CS522 Advanced Database Systems
Data Warehouse and OLAP

Chengyu Sun
California State University, Los Angeles

Operational Databases

- Handles day-to-day operations of an organization
- A.K.A. Online Transaction Processing (OLTP) systems
- Characterized by
  - Content – detailed and current
  - Users – client and employees
  - Access pattern – short, atomic, r/w transactions
  - Design – ER, normalized

Data Warehouse

- "A data warehouse is a subject-oriented, integrated, time-variant, and nonvolatile collection of data in support of management's decision making process" – W. H. Inmon

Characteristics of Data Warehouse

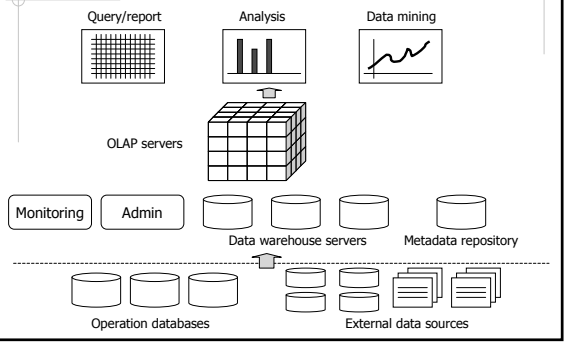
- Subject-oriented
- Integrated
- Time-variant
- Nonvolatile

- *To support decision making*

Data Warehouse vs. Operational Database

	Operational Database	Data Warehouse
Content	Detailed and current	??
Users	Clients and employees	??
Access Patterns	short, atomic, r/w transactions	??
Design	ER, normalized	??

Data Warehouse Architecture



Why The Multidimensional Model

- Decision support applications are dominated by queries involved aggregations and group-bys
- And such queries often can't be expressed or executed efficiently by OLTP databases

Standard SQL Aggregation Functions

- Operate on multiple rows and return a single result
  - `sum`
  - `avg`
  - `count`
  - `max` and `min`

GROUP BY

```
select category, count(id)
     from products
     group by category;
```

products

id	category	description	price
1	CPU	Intel Core 2 Duo	$200.00
2	CPU	Intel Pentium D	$98.99
3	CPU	AMD Athlon 64	$74.49
4	CPU	AMD Athlon 64x2	$115.98
5	HD	Seagate 320G	$77.49
6	HD	Maxtor 250G	$60.89

Understanding GROUP BY …

- Without aggregation/GROUP BY

select category, id from products;

category	id
CPU	1
CPU	2
CPU	3
CPU	4
HD	5
HD	6

… Understanding GROUP BY

- With aggregation/GROUP BY

select category, count(id) from products group by category;

Grouping attribute

category	id
CPU	1
CPU	2
CPU	3
CPU	4
HD	5
HD	6

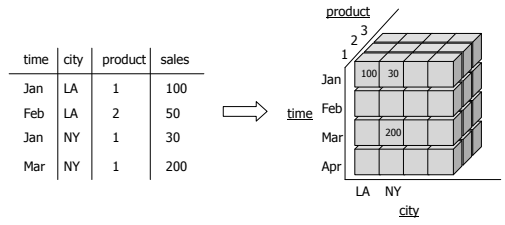
Aggregation attribute

count(id) = 4

count(id) = 2

The Multidimensional Model

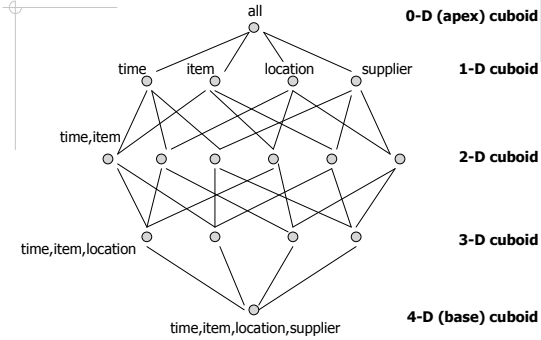
time	city	product	sales
Jan	LA	1	100
Feb	LA	2	50
Jan	NY	1	30
Mar	NY	1	200



Data Cube

- Dimensions
  - Time, product, city …
- Facts
  - Sales, units sold, expenses …

Data Cube as a Lattice of Cuboids



Data

time	item	location	supplier	sales
Jan	TV	LA	Sony	100
Jan	Laptop	NY	HP	120
Feb	TV	LA	Sony	110
Feb	TV	NV	Sony	200
Feb	Laptop	LA	HP	130
March	TV	LA	Sony	100

4-D Cuboid

- (time, item, location, supplier)
  - *Number of cells??*
  - *Cells with non-zero values??*

3-D Cuboids

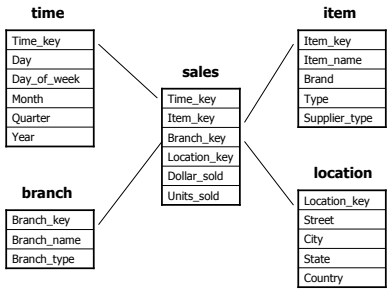
- (time,item,location)
- (time,item,supplier)
- (time,location,supplier)
- (item,location,supplier)

Number of cells in each cuboid??
Cells with non-zero values??

Observations about Data Cubes

- Given a n-dimensional data cube, with each dimension having `m` values
  - *Number of cuboids??*
  - *Number of cells??*

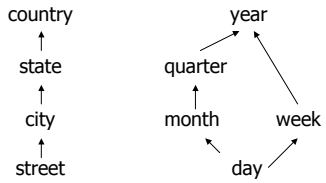
Star Schema …



… Star Schema

- One Fact Table
  - E.g. `sales`
- One Dimension Table per dimension
  - E.g. `time`, `item`, `branch`, and `location`
  - Concept Hierarchy
  - Redundancy in dimension tables

Concept Hierarchies



- Total order: `street < city < state < country`
- Partial order: `day < {month<quarter, week} < year`

Other Schemas for Multidimensional Databases

- Snowflake schema
  - Some dimensions are normalized
  - *Normalize the `location` dimension??*
- Fact Constellation schema
  - Dimension tables are shared by more than one fact tables

OLAP Storage Strategies

- Relational OLAP (ROLAP)
- Multidimensional OLAP (MOLAP)
- Hybrid OLAP (HOLAP)

A ROLAP Data Store

- Summary fact tables

RID	Item	◦◦◦	Day	Month	Quarter	Year	Sales
1001	TV		15	10	Q4	2003	250
1002	TV		23	10	Q4	2003	175
	◦◦◦			◦◦◦		◦◦◦	
5001	TV		all	10	Q4	2003	45,786

Aggregation Functions (Measures)

- Distributive
  - `sum`, `count`, `min`, `max`
- Algebraic
  - `avg = sum / count`
- Holistic
  - `median`

Estimate Median

23,5,12,20,2,11,14,16,18,19,8,1,21,25

1,2,5,8,11,12,14,16,18,19,20,21,23,25

$$median = L_{median} + \frac{N/2 - \sum_l freq}{freq_{median}} \bullet width_{median}$$

More About Aggregation Functions

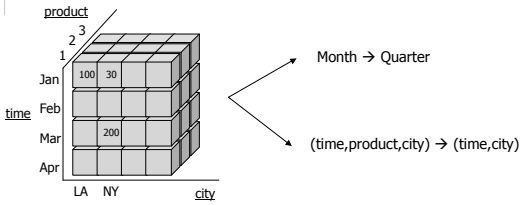
- Variance: $\frac{1}{N}\sum_{i=1}^{N}(x - \overline{x_i})^2$
- Incremental maintenance
  - E.g. `max`, `min`, `average`, `median`

OLAP Operations

- Roll-up
- Drill-down
- Slice and dice
- Pivot (rotate)

Roll-up

- Aggregation on a data cube by
  - Going up a concept hierarchy, or
  - Reducing dimension(s)



Drill-down

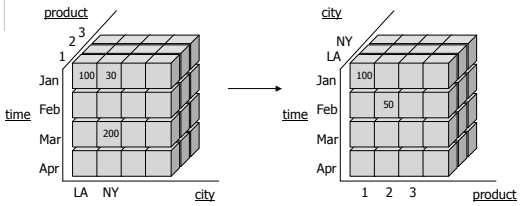
- Reverse of roll-up
  - Going down a concept hierarchy, or
  - Adding dimensions

Slice and Dice

- Slice: selection on one dimension
- Dice: selection on more than one dimensions
  - E.g. `(city = "LA") and (month = "Jan" or "Feb")`

Pivot (Rotate)

- Rotate the data axes to provide an alternative presentation of the data



Perform OLAP Operations Efficiently

- Indexing
- Cube pre-computation

Bitmap Indexing …

rid	item	city	month	sales
1001	TV	LA	Jan	100
1002	PC	LA	Jan	200
1003	PC	NY	Jan	150
1004	PC	NY	Feb	100
1005	Phone	NY	Jan	175
1006	TV	NY	Feb	200
1007	Phone	LA	Jan	300
1008	Phone	LA	Feb	120

Item: { TV, PC, Phone }

City: { LA, NY }

… Bitmap Indexing

Bitmap Index on Item: *Bitmap Index on City ??*

```
1   0   0
0   1   0
0   1   0
0   1   0
0   0   1
1   0   0
0   0   1
0   0   1
```

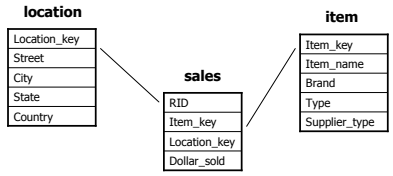
TV PC Phone

Using Bitmap Indexing

- List total sales in LA by item

```
select sum(sales), item
     from sales_table
     where city = 'LA'
     group by item;
```

Join Indexing …



… Join Indexing …

location

Location_key	Street	City	State	Country
1	123 Main St.	LA	CA	USA
2	456 Wall St.	NY	NY	USA
3	789 State St.	LA	CA	USA

item

Item_key	Item_name	Brand	Type
1	Bravia 42in	Sony	TV
2	Bravia 46in	Sony	TV
3	Pavilion A100	HP	PC
4	Pavilion A200	HP	PC
5	iPhone	Apple	Phone

Sales

RID	Item_key	Loc_key	Amount
1001	1	1	100
1002	3	3	120
1003	3	2	150
1004	4	2	110
1005	5	2	130
1006	2	2	170
1007	5	1	200
1008	5	3	100

… Join Indexing

Sales & Item type

rid	item_type
1001	TV
1006	TV
1002	PC
1003	PC
1004	PC
1005	Phone
1007	Phone
1008	Phone

Sales & Item type & City

rid	item	city
1001	TV	LA
1002	PC	LA
1007	Phone	LA
1008	Phone	LA
1006	TV	NY
1003	PC	NY
1004	PC	NY
1005	Phone	NY

Using Pre-computed Cuboids …

- Consider data cube `sales_cube [time,item,location] : sum(sales)`
  - Time: day < month < quarter < year
  - Item: item_name < brand < type
  - Location: street < city < state < country

… Using Pre-computed Cuboids

- Pre-computed cuboids
  - Cuboid 1: {year, item_name, city}
  - Cuboid 2: {year, brand, country}
  - Cuboid 3: {year, brand, state}
  - Cuboid 4: {item_name, state} where year = 2004
- Query
  - {brand, state} where year = 2004 ??

Summary

- Architecture
- Data
  - Multidimensional data model – Data Cube
  - Logical and physical data organization
- Operations
  - Aggregation functions
  - OLAP operations
  - Efficient execution
- *Readings: Chapter 3 of textbook*