CS 4440 A

Emerging Database Technologies

Lecture 15

03/31/25

Announcements

Guest lecture this Wednesday (Apr 2)

- Speaker bios in lecture announcement
- Attendance required

Exam 2

- Released: Wednesday Apr 2 @ 5PM
- Due: Friday Apr 4 @ 11:59PM

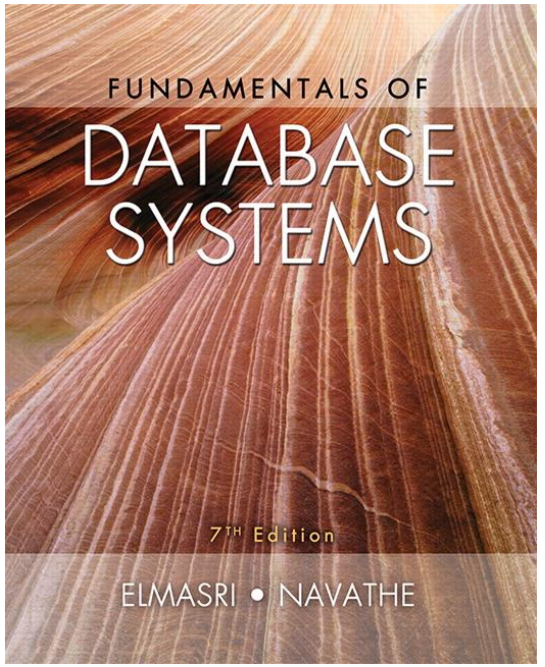
Agenda

1. OLTP vs OLAP
2. Data Cubes
3. HTAP

Reading Materials

Fundamental of Database Systems (7th Edition)

- Chapter 29 - Overview of Data Warehousing and OLAP



1. OLTP vs OLAP

So far we've been dealing with OLTP

OLTP: OnLine Transactional Processing

- Often used to store and manage relevant data to the day-to-day operations of a system or company
  - e.g., ATM transactions, online hotel bookings
- INSERT, UPDATE, DELETE commands
- Handles real-time transactions (response times often in milliseconds)
- ACID properties are often important

This is where relational databases shine!

Ok, so what's OLAP?

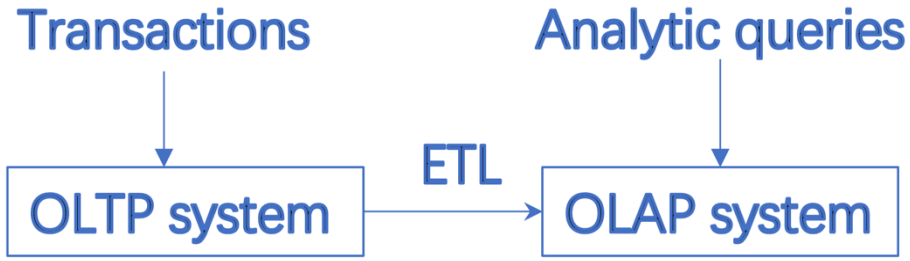
OLAP: OnLine Analytical Processing

- Also known as decision support or business intelligence (BI), but now BI has grown to include more (e.g., AI)
- A specialization of relational databases that prioritizes the reading and summarizing large volumes (TB, PB) of relational data to understand high-level trends and patterns
  - E.g., the total sales figures of each type of Honda car over time for each county
- "Read-only" queries

Contrast this to OLTP

- "Read-write" queries
- Usually touch a small amount of data
  - e.g., append a new car sale into the sales table

How is OLAP Performed?

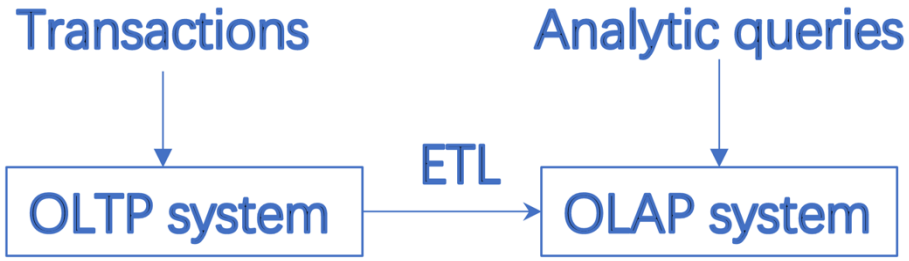


Usually, OLAP is performed on a separate data warehouse away from the critical path of OLTP transactions (a live/transactional database).

This data warehouse is periodically updated with data from various sources (e.g., once an hour or once a day)

- This is through a process of ETL (Extract, Transform, Load)
  - Extract useful business that needs to be summarized, transform it (e.g., canonicalize values, clean it up), load it in the data warehouse
- By doing it periodically, this data warehouse can become stale

How is OLAP Performed?



Usually, OLAP is performed on a separate data warehouse away from the critical path of OLTP transactions (a live/transactional database).

Why?

- Because OLAP queries end up reading most of the data, and will prevent OLTP queries from taking precedence
  - it is more important to ensure that sales are not prevented than to make sure that a report for a manager is generated promptly
  - the latter will anyway take a long time, so might as well have them wait a bit longer
- It's OK if the warehouse data is a bit stale.

OLAP in Data Warehouses

Here are some data warehouses that you might have heard of

Data Warehouse vs Data Lake

Data warehouse:

- structured data (schema-on-write)
- expensive for large data volumes
- managers and business analysts

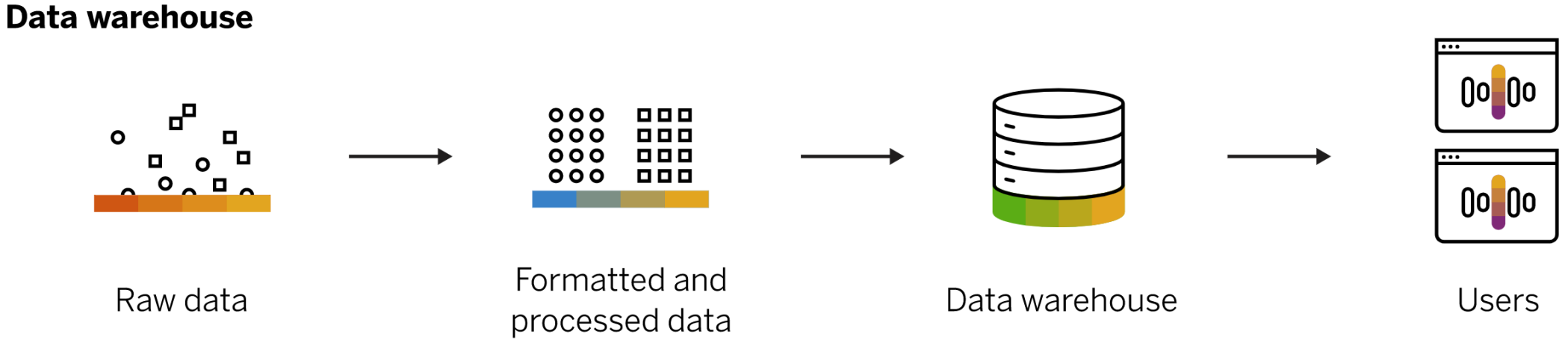


Image source: https://www.sap.com/insights/what-is-a-data-lake.html

Data Lake vs Data Warehouse

Data lake:

raw data, can be unstructured (schema-on-read)

low-cost storage, but no transactions, data quality checks

data scientists and engineers

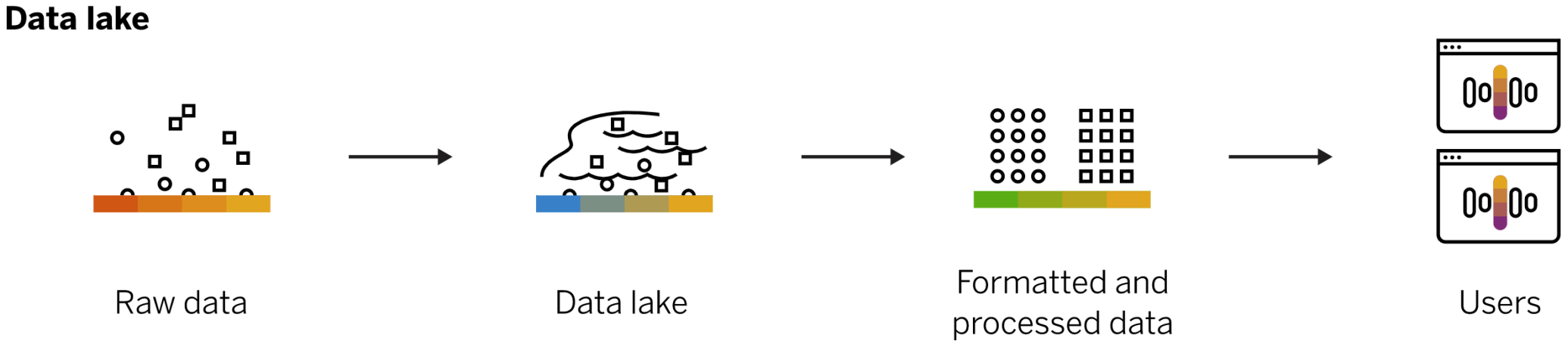


Image source: https://www.sap.com/insights/what-is-a-data-lake.html

We will focus on the following two aspects of OLAP systems

#1 Data Model

- Relational vs multi-dimensional schema

#2 Storage Format

- Row vs column store

#1 Data Model: Multi-dimensional Model

The multi-dimensional data model includes two types of tables:

- **Fact table**
  - Contain the actual metrics or measurements of business processes. Store quantitative data (numbers that can be aggregated).
    - E.g., Sales amount, Quantity sold
  - Typically has many rows but fewer columns
  - Usually contains foreign keys that link to dimension tables
- **Dimension table**
  - Contain descriptive attributes that provide context to facts
  - Store qualitative data
    - E.g., Product dimension (name, category, brand), Time dimension (date, month, year)

Multi-dimensional Schemas

Star schema:

- Consists of a fact table with a single table for each dimension.

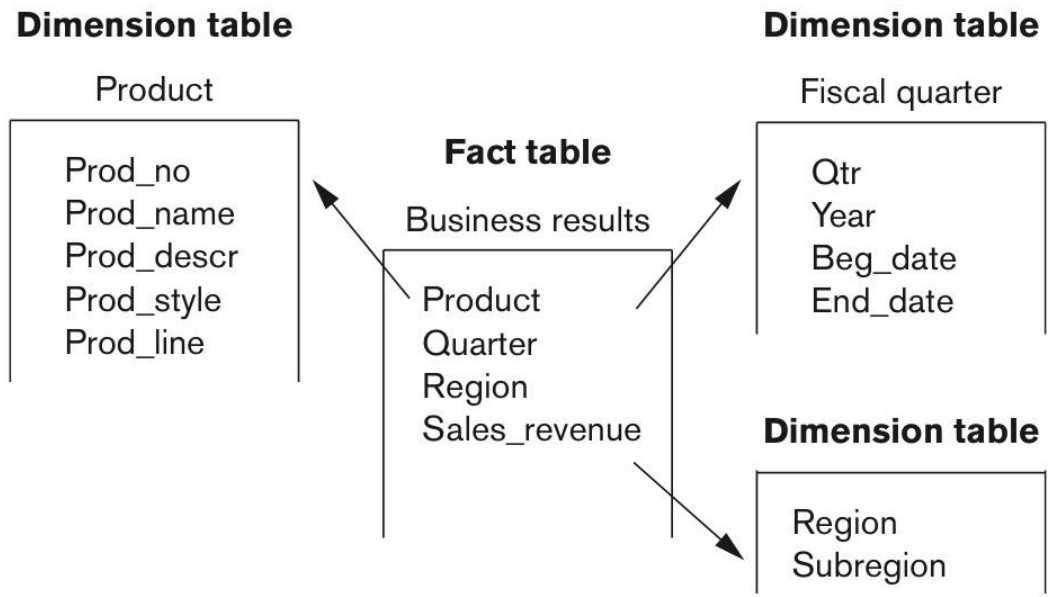


Figure 29.7 A star schema with fact and dimensional tables.

Multi-dimensional Schemas

Snowflake Schema:

- It is a variation of star schema, in which the dimensional tables from a star schema are normalized to eliminate redundancy

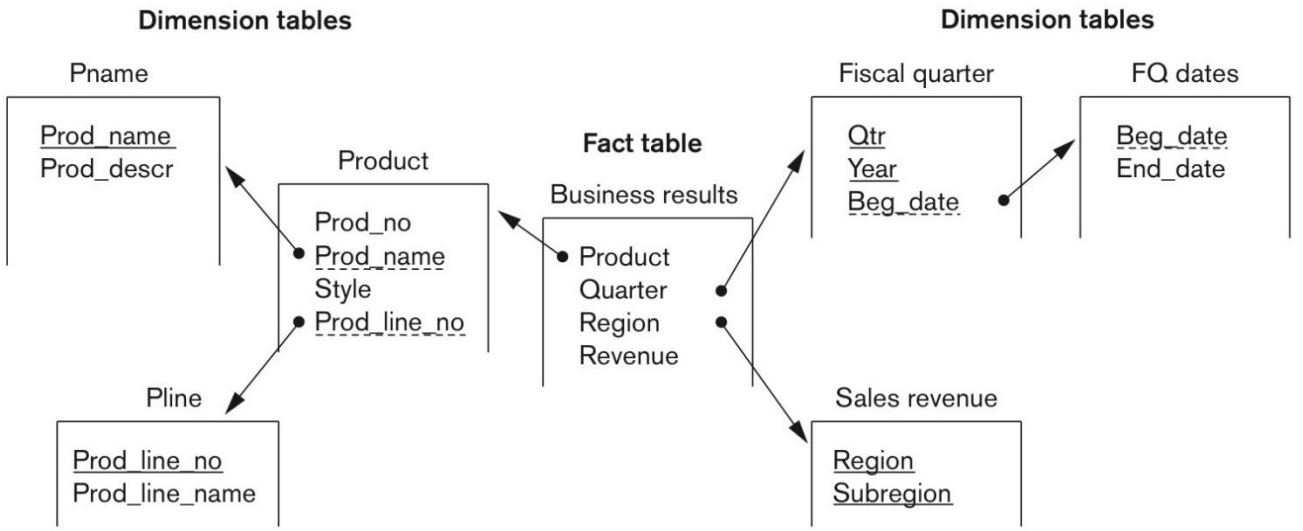


Figure 29.8 A snowflake schema.

Comparison with the Relational Model

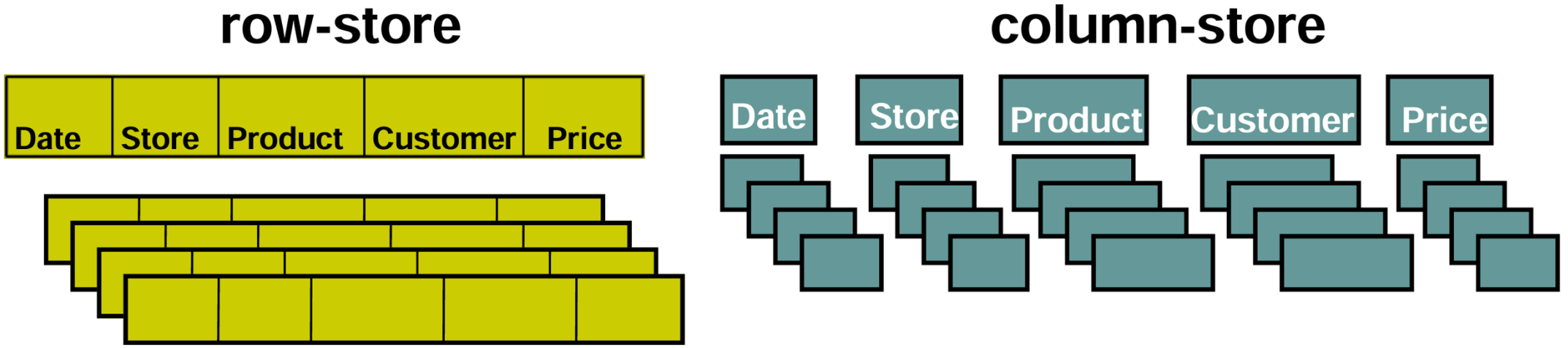
The relational model (used by OLTP) keeps tables normalized

- Minimal updates required for data inserts and deletes => help improve transaction performance

In the multi-dimensional model, the fact table is often in 3NF, but the dimensional tables are denormalized.

- This means columns in a table contains data which is repeated throughout the table => help improve read performance.
- Data is stored in fewer tables, which removes the overhead of having to perform complex joins => helps improve read performance.

#2 Storage Format: Row vs Column Store



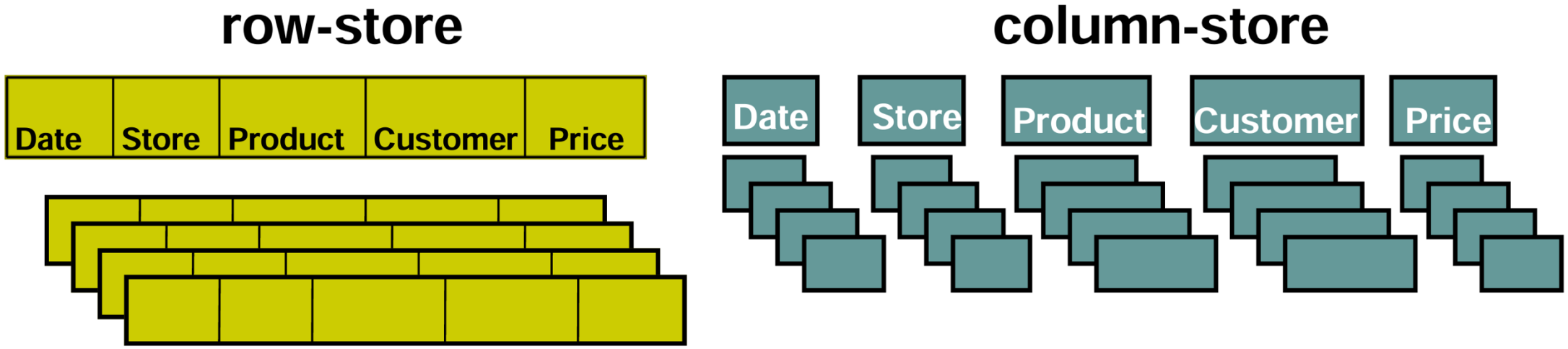
- Store all attributes of a tuple together
- Storage like “row-major order” in a matrix

- Store all rows for an attribute together
- Storage like “column-major order” in a matrix

Q: Which format do you think is better suited for OLTP vs OLAP?

Acknowledgement: Slides adapted from VLDB 2009 Tutorial on Column Stores

#2 Storage Format: Row vs Column Store



+ easy to add/modify a record

- need to read unnecessary data

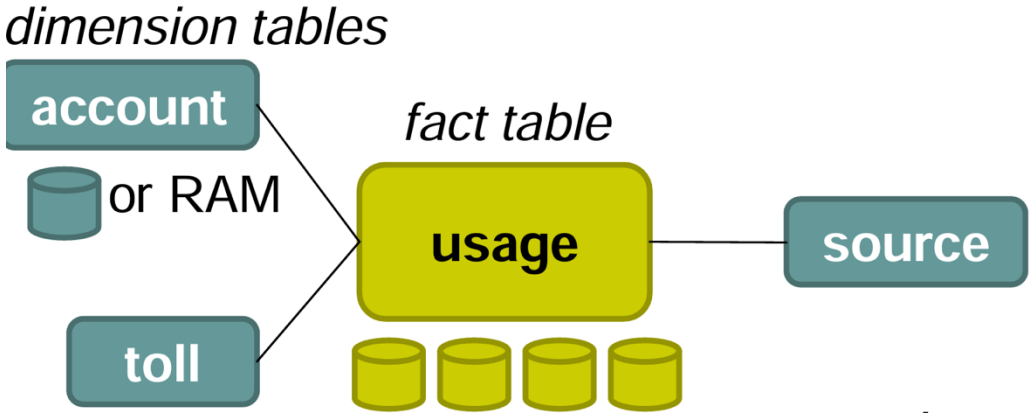
+ only need to read in relevant data

- tuple write require multiple accesses

Acknowledgement: Slides adapted from VLDB 2009 Tutorial on Column Stores

Telco Data Warehousing example

"One Size Fits All? - Part 2: Benchmarking Results"
Stonebraker et al. CIDR 2007



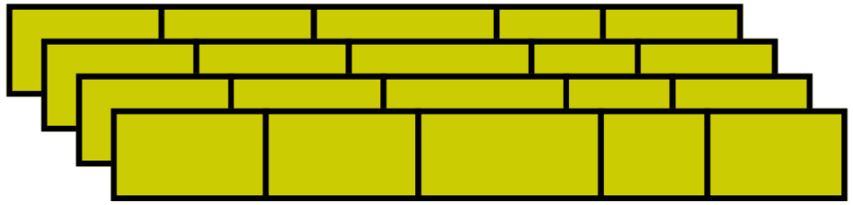

```
QUERY 2
SELECT account.account_number,
sum (usage.toll_airtime),
sum (usage.toll_price)
FROM usage, toll, source, account
WHERE usage.toll_id = toll.toll_id
AND usage.source_id = source.source_id
AND usage.account_id = account.account_id
AND toll.type_ind in ('AE'. 'AA')
AND usage.toll_price > 0
AND source.type != 'CIBER'
AND toll.rating_method = 'IS'
AND usage.invoice_date = 20051013
GROUP BY account.account_number
```

	Column-store	*Row-store*
Query 1	*2.06*	*300*
Query 2	*2.20*	*300*
Query 3	*0.09*	*300*
Query 4	*5.24*	*300*
Query 5	*2.88*	*300*

Acknowledgement: Slides adapted from VLDB 2009 Tutorial on Column Stores

Telco example explained: read efficiency

row store

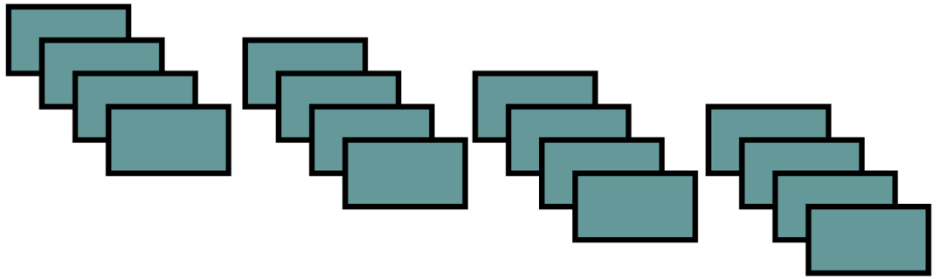


read pages containing entire rows

one row = 212 columns!

is this typical? (it depends)

column store



read only columns needed

in this example: 7 columns

caveats:

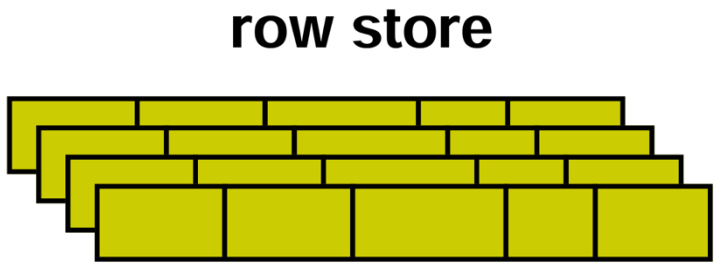
- "select * " not any faster
- clever disk prefetching
- clever tuple reconstruction

Acknowledgement: Slides adapted from VLDB 2009 Tutorial on Column Stores

Telco example explained: compression efficiency

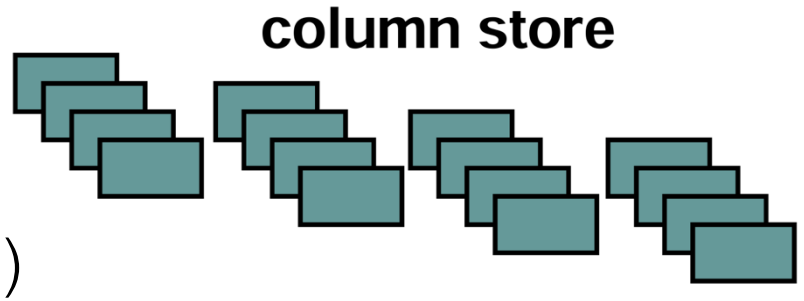
Columns compress better than rows

- Typical row-store compression ratio 1 : 3
- Column-store 1 : 10



Why?

- Rows contain values from different domains
  - More entropy, difficult to dense-pack
- Columns are much more homogeneous
- Caveat: CPU cost (use lightweight compression)

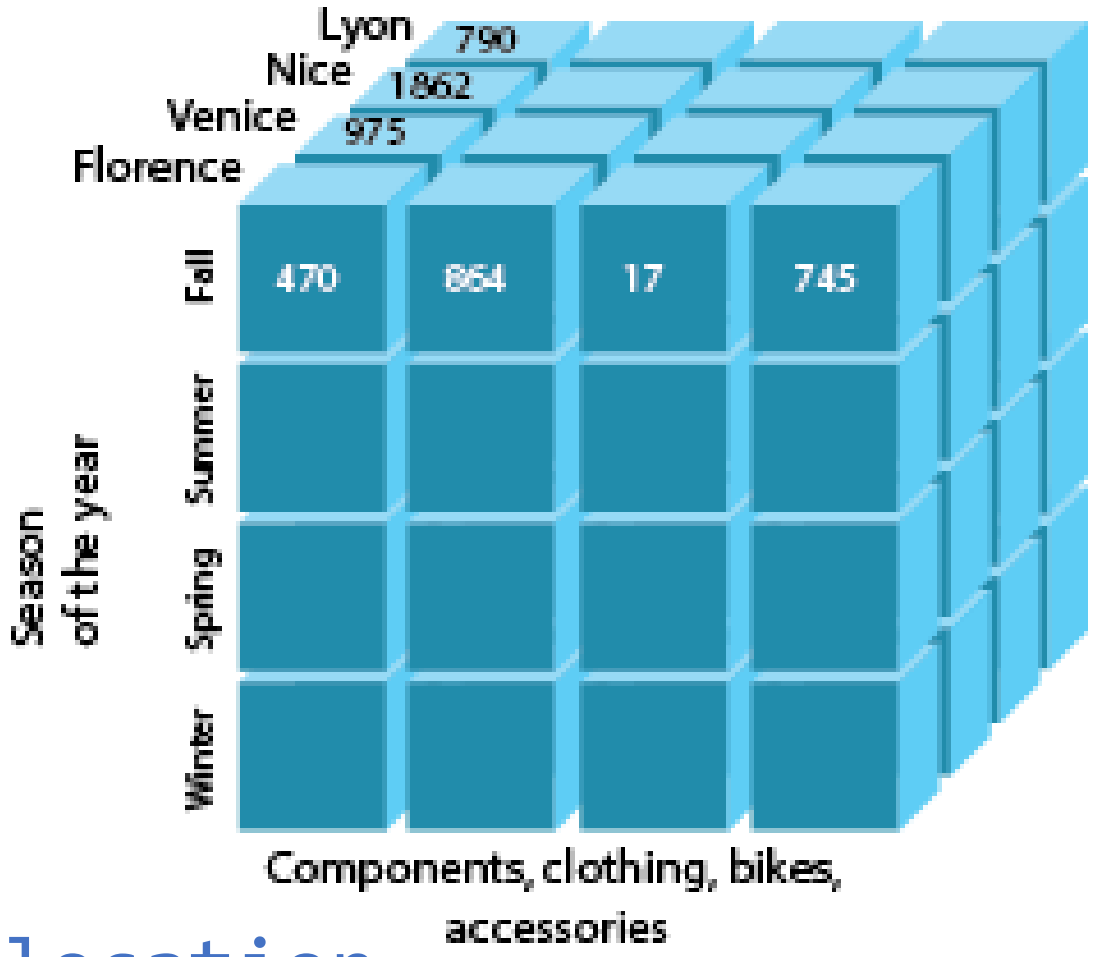


2. Data Cubes

Introducing Data Cubes

Dimensions
Item
season
location

Measure
volume

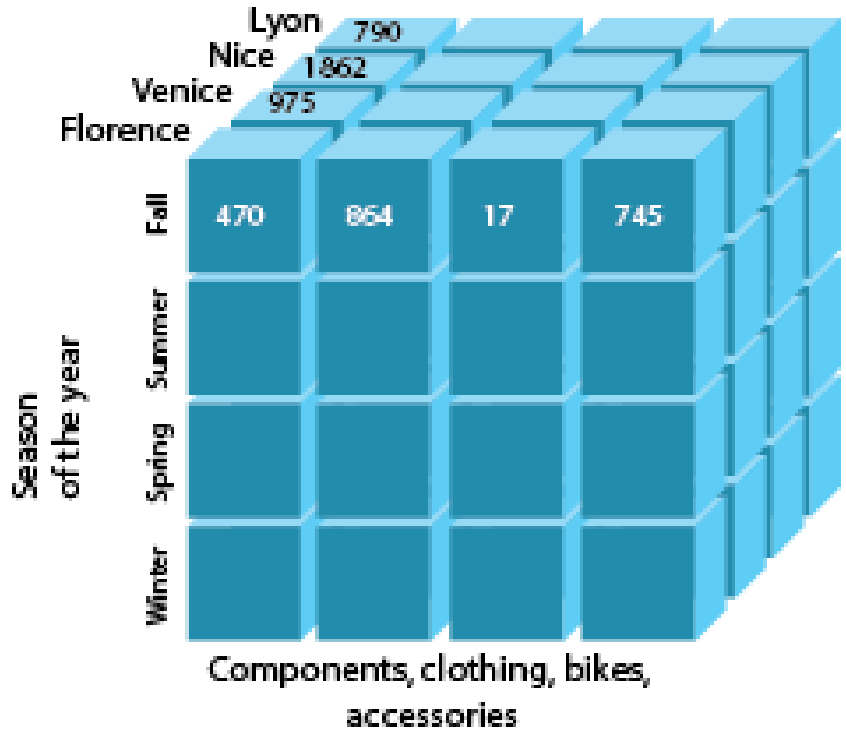



```
SELECT SUM(volume) …
GROUP BY item, season, location
```

Image source: https://galaktika-soft.com/blog/olap-operations-in-data-mining.html

Operations: Slicing and Dicing

Slicing: select a single value in one of the dimensions to create a smaller cube with one less dimension



Slice for
(season = winter)

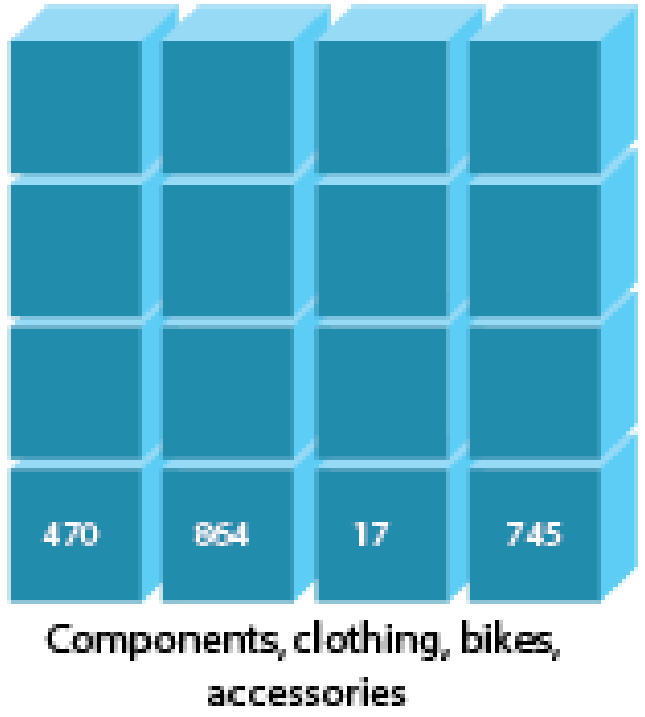
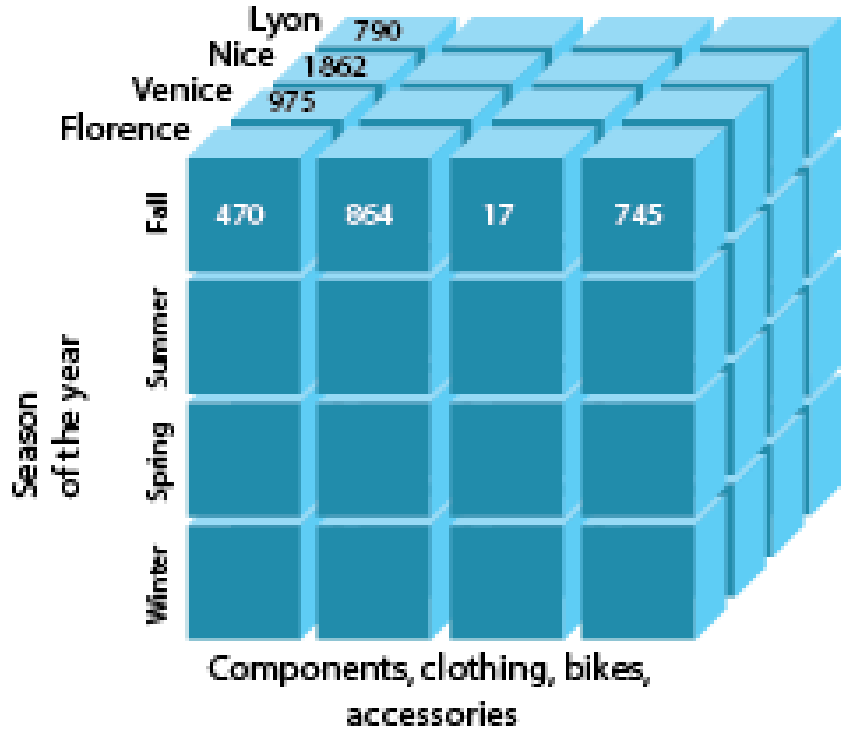


Image source: https://galaktika-soft.com/blog/olap-operations-in-data-mining.html

Operations: Slicing and Dicing

Dicing: select specific values of multiple dimensions to create a new subcube



Dice for
(season = winter or spring)
and
(location = Venice or Florence)
and
(item = components or clothing)

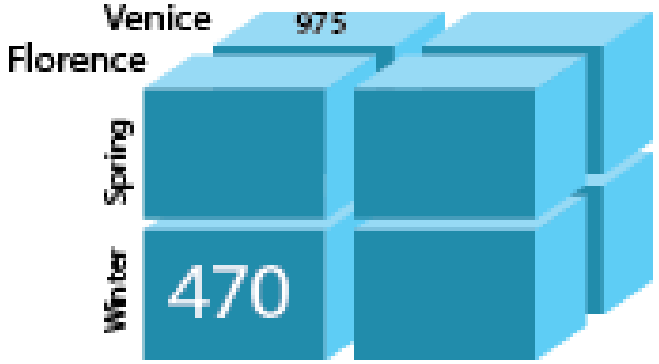


Image source: https://galaktika-soft.com/blog/olap-operations-in-data-mining.html

Operations: Roll up and drill down

Rollup: Moving from a finer to a coarser granularity

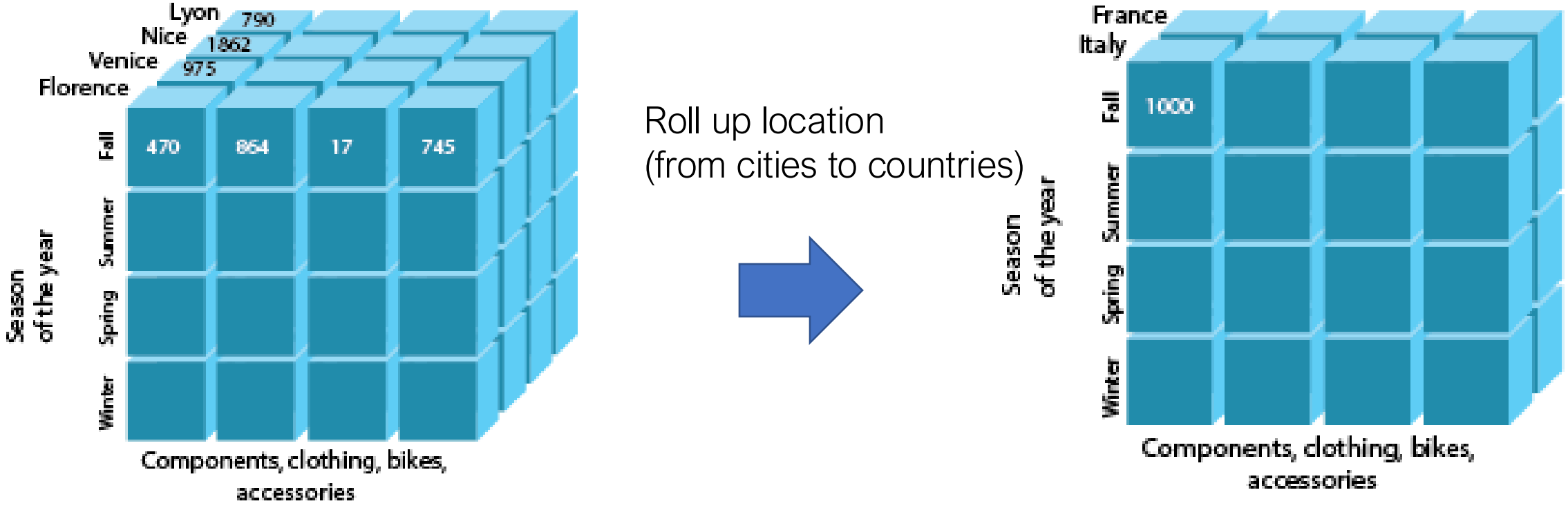
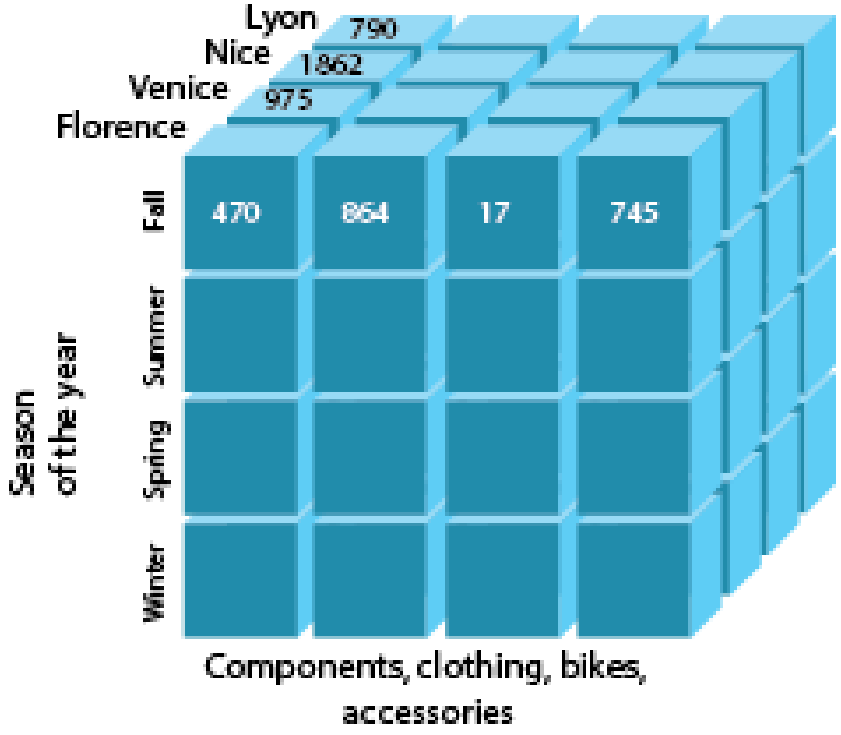


Image source: https://galaktika-soft.com/blog/olap-operations-in-data-mining.html

Operations: Roll up and drill down

Drill-down: Moving from a coarser to a finer granularity



Drill down time
(from quarters to months)

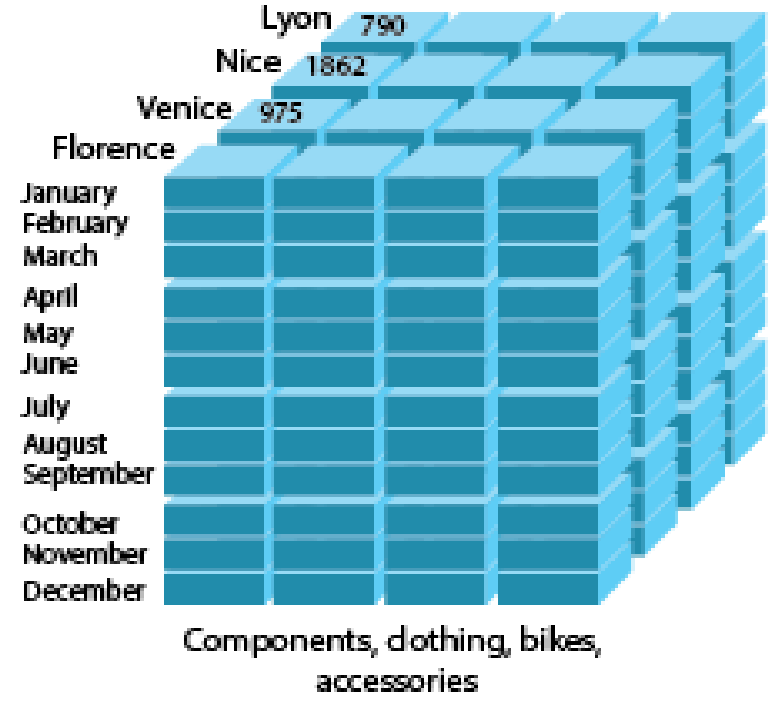


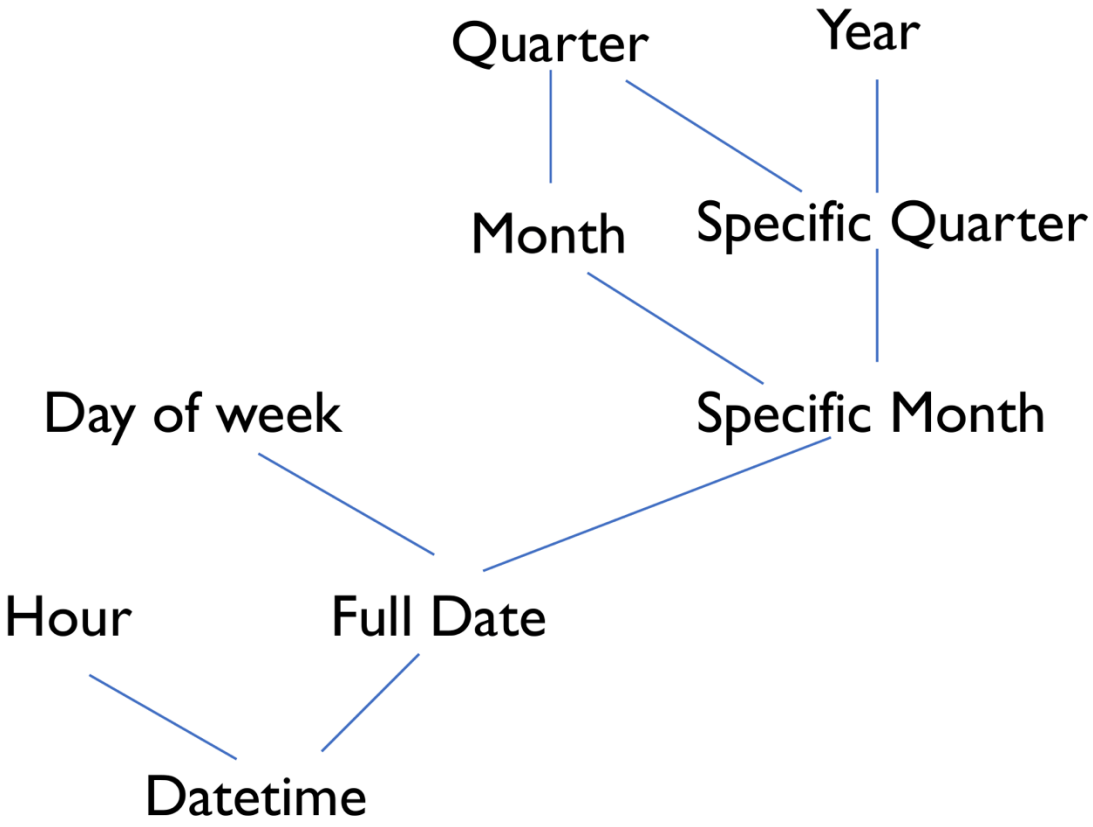
Image source: https://galaktika-soft.com/blog/olap-operations-in-data-mining.html

Partitioning granularity in data cubes

Different partitioning may be useful for different applications

- Q: I want the aggregates per month. What if I set my partitioning based on Full Date and computed the data cube? Can I avoid recomputing the cube?

- Q: What about if I had set my partitioning based on Year?



How to pick the right partitioning granularity?

First, why does this matter?

- If this query is being run once on a petabyte sized warehouse, it is important to get it right!

Think about all the ways you want to slice and dice your data

Pick as fine a granularity of partitioning so that you can recreate all the aggregates, but not so fine as to blow up the query result

- The query result size grows exponentially in the attributes
- This is known as the curse of dimensionality

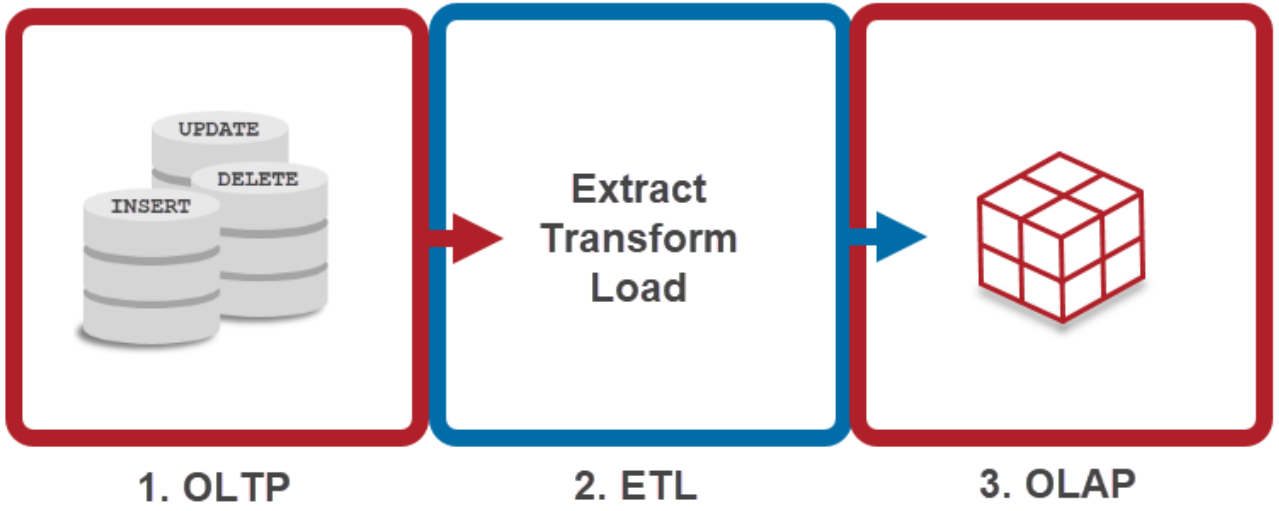
OLAP summary

OLAP is a specialization of relational databases to support analytical processing and report generation

- Typically done in large “batch” operations on the entire database
- Rule of thumb: pick as “coarse-grained” materialized views or query results as will allow you to construct all the cross-tabs that may be necessary

The concepts of OLAP data cubes, hierarchies, slicing/dicing, and rollup/drilldown are valuable to describe what you’re doing when you are exploring your data

OLAP summary



OLAP vs OLTP

- Data Model
- Storage Format

Data warehouses vs Data Lakes

Data cubes

- Slicing and dicing
- Rollup and drill-down
- Selecting partitioning granularity

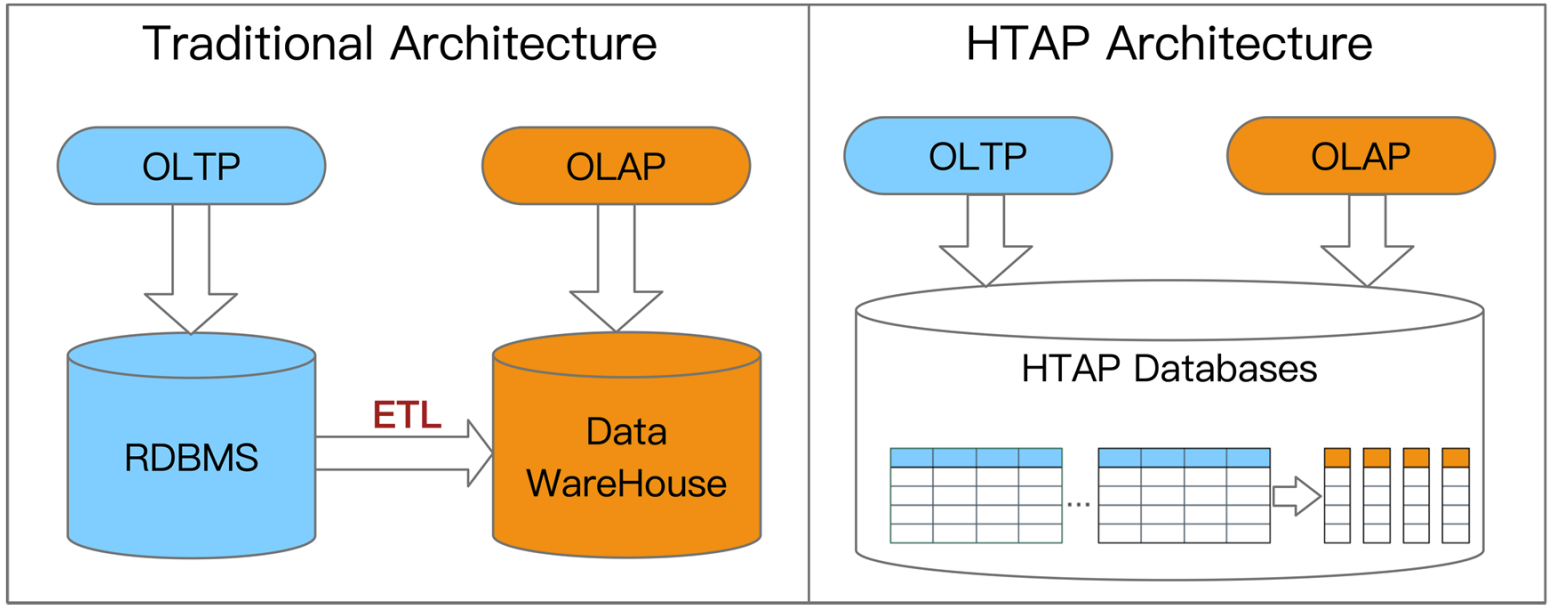
3. A brief intro to HTAP Databases

Slides adapted from *SIGMOD'22 Tutorial HTAP Databases: A Tutorial* by Guoliang Li and Chao Zhang

Motivation

HTAP: Hybrid Transaction Analytical Processing

- Gartner's definition in 2018: supports weaving analytical and transaction processing techniques together as needed to accomplish the business task.

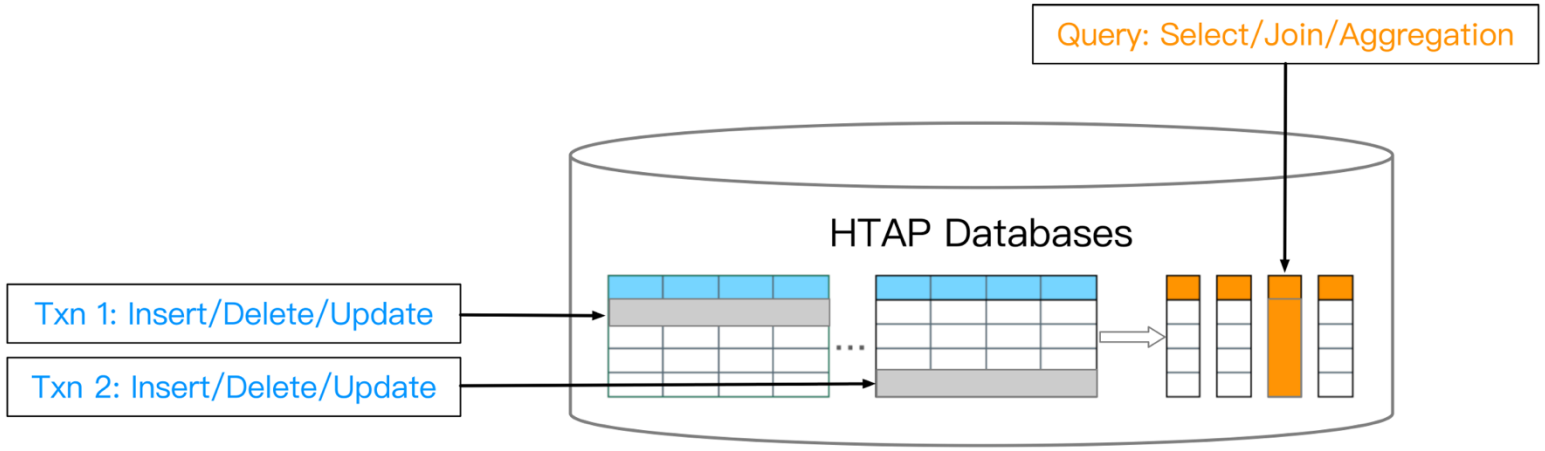


Slides Adapted from SIGMOD'22 Tutorial

Motivation

- Rule of thumb 1: row store is ideal for OLTP workloads
  - Row-wise, update-heavy, short-lived transactions

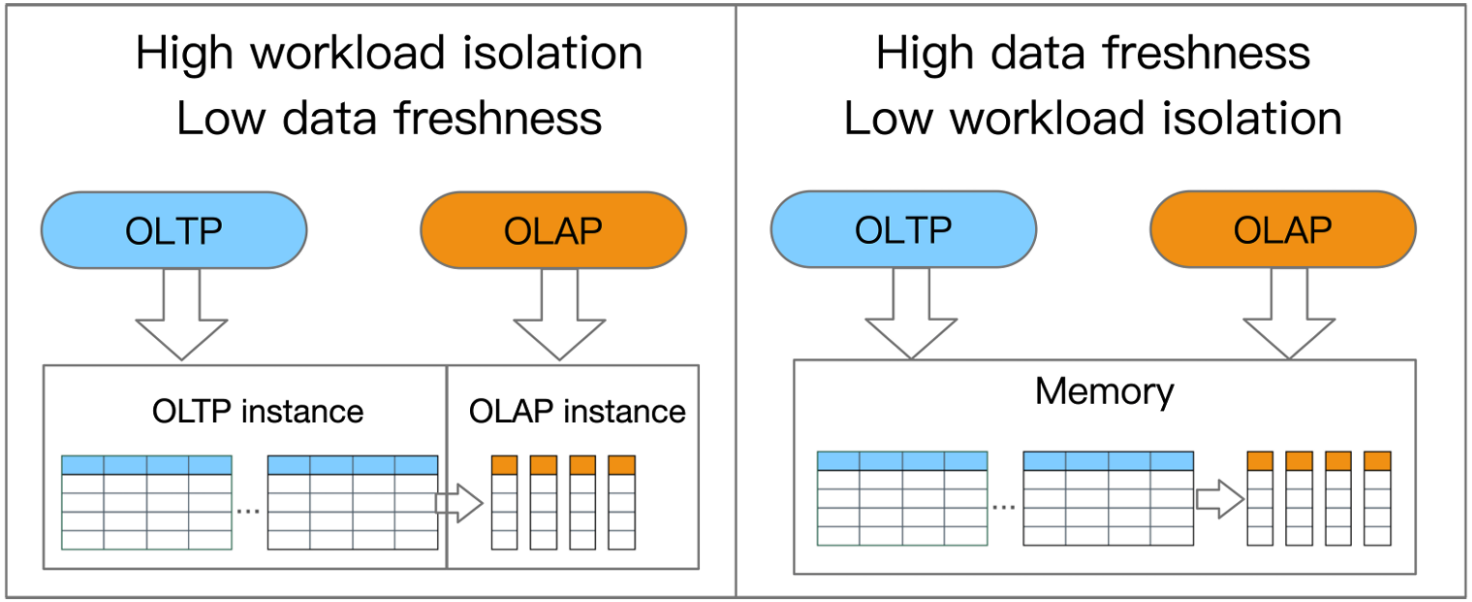
- Rule of thumb 2: column store is best suited for OLAP workload
  - Column-wise, read-heavy, bandwidth-intensive queries



Slides Adapted from SIGMOD'22 Tutorial

A trade-off for HTAP databases

- Workload isolation: the isolation level of handling the mixed workloads
- Data freshness: the portion of latest transaction data that is read by OLAP
- Trade-off for workload isolation and data freshness
  - High workload isolation leads to low data freshness
  - Low workload isolation results in high data freshness



Slides Adapted from SIGMOD'22 Tutorial

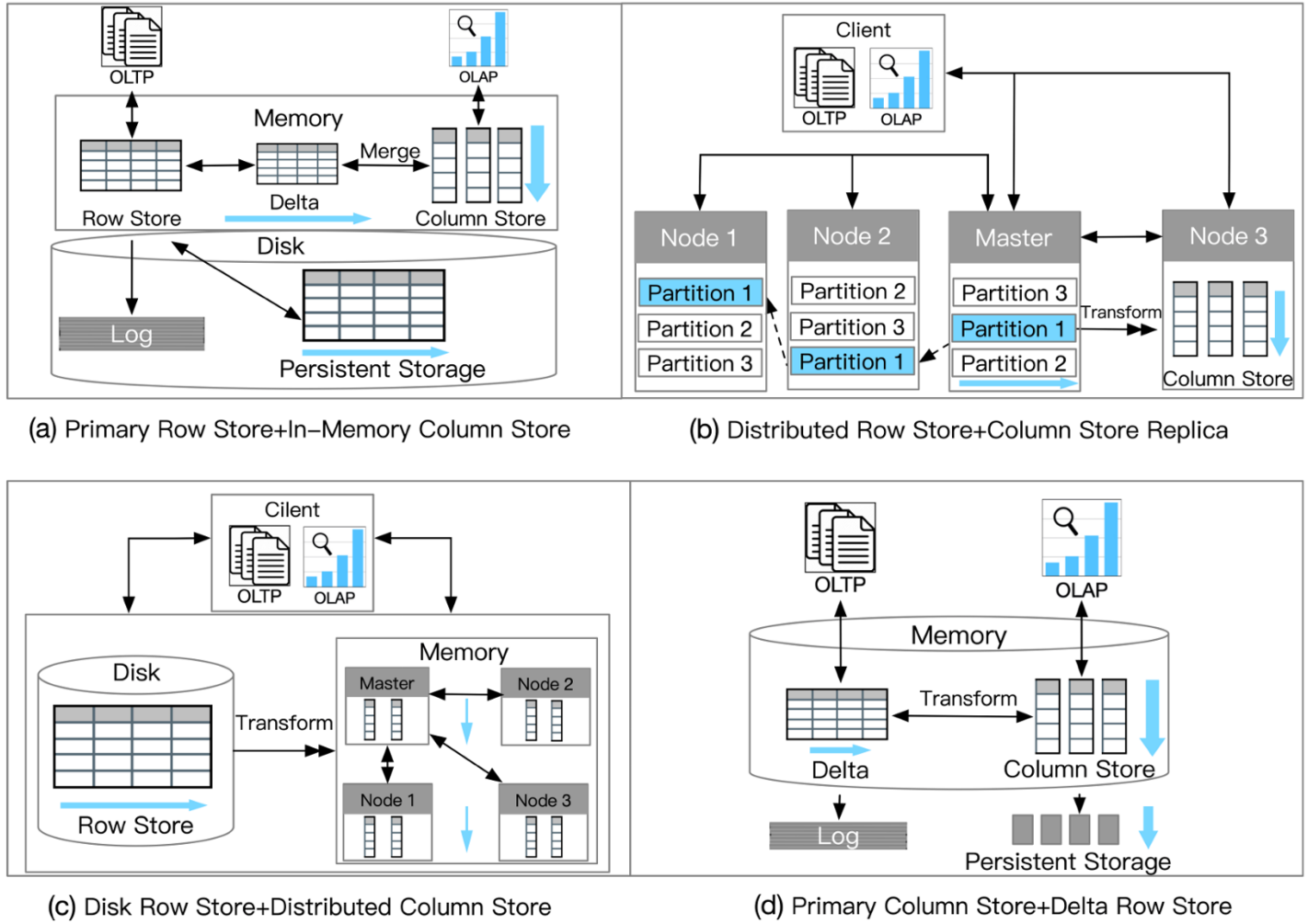
An Overview of HTAP Architectures

(a) **Primary Row Store + In-Memory Column Store**

(b) **Distributed Row Store + Column Store Replica**

(c) **Disk Row Store + Distributed Column Store**

(d) **Primary Column Store + Delta Row Store**



(a) Primary Row Store+In–Memory Column Store

(b) Distributed Row Store+Column Store Replica

(c) Disk Row Store+Distributed Column Store

(d) Primary Column Store+Delta Row Store

A summary of HTAP databases

Category	HTAP Databases	OLTP Throughput	OLAP Throughput	OLTP Scalability	OLAP Scalability	Workload Isolation	Data Freshness
Primary Row Store+ In Memory Column Store	Oracle Dual-Format SQL Server, DB2 BLU	High	High	Medium	Low	Low	High
Distributed Row Store + Column Store Replica	TiDB, F1 Lightning SingleStore	Medium	Medium	High	High	High	Low
Disk Row Store + Distributed Column Store	MySQL Heatwave, Oracle RAC	Medium	Medium	Medium	High	High	Medium
Primary Column Store + Delta Row Store	SAP HANA (without scale-out), Hyper	Medium	High	Low	Medium	Low	High

* Readings: HTAP Databases: What is New and What is Next