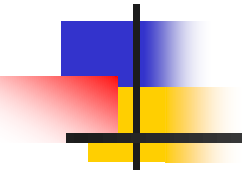
第17章数据仓库与联机分析处理



- 数据仓库
- 联机分析处理技术
- 数据挖掘

实例

在实际应用中：

- 某个客户：某某酒吧销售某某啤酒的价格是多少？操作型数据
- 某个投资商：每个酒吧在过去三个月里销售所有啤酒的平均价格是多少？ 分析型数据

操作型数据 OLTP

传统数据库：

- 细节的
- 在存取的时间时是正确的
- 可更新
- 性能要求高
- 事务驱动
- 面向应用
- 一次操作的数据量小
- 支持日常操作
- 。。。

分析型数据 OLAP

- 综合的，提炼过的
- 代表过去数据
- 不经常更新
- 性能要求宽松
- 分析驱动
- 面向分析
- 一次操作数据量大
- 支持管理决策

OLTP 实例

- 简单的，经常被查询到的，涉及数量不多的元组。
- 例如：某某酒吧销售某某啤酒的价格是多少？

OLAP 实例

- 复杂的查询，涉及大量的数据，可能需要运行几个小时的查询。
- 实例：过去一年某条街上酒吧销售的总量是多少？
- 查询不一定基于当前的数据库信息，可以基于前一个月的数据库信息。

数据仓库

目的：构建新的分析处理环境而出现的一种数据存贮和组织技术。

方法：

- 数据集成：

1. 拷贝很多数据源到一个数据仓库。
2. 不时更新数据仓库中的数据。
3. 主要用于数据的分析。

数据仓库的特点

- 数据是面向主题的
- 数据是集成的
- 数据是不可实时更新的
- 数据是随时间变化的

数据仓库的数据组织

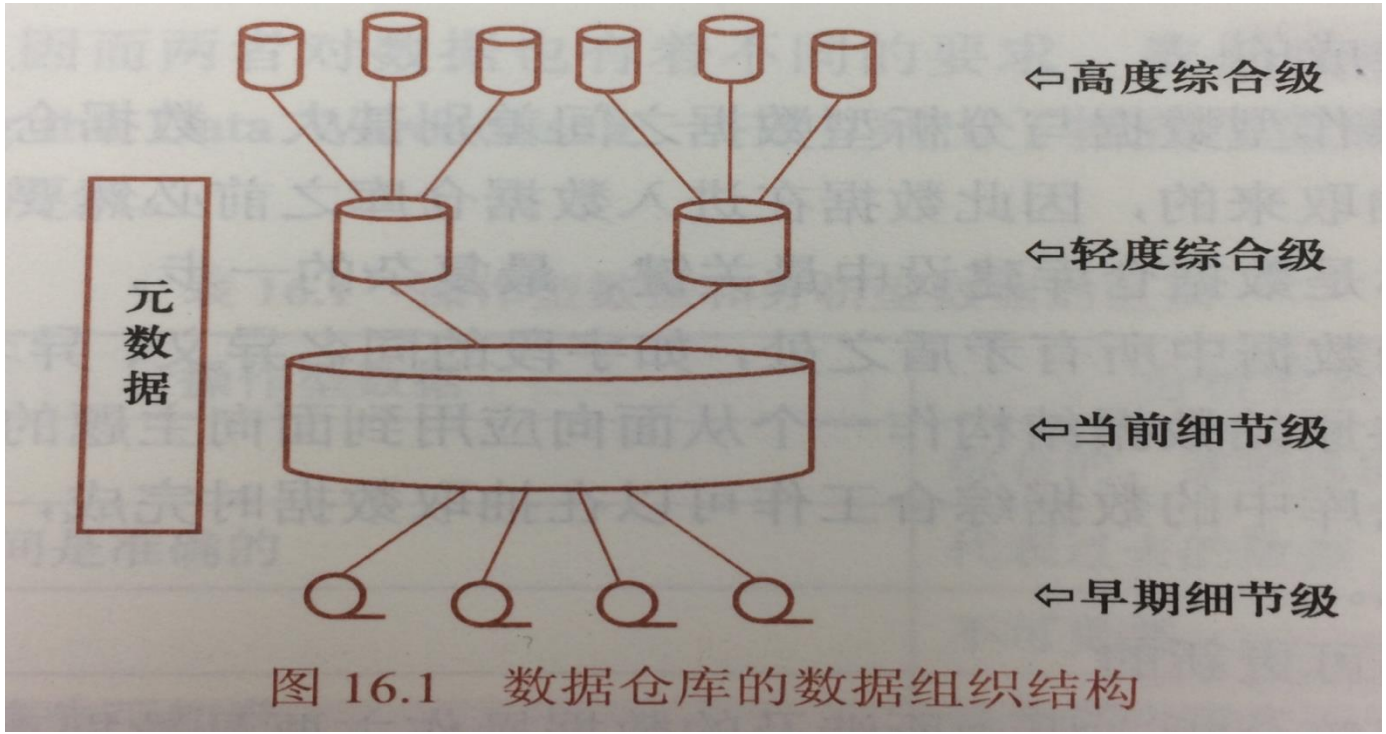


图 16.1　数据仓库的数据组织结构

数据仓库系统的体系结构

- 数据仓库的后台工具
- 数据仓库服务器
- OLAP服务器
- 前台工具

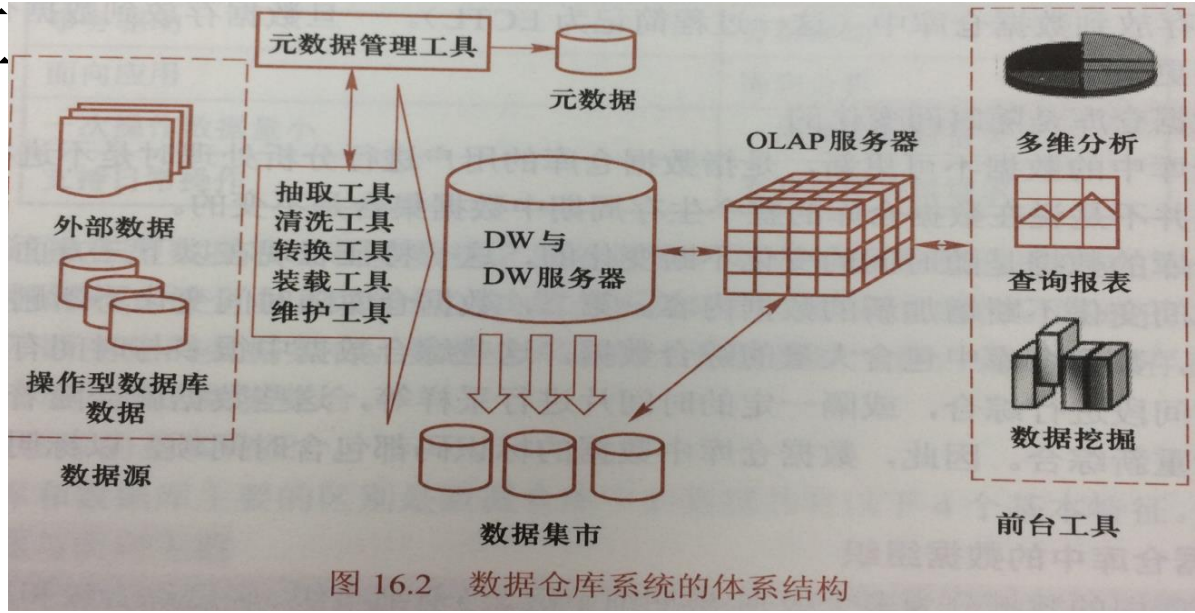


图 16.2　数据仓库系统的体系结构

应用场景

- 分店进行 OLTP.
- 分店的数据晚上拷贝到中央数据仓库
- 分析师对数据仓库进行 OLAP.

数据仓库数据模型：Star Schemas 星型模式

- 星型模式是最常用的数据仓库模型:

1.事实表Fact table：非常大的，带有各个维度的一个表。“insert-only.”

2. 维表Dimension tables：小的，关于各个实体详细的，静态的信息表。

Example: Star Schema

- 假设我们要基于酒吧数据库管理系统，分析每个酒吧，每一种啤酒，哪些客户每天的销售情况：
- 我们要建立的事实表如下：

Sales(bar, beer, drinker, day, time, price)

Example -- Continued

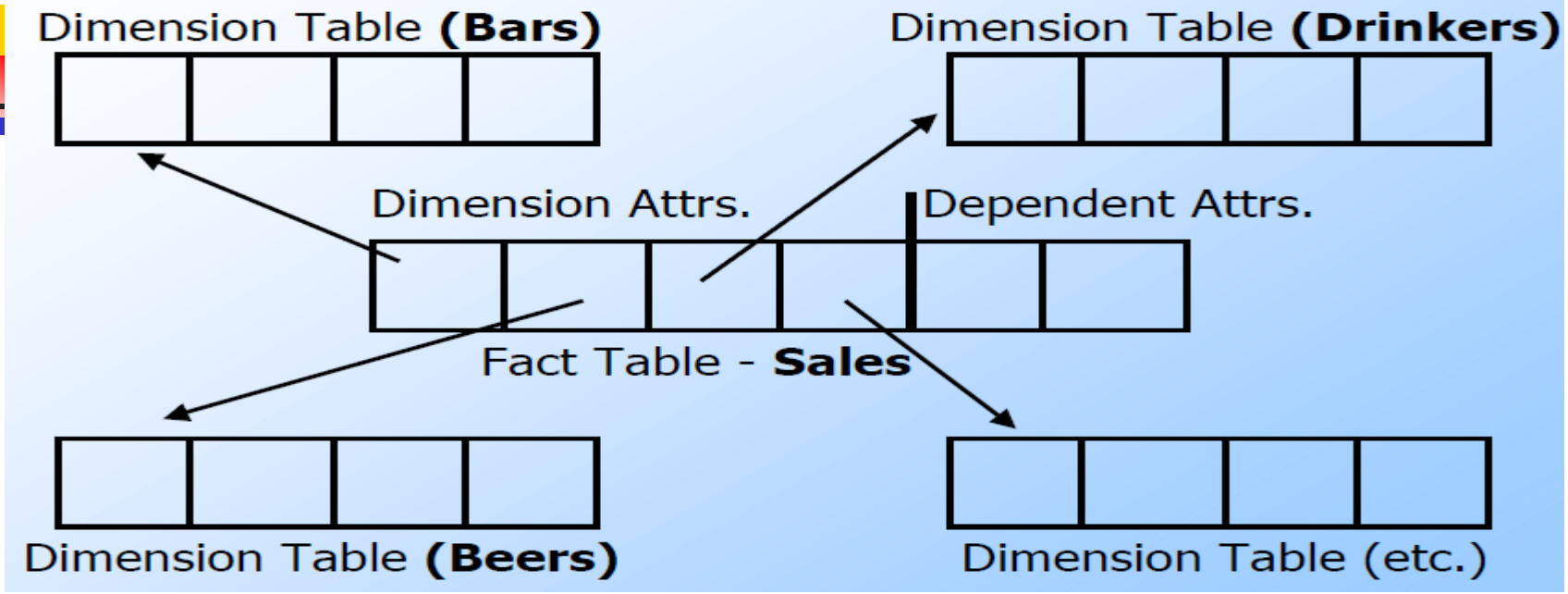
- 维表 如下：

Bars(bar, addr, license)

Beers(beer, manf)

Drinkers(drinker, addr, phone)

Visualization – Star Schema



时间维度

Days(day,week,month,year)

两种属性：维度属性和度量属性

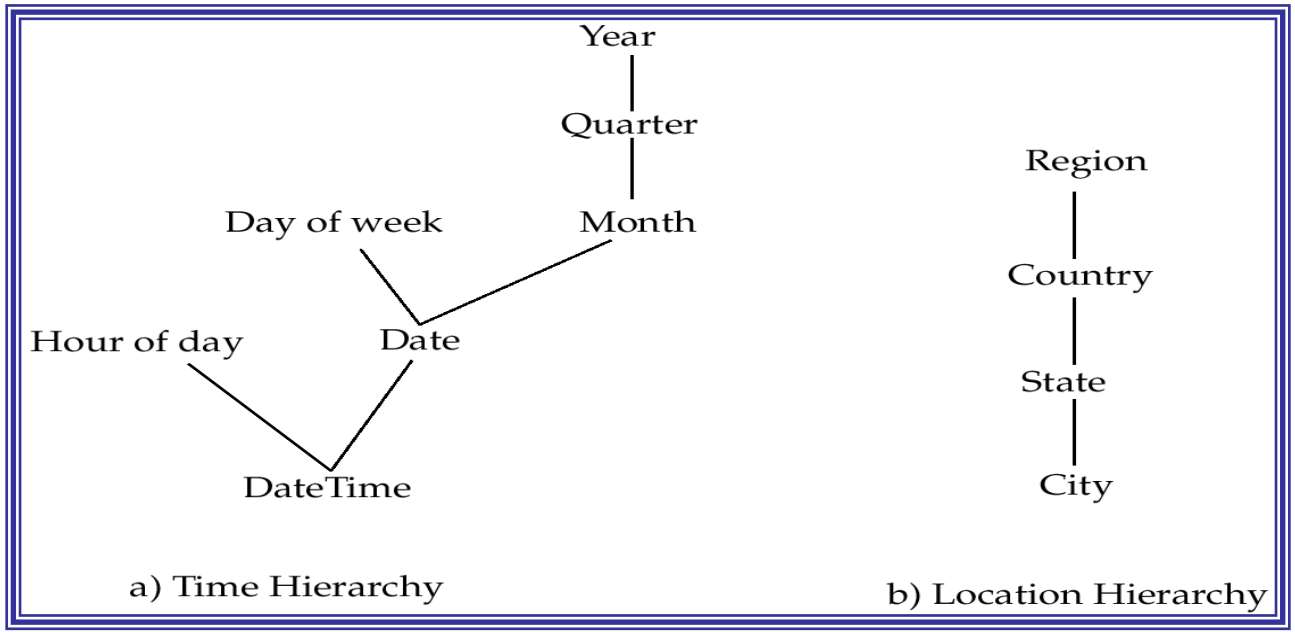
事实表中的属性分为：

1.维度属性：某一个维表中的码

2.度量属性：被分析衡量的属性，通常是数字值，由各个维度共同来决定

维度属性可以有层次的

- 可以在不同层次上查看维表的数据。



度量属性

- Price 是这个系统中需要衡量的指标。
- 它是基于不同维度的组合。

- 例如：价格可以从 酒吧，啤酒，喝酒人和时间共同来决定 价格。

数据仓库数据模式：
星型模型和雪花模式（维表可以有层次）

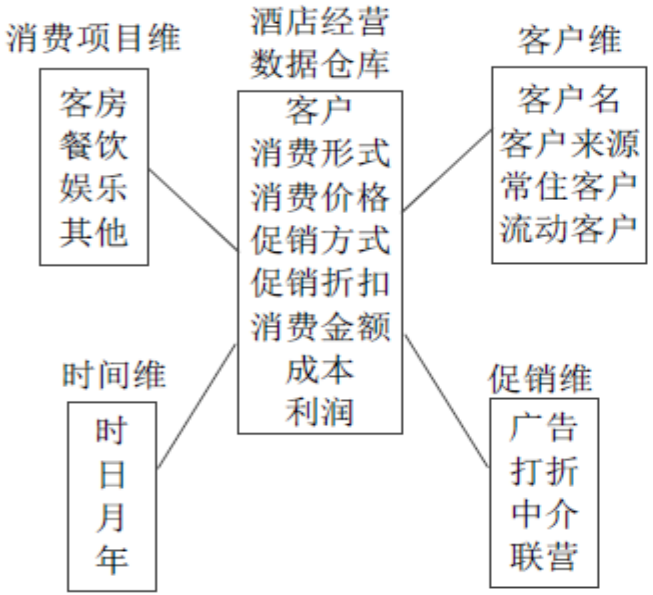


Fig.11 Star data model instance

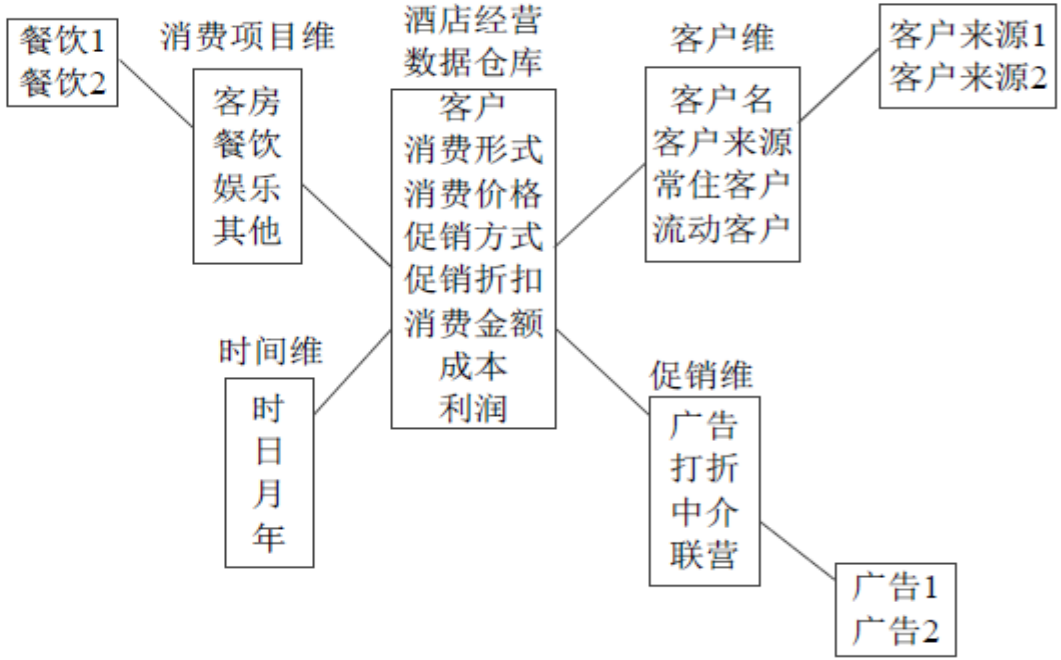


Fig.12 Snowflake data model instance

建立数据仓库的方法

1. ROLAP = “relational OLAP”: 应用关系数据库来管理多维数据，星型模型，雪花模型。
2. MOLAP = “multidimensional OLAP”: 用一个特殊的 DBMS （例如：数据立方体）来实现。

ROLAP 实现技术

1. *Bitmap indexes* **位图索引**：维表的每一个码建立一个位向量，说明事实表中记录具有哪个码值。
2. *Materialized views* ***物化视图***：存贮视图的结果。

位图索引实例

事实表：记录数非常大。

bar	drinkers	beers	price	date
Joe's	david	A	3.5	
Mary's	tim	A	4	
..	..	..	..	

位图索引：它的行数和事实表相同，列数和属性值个数同

Joe's	Mary's
1	0
0	1
..	..

david	tim	wang
1	0	0
0	1	0
..	..	..

假设2个酒吧，3个喝酒人

Select * from fact where bar='Joe"s' and drinkers='david'

物化视图

- 为什么要用物化视图呢？见下面一个实例。

典型的 OLAP 查询：

- 事实表和维表的连接
- Example:

```
SELECT *
FROM Sales, Bars, Beers, Drinkers
WHERE Sales.bar = Bars.bar AND
  Sales.beer = Beers.beer AND
  Sales.drinker = Drinkers.drinker;
```

一般OLAP 的查询

- OLAP 查询的步骤：
  1. 开始进行星型连接。
  2. 根据维度数据，选择感兴趣的元组
  3. 根据某一个或多个维度进行分组.
  4. 根据分组属性，进行聚合操作。

例如: OLAP 查询

对于在某条街上（Dong Chuan Road）的酒吧,查找由AB （Anheuser-Busch）制造商生产的每种啤酒*总的销量*.

- 条件: *addr* = “Dong Chuan” and *manf* = “Anheuser-Busch”.
- 分组: by bar and beer.
- 聚合: Sum of *price*.

SQL 查询语句

```
SELECT bar, beer, SUM(price)
FROM Sales NATURAL JOIN Bars
  NATURAL JOIN Beers
WHERE addr = 'Dong Chuan' AND
  manf = 'Anheuser-Busch'
GROUP BY bar, beer;
```

当事实表数据很大时，该查询需要的时间很长。

使用物化视图来实现

- 事实表和维表的连接可能化很长时间
- 如果我们有事先计算好的视图（materialized view），它有足够的信息满足我们这个查询，那么我们可以直接对这个视图进行查询。

查询：对于在某条街上（DongChuan）的酒吧,查找由AB（Anheuser-Busch）制造商生产的每种啤酒总的销量.

- 该视图的关键点:
  1. 至少要连接Sales, Bars, and Beers.
  2. 分组属性至少有 bar and beer.
  3. 酒吧地址是 DongChuan ， 制造商是 Anheuser-Busch beers.
  4. *addr* 和 *manf* 属性一定要保留

实例：视图 BABMS

```
CREATE VIEW BABMS(bar, addr, beer, manf,
sales) AS
(SELECT bar, addr, beer, manf,SUM(price) as
sales  FROM Sales NATURAL JOIN Bars NATURAL
JOIN Beers
GROUP BY bar, addr, beer, manf);
```

对比两个查询

- 基于该视图：

```
SELECT bar, beer,
sales FROM BABMS
   WHERE addr =
   'Dong Chuan' AND
   manf = 'Anheuser-
   Busch';
```

- 没有物化视图的查询：

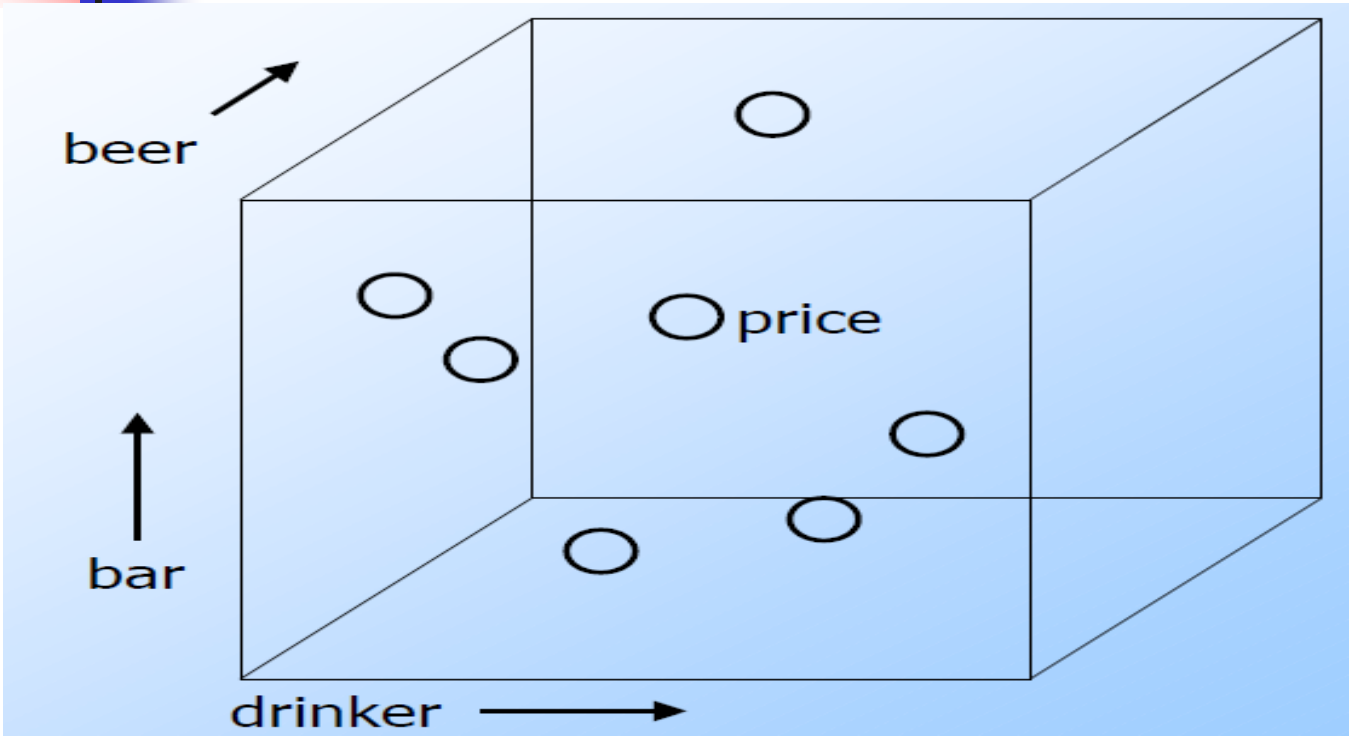
```
SELECT bar, beer, SUM(price)
FROM Sales NATURAL JOIN Bars
NATURAL JOIN Beers
WHERE addr = 'Dong Chuan'
AND manf = 'Anheuser-Busch'
GROUP BY bar, beer;
```

结论：基于物化视图的查询，可以节省大量的时间。

数据立方体模型

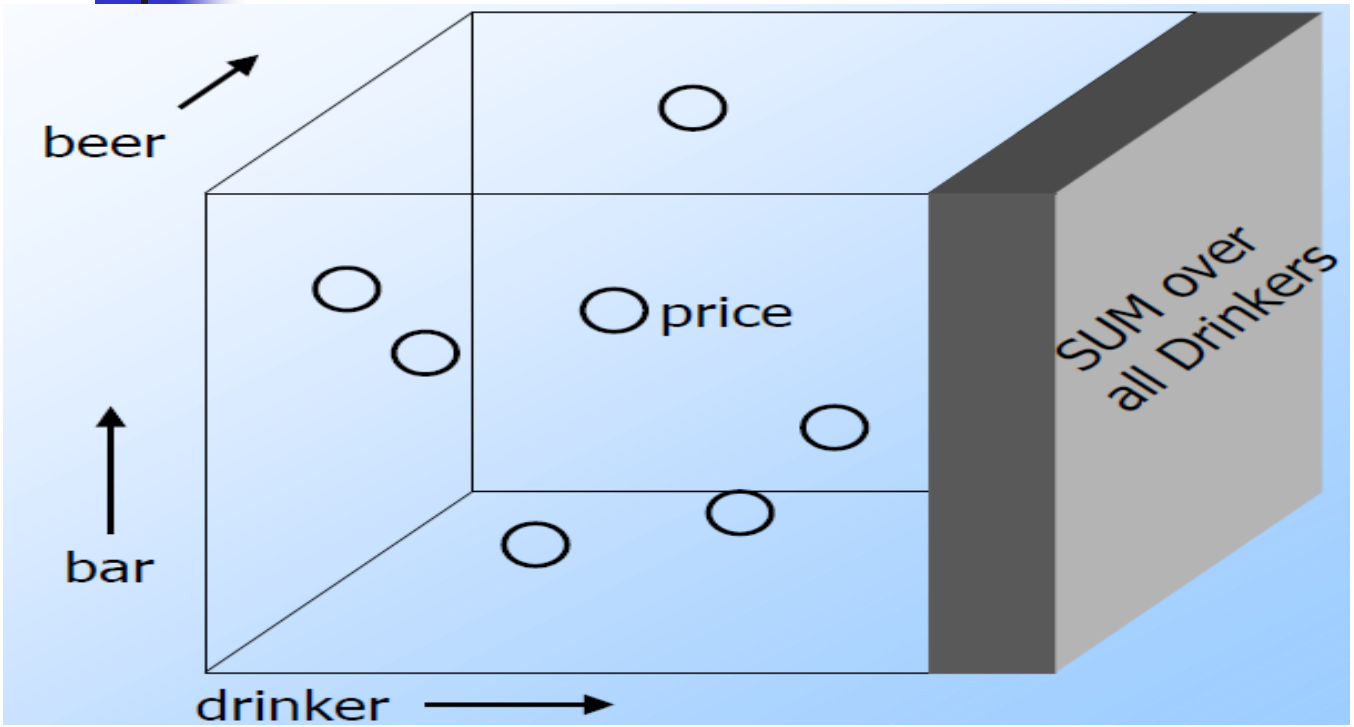
- 以多维立方体来组织数据，多维数组来存贮数据，支持对多维数据的操作。
- 超立方体的维表是 bar, beer, drinker, and time.
- 衡量的属性(price) 就是超立方体上的点

数据立方体：



- 维度数据形成了立方体的轴。
- 度量值是哪些单元的值。

数据立方体的可视化



在边，角和角落上的是聚合数据。

数据立方体

三维的立方体可以显示，多于三维的如何显示呢？

- 如何显示?
- 如何显示聚合数据 aggregated values?

假设我们需要描述酒吧，喝酒人，啤酒以及日期四维数据，即在什么时间，哪个酒吧，哪个人喝了哪种啤酒，化了多少钱？

立方体的结构---数组形式

- 每一维看作是一个元组的一个组成元素

Sales(‘joe’sbar’,’Bud’,’Mary’,2020-05-27)

- 每一维有一个特殊的值*，表示该维是聚合的。

例如：Sales(”Joe’s Bar”, ”Bud”, *, *) 表示Joes酒吧所有客人，花在Bud 啤酒上的开销。

多维分析操作

- 切片 Slice
- 切块 Dice
- 旋转 Pivot
- 向上综合 Roll-up
- 向下钻取 Drill-down

Drill-Down向下钻取

- Drill-down = “de-aggregate” = 分解聚合数据到各个组成部分。
- 例如: Joe’s Bar 酒吧销售非常少的来自制造商 Anheuser-Busch的啤酒,原因是什么呢? 想了解一下具体是哪几个品牌?

Roll-Up

- Roll-up = 按照某一个或某几个维度聚合

- Example:已知每个消费者在每个酒吧喝BUD啤酒的数目,想了解所有消费者一共花费在BUD 啤酒上的总开销是多少。

例子: Roll Up 和 Drill Down

$ of Anheuser-Busch by drinker/bar

	Jim	Bob	Mary
Joe's Bar	45	33	30
Nut-House	50	36	42
Blue Chalk	38	31	40

Roll up by Bar

$ of A-B / drinker

Jim	Bob	Mary
133	100	112

Drill down by Beer

$ of A-B Beers / drinker

	Jim	Bob	Mary
Bud	40	29	40
M'lob	45	31	37
Bud Light	48	40	35

$ means the price charged.

Examples: Roll Up by bar

```
Select bar, drinker, sum(price)
From sales natural join beers
Where manf="AB"
Group by bar, drinker;
```

```
Select drinker, sum(price)
From sales natural join beers
Where manf="AB"
Group by drinker;
```

Example: Drill Down on beer

```
Select drinker, sum(price)
From sales natural join beers
Where manf="AB"
Group by drinker;
```

```
Select beer,drinker, sum(price)
From sales natural join beers
Where manf="AB"
Group by beer, drinker;
```

“slicing and dicing”的形式

Slicing: 切片 让某一个维度具有某个固定的值.
Dicing: 切块，固定某一个或某几个维度.

Select <*grouping attributes and aggregations*>
From <*fact table joined with some dimension tables*>
Where<*certain attributes are constant*>
Group by<*grouping attributes*>;

例如：

$ of Anheuser-Busch by drinker/bar

	Jim	Bob	Mary
Joe's Bar	45	33	30
Nut-House	50	36	42
Blue Chalk	38	31	40

Roll up by Bar

$ of A-B / drinker

Jim	Bob	Mary
133	100	112

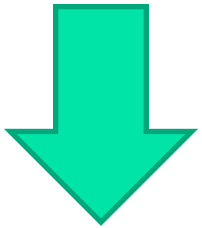
Drill down by Beer

$ of A-B Beers / drinker

	Jim	Bob	Mary
Bud	40	29	40
M'lob	45	31	37
Bud Light	48	40	35

Examples: Slicing

```
Select manf, bar, drinker, sum(price)
From sales natural join beers
Group by manf, bar, drinker;
```



```
Select bar, drinker, sum(price)
From sales natural join beers
Where manf='Anheuse-Busch'
Group by bar, drinker;
```

Example: Dicing

Find the sales of those bar located in Dong Chuan and sold beers from manufacture of "A-B":

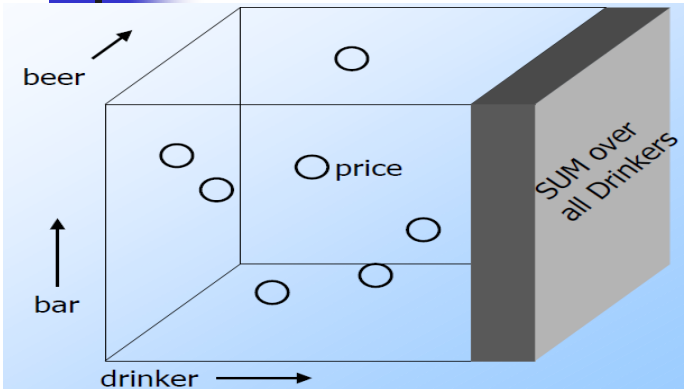
Select bar, drinker, sum(price)

From sales natural join beers natural join bars

Where manf='Anheuse-Busch' and addr='Dong Chuan'

Group by bar, drinker;

SQL中关于立方体的操作



物化视图看作为数据立方体

Create materialized view salesCube as

Select drink,bar,beer, sum(price)

From sales

Group by drink,bar,beer WITH CUBE;

(Jim, null, Bud, 20)

(Jim, Joe'sbar, null, 45)

(Jim, null,null,133)

(null, null, null,345)

Null 类似于前面多维数据库中的 *

SQL中关于立方体的操作

```
Create materialized view salesRollup as
Select drink,bar,beer, sum(price)
From sales
Group by drink,bar,beer WITH ROLLUP;
```

产生元组如下:
(Jim, Joe'sbar, Bud, 20)
(Jim, Joe'sbar, null, 45)
(Jim, null,null,133)
(null,null,null, 345)

问题:
它们之间的区别在哪里?

扩展的SQL语句

计算所有分组属性的各个子集

select *drinker, bar, beer,* **sum**(*price*)
from *sales*
group by cube(*drinker, bar, beer*)

共有如下8个分组的子集:

{ (*drinker, bar, beer*), (*drinker, bar*),
(*drinker, beer*), (*bar, beer*),
(*drinker*), (*bar*),
(*beer*), ( ) }

最后一个空代表没有分组，只有一个值。

扩展的SQL语句

- 计算分组属性中 有序的子集

  select *drinker*, *bar*, *beer*, **sum**(*price*)
  from *sales*
  group by rollup(*drinker, bar, beer*)

- **产出了4个子集:**

  { (*drinker, bar, beer*), (*drinker, bar*), (*drinker*), ( ) }

- 用于层次化的分组。

数据挖掘

- 目的：发现并提取隐藏在内，人们事先不知道的，但可能有用的信息和知识。

数据挖掘功能

- 概念描述：归纳数据的某些特征
- **关联分析**：发现两个或多个变量取值之间的某种规律性
- **分类与预测**
- 聚类
- 孤立点检测
- 趋势和演变分析

关联分析

- 找相似客户买的一系列书籍. 如果一个新客户买了一本书,系统会推荐他买其它的书.
  - *一个用户买了 Database System Concepts* 很有可能会买 *Operating System Concepts*.
  - 超市里,买了面包的人一般都会买牛奶
- 因果分析: X光和癌症的关系

如何找关联规则呢?

- **算法: a priori** technique 来找频繁的数据项集合。
- **两个评测指标：**
- **Support**：超市里有多少人买了X，又买了Y？ $sup = \Pr(X \cup Y)$.
  - 支持度(support)=(X,Y).count/T.count
- **Confidence**：买了**X**的人有多少人买**Y**？ $conf = \Pr(Y \mid X)$
  - 置信度(confidence)=(X,Y).count/X.count

实例

t1: Beef, Chicken, Milk
t2: Beef, Cheese
t3: Cheese, Boots
t4: Beef, Chicken, Cheese
t5: Beef, Chicken, Clothes, Cheese, Milk
t6: Chicken, Clothes, Milk
t7: Chicken, Milk, Clothes

- 假设数据如上
- Assume:
  minsup = 30%
  minconf = 80%
- 频繁项集 frequent *itemset*:

{Chicken, Clothes, Milk} [sup = 3/7]

- 关联规则:

Clothes → Milk, Chicken [sup = 3/7, conf = 3/3]
…
Clothes, Chicken → Milk, [sup = 3/7, conf = 3/3]

分类与预测

保险公司如何判断一个客户投保的信用呢? 通过对已有客户数据的分类来预测

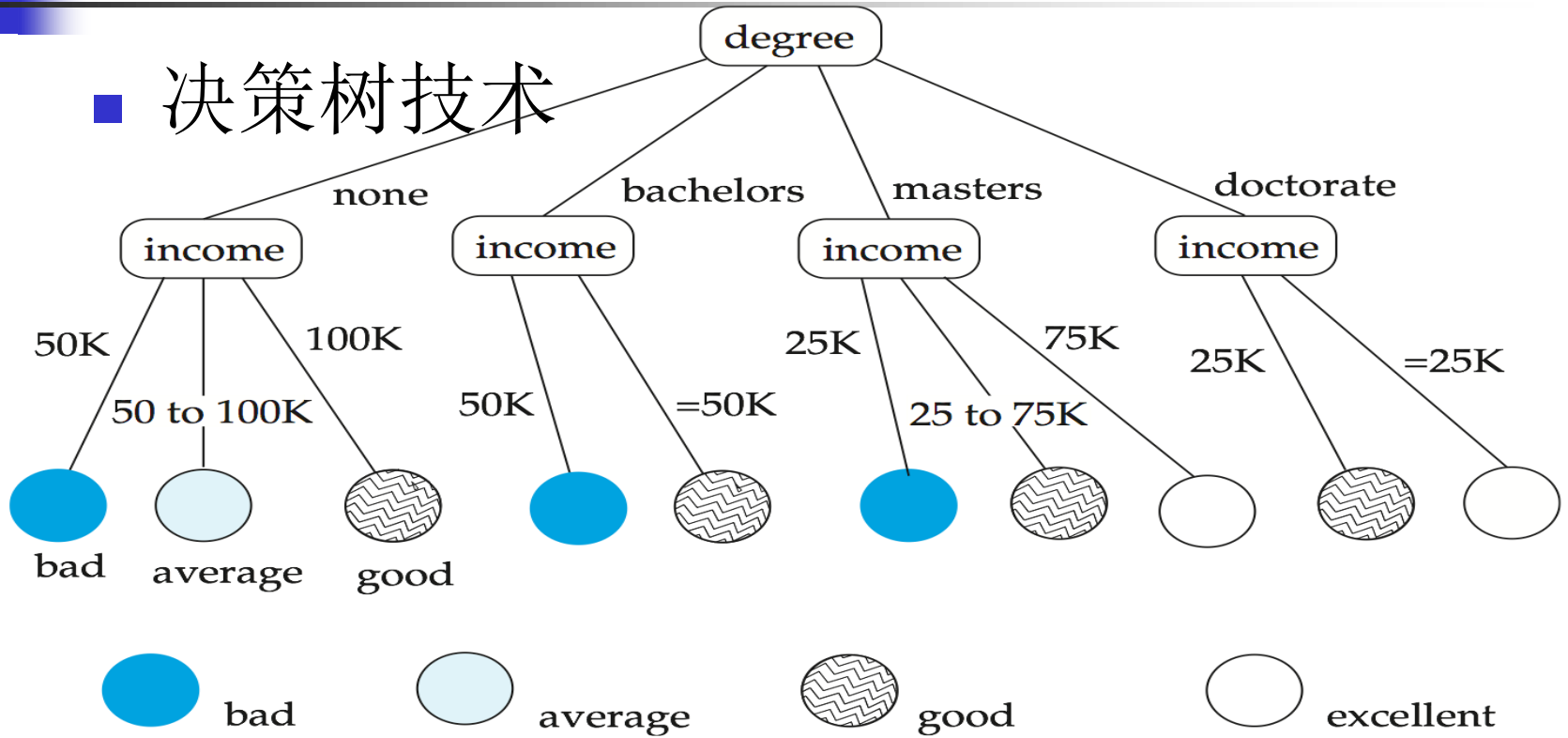
例如:

- 分类规则可以使用很多数据，例如学历，年薪，年龄等.

  - $\forall$ person P, P.degree = masters **and** P.income > 75,000
    $\Rightarrow$ P.credit = excellent
  - $\forall$ person P, P.degree = bachelors **and**
    (P.income $\geq$ 25,000 and P.income $\leq$ 75,000)
    $\Rightarrow$ P.credit = good

如何建立这些预测规则呢？

- 决策树技术



小结

- 数据仓库如何建立
- 联机数据分析处理技术
- 数据挖掘