L11: Data Warehousing, Data Analytics, and Transactions

DSAN 6300/PPOL 6810: Relational Databases and SQL Programming

Irina Vayndiner

November 9 and 13, 2023

GEORGETOWN UNIVERSITY

Classes left? Reminder: slide 14 from Lecture 1

	DSAN-6300-01 Mon	DSAN-6300-02& PPOL-6810 Thu
1	8/28	8/31
2	9/5 **(Tue!)**	9/7
3	9/11	9/14
4	9/18	9/21
5	9/25 (recording)	9/28
6	10/2	10/5
7	10/16	10/12
8 (midterm)	**10/23**	**10/19**
9	10/30	10/26
10	11/6	11/2
11	11/13	11/9
12	11/20 (on zoom)	11/16
13	11/27	11/30
14 (test)	**12/5 (Tue, 10:30am)**	**12/5 (Tue, 10:30am)**

No classes
Thu 11/24 Thanksgiving
Mon 12/4 – class on 12/5 instead

Logistics

- HW4 (mini-project) is now published
  - 130 points
  - FAA data
  - Due Wed, 12/6 (no extensions!)
    - If you submit by Mon 12/4, you will get 3 bonus points ☺
    - Plan accordingly
  - Multiple queries for one question on mini-project (only)? YES
  - Use Discussion Board
  - You will download your data set today

- After that one more assignment left:
  - Q04 to be available after the next class

Agenda for today's class

- Today:
  - Lecture: Data Warehousing, Data Analytics and Transactions
  - Lab: Rollups, etc.

Outline

- Data Warehousing and Data Analytics
- Online Analytical Processing (OLAP)
- Furter information on Transactions

Overview of Data Analytics in RDBMS

- Using RDBMS for Data analytics
  - Used to make business decisions, e.g.
    - Per individual customer => Online Transaction Processing (OLTP)
      - E.g. what product to suggest for purchase to this customer
    - Across all customers, aggregated => OLAP (Online Analytical Processing)
      - E.g. what products to manufacture/stock, in what quantity
  - Both are important for businesses

Common Steps in Data Analytics

- Gather data from multiple sources into one location
  - Data warehouses sometimes integrate data into a common schema
  - Data often needs to be **extracted** from source formats, **transformed** to common schema, and **loaded** into the data warehouse
    - Can be done as **ETL (extract-transform-load)**, or **ELT (extract-load-transform)**
    - Example: AWS Glue https://aws.amazon.com/glue/

Common Steps in Data Analytics (continued)

- Generate aggregates and reports summarizing data
  - Dashboards showing graphical charts/reports
  - **Online analytical processing (OLAP) systems** allow dice & slice querying
  - Statistical analysis using tools such as R/Python/SAS/SPSS
    - Including extensions for parallel processing of big data

Common steps in data analytics (Continued)

- **Business Intelligence (BI)** is a "standard" Data Analytics
  - **Business intelligence** is "a set of methodologies, processes, architectures, and technologies that transform raw data into meaningful and useful information used to enable more effective strategic, tactical, and operational insights and decision-making." Forrester Research
  - Often – descriptive (e.g. who did what when) or diagnostic (why is it happening)
  - The term **Decision Support (DS)** focuses on reporting and aggregation
  - Often includes Data Visualization

- **Machine learning (ML) analytics**

DATA WAREHOUSING

Data Warehousing

- Data sources often store only current data, not historical data
- Corporate decision making often requires a unified view of all organizational data, including historical data
- A **Data Warehouse(DW)** is a repository of information gathered from multiple sources, often stored under a unified schema of **RDBMS**
  - Examples:
    - AWS Cloud Redshift
    - Teradata
    - Oracle
    - MS SQL Server (also in the Azure Cloud)
    - Snowflake
    - Google Cloud BigQuery

DW Example: AWS Redshift

- Started as ParAccel
- Initial Redshift Release 2012
- RDBMS
- Based on Postgres
- Works well with many BI products, including data integration and visualization
- For Big Data Processing
- “Start small and scale up to Terabytes or Petabytes”

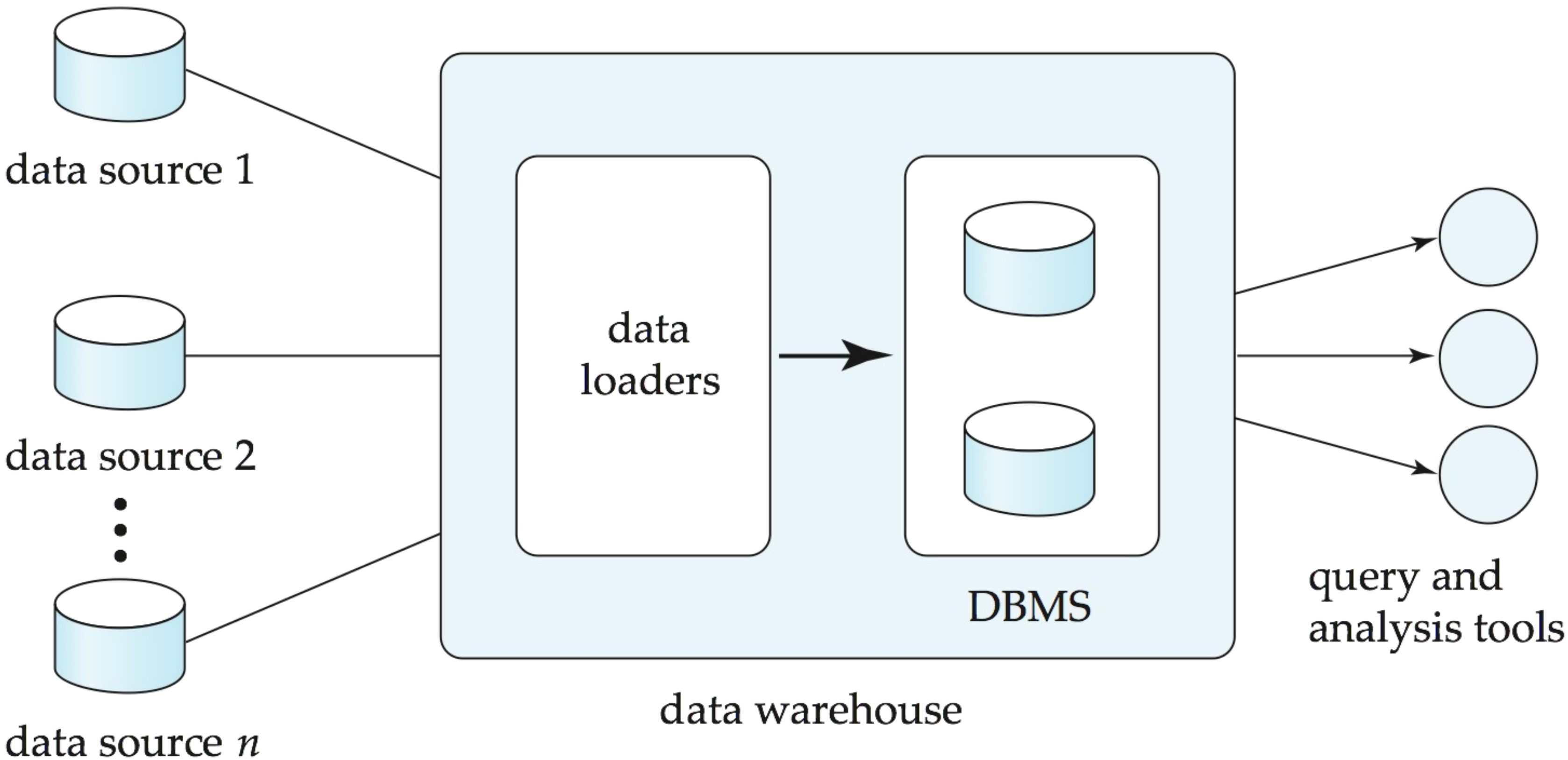
Example DW: Snowflake

- “Accelerate your analytics with the data platform built to enable the modern cloud data warehouse” Snowflake
  - Founded 2012, “DW as a Service”
    - The company's name was chosen as a tribute to the founders' love of winter sports
  - Cloud Storage
    - Automates data warehouse administration and maintenance
    - Runs on Amazon S3, on Microsoft Azure, and on the Google Cloud Platform
  - ANSI SQL compatible, with support for semi-structured data
    - Robust support for JSON-based functions
  - Optimized direct connectors for BI and Analytics tools
    - “Access charts and SQL analytics via Snowsight, the built-in visualization UI for Snowflake”.
  - Sept 2020, Snowflake made an IPO

Example DW: Google Cloud BigQuery

- Enterprise Data Warehouse in Google Cloud
- "BigQuery is a serverless, highly-scalable, and cost-effective cloud data warehouse with an in-memory BI Engine and AI Platform built in." "You can focus on uncovering meaningful insights using familiar SQL without the need for a database administrator." Google
- Has: Easy searchable query list, automated temp table expiration
- Built-in ML capabilities
- Google BigQuery queries need to be optimized to avoid high costs when pulling data
    - One of next lectures is on Database Optimization
    - Needs knowledge of SQL coding to leverage its data analysis capabilities.
        - You can do it! ☺
- More info: https://cloud.google.com/bigquery

Data Warehousing



Data Warehouse Design Issues

- *When and how to gather data*
  - **Source driven architecture**: data sources transmit new information to warehouse
    - Either continuously or periodically (e.g. at night)
  - **Destination driven architecture:** warehouse periodically requests new information from data sources
  - **Synchronous** vs **asynchronous replication**
    - Keeping warehouse exactly synchronized with data sources (e.g. using two-phase commit) is often too expensive
    - Usually OK to have slightly out-of-date data at warehouse
    - Data/updates are periodically downloaded from online transaction processing (OLTP) systems.
- *What schema to use*
  - Schema integration

Data Warehouse Design Issues (continued)

- **Data transformation** and **data cleansing**
  - E.g. correcting inconsistences in addresses (misspellings, zip code errors)
  - E.g. Merge address lists from different sources and purge duplicates
- *How to propagate updates*
  - Warehouse schema may be a (materialized) view of schema from data sources
    - View maintenance
- *What data to summarize*
  - The raw data generated by a transaction-processing system may be too large to store online.
  - However, we can answer many queries by maintaining just summary data obtained by aggregation on a relation
  - For example, instead of storing data about every sale of clothing, we can store total sales of clothing by item name and category.

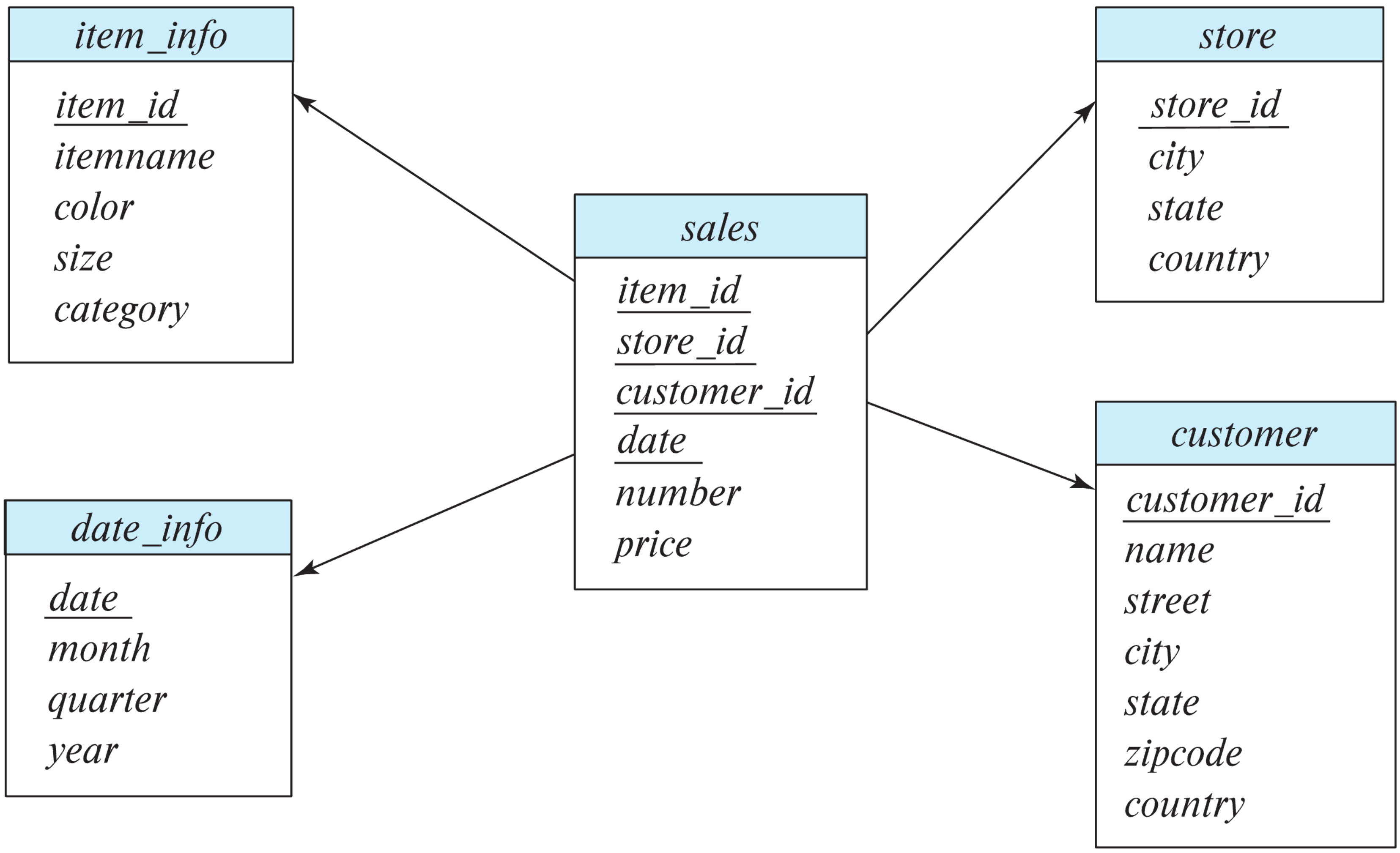
Multidimensional Data and Warehouse Schemas

- Important for HW4!
- Data in warehouses schemas can usually be divided into
  - **Fact tables**, which are large, e.g.
    - *sales*(*item_id, store_id, customer_id, date, number, price*)
  - **Dimension tables**, which are relatively small
    - Store extra information about stores, items, etc.
- Attributes of fact tables can be usually viewed as
  - **Measure attributes**
    - Measure some value, and can be aggregated upon
      - e.g., the attributes such as *number* or *price* of the *sales* relation
  - **Dimension attributes**
    - Dimensions on which measure attributes are viewed
      - e.g., attributes *item_id, color,* and *size* of the *sales* relation
    - Usually ids that are foreign keys to dimension tables

Star Schema

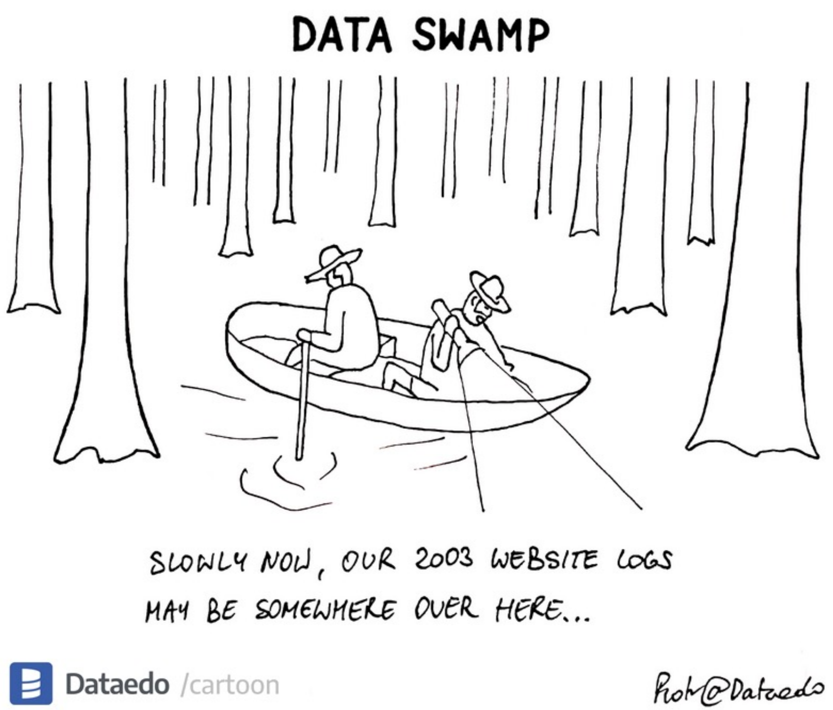
- Fact/dimension schema is called a **star schema**
  - More complicated schema structures
    - **Snowflake schema**: multiple levels of dimension tables
    - May have multiple fact tables
- Typically
  - Fact table joined with dimension tables
    - Need keys or star index for performance
  - Then aggregate on measure attributes of fact table

Data Warehouse Star Schema Example



Data Lake

- Some applications do not find it worthwhile to bring data to a common schema
  - **Data lakes** are repositories which allow data to be stored in multiple formats, without schema integration
  - Less upfront effort, but more effort during querying
    - A **Data swamp** is a deteriorated and unmanaged data lake that is either inaccessible to its intended users or is providing little value



Data Lake/house

- **Data Lakehouse**: newer data management paradigm
  - Combines the capabilities of data lakes and data warehouses
    - Enabling BI and ML on all data
    - Merging DW and Data Lake together into a single system means that data teams can move faster as they are able to use data without needing to access multiple systems.

Database Support for Data Warehouses

- Data in warehouses usually append only, not updated
  - Can avoid concurrency control overheads
- Data warehouses often use **column-oriented storage**
  - Columns are compressed, reducing storage, IO and memory costs significantly
  - Queries can fetch only attributes that they care about, reducing IO and memory cost
- Data warehouses often use **parallel** (MPP) storage and query processing infrastructure

The Modern Data Warehouse

1. Use Cloud Data Warehouse (vs On-Prem)

- Flexible and scalable environment
  - Minimal setup and maintenance (e.g. replication, backups, etc)
  - Storage and compute resources suitable for Big Data scale
    - On demand, automatic scalability is useful for highly unpredictable, ad-hoc query workloads that result from self-service practices.
  - Separates computing from storage
    - One can provision extra computing for the duration of batch job only
      - e.g. on weekends
  - Easy to integrate with other cloud services
  - Can be lower cost
- Cloud Models
  - Fully in the cloud
  - Hybrid (stable part "on prem", dynamic in the cloud)
  - Multiple-clouds DW

Adopted from Fern Halper "The Modern Data Warehouse and Analytics Stack, Six keys for success"

The Modern Data Warehouse (continued)

2. Make Use of Data Variety

- Modern data warehouses support a wide range of data types and analytics
  - Structured is still a top data source, e. g. from multiple RDBMSs
  - Semi- (JSON, XML) and un-structured (text, multimedia) data from Data lakes
  - Files from mainframes
  - Geospatial data
  - Sensors and IoT devices
- Data generated in the cloud, e.g. demographics, weather, home sale prices, etc.
  - Analyze data where it lives, aka "data gravity"
    - Moving large volumes of data is resource intensive

Adopted from Fern Halper "The Modern Data Warehouse and Analytics Stack, Six keys for success"

The Modern Data Warehouse (continued)

3. Use ELT (vs ETL)

- Data is first extracted from multiple sources, loaded, then transformed
  - Transformation can use parallel processing power of the cloud platform
  - Data loading tools use SQL
  - Workflows can be saved and re-used

4. Integrate tools for data discovery and analytics

- Movement toward self-service
  - Many self-service tools can enable immersive, “speed-of-thought” experiences with data and analytics.
  - Modern self-service platforms often provide external integration with R and Python.
- Automation
  - E.g. Data preparation, finding Insights
  - Machine learning infused into the cloud data warehouse to help with optimization

Adopted from Fern Halper “The Modern Data Warehouse and Analytics Stack, Six keys for success”

The Modern Data Warehouse (continued)

5. Data Governance

- Set of processes, roles, standards, and measures that ensure important data assets are formally and consistently managed throughout the enterprise
  - Trusted, curated environment for Data
    - Data quality, availability, usability, consistence, integrity, and security
- For Data Governance in the Cloud the following is needed
  - Visibility into Cloud, including authentication attempts, queries ran, etc
  - Cloud **metadata** that helps with data consistency
    - Where the data was created, who owns it, what it is about, what it is used for, how it is organized, where it is located
      - Data Catalog, including tables, indices, views, etc.
      - Data Provenance or Lineage
        - » Where data originated and how it has been changed and transformed in Data Warehouse

Adopted from Fern Halper "The Modern Data Warehouse and Analytics Stack, Six keys for success"

The Modern Data Warehouse (continued)

6. Integrated Analytics Stack

- Requirements, functionality
  - E.g. how many data sources, what kind of analytics
- Cost considerations
  - Based per Time/Query/Cluster(?)
  - Tune your queries!
  - Plan for how much data you are pulling
  - Tight integration with other tools including security tools

Adopted from Fern Halper “The Modern Data Warehouse and Analytics Stack, Six keys for success”

OLAP

Data Analysis and OLAP

- **Online Analytical Processing (OLAP)**
  - Interactive analysis of data, allowing Big Data data to be summarized and viewed in different ways with negligible delay
    - Analyze multidimensional data from multiple perspectives
  - Optimized for basic analytic operations, e.g.
    - Consolidation (roll-up)
    - Drill-down (navigate through details)
    - Slicing and dicing (data cubes)

- We will use the following relation to illustrate OLAP concepts
  - *sales* (*item_name*, *color*, *clothes_size*, *quantity*)

Example *sales* relation

item_name	*color*	*clothes_size*	*quantity*
dress	dark	small	2
dress	dark	medium	6
dress	dark	large	12
dress	pastel	small	4
dress	pastel	medium	3
dress	pastel	large	3
dress	white	small	2
dress	white	medium	3
dress	white	large	0
pants	dark	small	14
pants	dark	medium	6
pants	dark	large	0
pants	pastel	small	1
pants	pastel	medium	0
pants	pastel	large	1
pants	white	small	3
pants	white	medium	0
pants	white	large	2
shirt	dark	small	2
shirt	dark	medium	6
shirt	dark	large	6
shirt	pastel	small	4
shirt	pastel	medium	1
shirt	pastel	large	2
shirt	white	small	17
shirt	white	medium	1
shirt	white	large	10
skirt	dark	small	2
skirt	dark	medium	5
...	...	...	...
...	...	...	...

Cross Tabulation of sales by item_name and color

clothes_size **all**

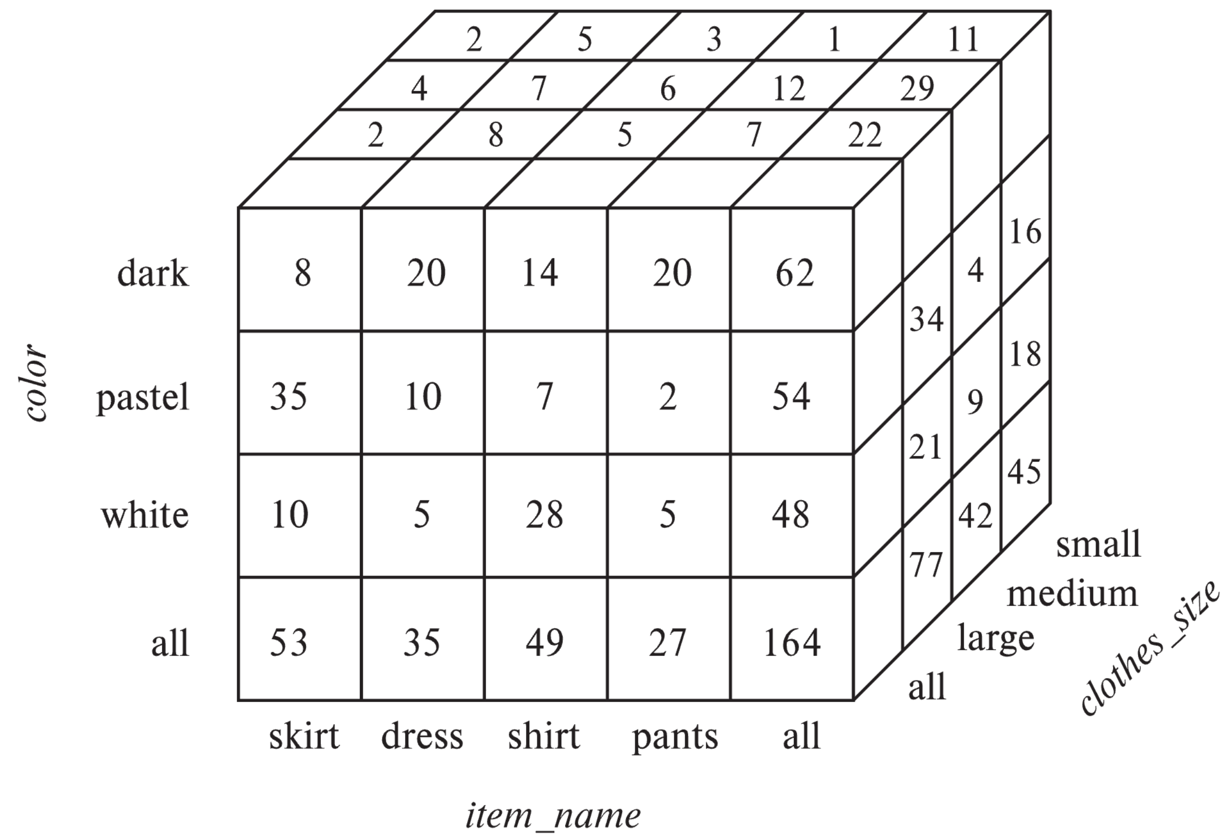
color

item_name	dark	pastel	white	total
skirt	8	35	10	53
dress	20	10	5	35
shirt	14	7	28	49
pants	20	2	5	27
total	62	54	48	164

- The table above is an example of a **cross-tabulation** (**cross-tab**), also referred to as a **pivot-table**.
  - Values for one of the dimension attributes form the *row headers*
    - *Color in the above example*
  - Values for another dimension attribute form the *column headers*
  - Values in individual cells are (aggregates of) the values of the dimension attributes that specify the cell.

Data Cube

- A **data cube** is a multidimensional generalization of a cross-tab
- Can have n dimensions; we show 3 below

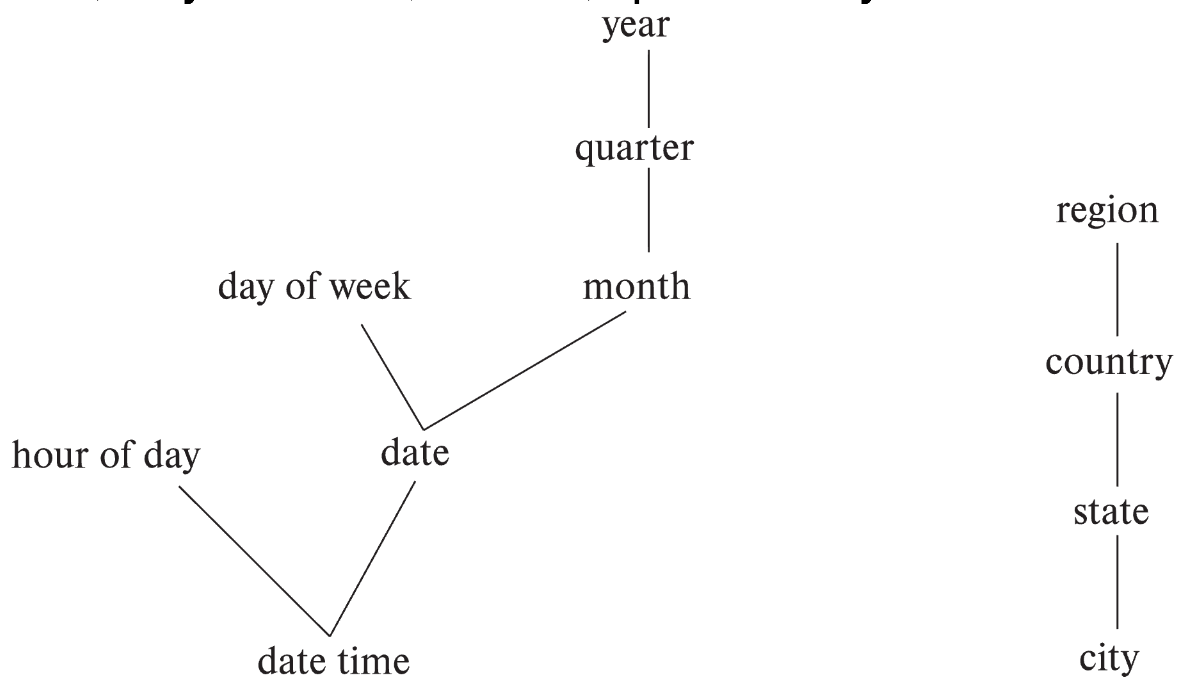


Online Analytical Processing (OLAP) Operations

- **Pivoting:** changing the dimensions used in a cross-tab
  - E.g. moving *colors* to column names
- **Slicing:** creating a cross-tab for fixed values only
  - E.g. fixing *color* to white and *size* to small
  - Sometimes called **dicing**, particularly when values for multiple dimensions are fixed.
  - Goal: divide a quantity of information up into smaller parts, especially in order to analyze it more closely or in different ways.
- **Rollup:** moving from finer-granularity data to a coarser granularity
  - Advanced Aggregation
    - E.g. moving from aggregates by day to aggregates by month or year
- **Drill down:** The opposite operation - that of moving from coarser-granularity data to finer-granularity data

Hierarchies on Dimensions

- **Hierarchy** on dimension attributes: lets dimensions be viewed at different levels of detail
- E.g., the dimension *datetime* can be used to aggregate by hour of day, date, day of week, month, quarter or year



(a) time hierarchy

(b) location hierarchy

Cross Tabulation With Hierarchy

- Cross-tabs can be easily extended to deal with hierarchies
  - Can drill down or roll-up on a hierarchy
  - E.g. hierarchy: *item_name* → *category*

clothes_size: **all**

category	*item_name*	*color*				
		dark	pastel	white	total	
womenswear	skirt	8	8	10	53	
	dress	20	20	5	35	
	subtotal	28	28	15		88
menswear	pants	14	14	28	49	
	shirt	20	20	5	27	
	subtotal	34	34	33		76
total		62	62	48		164

Relational Representation of Cross-tabs

- Cross-tabs can be represented as relations
- We use the value **all** to represent aggregates.
- The SQL standard actually uses *null* values in place of **all**
  - Works with any data type
  - But can cause confusion with regular null values.

item_name	*color*	*clothes_size*	*quantity*
skirt	dark	**all**	8
skirt	pastel	**all**	35
skirt	white	**all**	10
skirt	**all**	**all**	53
dress	dark	**all**	20
dress	pastel	**all**	10
dress	white	**all**	5
dress	**all**	**all**	35
shirt	dark	**all**	14
shirt	pastel	**all**	7
shirt	white	**all**	28
shirt	**all**	**all**	49
pants	dark	**all**	20
pants	pastel	**all**	2
pants	white	**all**	5
pants	**all**	**all**	27
all	dark	**all**	62
all	pastel	**all**	54
all	white	**all**	48
all	**all**	**all**	164

OLAP IN SQL

Pivot Operation using SQL

- ```
  select *
  from sales
  pivot (
        sum(quantity)
        for color in ('dark','pastel','white')
  )
  order by item name;
  ```
- Not available in MySQL

item_name	*clothes_size*	*dark*	*pastel*	*white*
dress	small	2	4	2
dress	medium	6	3	3
dress	large	12	3	0
pants	small	14	1	3
pants	medium	6	0	0
pants	large	0	1	2
shirt	small	2	4	17
shirt	medium	6	1	1
shirt	large	6	2	10
skirt	small	2	11	2
skirt	medium	5	9	5
skirt	large	1	15	3
  ```

Cube Operation

- The **cube** operation computes union of **group by**' s on every subset of the specified attributes
- E.g. consider the query

  select *item_name, color, size,* **sum**(*number*)
  from *sales*
  group by cube(*item_name, color, size*)

  This computes the union of **eight** different groupings of the *sales* relation:

  { (*item_name, color, size*), (*item_name, color*),
  (*item_name, size*), (*color, size*),
  (*item_name*), (*color*),
  (*size*), ( ) }

  where ( ) denotes an empty **group by** list.
- For each group, the result contains the null value for attributes not present in the group.
- Not available in MySQL

Cube Operation Example

select *item_name*, *color*, **sum**(*number*)
  from *sales*
group by cube(*item_name*, *color*)

The SQL standard actually uses the **null** value in place of **all**, but to avoid confusion with regular null values, we used **all**

4 groups:
{ (*item_name, color*),
  (*color*),
  (*item_name*),
  ( ) }

item_name	*color*	*clothes_size*	*quantity*
skirt	dark	**all**	8
skirt	pastel	**all**	35
skirt	white	**all**	10
skirt	**all**	**all**	53
dress	dark	**all**	20
dress	pastel	**all**	10
dress	white	**all**	5
dress	**all**	**all**	35
shirt	dark	**all**	14
shirt	pastel	**all**	7
shirt	white	**all**	28
shirt	**all**	**all**	49
pants	dark	**all**	20
pants	pastel	**all**	2
pants	white	**all**	5
pants	**all**	**all**	27
all	dark	**all**	62
all	pastel	**all**	54
all	white	**all**	48
all	**all**	**all**	164

Grouping Function

- The function **grouping()** can be applied to an attribute
  - Returns
    - 1, if the value is a **null** value representing **all**
    - 0, in all other cases.
  - Helps to substitute nulls to all
  - **Grouping** (not cube!) works in MySQL
    - More info: https://dev.mysql.com/blog-archive/mysql-8-0-grouping-function/
  - Example of use:

```
select case when grouping(item_name) = 1 then 'all’
                 else item_name end as item_name,
            case when grouping(color) = 1 then 'all’
                  else color end as color,
            'all' as clothes size,
            sum(quantity) as quantity
from sales
group by cube(item name, color);
```

Grouping Function SQL Flavors

- Instead of *case* function:
  - Can also use *decode*() – in Oracle
  - Can also use if() – in MySQL

  in the **select** clause to replace nulls by a value, such as **all**

  - E.g., in mysql replace *item_name* in the first query by
    if(**grouping**(item_*name*), 'all', *item_name*) as item_name

Rollup Construct: Advanced Aggregation

- **Rollup** extension of the *group by* clause
  - Allows you to include extra rows, e.g. representing the subtotals along with the grand total.
  - You can use a single query to generate multiple grouping sets.
- Generates union on every prefix of specified list of attributes

  select *item_name*, *color*, *size*, **sum**(*number*)
  from *sales*
  group by rollup(*item_name, color, size*)

  Group by rollup generates union of four groups:

  { (*item_name, color, size*), (*item_name, color*), (*item_name*), ( ) }
- Notes: compare with Cube (was 8!)
  - Order of attributes is important!
  - Rollup is available in MySQL, cube is not

- More info: https://dev.mysql.com/doc/refman/8.0/en/group-by-modifiers.html

Rollup Construct: Advanced Aggregation (continued)

- Rollup can be used to generate aggregates at multiple levels of a hierarchy.

- E.g., suppose table *itemcategory*(*item_name, category*) gives the category of each item. Then

  select *category, item_name*, **sum**(*number*)
  from *sales, itemcategory*
  where *sales.item_name = itemcategory.item_name*
  group by rollup(*category, item_name*)

  would give a summary by (*category*, *item_name)* and by (*category).*

Rollup Construct SQL Flavors

In MySQL syntax of the **rollup** construct is slightly different.

E.g. compare standard SQL , e.g. in SQL Server:

select *item_name*, *color*, *size*, **sum**(*number*)
from *sales*
group by rollup(*item_name, color, size*)

To MySQL:

select *item_name*, *color*, *size*, **sum**(*number*)
from *sales*
group by *item_name, color, size* **with rollup**

Note: You can use *having* and *order by* with *group by ..with rollup*

OLAP Implementations

- OLAP systems that use multidimensional arrays in memory to store data cubes, and are referred to as **multidimensional OLAP (MOLAP)** systems.
- OLAP implementations using only relational database features are called **relational OLAP (ROLAP)** systems
- Hybrid systems, which store some summaries in memory and store the base data and other summaries in a relational database, are called **Hybrid OLAP (HOLAP)** systems.

Reporting and Visualization

- **Reporting tools** help create formatted reports with tabular/graphical representation of data
  - E.g. SQL Server reporting services, MS Power BI, Crystal Reports
- **Data visualization** tools help create interactive visualization of data
  - E.g Tableau, FusionChart, plotly, Datawrapper, Google Charts, etc. etc.
  - Front-end typically based on HTML+JavaScript

Acme Supply Company, Inc.
Quarterly Sales Report

Period: Jan. 1 to March 31, 2009

Region	Category	Sales	Subtotal
North	Computer Hardware	1,000,000	
	Computer Software	500,000	
	All categories		1,500,000
South	Computer Hardware	200,000	
	Computer Software	400,000	
	All categories		600,000
		Total Sales	2,100,000