

CS522 Advanced Database Systems

Introduction to 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 – clients, developers, DBA
 - Access pattern – short, atomic, r/w transactions
 - Design – ER, normalized

The Need for Data Warehouse

...

- ◆ Decision support applications, e.g.
 - Show the sales number by month, by day, region, and/or by product
- ◆ Reporting and analysis applications, e.g.
 - Web site analytics
 - Online ad tracking

... The Need for Data Warehouse

- ◆ These applications are dominated by queries involving 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

More About Aggregation Functions

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

Distributive Aggregation

	Count	Sum	Min	Max
[5,6,2,8,1,9]	6			
[11,12,14,16,18]	5			
[23,20]	2			
All	??			

Holistic Aggregation

	Median
[5,6,2,8,1,9]	??
[11,12,14,16,18]	14
[23,20]	??
All	??

Estimate Median ...

	Count	Min	Max
[5,6,2,8,1,9]	6	1	9
[11,12,14,16,18]	5	11	18
[23,20]	2	20	23
All	13	1	23

... Estimate Median

$$median = Min_m + \frac{N/2 - \sum_l Count}{Count_m} \cdot (Max_m - Min_m)$$

N: total count
 m: the median interval
 l: the intervals lower than the median interval

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

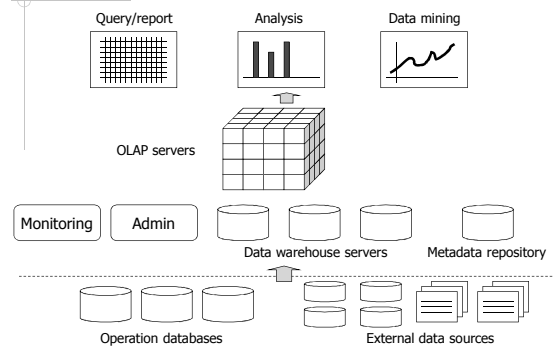
... Understanding GROUP BY

◆ With aggregation/GROUP BY

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

Grouping attribute	category	id	Aggregation attribute
}	CPU	1	count(id) = 4
	CPU	2	
	CPU	3	
	CPU	4	
}	HD	5	count(id) = 2
	HD	6	

Data Warehouse Architecture



Data Warehouse

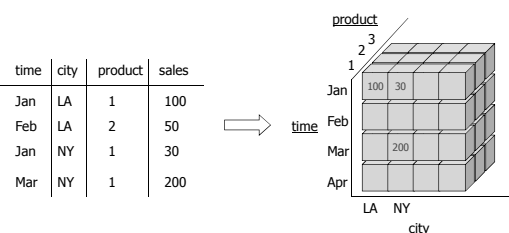
◆ "A data warehouse is a *subject-oriented, integrated, time-variant, and nonvolatile* collection of data in support of management's *decision making process*" – W. H. Inmon

◆ An Online Analytical Processing (OLAP) system

Data Warehouse vs. Operational Database

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

The Multidimensional Model



Terminology

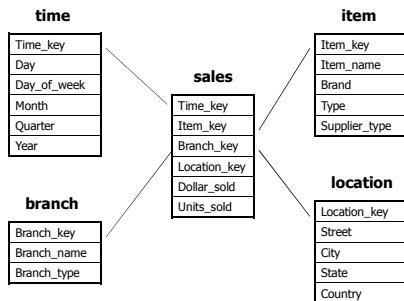
◆ Dimensions

- Time, product, city ...

◆ Facts

- Sales, units sold, expenses ...

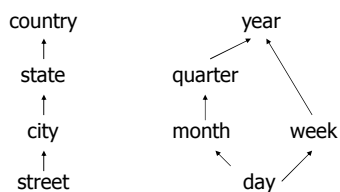
Star Schema ...



... 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
 - ◆ Why redundancy is not a problem here??

Concept Hierarchies



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

Other Schemas for Multidimensional Databases

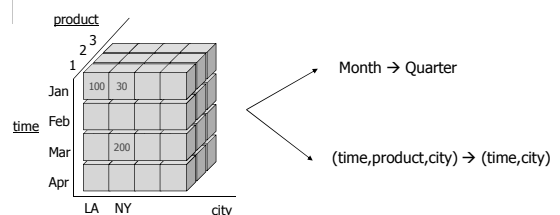
- ◆ Snowflake schema
 - Some dimensions are normalized
 - Normalize the location dimension??
- ◆ Fact Constellation schema
 - Dimension tables are shared by more than one fact tables

OLAP 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

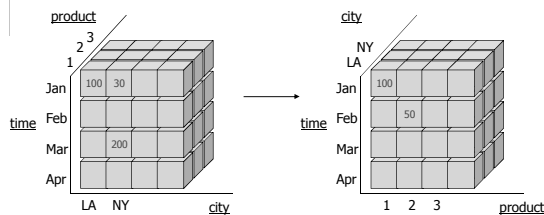
- ◆ Reverse of roll-up
 - Going down a concept hierarchy, or
 - Adding dimensions

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 ??*

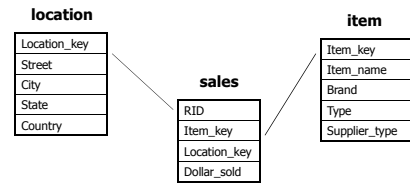
1	0	0
0	1	0
0	1	0
0	1	0
0	0	1
1	0	0
0	0	1
0	0	1
TV	PC	Phone

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					Sales			
Location_key	Street	City	State	Country	RID	Item_key	Loc_key	Amount
1	123 Main St.	LA	CA	USA	1001	1	1	100
2	456 Wall St.	NY	NY	USA	1002	3	3	120
3	789 State St.	LA	CA	USA	1003	3	2	150
item					1004	4	2	110
Item_key	Item_name	Brand	Type	1005	5	2	130	
1	Bravia 42in	Sony	TV	1006	2	2	170	
2	Bravia 46in	Sony	TV	1007	5	1	200	
3	Pavilion A100	HP	PC	1008	5	3	100	
4	Pavilion A200	HP	PC					
5	iPhone	Apple	Phone					

... Join Indexing

Sales & Item type		Sales & Item type & City		
rid	item_type	rid	item	city
1001	TV	1001	TV	LA
1006	TV	1002	PC	LA
1002	PC	1007	Phone	LA
1003	PC	1008	Phone	LA
1004	PC	1006	TV	NY
1005	Phone	1003	PC	NY
1007	Phone	1004	PC	NY
1008	Phone	1005	Phone	NY

Readings

- Textbook Chapter 3