An Overview of Data Warehousing and OLAP Technology

Slides modified by Michael (original: Otto Bian)
Discussion: Sarah

Motivation

- **Data is used to make decisions**
- However, businesses have a lot of data, operational data and facts.
- Data is usually in different databases and in different physical places.

- Decision makers need to access information (data that has been summarized) virtually on the single site.
- Access needs to be fast regardless of the size of data, and how data's age.

Decision Support

- Computerized information systems that support decision-making.
- Decision support systems consolidate data from heterogeneous sources to help knowledge workers **make better decisions.**

Data Warehouse

"<u>subject-oriented</u>, <u>integrated</u>, <u>time-varying</u>, <u>non-volatile</u> collection of data that is used primarily in organizational decision making."

- Major subject areas
- Data from different sources
- Changes with new data over time
- Retains history of old data

	OLTP	OLAP
Users	Clerk, IT professional	Knowledge worker
Function	Day to day operations	Decision support
DB Design	Application-oriented	Subject-oriented
Data	Current, up-to-date detailed.	Historical, summarized, multidimensional,…
Usage	repetitive	Ad-hoc
Access	Read/write	Lots of scans
Unit of work	Short, simple transaction	Complex query
# rec accessed	tens	Millions
# users	thousands	Hundreds
DB size	100 MB- GB	100 GB-TB
Metric	Transaction throughput	Query throughput

Discussion (in pairs)

- Now that we have discussed the differences between OLTP and OLAP, what are some real world uses of OLTP and OLAP? Which types of businesses require more out of OLAP vs OLTP? (From Jeffrey)

DW vs DB

- Performance reasons:
  - OLAP requires special data organization that supports multidimensional views.
  - OLAP queries would degrade operational DB.
  - OLAP is read only.
  - No concurrency control and recovery.
- Decision-support requires historical, consolidated data.

OLAP Architecture

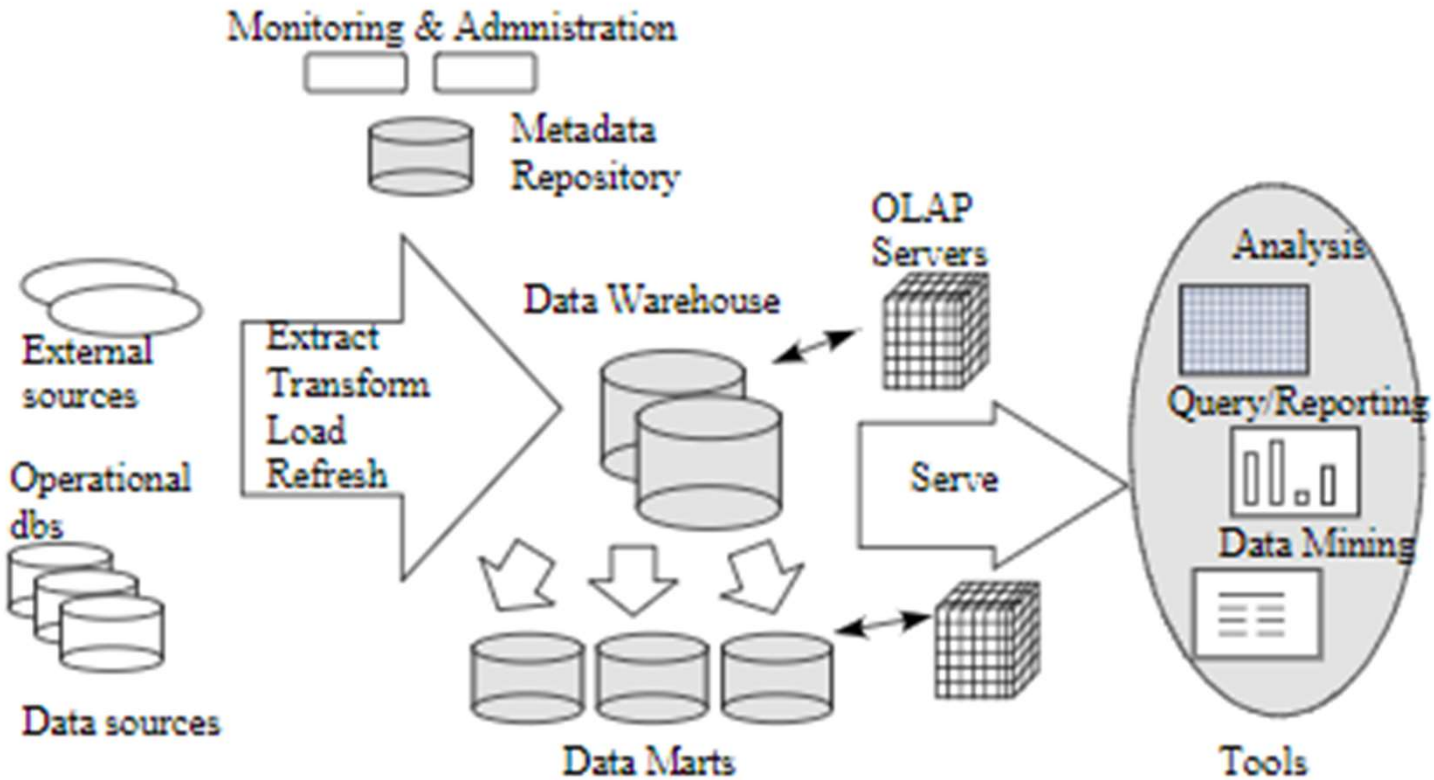


Figure 1. Data Warehousing Architecture

Database Design Methodology

- **Multidimensionality** is core to facilitate complex analyses and visualizations.
- **Star Schema**
  - **fact table** has a pointer to each of the dimensions (acts as a multidimensional coordinate with numerical measures).
  - **dimension tables** store attributes of the dimension
  - Fact table connects to all dimension tables with a multiple join. Each tuple in fact-table consists of a pointer to each of the dimension-tables.

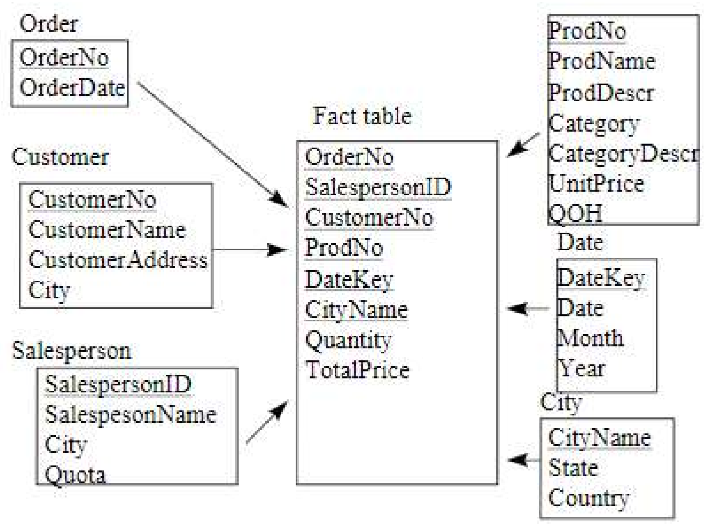


Figure 3. A Star Schema.

Database Design Methodology

- Each dimension is represented by one table in Star Schema.
- Un-normalized hierarchical structure introduces redundancy
  - (Vancouver, BC, Canada, North America)
  - (Victoria, BC, Canada, North America)
- Normalize table dimensions – **Snowflake Schema**

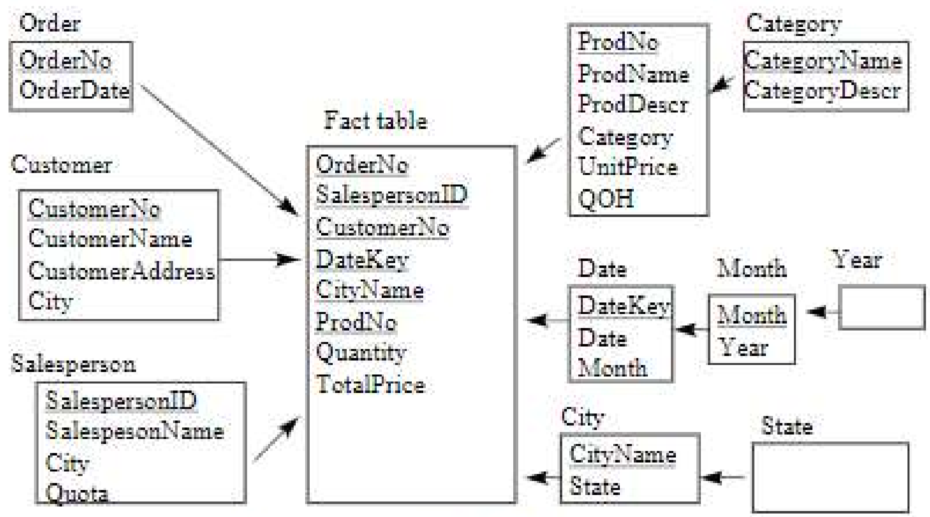


Figure 4. A Snowflake Schema.

Materialized Views

- Many decision support queries require summary data and therefore use aggregates –> use of materialized views.
- Challenges
  - Understanding which views to materialize.
  - Understand how to use such views to answer queries.
  - Efficiently updating materialized views during load and refresh.
- Choice can depend on workload characteristics, update costs, storage requirements.

Discussion (in groups of 3-4)

- Do you think that materialized views are more important for a data warehouse or in a relational database management system?
- Which one do you think it is easier to use materialized views in? Why?

Metadata Requirements

- Metadata management is important, as it reflects upon the use of data within the warehouse.
- **Administrative metadata**
- **Business metadata**
- **Operational metdata**

Discussion (in groups of 4)

- The paper emphasizes the industry and financial motivations behind OLAP and as the last paper discussed, commercial giants often decide what wins.
- To what extent then is database research more influenced by industry rather than academic curiosity and why? How about other computer science sectors such as the one you are in? (From Ryan)

Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals

Slides modified by Michael (original: Jim Cao)
Discussion:

Data Analysis and Applications

- **Looking for anomalies, patterns**
- 4 steps
  - **formulating** a query that extracts relevant data from a database
  - **extracting** the aggregated data from the database into a table
  - **visualizing** the results in a graphical way, and
  - **analyzing** the results and formulating a new query

Dimensionality Reduction

- Data Visualization (and other data analysis tools) do dimensionality reduction by summarizing dimensions that are left out.
- Represent N-dimension data in 2- or 3-D.
- Example:
  - A Car Sale might have information about date of sale, sales company, color of car, model of car, year of car, etc.
  - But we might only want to analyze sales based on a subset of these attributes (e.g. color, model, year).

Relational Representation

- 2D flat files can model an N-dimensional problem as a relation with N-attribute domains.

Table 1: Weather

Time (UCT)	Latitude	Longitude	Altitude (m)	Temp (c)	Pres (mb)
96/6/1:1500	37:58:33N	122:45:28W	102	21	1009
many more rows like the ones above and below					
96/6/7:1500	34:16:18N	27:05:55W	10	23	1024

However, consider…

- Reports commonly aggregate data at a coarse level and then finer levels.
  - Going up the levels is **called rolling-up** the data
  - Going down the levels is called **drilling-down** the data.
- The report on the right shows aggregated data at 3 distinct levels with subtotals.
- In this table, sales are rolled up by using totals and subtotals.
- Data is aggregated by Model, then by Year, then by Color.

Table 3.a: Sales Roll Up by Model by Year by Color

Model	Year	Color	Sales by Model by Year by Color	Sales by Model by Year	Sales by Model
Chevy	1994	black	50		
		white	40		
				90	
	1995	black	85		
		white	115		
				200	
					290

However, consider...

- The report shows data aggregated at three levels, that is, at Model level, Year level, and Color level.
- Data aggregated at each distinct level produces a sub-total.
- Problems
  - This data is not relational – the empty cells (NULL values) cannot form a key.
  - 2N aggregation columns for a roll-up of N elements.

Table 3.a: Sales Roll Up by Model by Year by Color

Model	Year	Color	Sales by Model by Year by Color	Sales by Model by Year	Sales by Model
Chevy	1994	black	50		
		white	40		
				90	
	1995	black	85		
		white	115		
				200	
					290

Date's Alternative

- Is relational, but rejected as the second problem still persists - it implies a large number of domains in resulting tables.

Table 3.b: Sales Roll-Up by Model by Year by Color as recommended by Chris Date [Date1].

Model	Year	Color	Sales	Sales by Model by Year	Sales by Model
Chevy	1994	black	50	90	290
Chevy	1994	white	40	90	290
Chevy	1995	black	85	200	290
Chevy	1995	white	115	200	290

Pivot Table

- Excel pivot table transposes a spreadsheet, aggregating cells based on values within the cells.
- Problem:
  - Pivot creates columns based on subsets of column values – this is a much larger set.
  - If one pivots on two columns with N and M values – pivot table has N x M values – many columns and obtuse column name.

Table 4: An Excel pivot table representation of Table 3 with Ford sales data included.

Sum	Year	Color					
Sales	1994		1994 Total	1995		1995 Total	Grand Total
Model	black	white		black	white		
Chevy	50	40	90	85	115	200	290
Ford	50	10	60	85	75	160	220
Grand Total	100	50	150	170	190	360	510

The ALL Value

- 3-dimensional roll-up with 3 unions.
- Do not extend to have new columns.
- The dummy value “ALL” has been added to fill in the super-aggregation items
  - Avoid exponential growth of columns.

Table 5.a: Sales Summary			
Model	**Year**	**Color**	**Units**
Chevy	1994	black	50
Chevy	1994	white	40
Chevy	1994	ALL	90
Chevy	1995	black	85
Chevy	1995	white	115
Chevy	1995	ALL	200
Chevy	ALL	ALL	290

The ALL Value

- Since prior table was a relation, it could be built with SQL.
- Roll-up is asymmetric (e.g. prior table aggregates by year but not colour)
  - A symmetric aggregation that captures both is called a cross-tabulation (cross tab)
- Expressing roll-up and cross-tab queries with conventional SQL is daunting! Why?
  - A 6-D cross tab requires a 64-way union of 64 different GROUP BY operators to build the underlying representation.
  - Too complex for optimization!

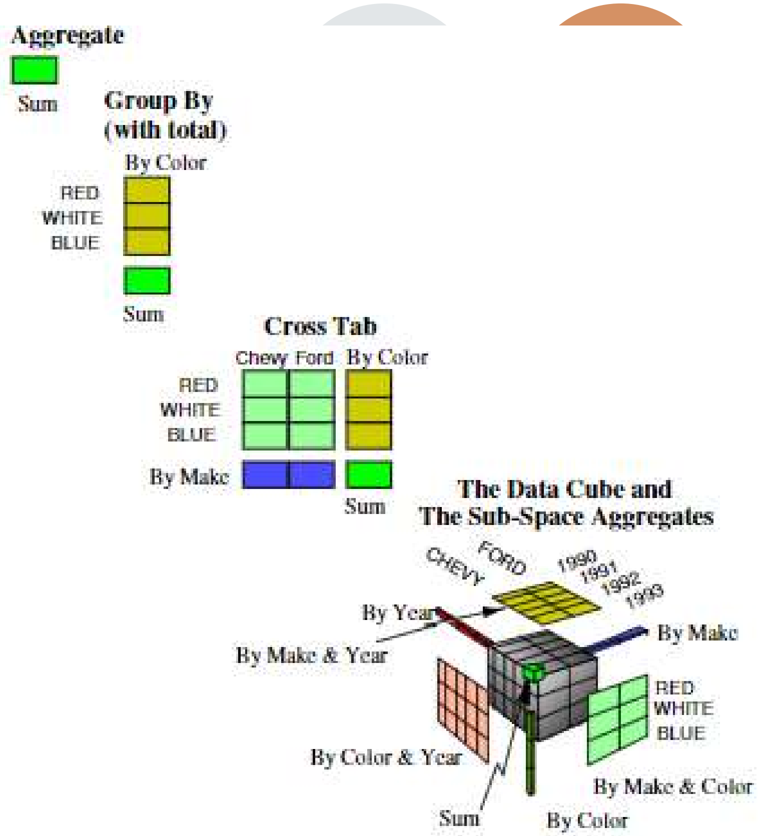
```
SELECT 'ALL', 'ALL', 'ALL', SUM(Sales)
   FROM      Sales
   WHERE     Model = 'Chevy'
UNION
SELECT Model, 'ALL', 'ALL', SUM(Sales)
   FROM      Sales
   WHERE     Model = 'Chevy'
   GROUP BY  Model
UNION
SELECT Model, Year, 'ALL', SUM(Sales)
   FROM      Sales
   WHERE     Model = 'Chevy'
   GROUP BY  Model, Year
UNION
SELECT Model, Year, Color, SUM(Sales)
   FROM      Sales
   WHERE     Model = 'Chevy'
   GROUP BY  Model, Year, Color;
```

Discussion (in groups of 4)

- The authors state that veteran SQL implementers will be terrified of the ALL value – like NULL, it will create many special cases.
- What are some the special cases you can imagine are created by NULL? How about ALL? Do you think ALL is a bigger or lesser concern than NULL?

Data CUBE

- **N-dimensional** generalization of simple aggregate functions
- **N-1 lower-dimensional aggregates** are points, lines, planes, cubes
- Data cube operator builds a table containing all these aggregate values
  - O-D data cube: a point.
  - 1-D data cube: a line & a point.
  - 2-D data cube: a cross tabulation, a plane, two lines, and a point.
  - 3-D data cube: a cube with three intersecting 2D cross tabs



The CUBE operator

SELECT Model, Year, Color, SUM (Sales) AS sales
FROM Sales
WHERE Model in ['Ford', 'Chevy']   AND year BETWEEN 1994 AND 1995
GROUP BY CUBE Model, Year, Color

- CUBE is a relational operator – GROUP BY and ROLL UP are degenerate forms of the operator.
- Creating a data cube requires generating the power set of all aggregation columns.

The CUBE operator

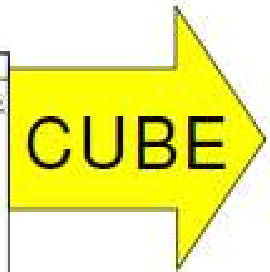
```
SELECT Model, Year, Color, SUM (Sales) AS sales
FROM Sales
WHERE Model in ['Ford', 'Chevy']    AND  year BETWEEN 1994 AND 1995
GROUP BY CUBE Model, Year, Color
```

- Aggregates over all the <select list> attributes in GROUP BY clause.
- UNIONS each super-aggregate of the global cube, substituting ALL for the aggregation columns.
- If there are N attributes in <select list>, there are 2^N – 1 super-aggregate values.
- Super-aggregates are produced through ROLLUP, like a sum or average

```
SELECT Model, Year, Color, SUM(sales) AS Sales
FROM Sales
WHERE Model in ('Ford', 'Chevy')
  AND Year BETWEEN 1990 AND 1992
GROUP BY CUBE Model, Year, Color;
```

SALES

Model	Year	Color	Sales
Chevy	1990	red	5
Chevy	1990	white	87
Chevy	1990	blue	62
Chevy	1991	red	54
Chevy	1991	white	95
Chevy	1991	blue	49
Chevy	1992	red	31
Chevy	1992	white	54
Chevy	1992	blue	71
Ford	1990	red	64
Ford	1990	white	62
Ford	1990	blue	63
Ford	1991	red	52
Ford	1991	white	9
Ford	1991	blue	55
Ford	1992	red	27
Ford	1992	white	62
Ford	1992	blue	39



DATA CUBE

Model	Year	Color	Sales
Chevy	1990	blue	62
Chevy	1990	red	5
Chevy	1990	white	95
Chevy	1990	ALL	154
Chevy	1991	blue	49
Chevy	1991	red	54
Chevy	1991	white	95
Chevy	1991	ALL	198
Chevy	1992	blue	71
Chevy	1992	red	31
Chevy	1992	white	54
Chevy	1992	ALL	156
Chevy	ALL	blue	182
Chevy	ALL	red	90
Chevy	ALL	white	236
Chevy	ALL	ALL	508
Ford	1990	blue	63
Ford	1990	red	64
Ford	1990	white	62
Ford	1990	ALL	189
Ford	1991	blue	55
Ford	1991	red	52
Ford	1991	white	9
Ford	1991	ALL	116
Ford	1992	blue	39
Ford	1992	red	27
Ford	1992	white	62
Ford	1992	ALL	128
Ford	ALL	blue	157
Ford	ALL	red	143
Ford	ALL	white	133
Ford	ALL	ALL	433
ALL	1990	blue	125
ALL	1990	red	69
ALL	1990	white	149
ALL	1990	ALL	343
ALL	1991	blue	106
ALL	1991	red	104
ALL	1991	white	110
ALL	1991	ALL	314
ALL	1992	blue	110
ALL	1992	red	58
ALL	1992	white	116
ALL	1992	ALL	284
ALL	ALL	blue	339
ALL	ALL	red	233
ALL	ALL	white	369
ALL	ALL	ALL	941

Data Cubes - Summary

- The cube operator generalizes and unifies several common and popular concepts: such as aggregates, group by, histograms, roll-ups and drill-downs and cross tabs.

- The cube operator is based on a relational representation of aggregate data using the ALL value to denote the set over which each aggregation is computed.

- The data cube is easy to compute for a wide class of functions
- SQL’s basic set of five aggregate functions needs careful extension to include

Discussion (in pairs)

- The abstract mentions that “many of the features are being added to the SQL standard”.
- Should CUBE be a standard SQL feature? More generally, how do we decide which functionality should be left to an extension and which functionality should be included in the standard? (From Carol)