CS6111
Advanced Database Systems
Spring 2014

Computer Science Department
Columbia University

OLAP and Data Warehousing

* Marked slides courtesy of:

Surajit Chaudhuri
Microsoft Research

Umeshwar Dayal
Hewlett-Packard Labs.

Additional slides courtesy of:

Julia Stoyanovich
Columbia University

What is OLAP?

- **O**n-**L**ine **A**nalytical **P**rocessing
- Information technology to help the knowledge worker (executive, manager, analyst) make faster and better decisions.
- OLAP is an element of decision support systems (DSS).

Running Example: Car Sales

- Cars: carId, make, model, color
- Dealers: dealerId, city, state
- Time of Sale: tid, year, month, day
- Sales: carId, dealerId, tid, price

OLTP Queries: Examples

- create a new sales record that indicates that a red VW Golf was sold in Boston, MA
- see how many black and silver VW Passats were sold at dealership #123 on April 11, 2013

5

OLAP Queries: Examples

- Analyze comparative sales of the different colors of VW Golf by state
- See which months are particularly favorable to the sale of different VW models and colors
- Rank VW dealerships by revenue, displaying a ranked list of dealerships and % differences in sales between each dealership and the one ranked 1 place higher

6

OLAP vs. OLTP

	OLTP	OLAP
User	Clerk, IT professional	Knowledge worker
Function	Day to day operations	Decision support
DB design	Application-oriented (E-R based)	Subject-oriented (Star, snowflake)
Data	Current, Isolated	Historical, Consolidated
View	Detailed, Flat relational	Summarized, Multidimensional
Usage	Structured, Repetitive	Ad hoc
Unit of work	Short, simple transaction	Complex query
Access	Read/write	Read mostly
Operations	Index/hash on prim. key	Lots of scans
# Records accessed	Tens	Millions
# Users	Thousands	Hundreds
Db size	100 MB - GB	100 GB - TB
Metric	Trans. throughput	Query throughput, response

Data Warehouse

- A decision support database that is maintained separately from the organization's operational databases.
- A data warehouse is a
  - subject-oriented,
  - integrated,
  - time-varying,
  - non-volatile

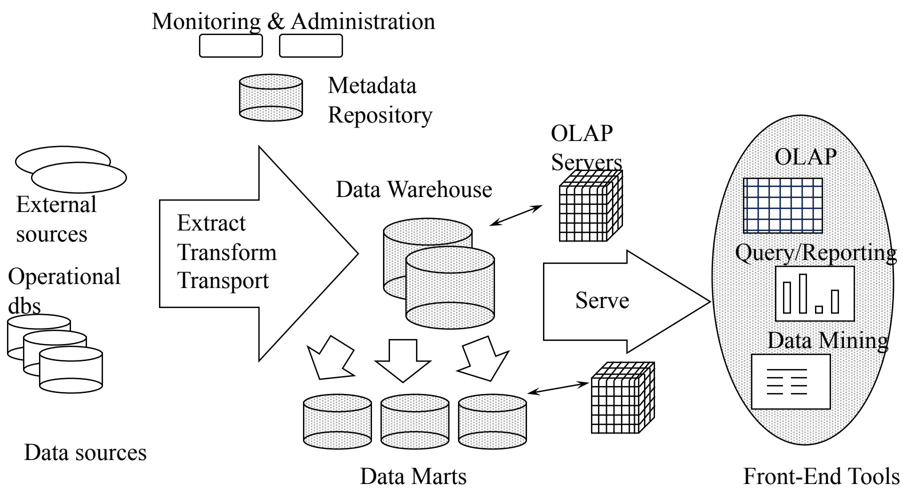
  collection of data that is used primarily in organizational decision making.

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

Why Separate Data Warehouse

- Performance
  - » Op dbs designed & tuned for known trans. workloads.
  - » Complex OLAP queries would degrade performance for operational transactions.
  - » Special data organization, access & implementation methods needed for multidimensional views & queries.
- Function
  - » Missing data: Decision support requires historical data, which op dbs do not typically maintain.
  - » Data consolidation: Decision support requires data consolidation (aggregation, summarization) from many heterogeneous sources: op dbs, external sources.
  - » Data quality: Different sources typically use inconsistent data representations, codes, and formats, which have to be reconciled.

Data Warehousing Architecture



OLAP Queries: Challenges

- Many AND, OR in the WHERE clause
- Self-join, nested sub-queries
  - » Last year's sales vs this year's sales for each product
  - » Show reps for whom every sale has been more than $15000
- Extensive use of aggregation, often on related datasets
- Aggregation over time periods
- Ranking
- Use of statistical functions
- Very large datasets
- Expectation of an interactive response time

OLAP Query Tools

- Goal of OLAP is to support ad-hoc querying for the business analyst (Power user)
- Business analysts are familiar with spreadsheets
- Extend spreadsheet analysis model to work with warehouse data
  - » Large data set
  - » Semantically enriched to understand business terms (e.g., time, geography)
  - » Combined with reporting features
- Multidimensional view of data is the foundation of OLAP.

Multidimensional Data Model

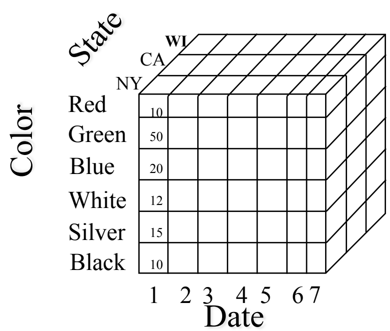
- Database is a set of **facts** (points) in a multidimensional space
- A fact has a **measure** dimension
  - » quantity that is analyzed, e.g., sale amount, budget
- A set of **dimensions** with respect to which data is analyzed
  - » e.g., store, product, date associated with a sale amount
- Dimensions form a sparsely populated coordinate system
- Each dimension has a set of attributes
  - » e.g., owner, city and county of store

Attribute Hierarchies

- Attributes of a dimension may be related
- An m:1 dependency is most common
- Dependency graph may be:
  - » Hierarchy: e.g.,
    city -> state -> country
  - » Lattice:
    date -> month -> year
    date -> week -> year
- Hierarchies are most common
- Dependencies influence choice of operations and data representation

Multidimensional Data

Sales volume as a function of product, time, geography



Fact data: Sales volume in $100

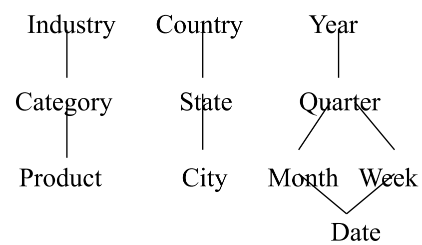
Dimensions

Color, State, Date

Attributes

date (year, month, day)

Attribute Hierarchies and Lattice



Operations on Multidimensional Data Model

- Aggregation (***roll-up***) of detailed data to create *summary data*
- Navigation to detailed data (***drill-down***) from summary
- Selection (***slice)*** defines a subcube
  - Project the cube on fewer dimensions by specifying coordinates of remaining dimensions
  - e.g., sales where state = NY and month = Jan
- Calculation
  - Within a dimension, e.g., (sales - expense) by state
  - Across dimensions
- Ranking
  - top 3% of states by average sales
- Window Queries

Roll-up and Drill-Down

- Roll-Up: Use of aggregation
  - » *dimension reduction:*
    - – e.g., total sales by state by color
    - – e.g., total sales by state
  - » *navigating attribute hierarchy:*
    - – e.g., sales by **city** -> total sales by **state** -> total sales by **country**
    - – e.g., total sales by **city** and year -> total sales by **state** and year -> total sales by **country**
- Drill-Down**:** Inverse operation of roll-up
  - » Provides the data set that was aggregated
    - – e.g., show "base" data for total sales figure for CA state

Slice and Dice

- What colors of Golf are not doing so well?

```
Select     color, sum(price)
From       SALES
Where      model = 'Golf'      slicing
Group By color                 dicing
```

- Keep slicing if results are uniform

Multiple Aggregations

- Create a 2-dimensional spreadsheet that shows sum of sales by year as well as by state
- Each subtotal requires a separate aggregate query

	STATE	
Y E A R		Sum by Year
	Sum By State	

 19

Example: Multiple Aggregations

	WI	CA	Total
2011	63	81	144
2012	38	107	145
2013	75	35	110
Total	176	223	399

20

Generalization: The Data Cube

- Base tuples
- Aggregate tuples:
  - » one aggregation for each subset of dimensions (powerset)
  - » exponential number of subsets, but can optimize the computation
- Example
  - » N = 3 dimensions
    - – model = {Golf, Jetta}
    - – color = {red, black, white}
    - – state = {NY, CA, WI}
  - » How many aggregate tuples in the data cube?
    - – face – 1D agg; edge – 2D agg; corner – 3D agg

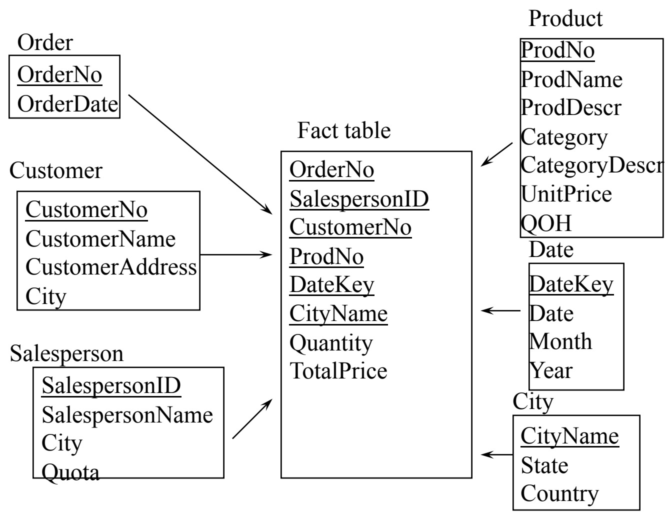
ROLAP and MOLAP

- Relational OLAP (ROLAP)
  - » Relational and Specialized Relational DBMS to store and manage warehouse data
  - » OLAP middleware to support missing pieces
    - – Optimize for each DBMS backend
    - – Aggregation Navigation Logic
    - – Additional tools and services
- Multidimensional OLAP (MOLAP)
  - » Array-based storage structures
  - » Direct access to array data structures

Warehouse Database Schema

- Entity-Relationship design techniques not appropriate
- Design should reflect multidimensional view
- Typical schemas:
  - » Star Schema
  - » Snowflake Schema
  - » Fact Constellation Schema

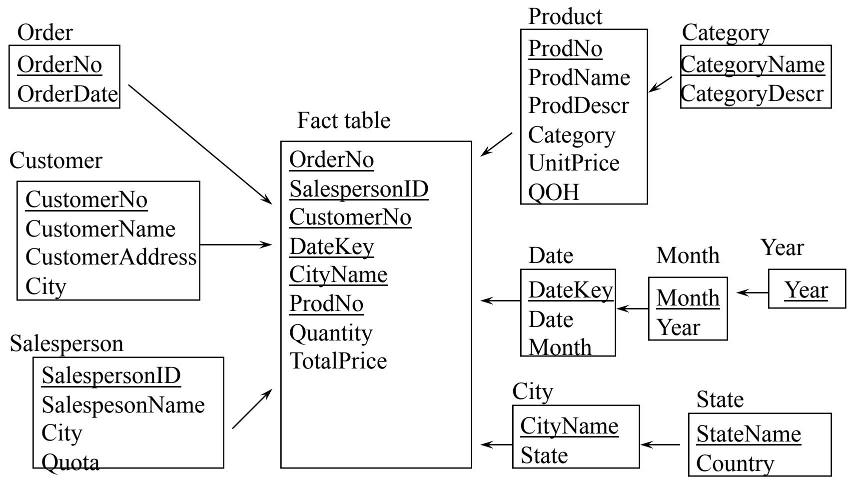
Example of a Star Schema



Star Schema and Variants

- A single fact table and a single table for each dimension
- Generated keys are used for performance and maintenance reasons
- *Fact constellation:* Multiple Fact tables that share common dimension tables
  - » Example: ProjectedExpense and ActualExpense may share dimensional tables
- *Snowflake Schema:* Represents dimensional hierarchy by normalization

Example of a Snowflake Schema



Performance Considerations

- Normalization for dimension tables
  - » Read-only data, so no update anomalies
  - » Fewer joins – better performance
- Pre-computation of summary tables
  - » Re-use can speed up performance
  - » How can we use pre-computed results effectively?
- Data is very large, dimension data often sparse
  - » Crucial to use indexes effectively
  - » Need for new indexing techniques: bitmap indexes, join indexes

27

Bit Map Index

- An alternative representation of RID-list
- Comparison, join and aggregation operations are reduced to *bit arithmetic*
- Specially advantageous for low-cardinality domains
  - » Significant reduction in space and I/O (30:1)
  - » Adapted for higher cardinality domains
  - » Compression (e.g., run-length encoding) exploited
  - » Upper Bound of 2R words for any bitmap over R rows [Hasan & Sinha, 1997]

28

Bit Map Index Example

Base Table

Cust	Region	Rating
C1	N	H
C2	S	M
C3	W	L
C4	W	H
C5	S	L
C6	W	L
C7	N	H

Region Index

RowID	N	S	E	W
1	1	0	0	0
2	0	1	0	0
3	0	0	0	1
4	0	0	0	1
5	0	1	0	0
6	0	0	0	1
7	1	0	0	0

Rating Index

RowID	H	M	L
1	1	0	0
2	0	1	0
3	0	0	1
4	1	0	0
5	0	0	1
6	0	0	1
7	1	0	0

Customers where ***Region = W*** ***and*** ***Rating = L***

Join Index

- Traditional index maps the value in a column to a list of rows with that value
- Join index maintain relationships between attribute value of a dimension and the matching rows in the fact table
- Join index may span multiple dimensions (composite join index)
  - » Use join index to identify regions of cartesian product that are of interest
  - » Few people in Southern California may buy umbrellas

Join Index over Star Schema