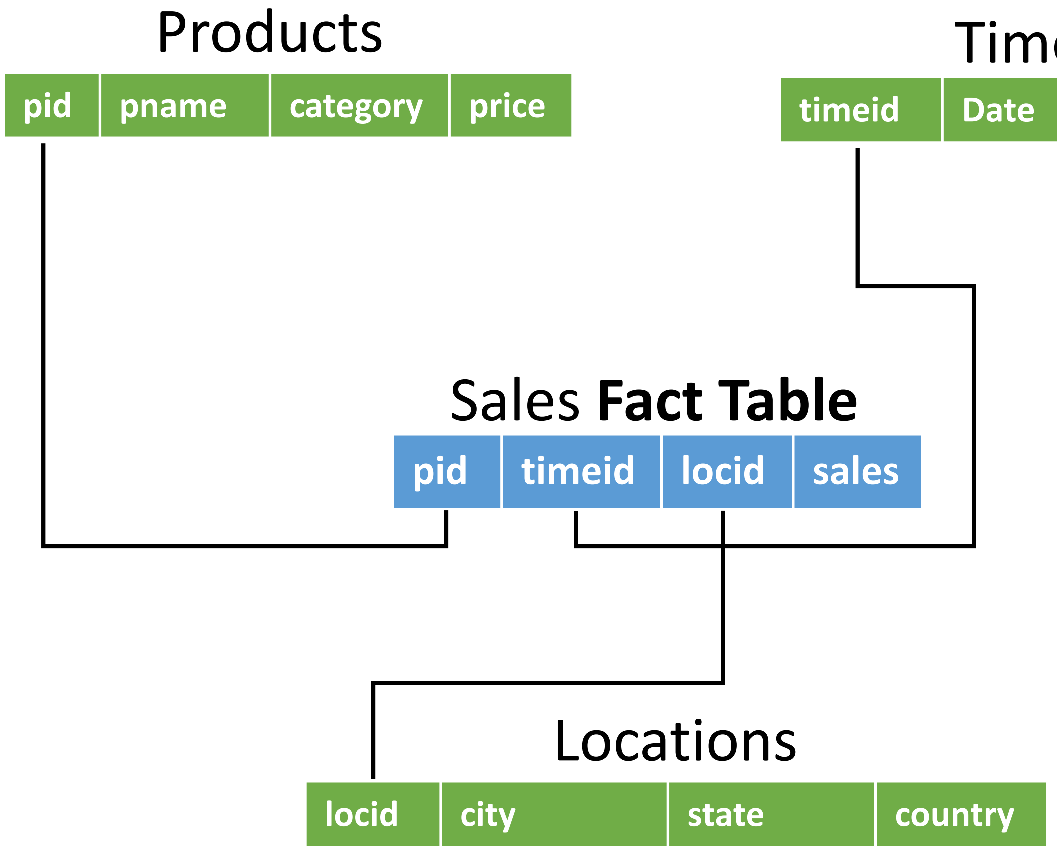
| pid | pname | category | price |
|---|---|---|---|
| 11 | Corn | Food | 25 |
| 12 | Galaxy 1 | Phones | 18 |
| 13 | Peanuts | Food | 2 |

Time

| timeid | Date | Day |
|---|---|---|
| 1 | 3/30/16 | Wed. |
| 2 | 3/31/16 | Thu. |
| 3 | 4/1/16 | Fri. |

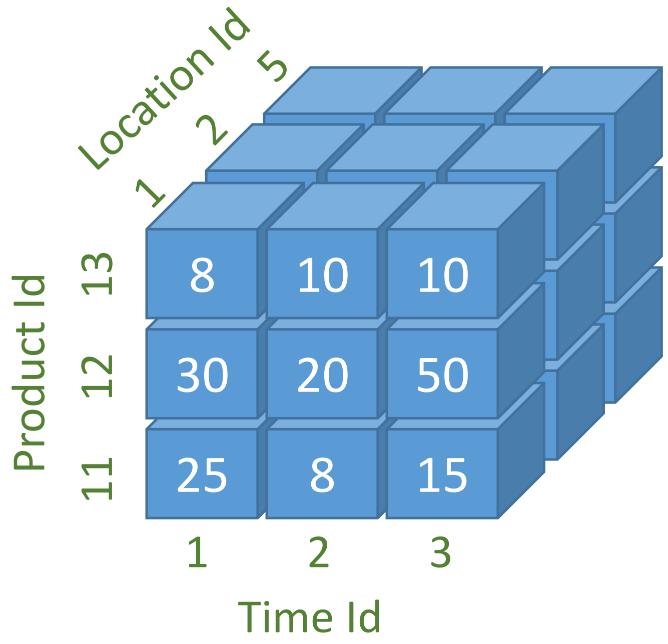
- Sales Fact Table
  - Contains only foreign keys → Efficient
- Easy to manage Dimensions
  - Galaxy1 → Phablet: no need to update **Fact Table**
- Normalization
  - Minimizing redundancy
  - More on this later …

# Multidimensional Data: **Star Schema**

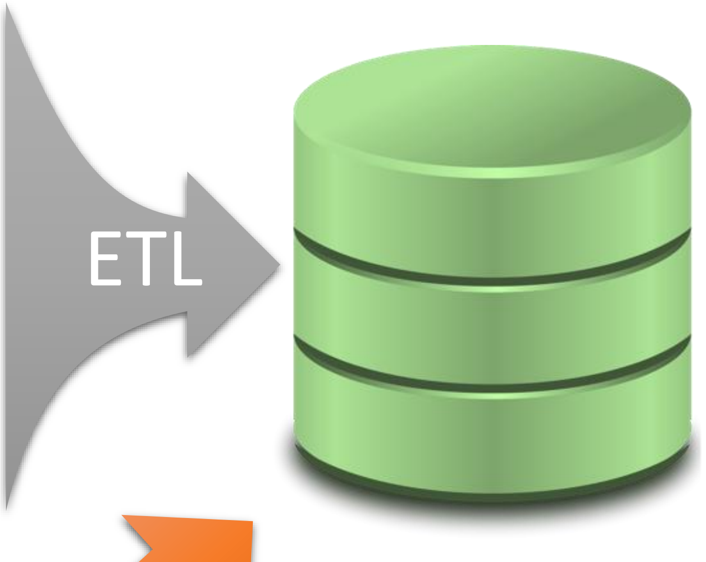



**Dimension Tables**

← This looks like a star …



# Data Warehouse



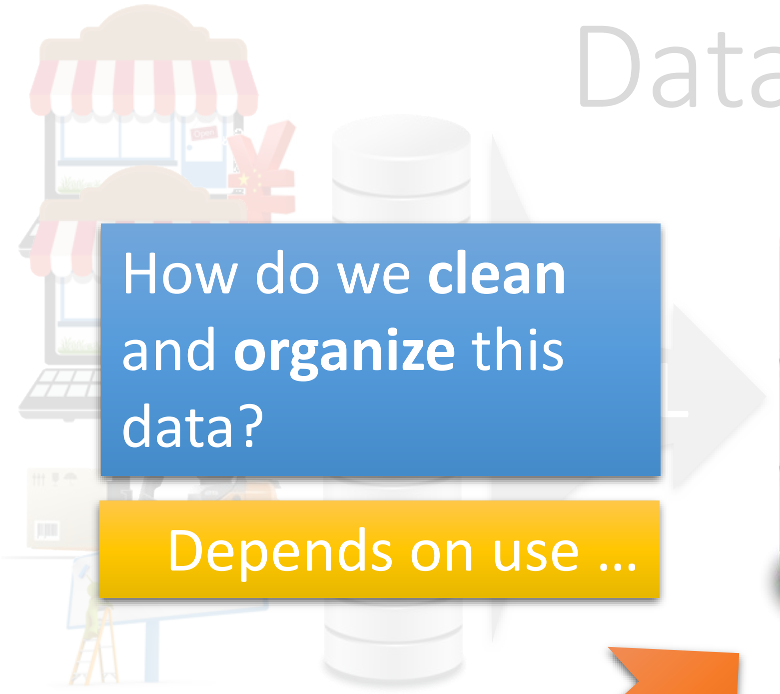


Text/Log Data

Photos & Videos


- How do we deal with semi-structured and unstructured data?
- Do we really want to force a schema on load?

# Data Warehouse



How do we **clean** and **organize** this data?

Depends on use …

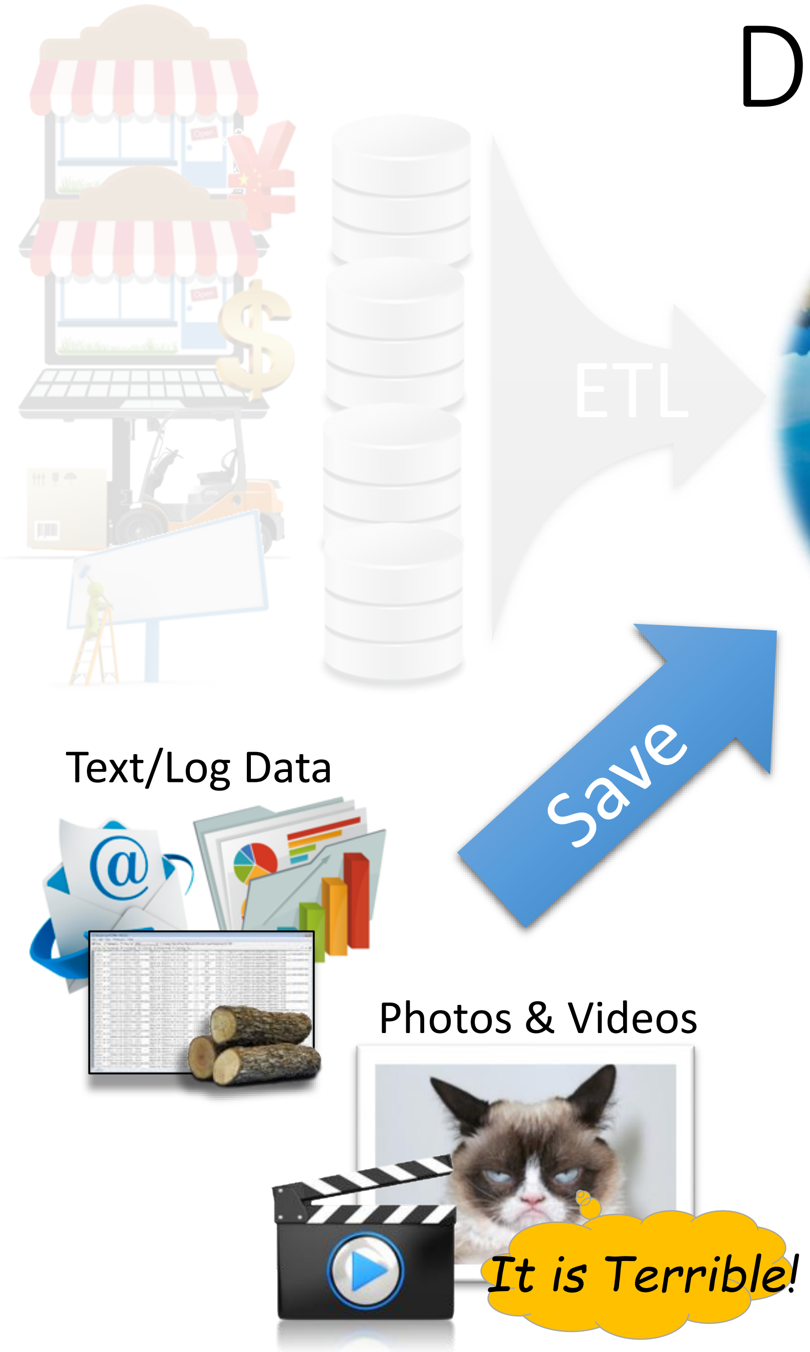

Text/Log Data

Photos & Videos


How do we **load** and **process** this data in a relation system?

Depends on use …
Can be difficult ...
Requires thought ...

# Data Lake*




*Still being defined…
[Buzzword Disclaimer]

**Big Idea:**

Maintain a copy of all the data in one place and *free** data consumers to choose how to transform and use it.

*free to solve all the problems themselves

# Origin of the Data Lake

- Attributed to James Dixon, CTO of Pentaho, 2010

“If you think of a **datamart** as a store of bottled water – **cleansed** and **packaged** and **structured** for **easy consumption** – the **data lake** is a **large body of water** in a more **natural state**.

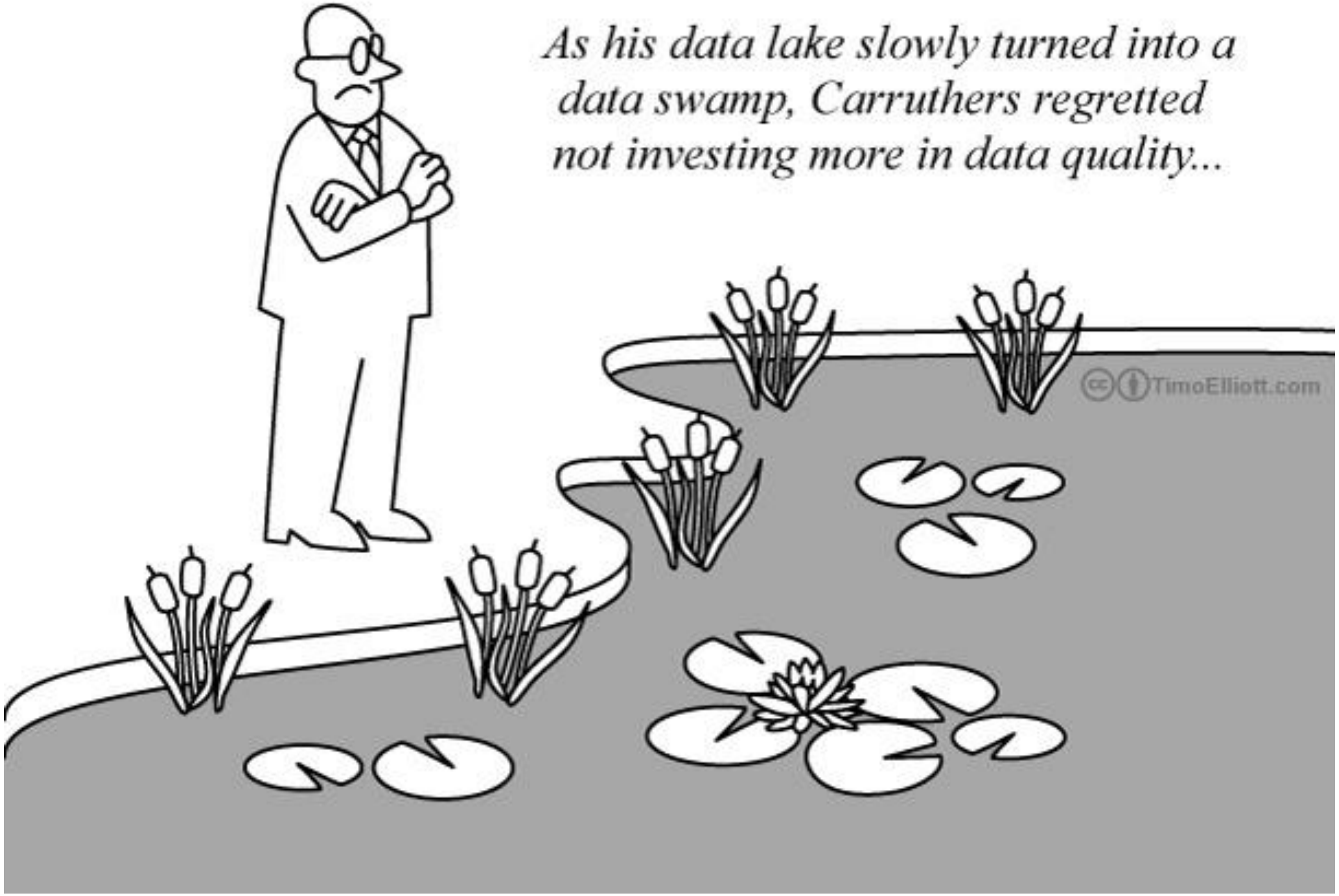
The contents of the data lake **stream in** from a source to fill the lake, and various users of the lake can come to examine, dive in, or **take samples**.”

https://jamesdixon.wordpress.com/2010/10/14/pentaho-hadoop-and-data-lakes/

# Data Lake


- Store unstructured data in **raw form**
  - Schema-on-**Read**: *determine the best organization* ***when data is used***
  - **Contrast:** Data Warehouses are Schema-on-Load (ET***<u>L</u>***)
    - Plan ahead (Fact tables and Dimensions)
- Often much **larger** than data warehouses
- Technologies
  - **Storage:** Large distributed file systems (e.g., HDFS)
    - Semi-structured formats (JSON, Parquet)
  - **Computation:** Map-Reduce
    - Recent trend to add SQL (or SQL like) functionality
- More Agile (?):
  - Don't worry about schema & verification when loading
  - Disaggregated compute and storage → BYOF
    - bring your own compute frameworks …

- **What could go wrong?**




http://timoelliott.com/blog/2014/12/from-data-lakes-to-data-swamps.html

# Data Lake → Data Swamp

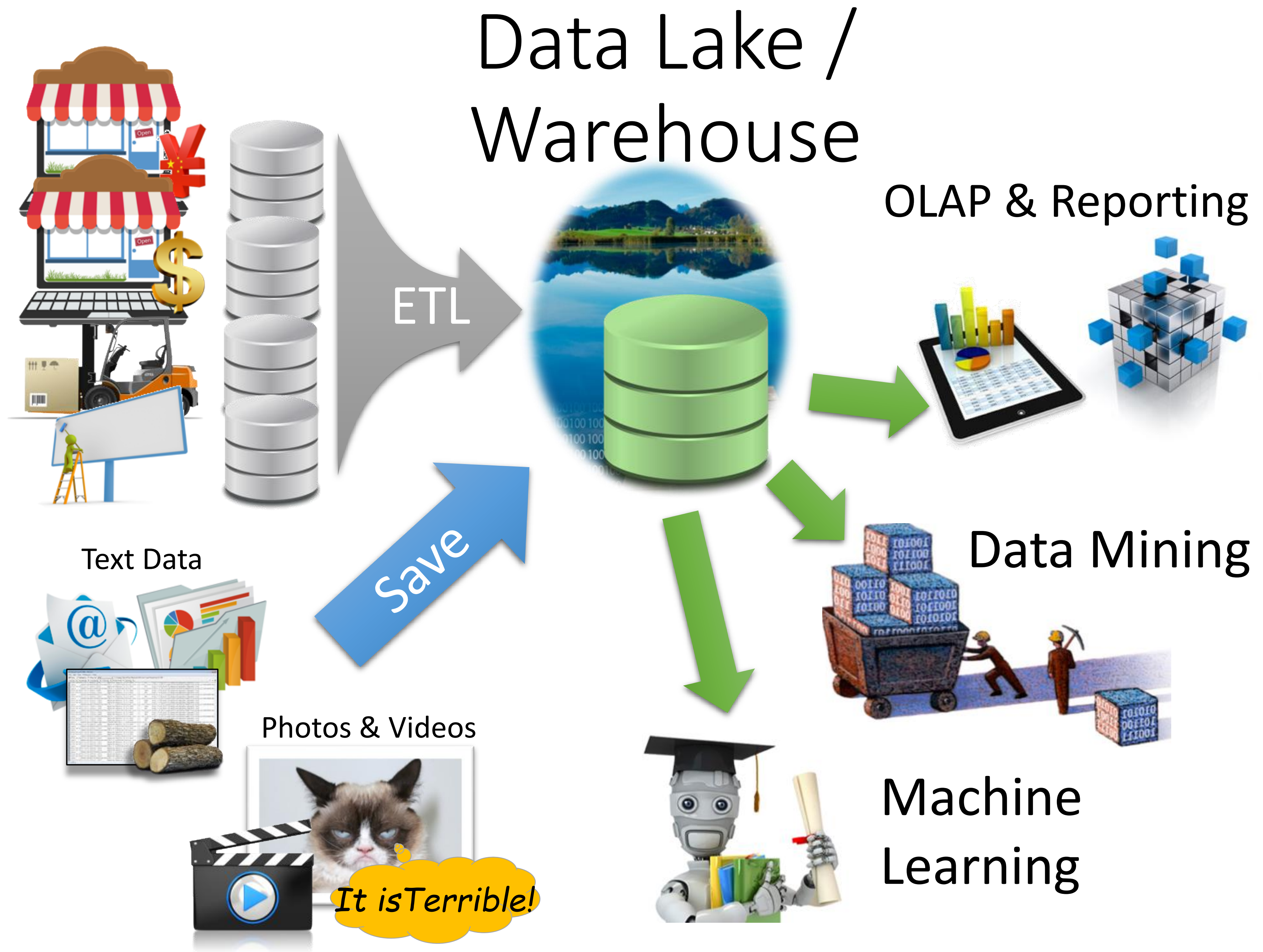

- Cultural shift: *Curate* → Save **Everything**!
  - Signal to Noise ratio drops …
- Limited data governance → more agile →
  - **What** does it contain? **What** are all the **"fields"**
  - **When** and **how** and **from where** was it created
- Without cleaning and verification we begin to collect a rich history of **dirty data**
- Limited compatible with traditional tools

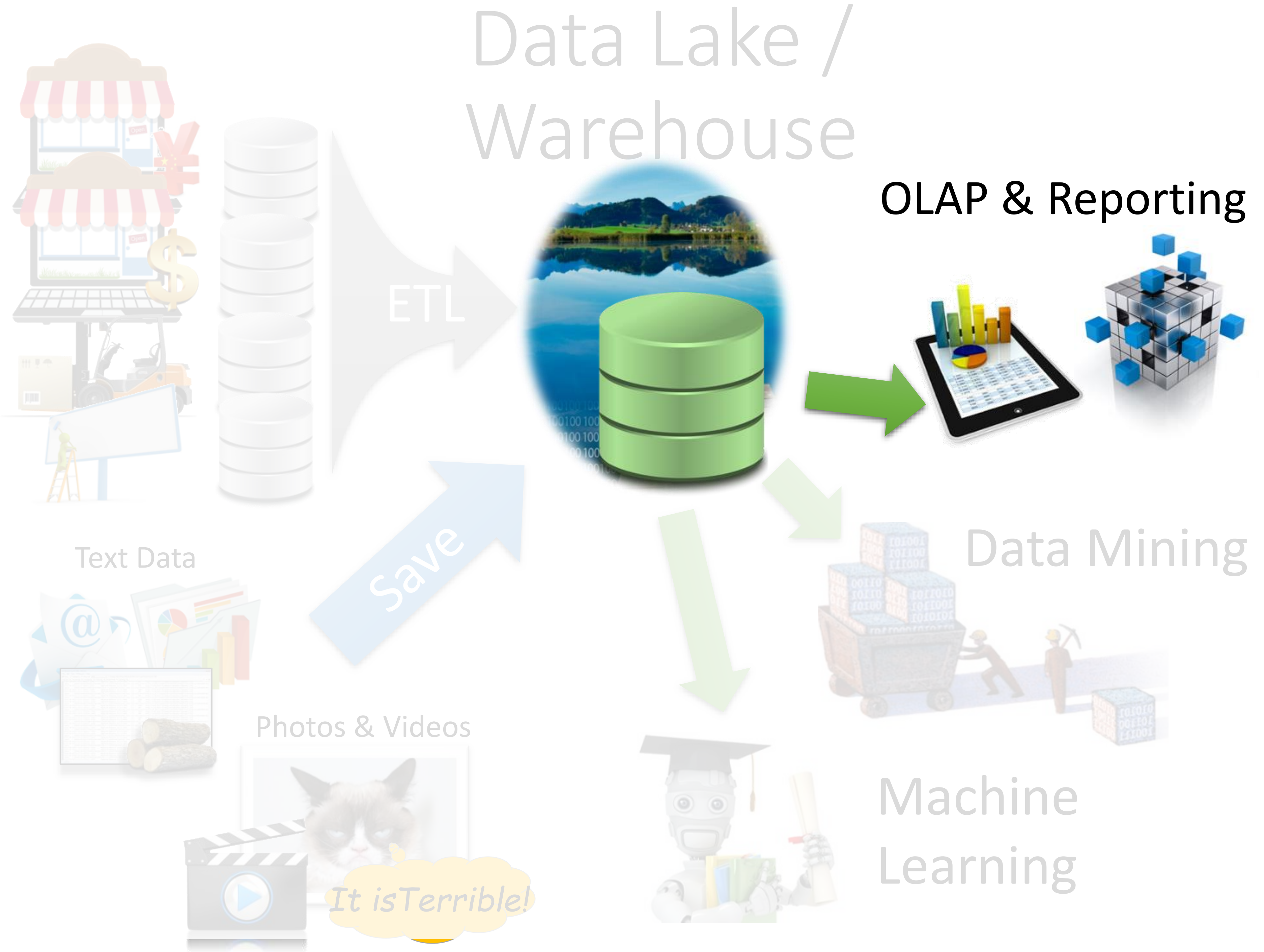
# Data Lakes *Appear* to be Maturing

- Relational data-models + SQL:
  - **Hive:** SQL on top of Hadoop Map-Reduce
  - **SparkSQL:** SQL on top of Spark
- Tools are Improving:
  - Better data cleaning
  - Catalog Managers
  - Improved semi-structured “raw” data formats
- Improved data governance
  - Organization are recognizing the issues

# Data Lake / Warehouse


What do we do with all this data?


Data Lake /
Warehouse
ETL
Save
OLAP & Reporting
Data Mining
Machine
Learning
Text Data
Photos & Videos
It isTerrible!


Data Lake / Warehouse
OLAP & Reporting
ETL
Text Data
Save
Data Mining
Photos & Videos
It isTerrible!
Machine Learning

# Online Analytics Processing (OLAP)

Users interact with multidimensional data:

- Constructing ad-hoc and often complex SQL queries
- Using graphical tools that to construct queries
- Sharing views that summarize data across important dimensions

# Cross Tabulation (Pivot Tables)

| Item | Color | Quantity |
|---|---|---|
| Desk | Blue | 2 |
| Desk | Red | 3 |
| Sofa | Blue | 4 |
| Sofa | Red | 5 |

| | | Item | | |
|---|---|---|---|---|
| | | Desk | Sofa | ***Sum*** |
| Color | Blue | 2 | 4 | ***6*** |
| | Red | 3 | 5 | ***8*** |
| | ***Sum*** | ***5*** | ***9*** | ***14*** |

- Aggregate data across pairs of dimensions
  - **Pivot Tables:** *graphical interface* to select dimensions and aggregation function (e.g., SUM, MAX, MEAN)
  - **GROUP BY** queries

➢Related to contingency tables and marginalization in stats.

- What about many dimensions?

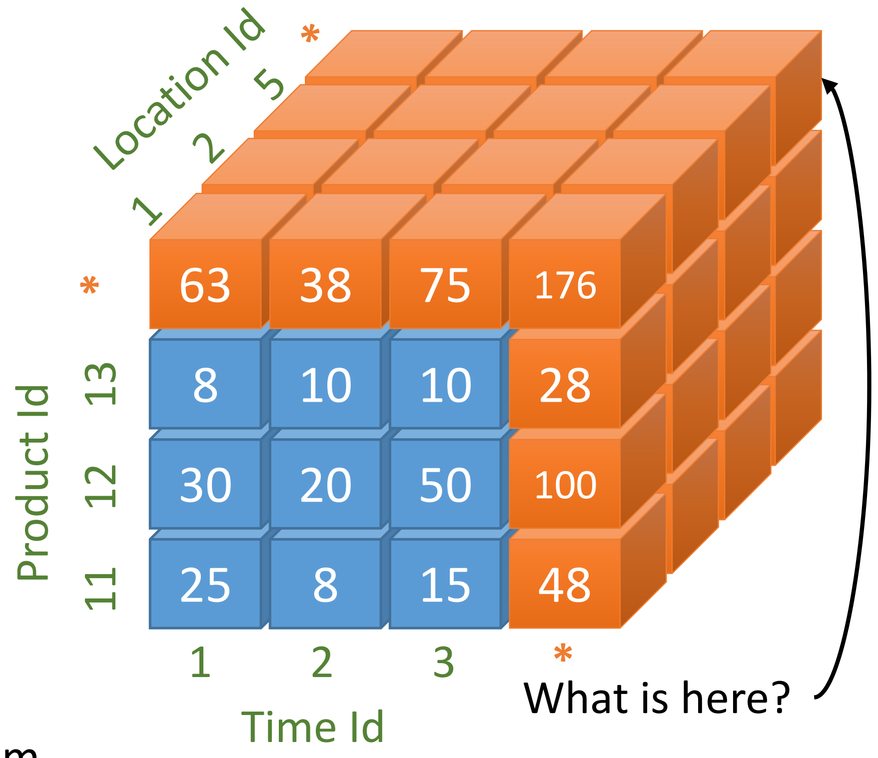
# Cube Operator

- Generalizes cross-tabulation to higher dimensions.

➢In SQL:

**SELECT** Item, Color, **SUM**(Quantity) **AS** QtySum
**FROM** Furniture
**GROUP BY** ***CUBE*** (Item, Color);

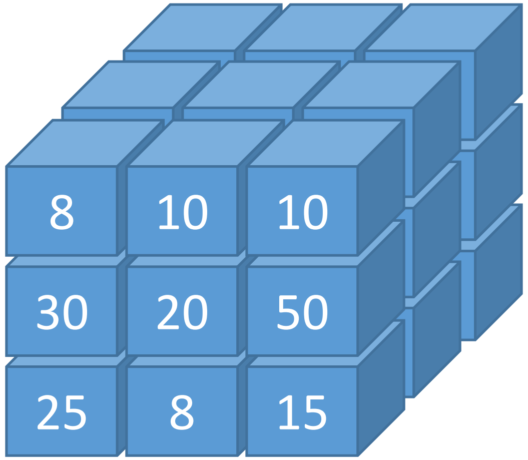
| Item | Color | Quantity |
|---|---|---|
| Desk | Blue | 2 |
| Desk | Red | 3 |
| Sofa | Blue | 4 |
| Sofa | Red | 5 |

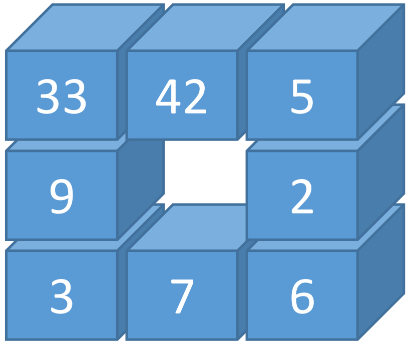



| Item | Color | QtySum |
|---|---|---|
| Desk | Blue | 2 |
| Desk | Red | 3 |
| Desk | * | 5 |
| Sofa | Blue | 4 |
| Sofa | Red | 5 |
| Sofa | * | 9 |
| * | * | 14 |
| * | Blue | 6 |
| * | Red | 8 |

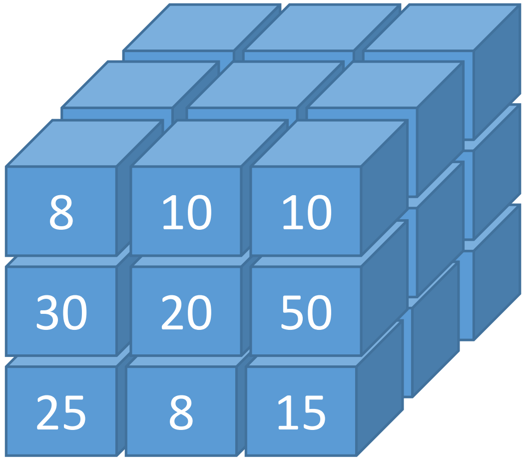
# OLAP Queries

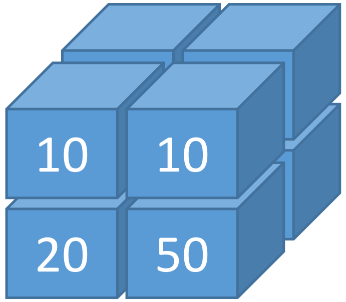
- **Slicing:** *selecting a value for a dimension*



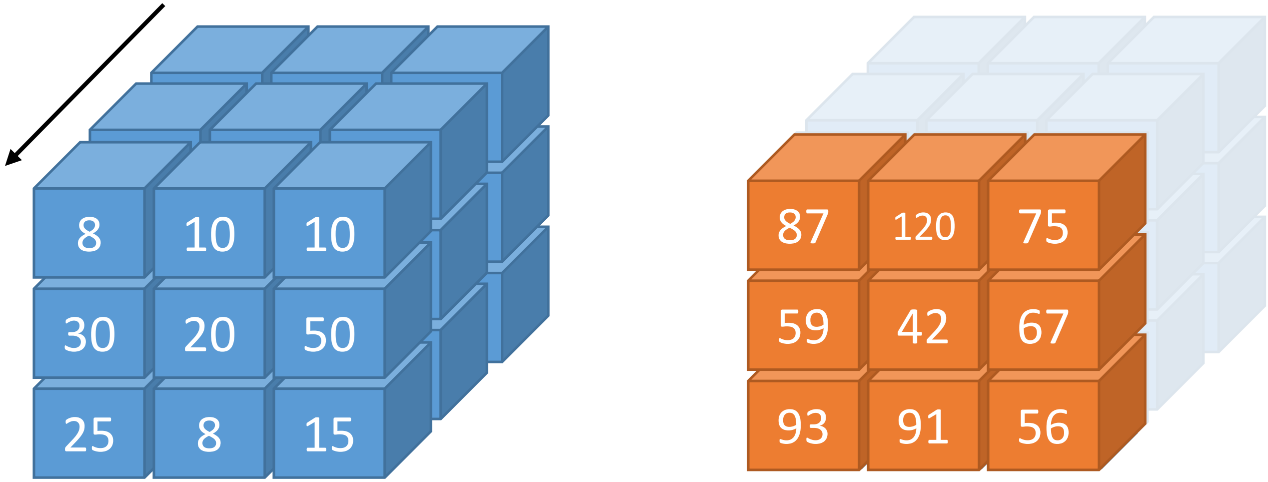

- **Dicing:** *selecting a range of values in multiple dimension*



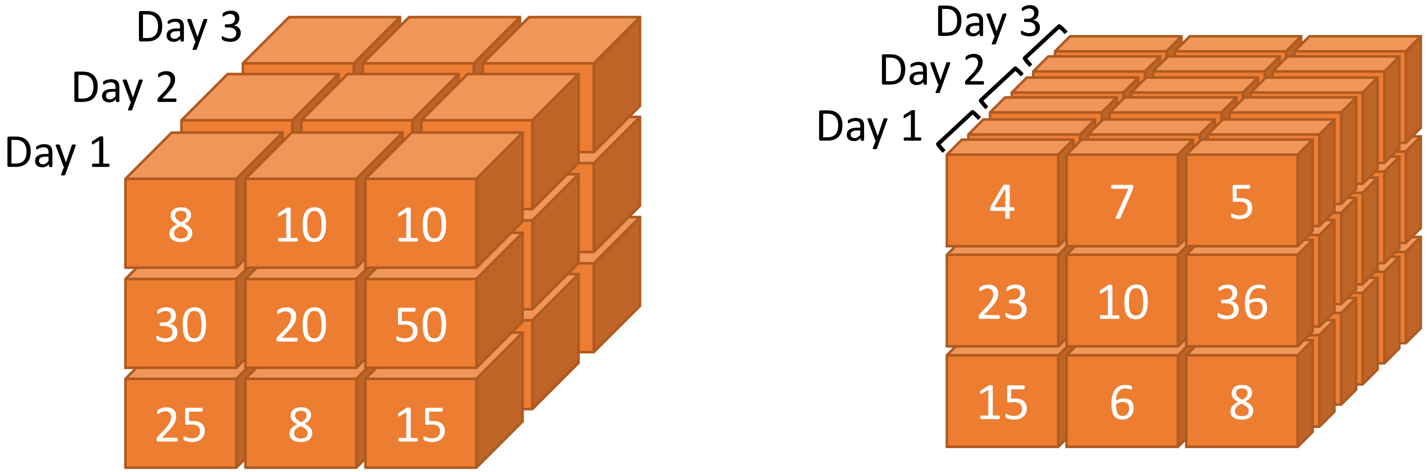



# OLAP Queries

- **Rollup:** *Aggregating along a dimension*

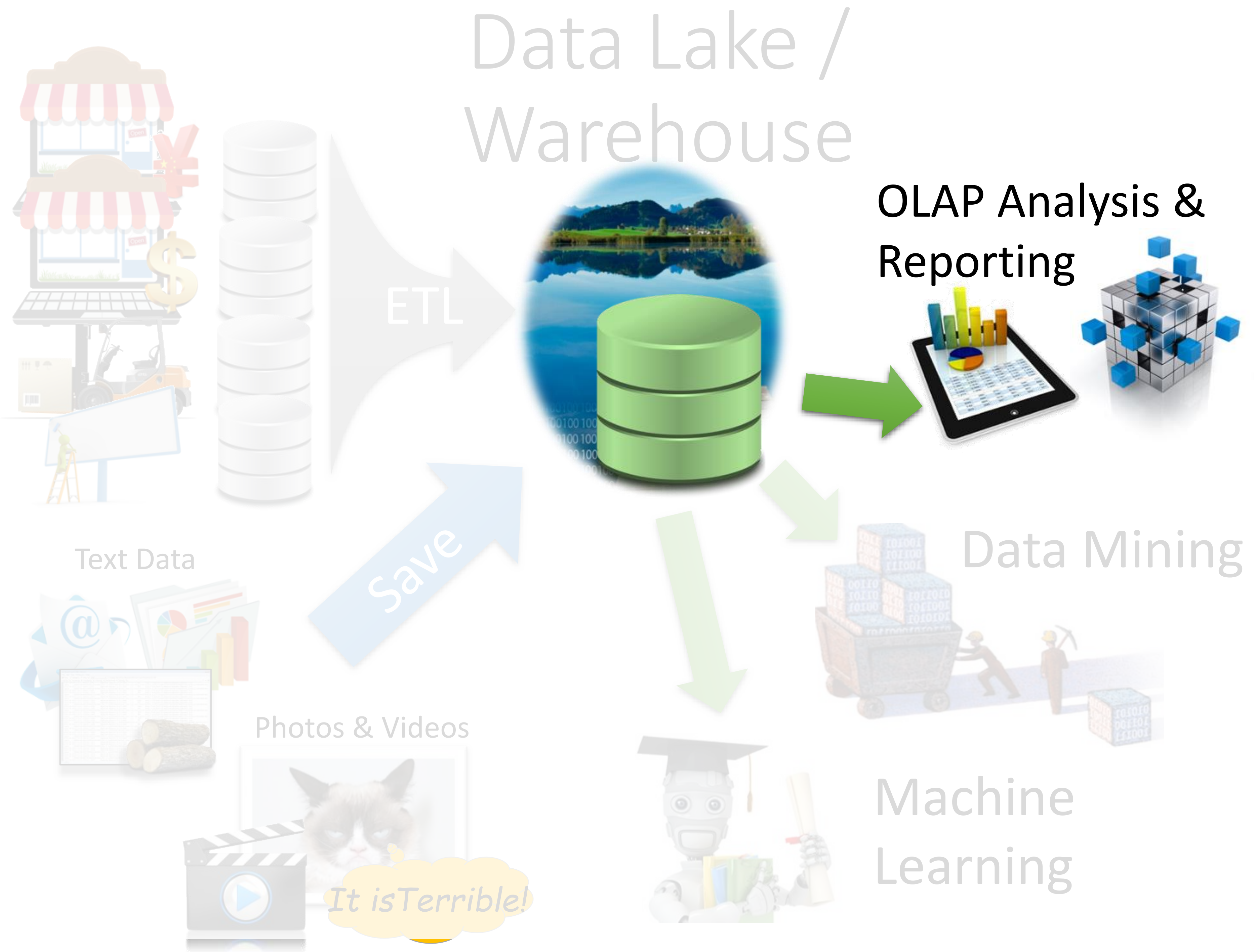


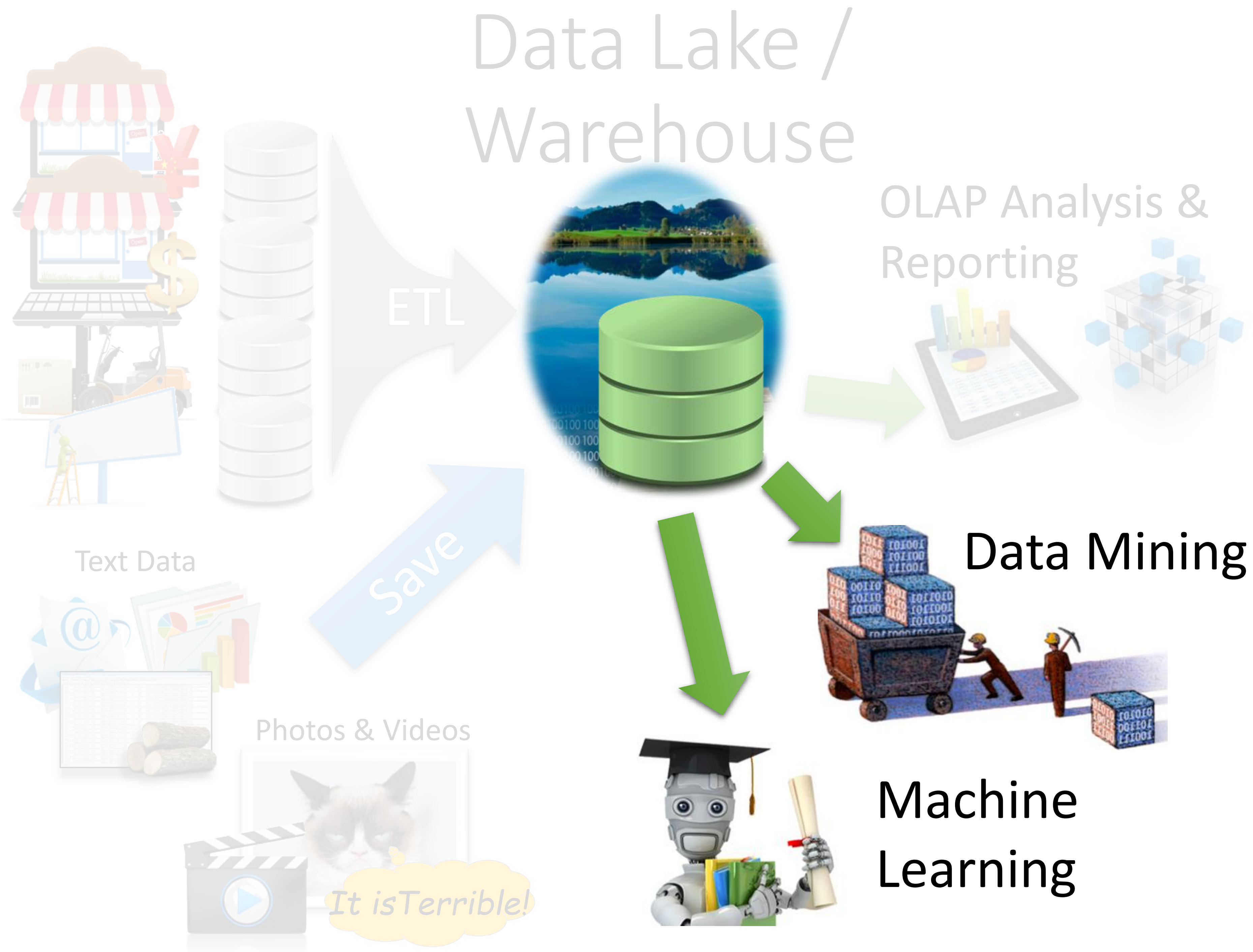

- **Drill-Down:** *de-aggregating along a dimension*



# Reporting and Business Intelligence (BI)

- Use high-level tools to interact with their data:
  - Automatically generate SQL queries
    - Queries can get big!
- Common!

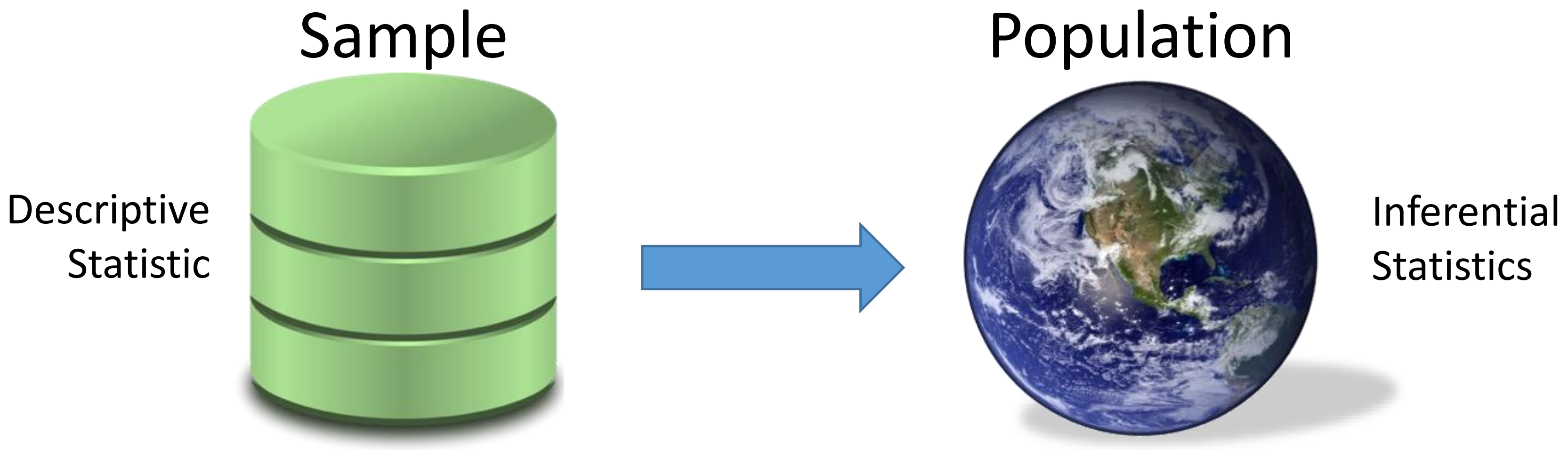

Data Lake / Warehouse
ETL
OLAP Analysis & Reporting
Save
Text Data
Photos & Videos
It isTerrible!
Data Mining
Machine Learning


Data Lake /
Warehouse
ETL
OLAP Analysis &
Reporting
Text Data
Save
Photos & Videos
It isTerrible!
Data Mining
Machine
Learning

# Knowledge Discovery in Databases (KDD)

- Process of extracting ***knowledge*** from a ***data***
  - What does this mean?

# **Descriptive** vs. **Inferential** Statistics




- **Descriptive Statistics:** *describe* the sample data
  - Example: *Average* sales last quarter
  - Can be **measured directly** from the database
- **Inferential Statistics:** *estimate* the population
  - Example: *Expected* sales next quarter
  - May be **estimated** using descriptive statistics

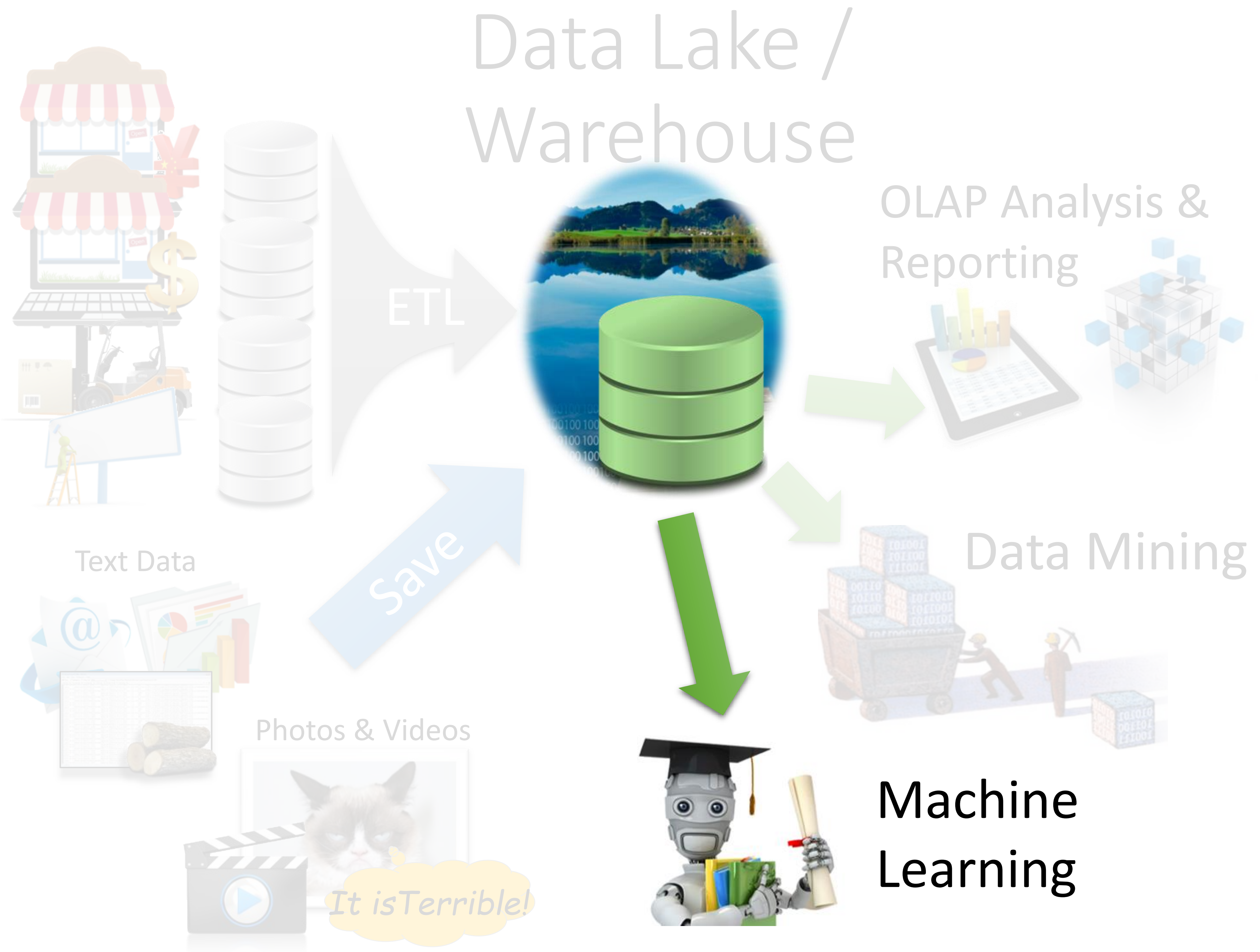
# The Basic KDD Process

- **Data Selection:** *What data do I need for a given task?*
  - If data was already collected, how was the data collected?

- **Data Cleaning:** *Preparing the data for a given task*
  - Typically most challenging (time consuming) part.
  - Why might ETL not be enough?

- **Data Mining & ML:** *Running algorithms to infer patterns*
  - The fun part!  Many tools, many options, complex tradeoffs.

- **Evaluation:** *Verifying that patterns are significant*
  - Algorithms will typically find patterns especially when none exist.



Data Lake /
Warehouse
ETL
OLAP Analysis &
Reporting
Data Mining
Text Data
Save
Photos & Videos
It isTerrible!
Machine
Learning

# What is Machine Learning?

Study of algorithms that:

- That improve their **performance**
  - Ability to understand what you are saying
- at some **task**
  - Voice recognition
- through **experience**
  - Transcribed speech data

-- Prof. Tom Mitchell, *CMU*

*"Machine Learning is the **second best** solution to any problem.*
*The **first best** is of course to **solve the problem** directly."*

-- Prof. Yaser S. Abu-Mostafa, *Caltech*

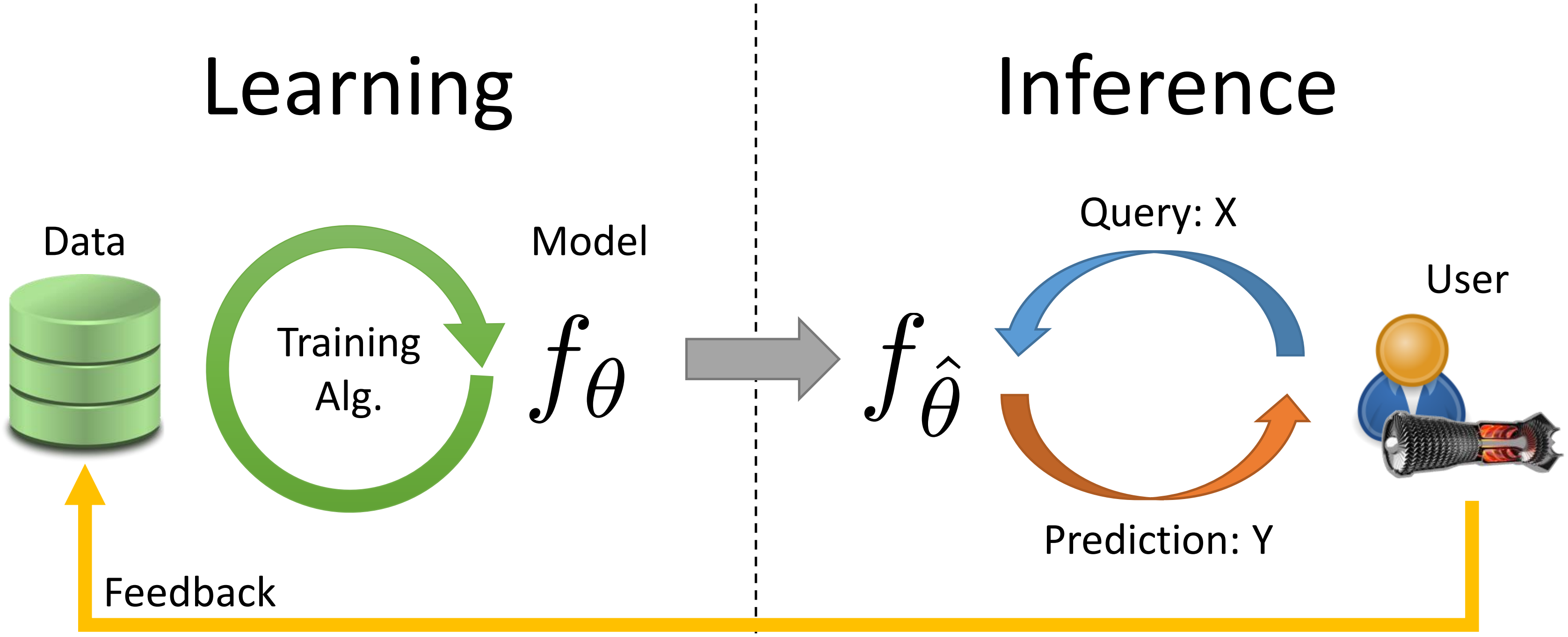
How would you write a program to recognize human speech?

# You use ML every day!

What machine learning do you use every day?

- Spam detection
- Voice recognition
- Face tagging on Facebook
- Ad Targeting
- Credit card fraud detection
- Others? …

# Machine Learning Lifecycle

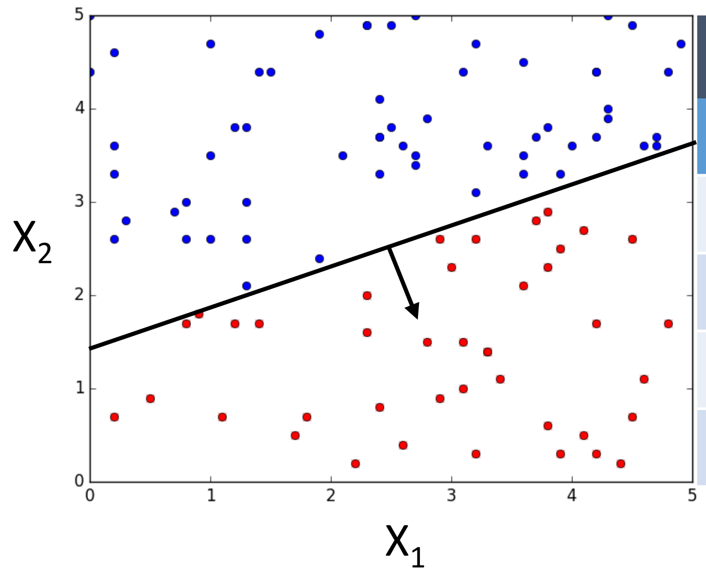



- Typically a time consuming iterative batch process
  - Feature engineering
  - Validation

➢ Focus is on making fast robust predictions
  - Monitoring and tracking feedback
  - Materialization + fast model inference

# Learning: *Fitting the Model*

- **Training Data**
  - **X**: Features
  - **Y**: Label/Obs.
- Learn a function that **generalizes** the relationship between *X* and *Y*




| X | | Y |
|---|---|---|
| $X_1$ | $X_2$ | |
| 3.9 | 3.1 | T |
| 0.2 | 0.1 | F |
| 4.2 | 0.9 | F |
| 1.1 | 2.9 | T |

Features

Function class / Model Family

$$f_\theta(X) \rightarrow Y$$

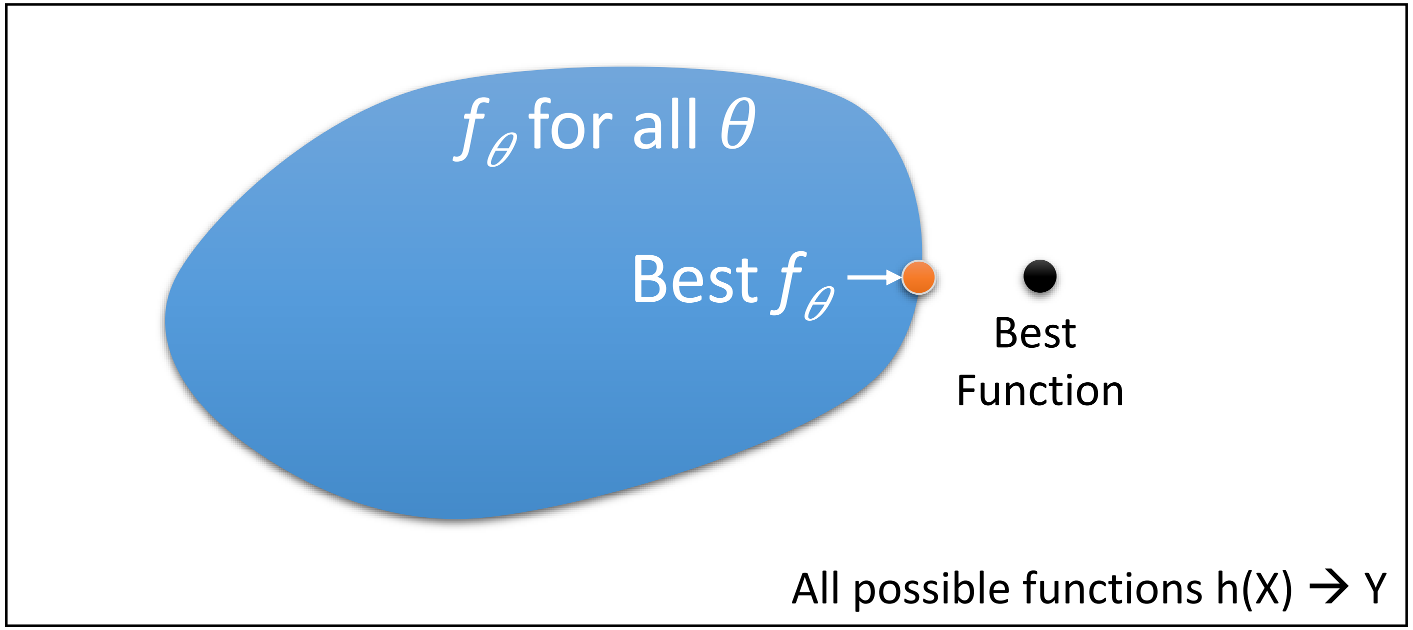
Labels / Observations

Model Parameters

# Finding the Best Parameters

$$f_\theta(X) \to Y$$

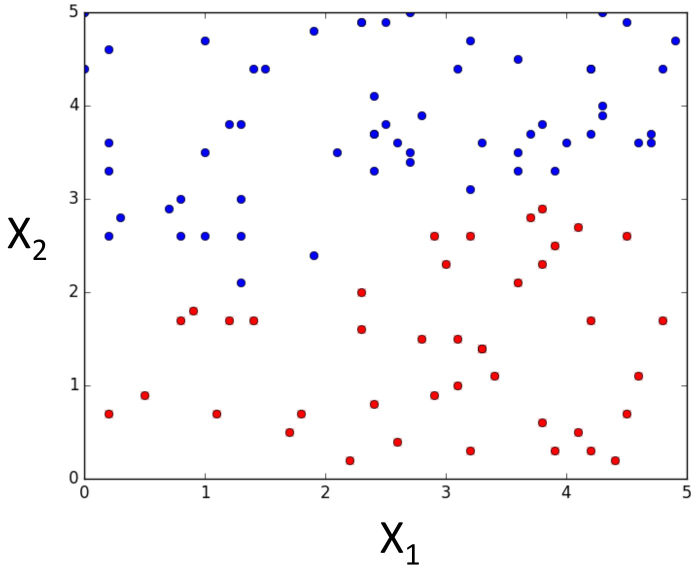
- Define some **objective** (e.g., prediction error)
- Search for best $\theta$ with respect to the objective

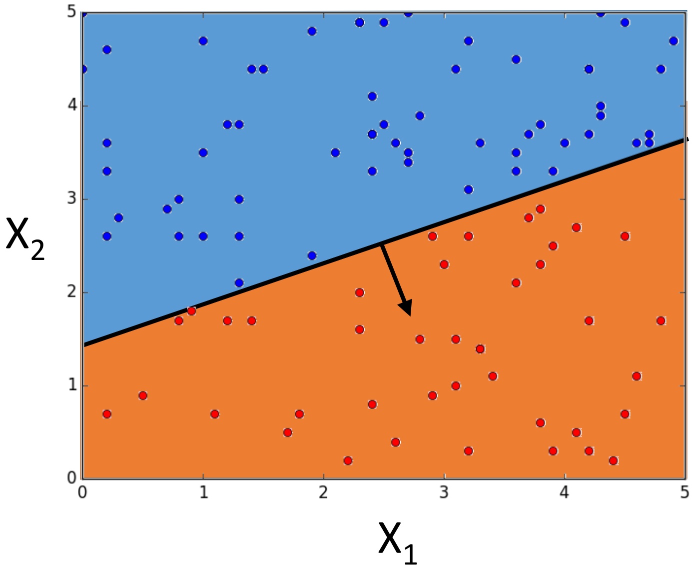


# Generalization …

Sample

Population



# Inference: *Rendering Predictions*

- Evaluating the model on input queries:

$$f_{\hat{\theta}}(X) \rightarrow Y$$

- Online vs Offline:
  - Pre-computed **offline**: *movie rankings*
  - Computed **online** with each query: *speech recognition*
- May want to track confidence in prediction
- May require additional pre and post-processing
  - Feature lookup, content ranking, etc…

# **Feedback**: *Incorporating New Data*

- After rendering a prediction we may get feedback on the results of the prediction:
  - **Explicit**: the *correct value* was "cat"
  - **Implicit**: the predicted animal was *incorrect*
  - Can be **noisy** …

- Watch out for **sample bias:**
  - Model affects the data is uses for training in the future
  - **Example**: only play top40 songs ...

# Taxonomy
## of Machine Learning

Labeled Data

Indirect (reward)

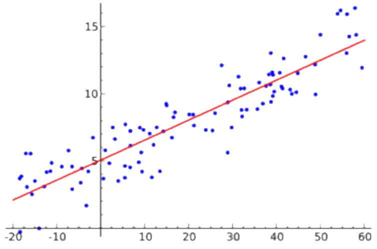
Unlabeled Data

**Supervised Learning**

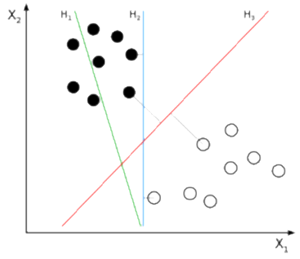
**Reinforcement & Bandit Learning**

**Unsupervised Learning**

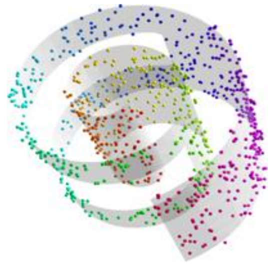
Regression



Classification



Dimensionality Reduction



Clustering