SYSTEMS AND METHODS FOR BIG AND UNSTRUCTURED DATA

DWH and Snowflake

Marco Brambilla

marco.brambilla@polimi.it

@marcobrambi

Agenda

OLTP, OLAP

Datawarehouse Architectures

Datawarehouse Models

Datawarehouse Operations

Snowflake

OLAP vs. OLTP

OLTP vs. OLAP

OLTP: On Line Transaction Processing

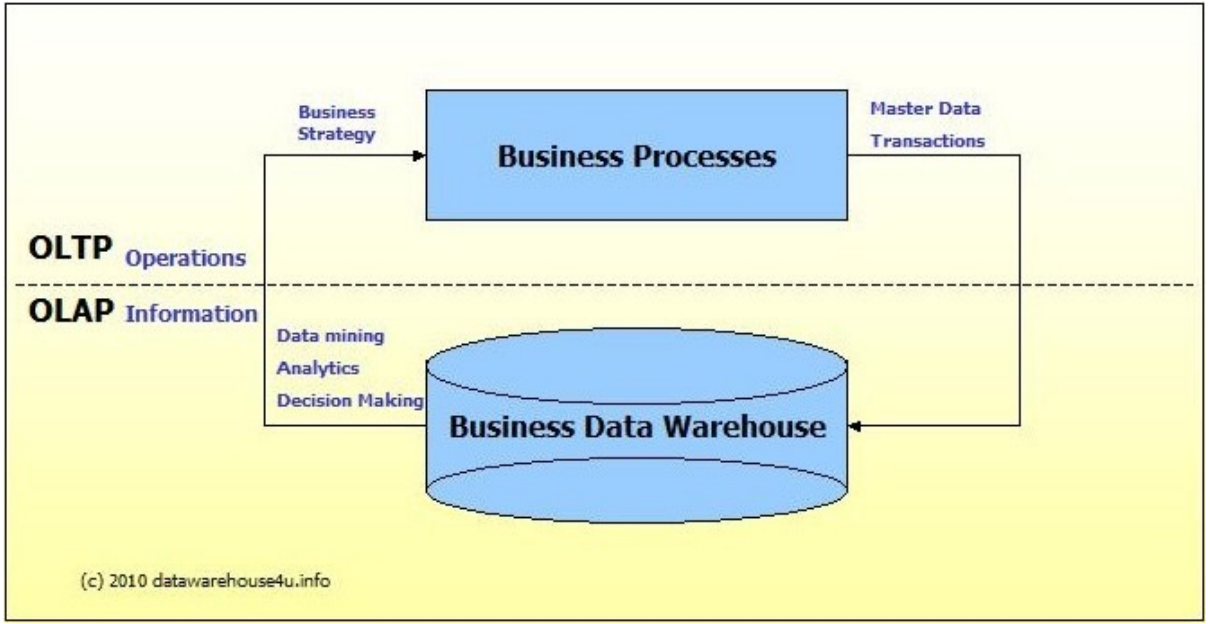
Describes processing at operational sites

OLAP: On Line Analytical Processing

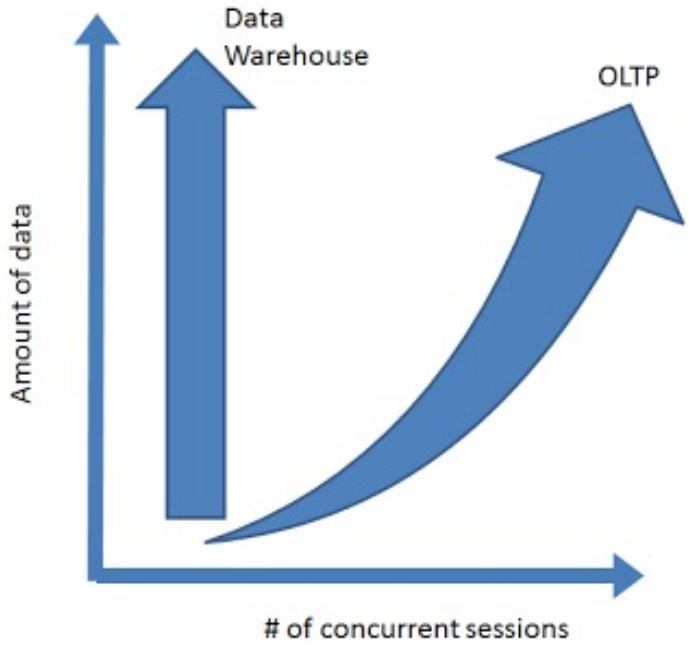
Describes processing at large, integrated data warehouses

OLTP vs. OLAP

We can divide IT systems into transactional (OLTP) and analytical (OLAP). We can assume that OLTP systems provide source data to data warehouses, whereas OLAP systems help to analyze it



Challenges of Scale Differ



Datawarehouse Architectures

Marco Brambilla.

Problem

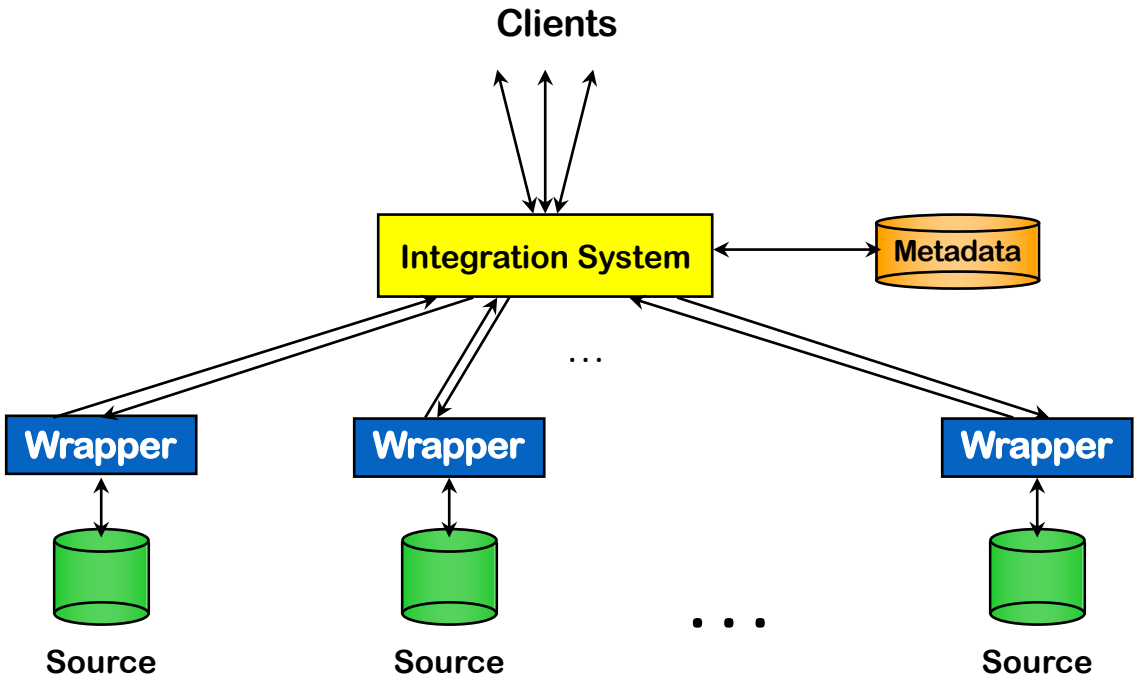
Integrated Data Analytics
Coherent Insights Extraction
Heterogeneous and distributed data sources

The Traditional Analytics Approach

Query driven

Lazy

On-demand



DataWarehouse: a Specialized DB

Standard DB (OLTP)

- Mostly updates
- Many small transactions
- Mb - Gb of data
- Current snapshot
- Index/hash on p.k.
- Raw data
- Thousands of users (e.g., clerical users)

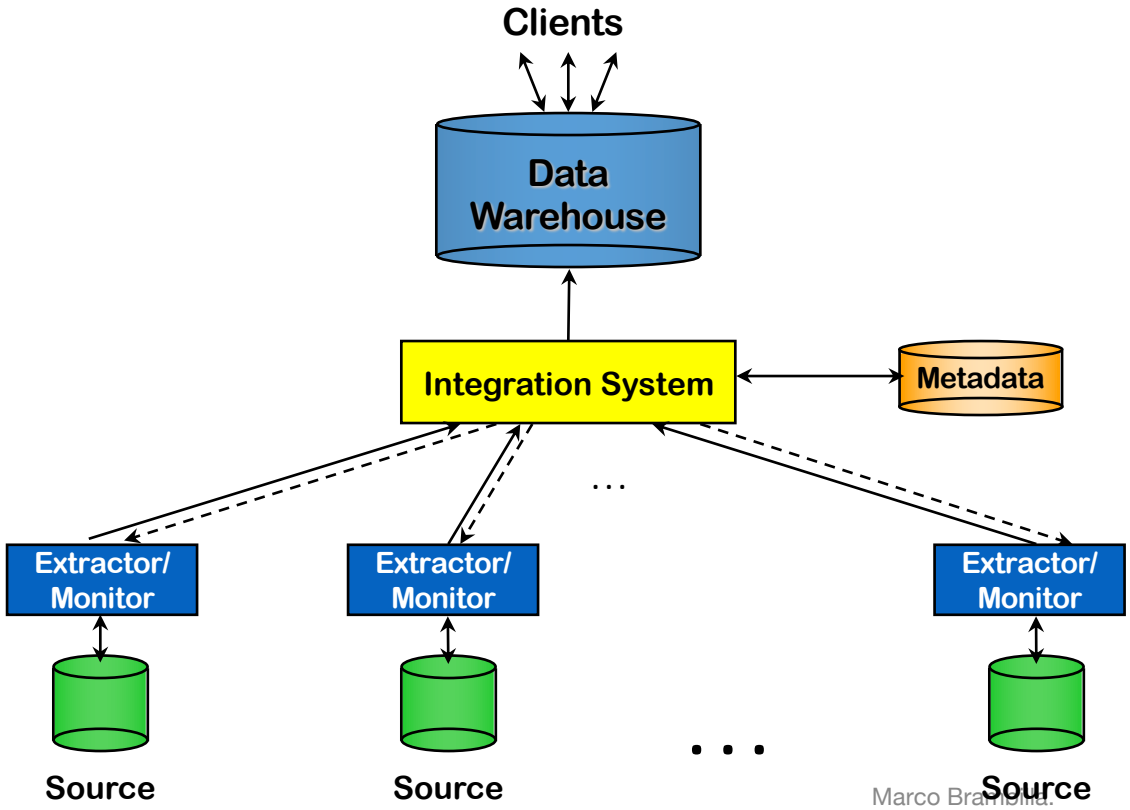
Warehouse (OLAP)

- Mostly reads
- Queries long and complex
- Gb - Tb of data
- History
- Lots of scans
- Summarized, reconciled
- Hundreds of users (e.g., decision-makers, analysts)

The Warehousing Approach

Information integrated in advance

Stored in dwh for direct querying and analysis



Advantages of Warehousing Approach

High query performance
- But not necessarily most current information

Doesn't interfere with local processing at sources
- Complex queries at warehouse
- OLTP at information sources

Information copied at warehouse
- Can modify, annotate, summarize, restructure, etc.
- Can store historical information

Widely adopted in industry

Not Either-Or Decision

Query-driven approach still better for

- Rapidly changing information
- Rapidly changing sources
- Truly vast amounts of data from large numbers of sources
- Clients with unpredictable needs

What is a Data Warehouse?

A Practitioners Viewpoint

"A data warehouse is simply
a single, complete, and consistent store of data
obtained from a variety of sources and made **available to end users** in a way they can **understand and use it**
in a business context."

-- Barry Devlin, *IBM Consultant*

What is a Data Warehouse?

An Analytical Viewpoint

"A DW is a
subject-oriented,
integrated,
time-varying,
non-volatile
collection of data that is used primarily in organizational decision making."

-- W.H. Inmon, Building the Data Warehouse, 1992

A Data Warehouse is...

- Stored collection of diverse data
  - A solution to data integration problem
  - Single repository of information
- Subject-oriented
  - Organized by subject, not by application
  - Used for analysis, data mining, etc.
- Optimized differently from transaction-oriented db
- User interface aimed at executive

A Data Warehouse means ...

Large volume of data (Gb, Tb)

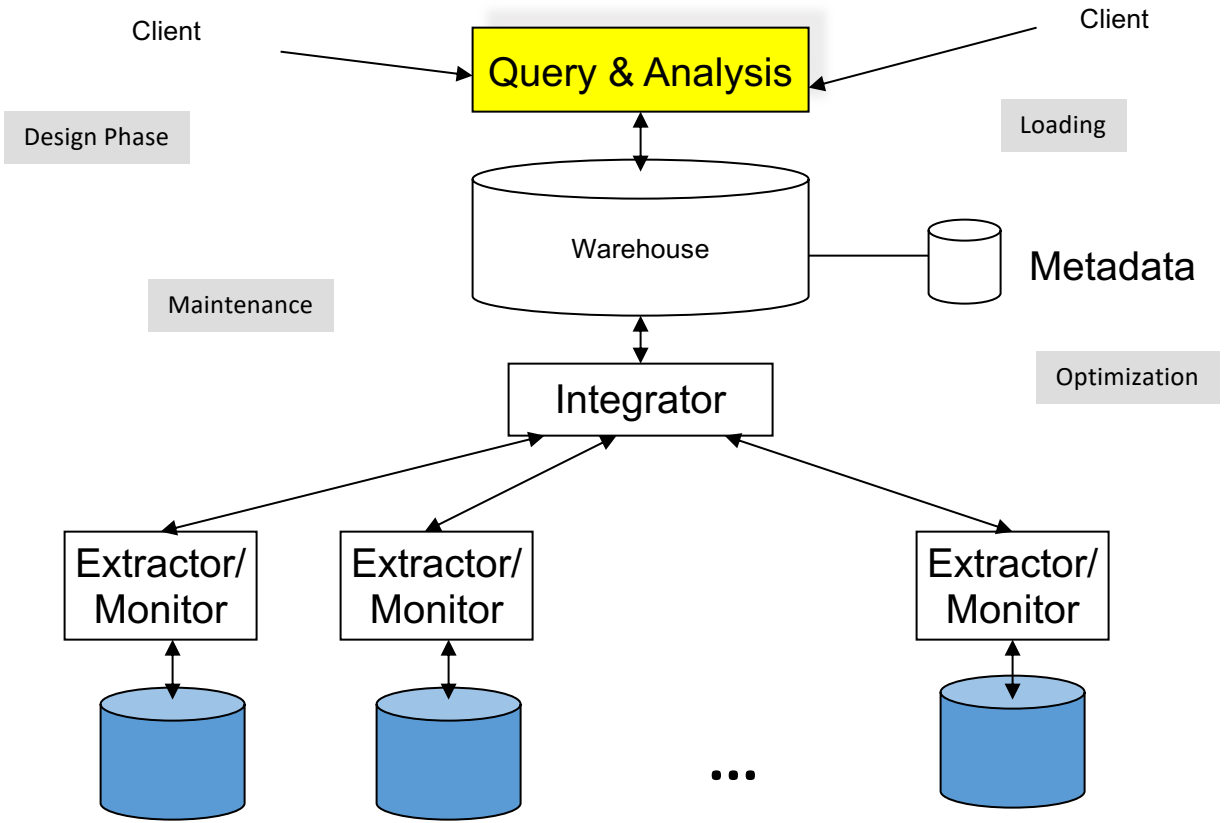
Non-volatile
- Historical
- Time attributes are important

Updates infrequent

Maybe append-only

Examples
- All transactions ever at a bank
- Complete client histories at insurance firm
- Financial information and portfolios of customers

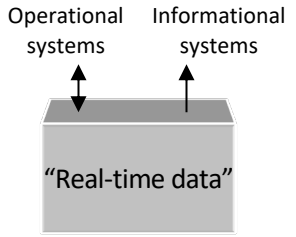
Generic Warehouse Architecture



Data Warehouse Architectures: Conceptual View

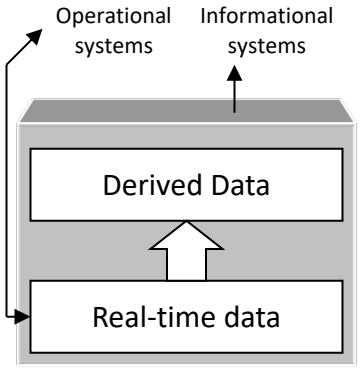
Single-layer

- Every data element is stored once only
- Virtual warehouse



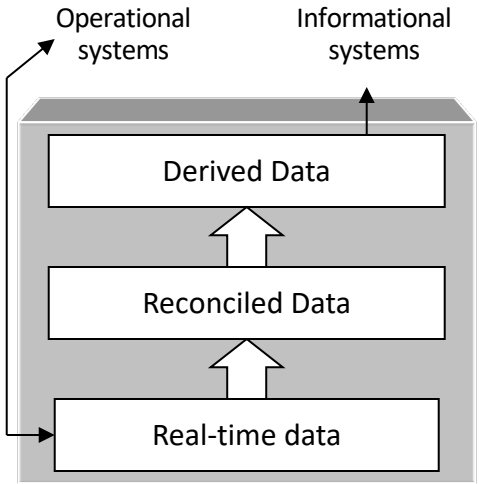
Two-layer

- Real-time + derived data
- Most used approach in industry



Three-layer Architecture: Conceptual View

Transformation of real-time data to derived data really requires two steps



View level
"Particular informational needs"

Physical Implementation of the Data Warehouse

Issues in Data Warehousing

- Warehouse Design
- Extraction
  - Wrappers, monitors (change detectors)
- Integration
  - Cleansing & merging
- Optimization
- Maintenance

Decision Support Systems (DSS)

Information technology to help the knowledge worker (executive, manager, analyst) make faster & better decisions

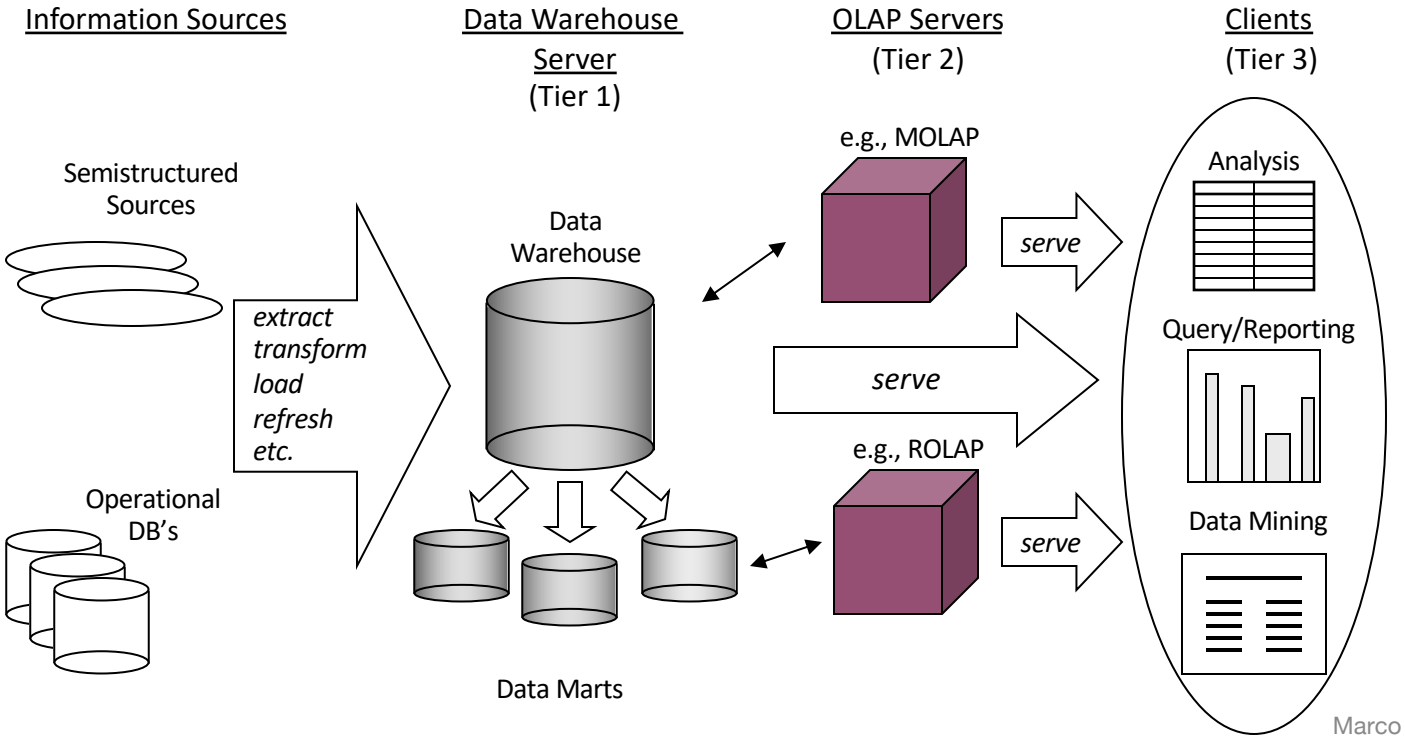
"What were the sales volumes by region and product category for the last year?"

"How did the share price of comp. manufacturers correlate with quarterly profits over the past 10 years?"

"Which orders should we fill to maximize revenues?"

On-line analytical processing (OLAP) is an element of decision support systems (DSS)

Decision Support Systems (DSS) Architecture



Approaches to OLAP Servers

Relational DBMS as Warehouse Servers

(rarely flat files)

Two possibilities for OLAP servers

(1) Relational OLAP (ROLAP)

- Relational specialized DBMS to store and manage warehouse data
- OLAP middleware to support missing pieces

(2) Multidimensional OLAP (MOLAP)

- Array-based storage structures
- Direct access to array data structures
- implements multidimensional data and operations

Data Warehouse vs. Data Marts

Enterprise warehouse: collects all information about subjects that span the entire organization

Data Marts: Departmental subsets that focus on selected subjects

Marketing data mart: customer, product, sales
Faster roll out, but complex integration in the long run

Virtual warehouse: views over operational DBs

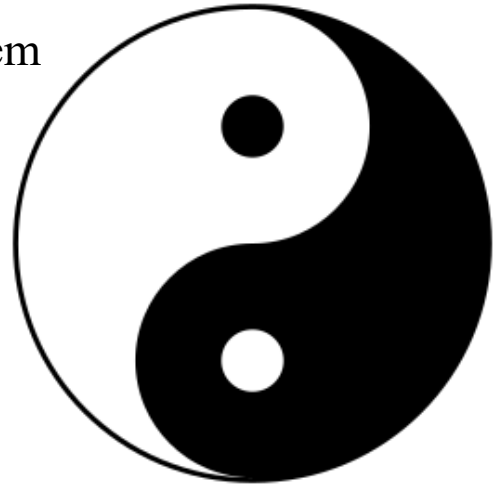
Materialize sel. summary views for efficient query processing
Easy to build but require excess capability on operat. db servers

Comparison Chart of Database Types

Data warehouse (OLAP)	Operational system (OLTP)
Subject oriented	Transaction oriented
Large (hundreds of GB up to several TB)	Small (MB up to several GB)
Historic data	Current data
De-normalized table structure (few tables, many columns per table)	Normalized table structure (many tables, few columns per table)
Batch updates	Continuous updates
Usually very complex queries	Simple to fairly complex queries

Supporting a Complete Solution

OLTP
Operational System
Data Entry



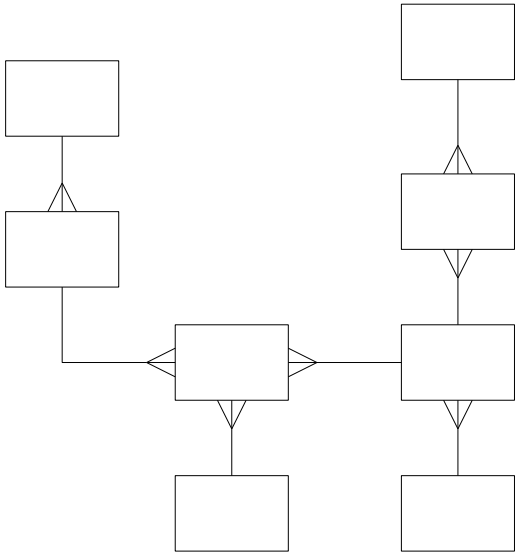
OLAP
Data Warehouse
Data Retrieval

Datawarehouse Models

Marco Brambilla.

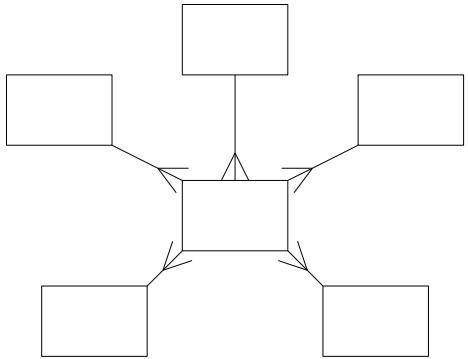
Design Differences

Operational System



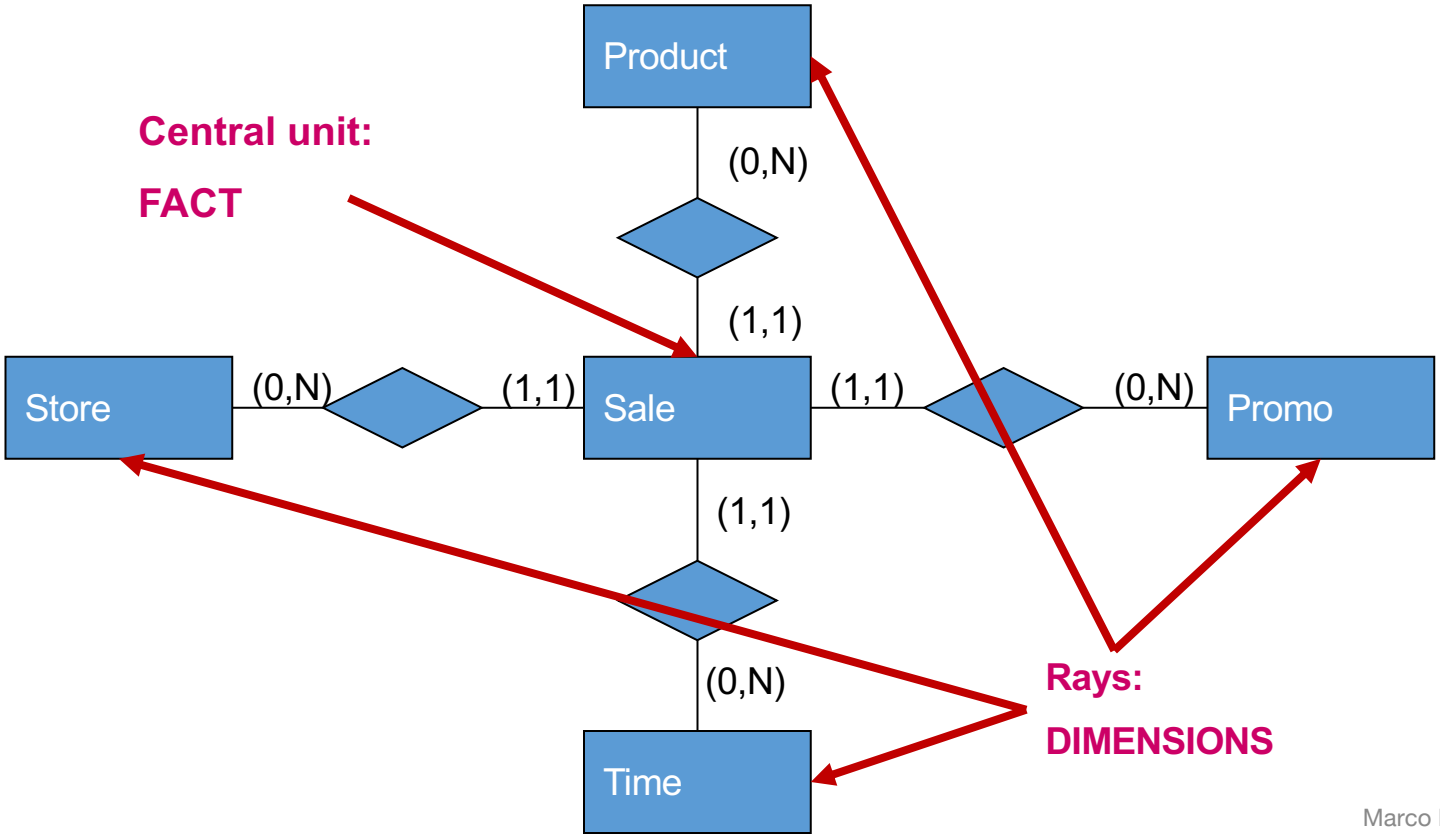
ER Diagram

Data Warehouse



Star Schema

Star schema


Product
Central unit:
FACT
(0,N)
(1,1)
Store
(0,N)
(1,1)
Sale
(1,1)
(0,N)
Promo
(1,1)
(0,N)
Time
Rays:
DIMENSIONS

About the star schema

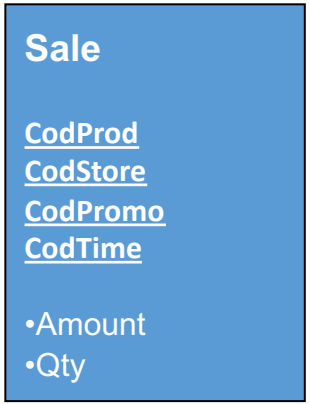
Attributes all over

FACT is an aggregate

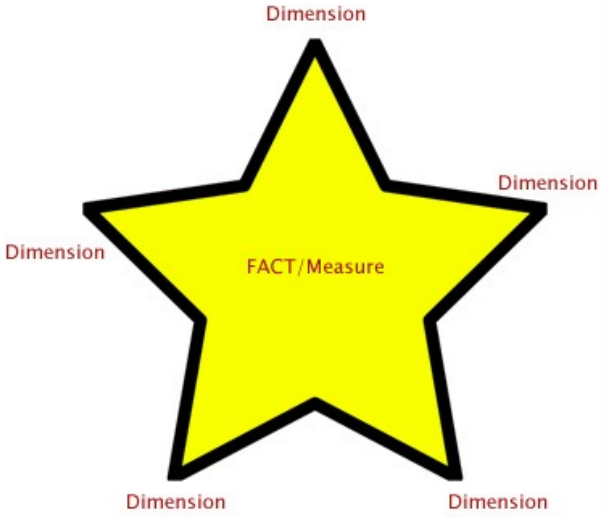
FACT has composed key (from dimensions)

FACT is normalized (BCNF)

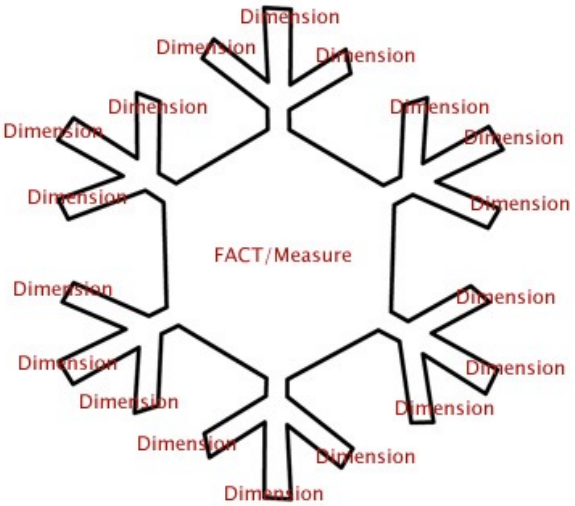
DIMENSION is NOT normalized



Star vs. Snowflake



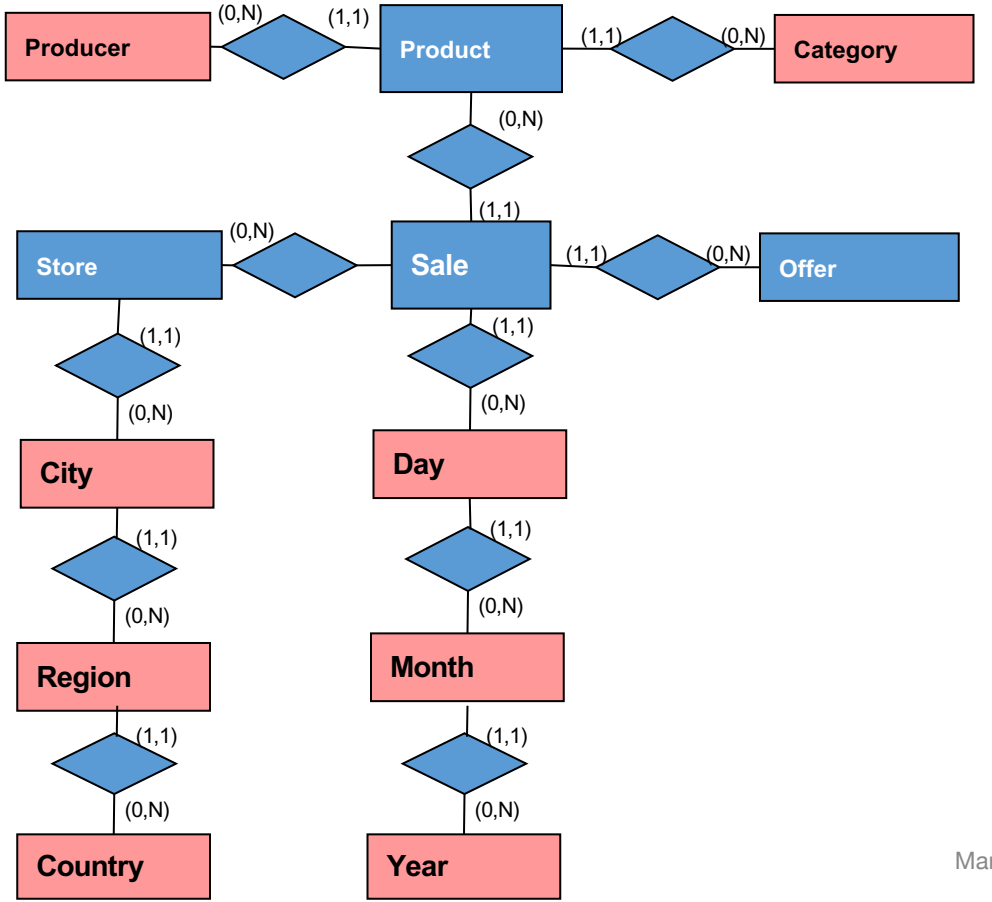
Star Schema



SnowFlake Schema

Snowflake Schema

Hierarchies for not normalized dimensions



Operations

Drill-down

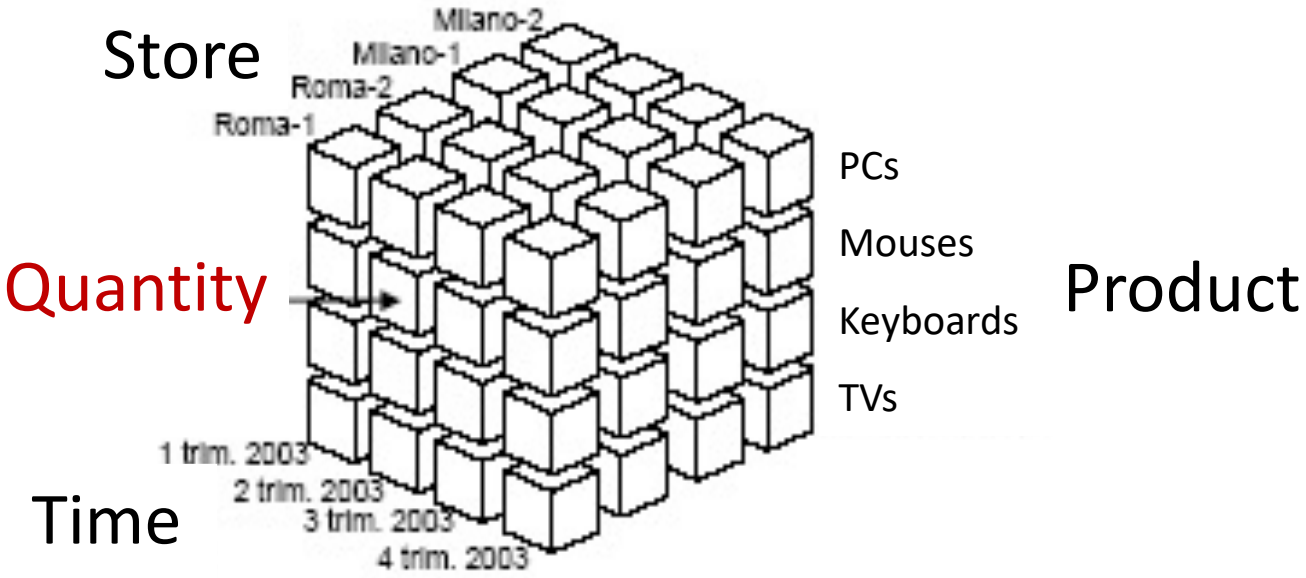
Add one analysis dimension (disaggregation)

Roll-up

Remove one analysis dimension (aggregation)

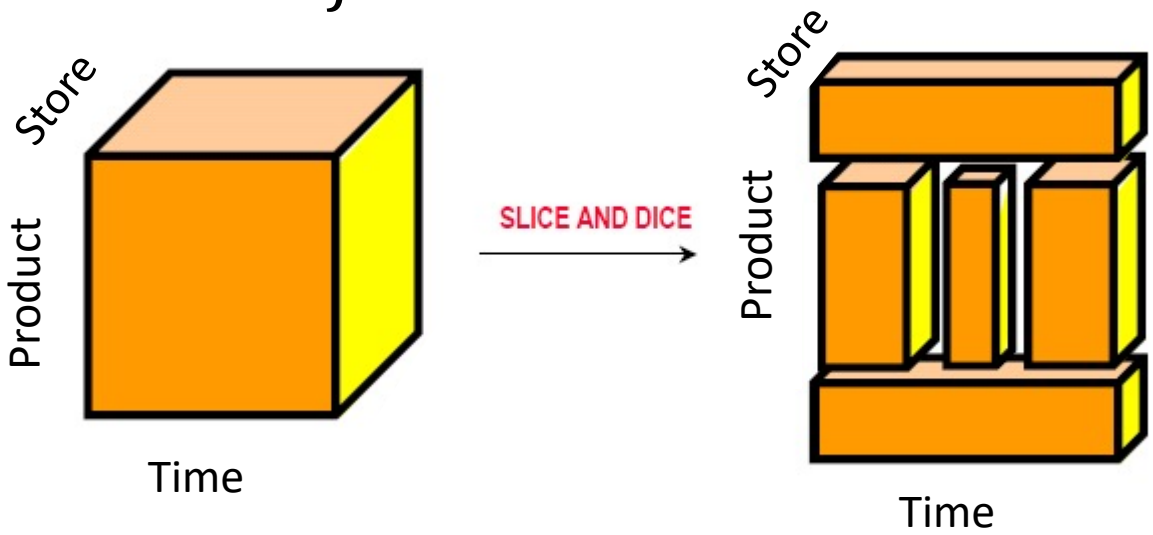
Datacube (slice and dice)

Multidimensional Cube



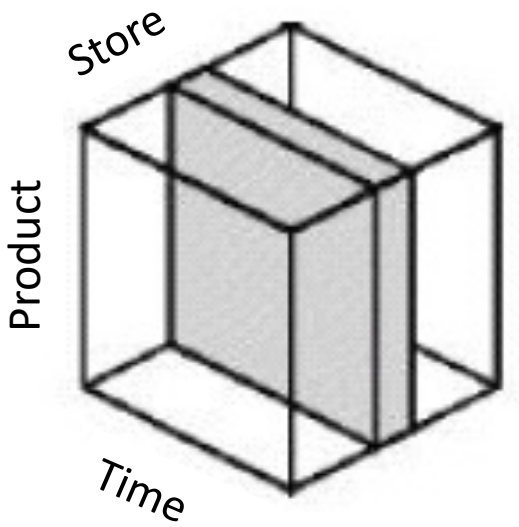
Slice-and-Dice

Select and Project



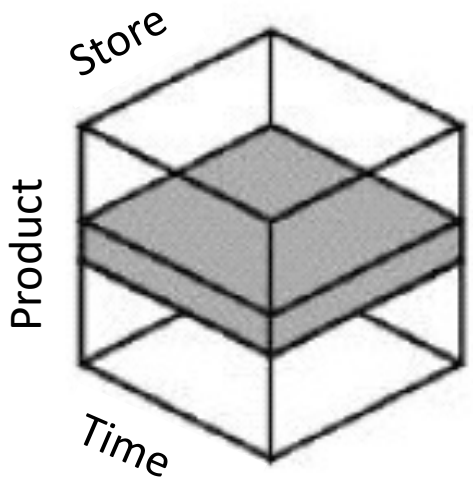
Slice-and-Dice

Sales in an area



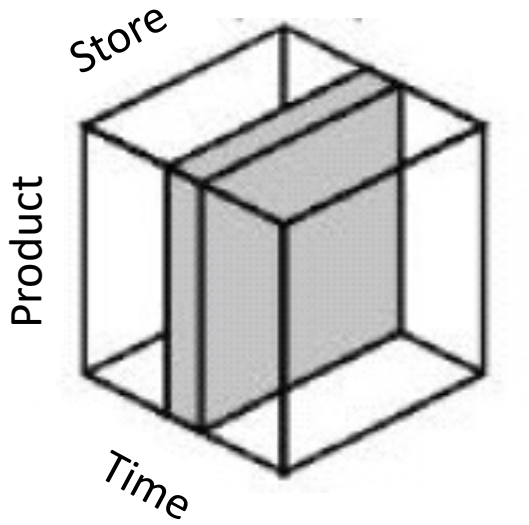
Slice-and-Dice

Sales of a product



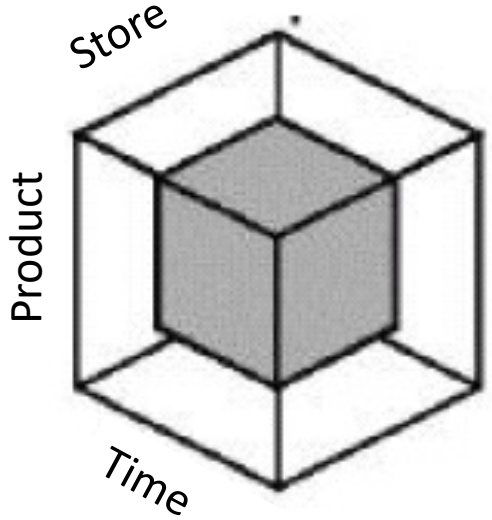
Slice-and-Dice

Sales in a period



Slice-and-Dice

STRATEGIC DECISION SUPPORT:
Category of products in a region and mid-term time span



.. So old school ...

Snowflake Key Features

Enterprise-ready cloud data warehouses that automatically **scales** for balancing performances and costs

Separation between **compute** and **storage**

- Other databases combine the two together, meaning you must size for your largest workload and incur the cost that comes with it

Single place for storing all the data

Why is data warehousing important?

Consistency

Uniform format applied to all collected data

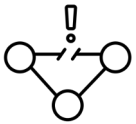
Easier for corporate decision-makers to analyze and share data insights

Security and Data Protection

Multi-Factor Authentication (MFA), federal authentication, Single Sign-on (SSO), traffic encrypted with TLS

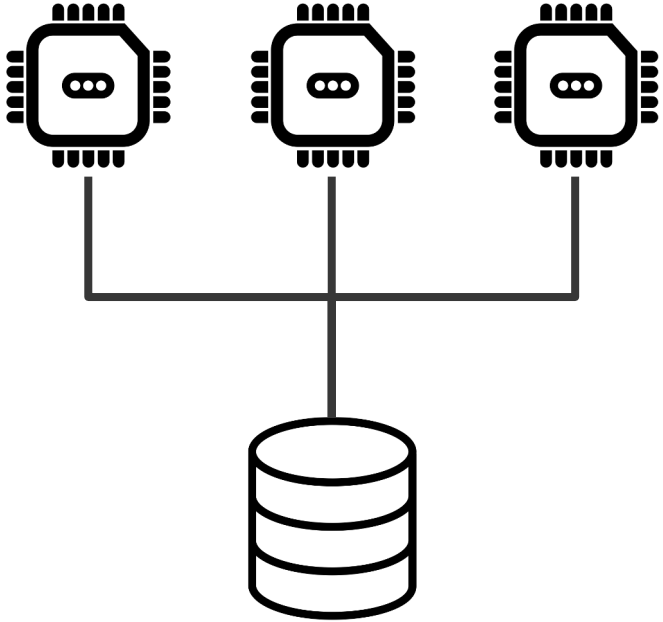
Connectivity

Client connectors and drivers such as Python connector, Spark connector, Node.js driver, .NET driver, etc.

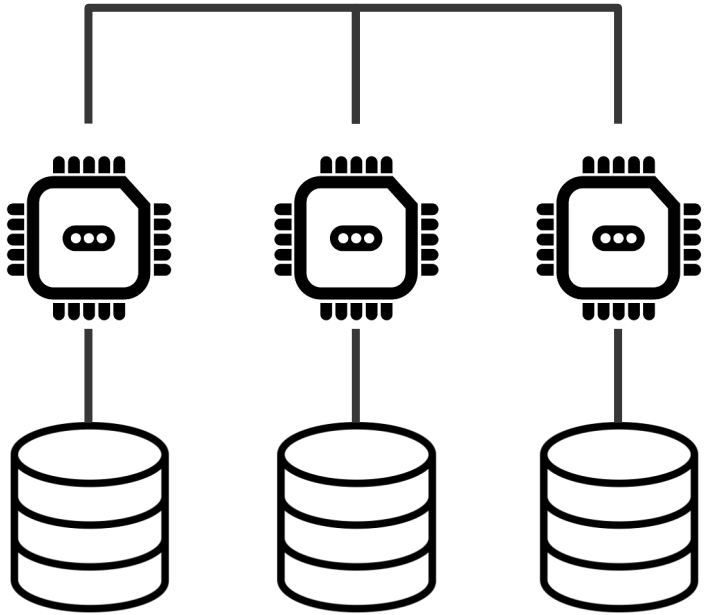


Fault Tolerance

Traditional Database Architectures

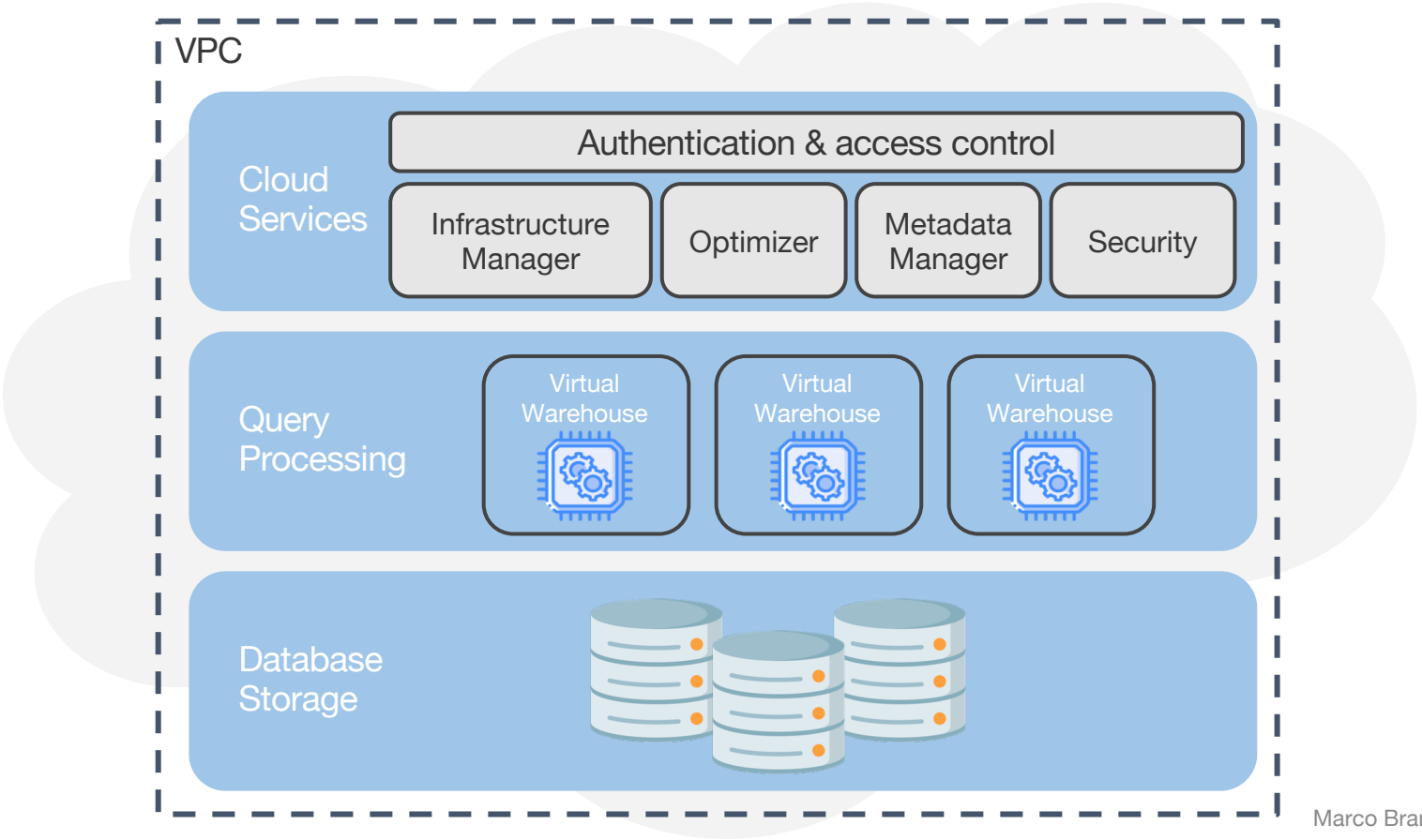


Shared-Disk Architecture



Shared-Nothing Architecture

Snowflake Architecture

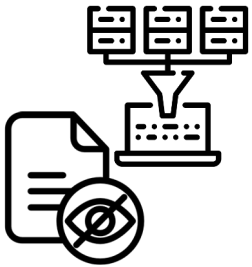


Marco Brambilla.

Database Storage

Data in Snowflake is organized into the internal optimized, compressed, columnar format.
Snowflake manages all aspects of how this data is stored:

- Organization, file size, structure, compression, metadata, statistics



Data visible or accessible by customers only through SQL query executed in Snowflake

Query Processing

Snowflake processes queries using **virtual warehouses**

- MPP compute cluster composed of **multiple compute nodes** allocated by Snowflake from a cloud provider

Virtual warehouses are independent compute cluster and does not share compute resources with other virtual warehouses

- No impact on the performance of other virtual warehouses

Cloud Services

Services that coordinate activities across Snowflake for processing user requests, from login to query dispatch.

Services managed in this layer:

- Authentication
- Infrastructure management
- Metadata management
- Query parsing and optimization
- Access control

Connecting to Snowflake

Web-based user interface

Command line clients

ODBC and JDBC drivers

Native connectors

Third-party connectors (ETL, BI tools)

Ingesting Data

Types of data supported for loading:

- Data with supported character encoding
- Compressed files
- CSV, TSV, etc.
- JSON, Avro, ORC, Parquet, and XML format
- Amazon S3, Google Cloud Storage, or Microsoft Azure

Ingesting Data

Bulk Load

- Snowflake provides the COPY command for batch loading
  - COPY command uses Virtual Warehouse computing resources
  - Managed manually
- Allows basic transformations such as reordering and excluding columns, data typing, truncating strings

Continuous Load

- Snowpipe used for loading streaming data
- Scaling up / down is done automatically
- Doesn’t use the virtual warehouse computing resources

Query data without loading data

External Tables

- It is possible to use external tables to access data from Snowflake
- Useful when there is a lot of data externally but we want to query only a small subset of data
- Performances and costs can be optimized by creating materialized views

Loading bulk data from cloud & local storage

1. **Prepare your files**
   Pre-processing phase

1. **Stage the data**
   Make snowflake aware of the data

1. **Execute COPY command**
   Copy Data into the table

1. **Managing Regular Loads**
   Scheduling

Stage the Data

Staging area is an intermediate, transient place used to process data for extracting, transforming and loading processes

Classic method: stage data in a **S3 bucket** or an **Azure blob store**

- Long term cheap storage for raw data

Data can also be staged on local file system

```
CREATE STAGE <stage_name> url='cloud_storage_url'
credentials='login_password'
```

Execute COPY Command

COPY INTO command is the common mechanism for loading data into Snowflake with batch mode

```
COPY INTO <table_name>
        from @<stage_name>
        pattern = '.*.csv'
        file_format = (type = csv field_delimiter = '|' skip_header =
1);
```

JSON file:

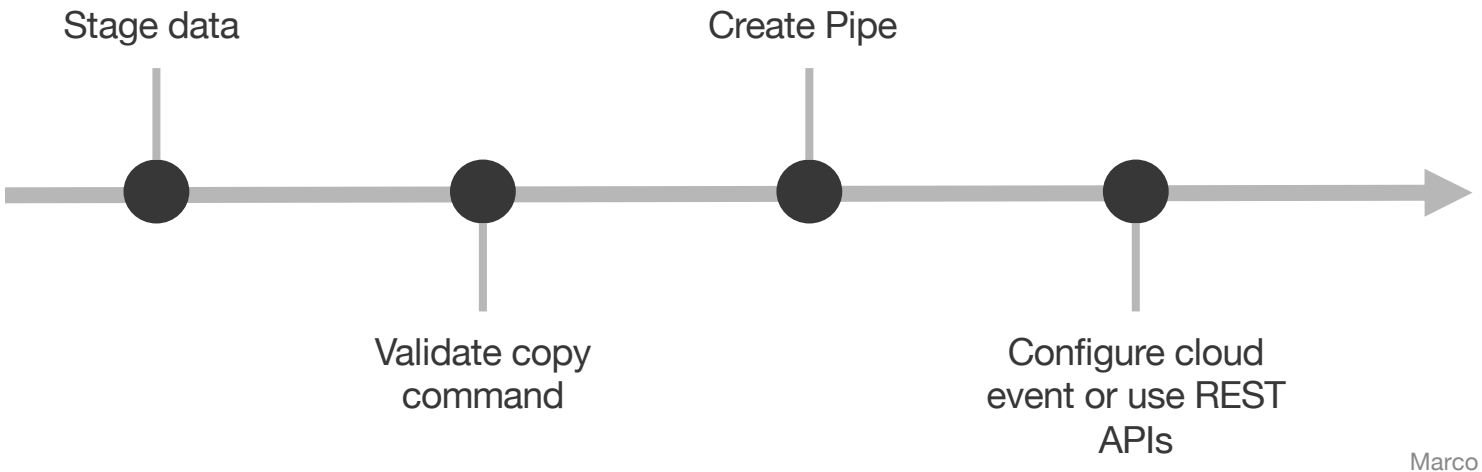
```
COPY INTO <table_name>
        from @<stage_name>
        file_format = (type = json);
```

Snowpipe

Mechanism to enable loading of streaming data

Serverless architecture:

- It has its own processing
- No virtual warehouse instances involved



Hands On with Snowpipe

```
-- create a database if it doesn't already exist
CREATE DATABASE ingest_data;

USE DATABASE ingest_data;


-- create an external stage using an S3 bucket
CREATE OR REPLACE STAGE snowpipe_copy_example_stage
url='s3://snowpipe-streaming/example_table';

-- list the files in the bucket
LIST @snowpipe_copy_example_stage;
```

Hands On with Snowpipe

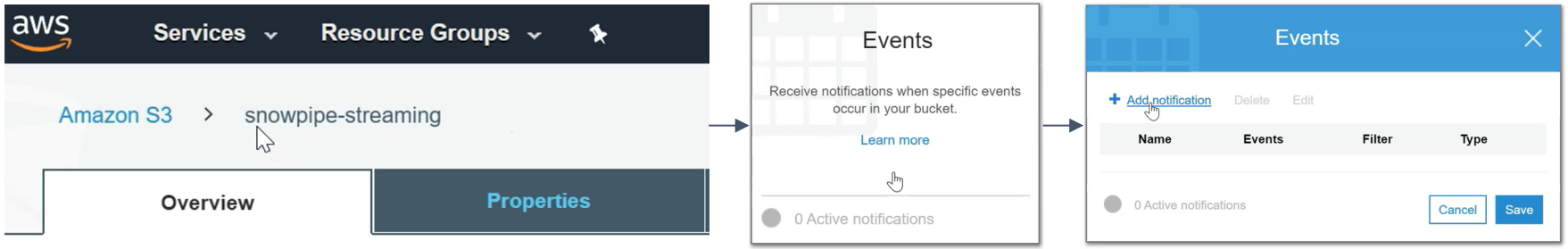
```
CREATE TABLE transactions
(
      Transaction_Date DATE,
      Customer_ID NUMBER,
      Transaction_ID NUMBER,
      Amount NUMBER
);

CREATE OR REPLACE PIPE transaction_pipe
auto_ingest = true
AS COPY INTO transactions FROM
@snowpipe_copy_example_stage
file_format = (type = csv field_delimiter = '|' skip_header = 1);
```

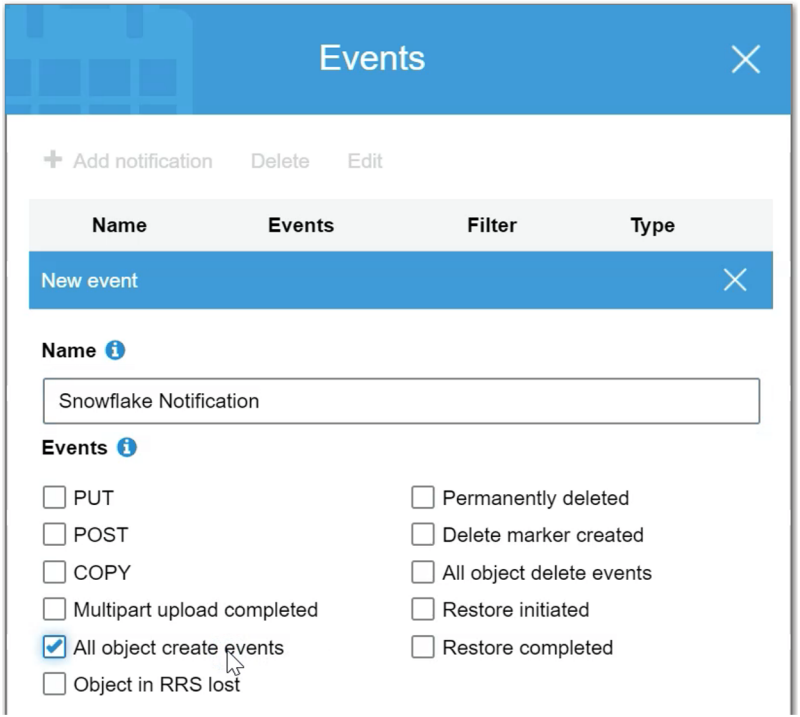
Hands On with Snowpipe

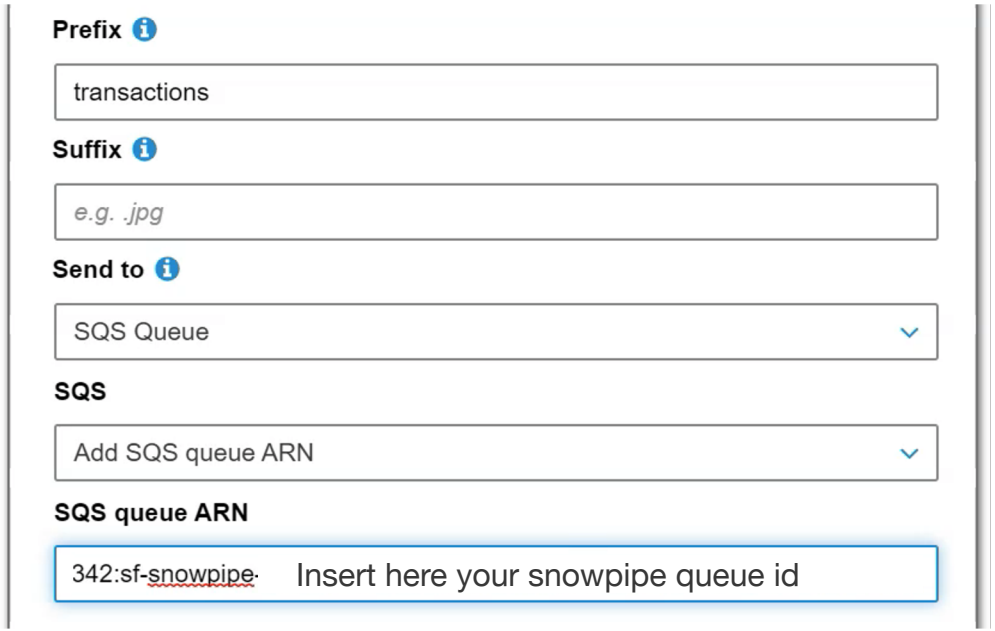
To setup notifications on S3 bucket:

Amazon S3 > bucket_name > properties > events



Hands On with Snowpipe





Hands On with Snowpipe

Now if a new file is uploaded on S3, the data arrives automatically on Snowflake

```
SELECT * FROM transactions;

SHOW PIPES;

-- setup S3 event notification here

SELECT COUNT(*) FROM transactions;
```

Performance Optimization

1. Use **dedicated virtual warehouses** for different workloads

1. Scale up for **know large workloads**

1. Scale up with virtual warehouses for **unknown workloads**

1. Design the production workflow to **maximize cache usage**

1. Use **cluster keys** for partitioning large tables

Dedicated virtual warehouses

```
-- grant usage to a database to PUBLIC role
GRANT USAGE ON DATABASE INGEST_DATA TO ROLE PUBLIC;
GRANT USAGE ON SCHEMA INGEST_DATA.PUBLIC TO ROLE PUBLIC;
GRANT SELECT ON TABLE INGEST_DATA.PUBLIC.TRANSACTIONS TO ROLE
PUBLIC;

-- virtual warehouse for data scientists
CREATE WAREHOUSE DATASCIENCE_WH WITH WAREHOUSE_SIZE = 'SMALL'
WAREHOUSE_TYPE = 'STANDARD' AUTO_SUSPEND = 300 AUTO_RESUME =
TRUE;

-- virtual warehouse for DBAs. xsmall because of queries size
CREATE WAREHOUSE DBA_WH WITH WAREHOUSE_SIZE = 'XSMALL'
WAREHOUSE_TYPE = 'STANDARD' AUTO_SUSPEND = 300 AUTO_RESUME =
TRUE;
```

Dedicated virtual warehouses

```
CREATE ROLE DATA_SCIENTIST;
GRANT USAGE ON WAREHOUSE DATASCIENCE_WH TO ROLE
DATA_SCIENTIST;

CREATE ROLE DBA;
GRANT USAGE ON WAREHOUSE DBA _WH TO ROLE DBA;

-- create login credentials for data scientist DS_1
CREATE USER DS_1 PASSWORD  = 'DS_1'
LOGIN_NAME = 'DS_1' DEFAULT_ROLE = 'DATA_SCIENTIST'
DEFAULT_WAREHOUSE = 'DATASCIENCE_WH'
MUST_CHANGE_PASSWORD = TRUE

GRANT ROLE DATA_SCIENTIST TO USER DS_1;
```

Scale Up - Multi Cluster Warehouses

Virtual warehouses size can be **scaled up** in response to a **workload change**

Possible scenarios:

- Ad-hoc event e.g. urgent business requirement that implies several complex queries
- Recurrent / scheduled pattern

This is achieved by automatically create a **replica of virtual warehouse: Multi Cluster Warehouses**

Requirements:

- Set a minimum and maximum number of virtual warehouses
- Enterprise Edition of Snowflake

Scale Up - Multi Cluster Warehouses

```
CREATE WAREHOUSE AS_WH
WITH WAREHOUSE_SIZE = 'XSMALL'
MIN_CLUSTER_COUNT = 1
MAX_CLUSTER_COUNT = 3
SCALING_POLICY = 'STANDARD'
AUTO_SUSPEND = 300
AUTO_RESUME = TRUE;
```

Maximizing Cache Usage

In Snowflake **caching is automatic**

- Results are cached for 24 hours
- Snowflake knows when data has changed and it re-executes the query if necessary

To Maximize cache usage:

- Ensure **similar queries** go to the **same virtual warehouse**

Clustering Keys

For very large tables, automatic partitioning scheme **may not be optimal**

It is possible to create **custom cluster keys** to partition table according to the needs

- Normally used for columns involved frequently in WHERE, JOIN or ORDER BY statements

Only very large tables (multi terabyte size) will benefit from clustering

Clustering Keys

```
CREATE TABLE transactions_clustered_date
(
    Transaction_date DATE
    Transaction_id Integer
    Customer_id String
    Amount integer
)     CLUSTER BY (Transaction_date);
```

POLITECNICO
MILANO 1863

THANKS!
QUESTIONS?

Marco Brambilla @marcobrambi marco.brambilla@polimi.it
http://datascience.deib.polimi.it http://home.deib.polimi.it/marcobrambi