

数据库系统概论新技术篇

数据仓库与联机分析处理技术(2)

陈红

中国人民大学信息学院

数据仓库与OLAP技术

- ❖ 从数据库到数据仓库
- ❖ 数据仓库的特征与体系结构
- ❖ 数据仓库与OLAP的关键技术
- ❖ 新的研究方向



数据仓库与OLAP的关键技术

- ❖ 多维数据模型
- ❖ CUBE计算技术
- ❖ 实体化视图技术
- ❖ 精简数据方体技术
- ❖ 索引技术



数据仓库与OLAP的关键技术

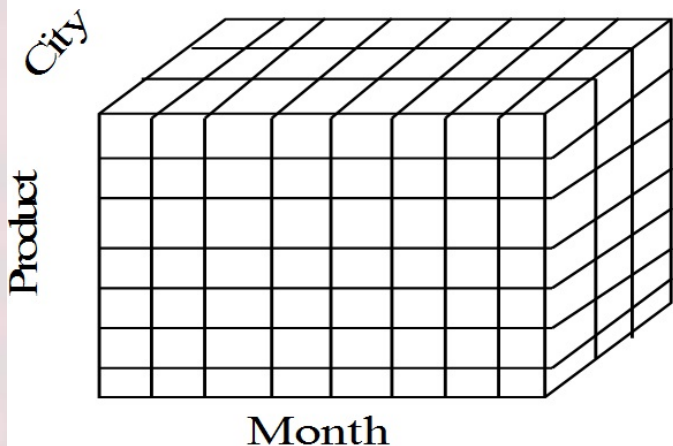
- ❖ 多维数据模型
- ❖ CUBE计算技术
- ❖ 实体化视图技术
- ❖ 精简数据方体技术
- ❖ 索引技术



多维数据模型

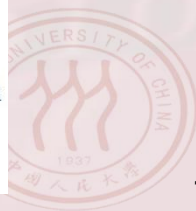
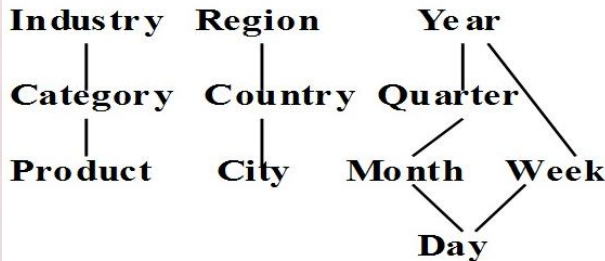
- ❖ 数据仓库和OLAP服务器基于多维数据模型。
- ❖ 多维数据模型将数据看作数据方体(Data Cube), 它通过维(dimension)和度量(measure)进行定义。
- ❖ 维可以有层次。

Month, Product, City上的数据立方体



多维数据模型实际上是把度量看成是由维组成的多维空间上的值

维Product, Location, Time的层次组织



多维数据模型的实现

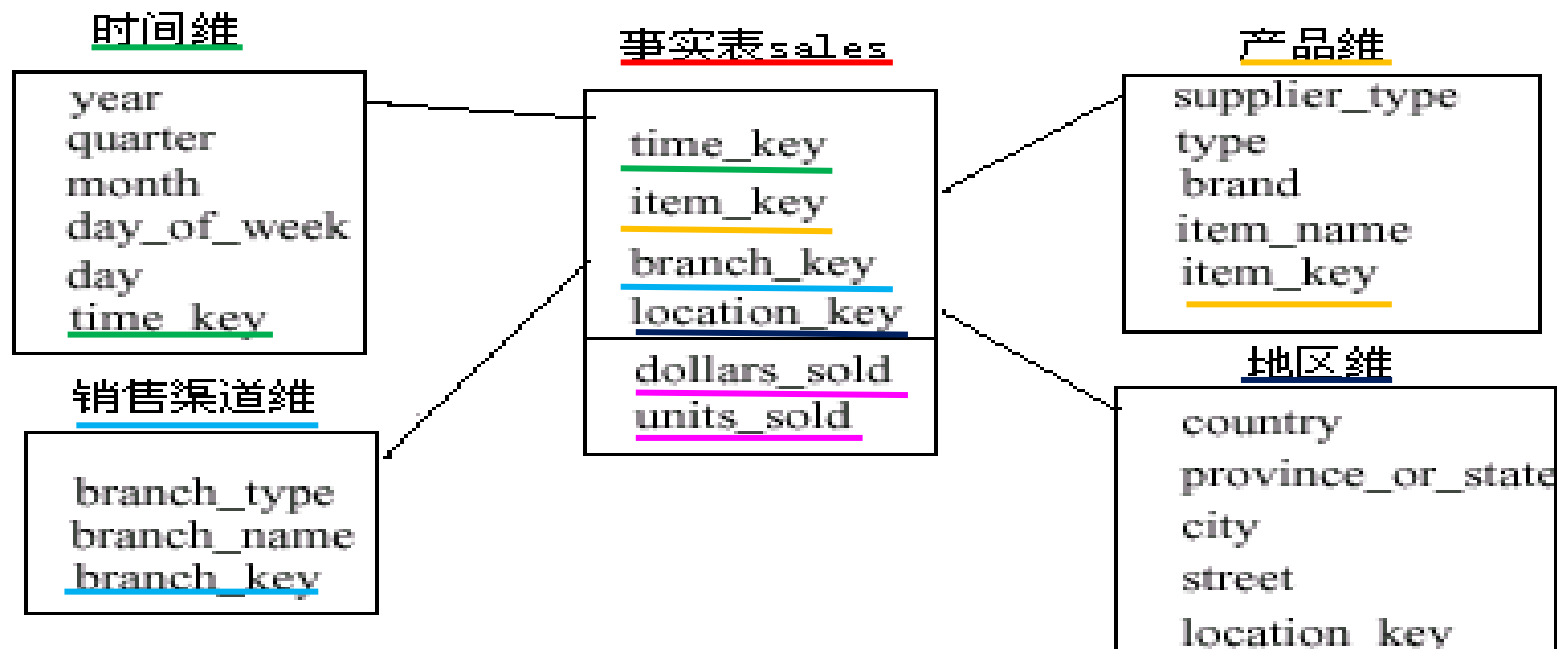
❖ 基于关系数据库

- 星型模式
- 雪片模式
- 事实群模式

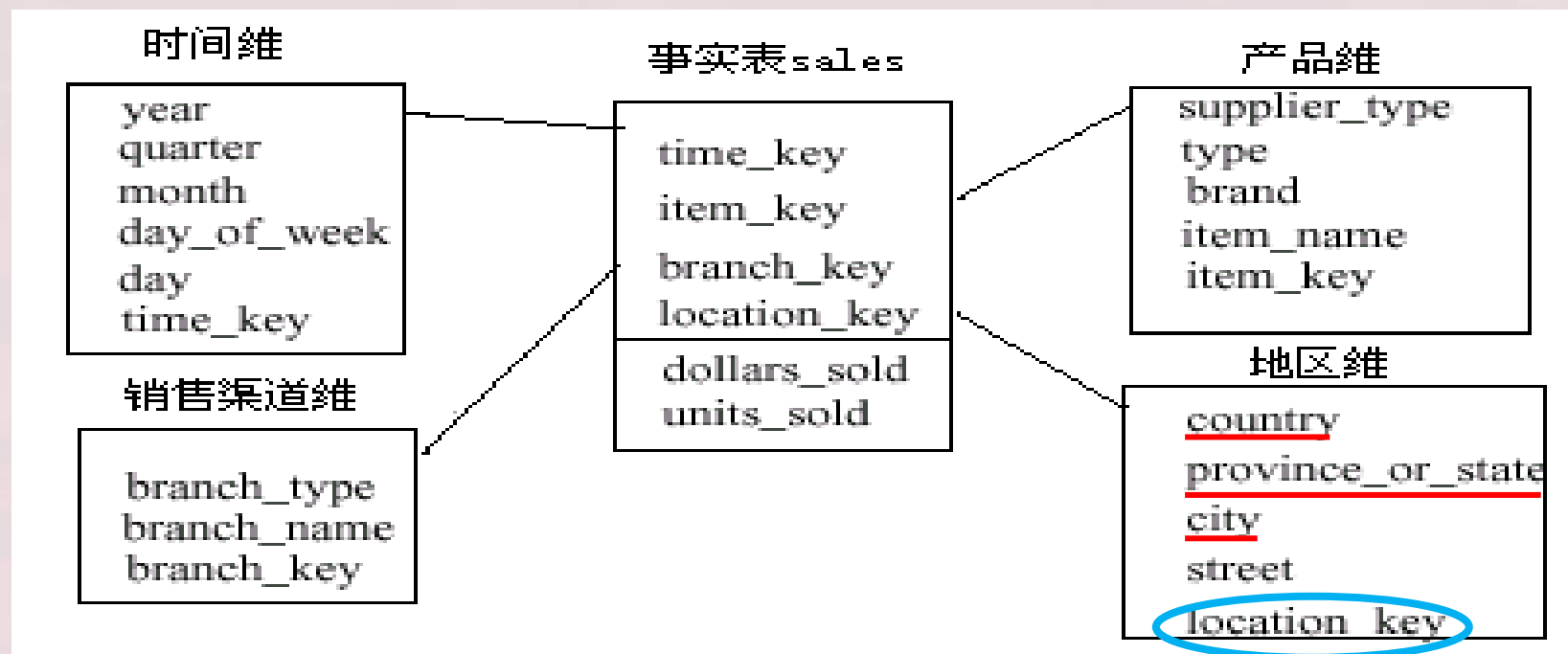
❖ 基于多维数组



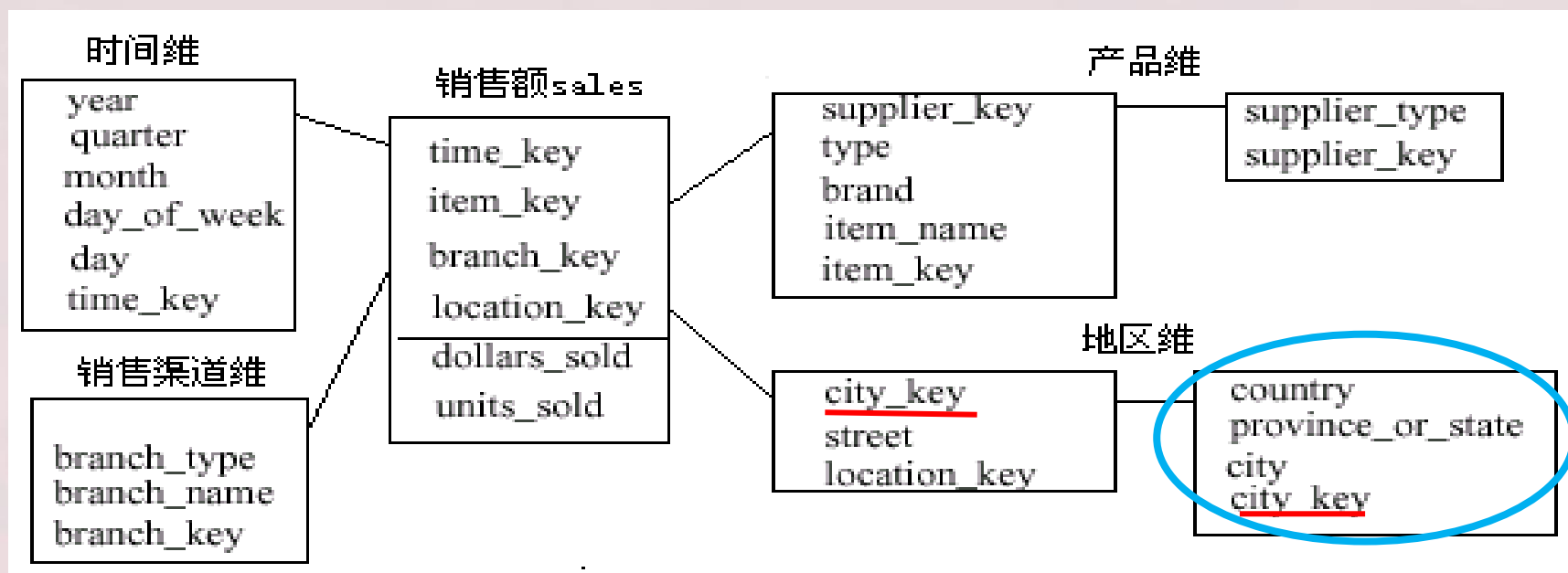
星型模式



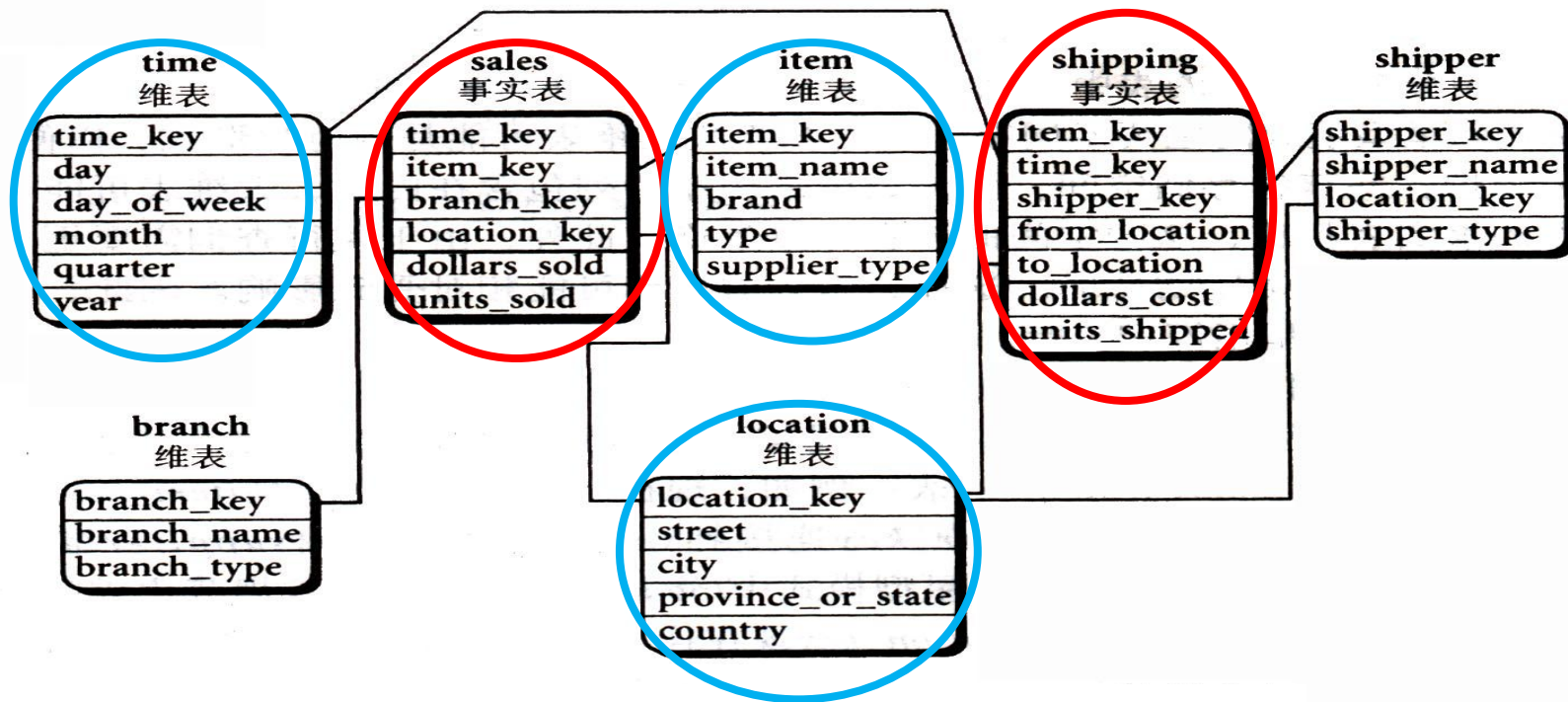
星型模式



雪片模式



事实群模式



多维数组存储

❖ 存储方法

- 多维数组只存储数据方体的度量值，维值由数组的下标隐式给出。

Fact table view:

sale	prodId	storeId	amt
	p1	c1	12
	p2	c1	11
	p1	c3	50
	p2	c2	8



Multi-dimensional cube:

	c1	c2	c3
p1	12	8	50
p2	11	8	

dimensions = 2

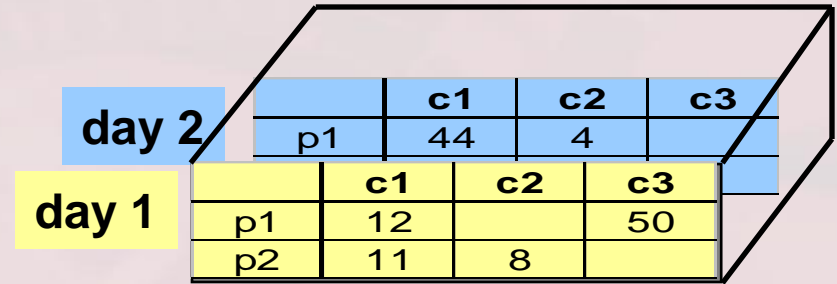


3-D Cube

Fact table view:

sale	prodId	storeId	date	amt
	p1	c1	1	12
	p2	c1	1	11
	p1	c3	1	50
	p2	c2	1	8
	p1	c1	2	44
	p1	c2	2	4

Multi-dimensional cube:



dimensions = 3



两类方法的比较

❖ 关系存储

- 👍 适应性、伸缩性和扩展性好
- 👍 不存在稀疏数据问题
- 👎 访问速度不如多维数组快

❖ 多维数组存储

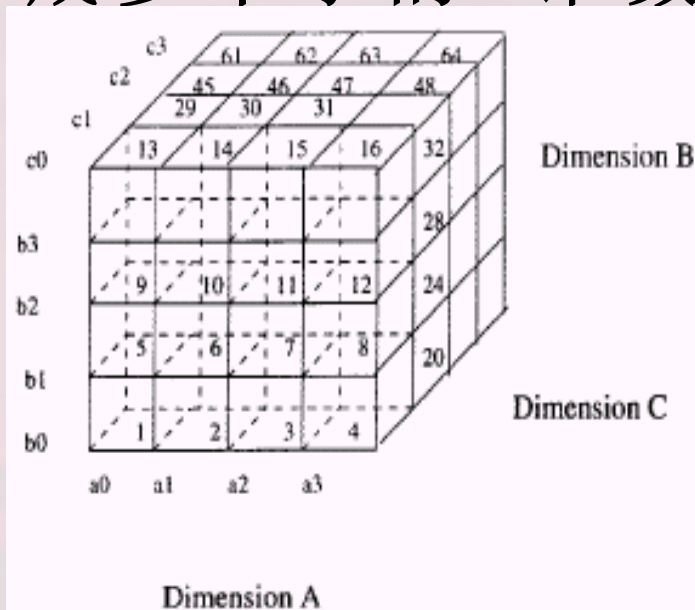
- 👍 存储效率高
- 👍 访问速度快
- 👎 不同维的访问效率差别很大
- 👎 在数据稀疏的情况下，由于大量无效值的存在，存储效率下降



多维数组存储

❖ 解决不同维的访问效率差别大的问题

■ 将一个 n 维数组分成多个小的 n 维数据块 (chunk) 的方法。



多维数组存储


❖ 解决不同维的访问效率差别大

■ 将一个 n 维数组分成多个小 (**chunk**) 的方法。

❖ 解决数据稀疏造成空间浪费的

■ 采用数据压缩技术，如头文件压缩方法等。

A	B	M
0	0	X
1	0	X
2	0	8
3	0	X
4	0	X
0	1	X
1	1	9
2	1	X
3	1	X
4	1	X



2
8
6
9

多维分析操作

- ❖ 建立在关系聚集操作上的一组复合操作
- ❖ 基本的分析是求聚集函数(**aggregation**)



多维分析的基础：聚集

- 例：求第一天的销售总额

In SQL: **SELECT sum(amt) FROM SALE**
WHERE date = 1

sale	prodId	storeId	date	amt
	p1	c1	1	12
	p2	c1	1	11
	p1	c3	1	50
	p2	c2	1	8
	p1	c1	2	44
	p1	c2	2	4



81



多维分析的基础：聚集

- 例：按照每天求销售总额

In SQL: **SELECT date, sum(amt) FROM SALE**
GROUP BY date

sale	prodId	storeId	date	amt
	p1	c1	1	12
	p2	c1	1	11
	p1	c3	1	50
	p2	c2	1	8
	p1	c1	2	44
	p1	c2	2	4



ans	date	sum
	1	81
	2	48



多维分析的基础：聚集

- 例：按照每天和产品求销售总额

In SQL:

```
SELECT prodlid, date, sum(amt) FROM SALE  
GROUP BY prodlid, date
```



聚集函数

❖ 分布型

- 可以分布计算的聚集函数。
- 例如: **sum()**, **count()**, **max()**, **min()**

❖ 代数型

- 可以由一个具有M个参数的代数函数计算得到, 其中每个参数可以用一个分布型聚集函数得到。
- 例如: **AVG()**

❖ 整体型

- 描述它的子聚集所需的存储没有一个常数界。
- 例如: **median()**; **rank()**



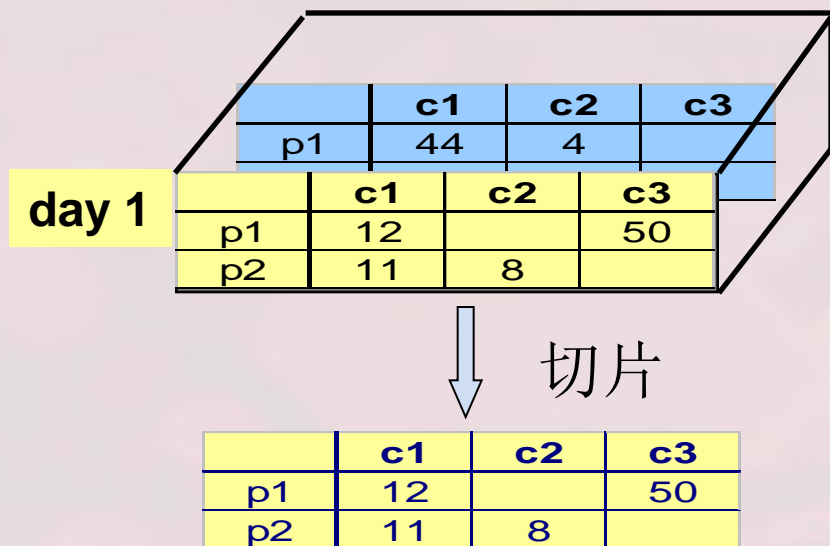
多维分析操作

- ❖ 切片(slice) & 切块(dice)
- ❖ 上卷(roll-up), 下钻(drill down)
- ❖ 旋转(pivoting)



切片/切块操作

❖ 实质上对应于where/having 子句



钻取操作

- 钻取操作就是在不同粒度表之间的切换

sale	prodId	storeId	date	amt
	p1	c1	1	12
	p2	c1	1	11
	p1	c3	1	50
	p2	c2	1	8
	p1	c1	2	44
	p1	c2	2	4

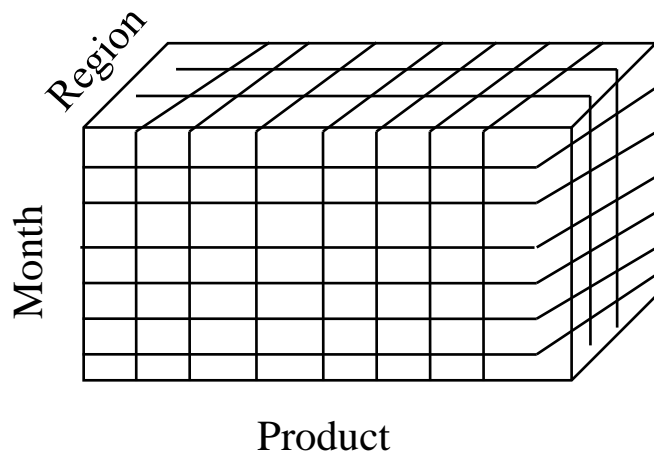
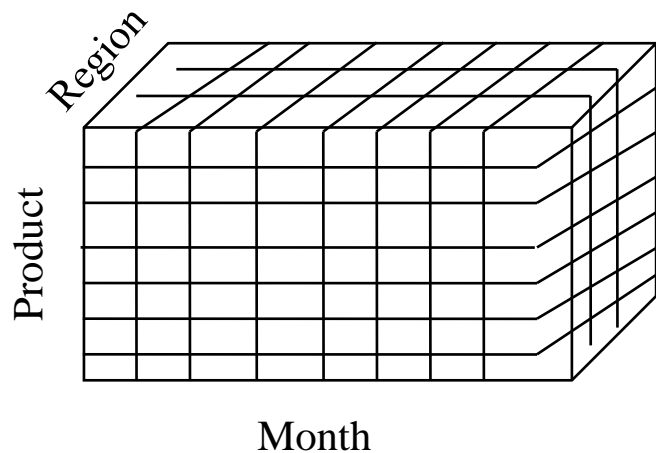
rollup
drill-down

sale	prodId	date	amt
	p1	1	62
	p2	1	19
	p1	2	48



旋转操作

- 旋转操作就是转动观察数据的视角，提供数据的另一种展现方式。



OLAP服务器

- ❖ 多维数据存储
- ❖ 多维数据操作



OLAP服务器基本实现

❖ ROLAP:

Relational On-Line Analytical Processing

❖ MOLAP:

Multi-Dimensional On-Line Analytical Processing

❖ HOLAP

Hybrid On-Line Analytical Processing

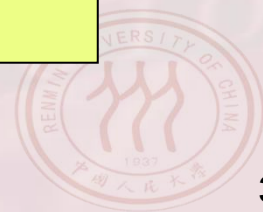


ROLAP Server

❖ ROLAP: Relational On-Line Analytical Processing

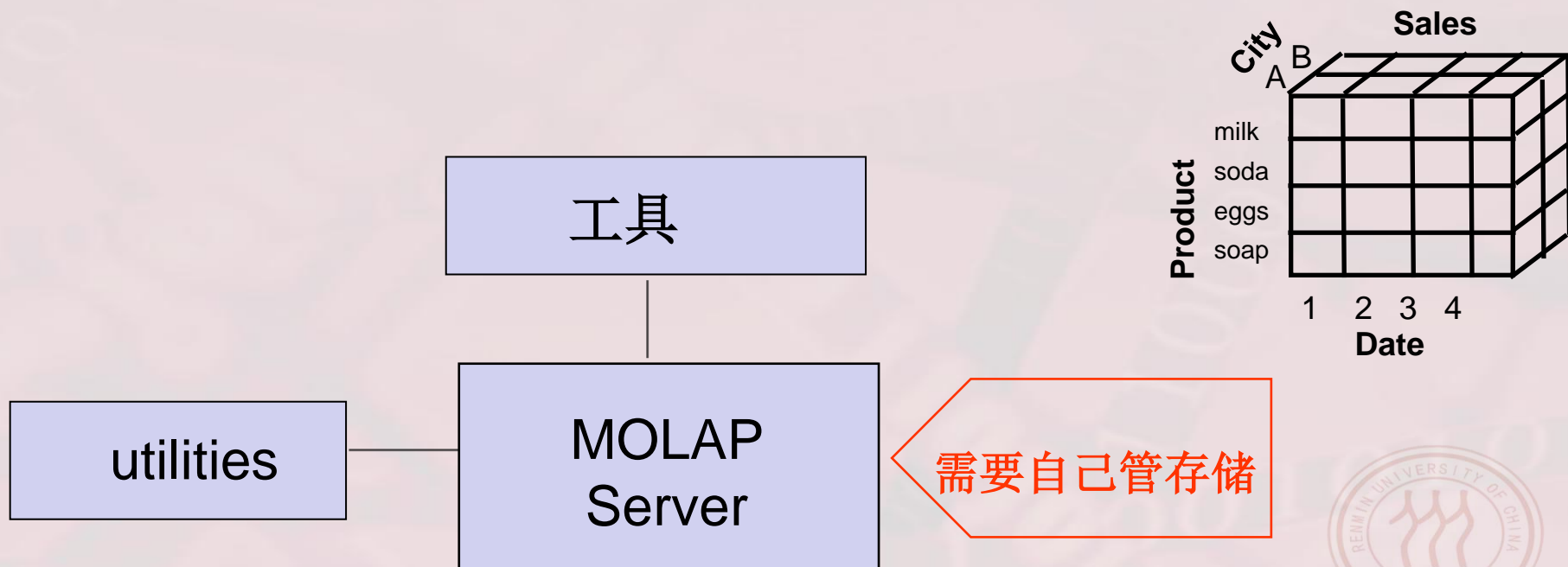
❖ ROLAP数据存取过程

sale	prodId	date	sum
	p1	1	62
	p2	1	19
	p1	2	48



MOLAP Server

❖ MOLAP: Multi-Dimensional On-Line Analytical Processing



HOLAP结构

❖ **HOLAP: Hybrid On-Line Analytical Processing**

❖ **HOLAP将ROLAP和MOLAP结合起来**

例如，将细节数据存在关系数据库中，而将综合数据存在**MOLAP**服务器中

既利用了**ROLAP**可扩展性好的优点，也利用**MOLAP**计算速度快的优点。



小结

❖ 多维数据模型的概念

❖ 多维数据模型的实现方法

- 基于关系数据库
- 基于多维数组

❖ 多维分析操作

- 切片& 切块
- 上卷, 下钻
- 旋转

❖ OLAP服务器基本实现

- ROLAP
- MOLAP
- HOLAP



