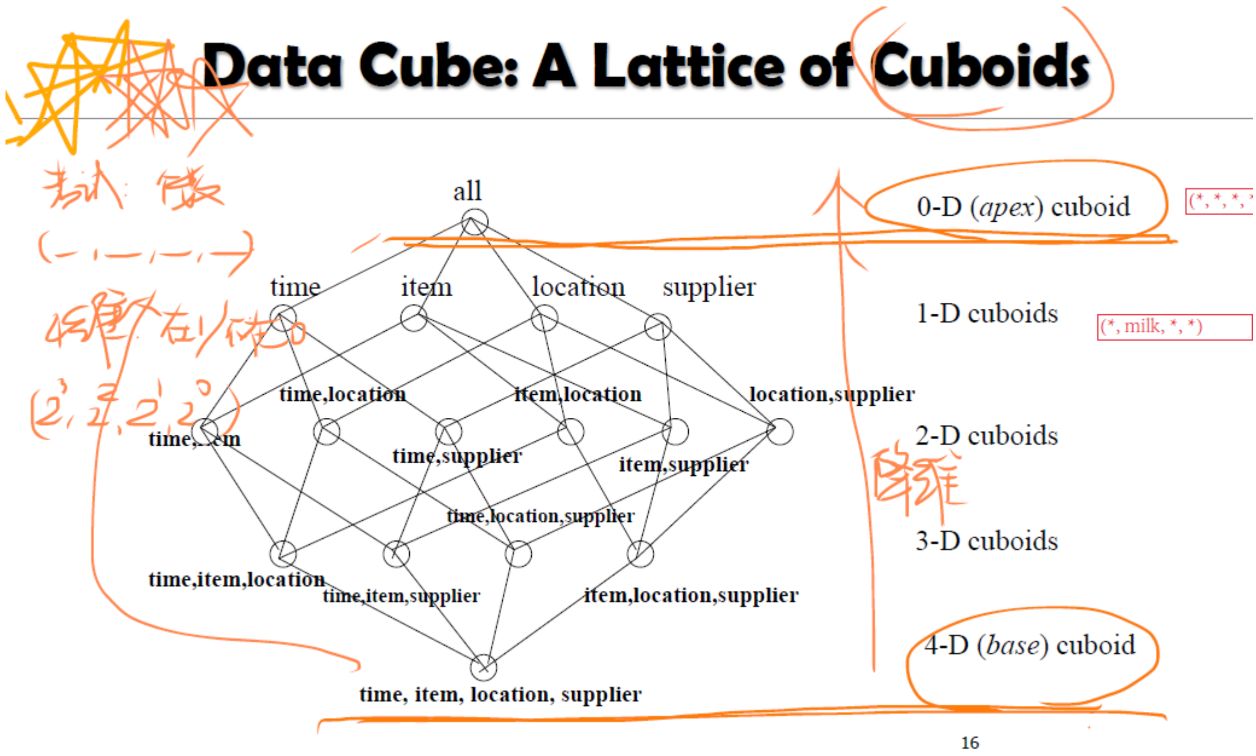
# Chap 4 Data Warehousing and On-line Analytical Processing

- Table
  - Dimension Table
  - Fact Table: contains measures
- Spreadsheet
- Data Cube ( P15-16)
  - Base Cuboid: n-D base cubes
  - Apex Cuboid: highest-level of summarization

### 1. Data Warehouse: A multi-dimensional model of a data warehouse

#### 1) A Data Cube consists of dimensions & measures: (P15-16)

- Base Cuboid: n-D base cube
- Apex Cuboid: highest-level of summarization



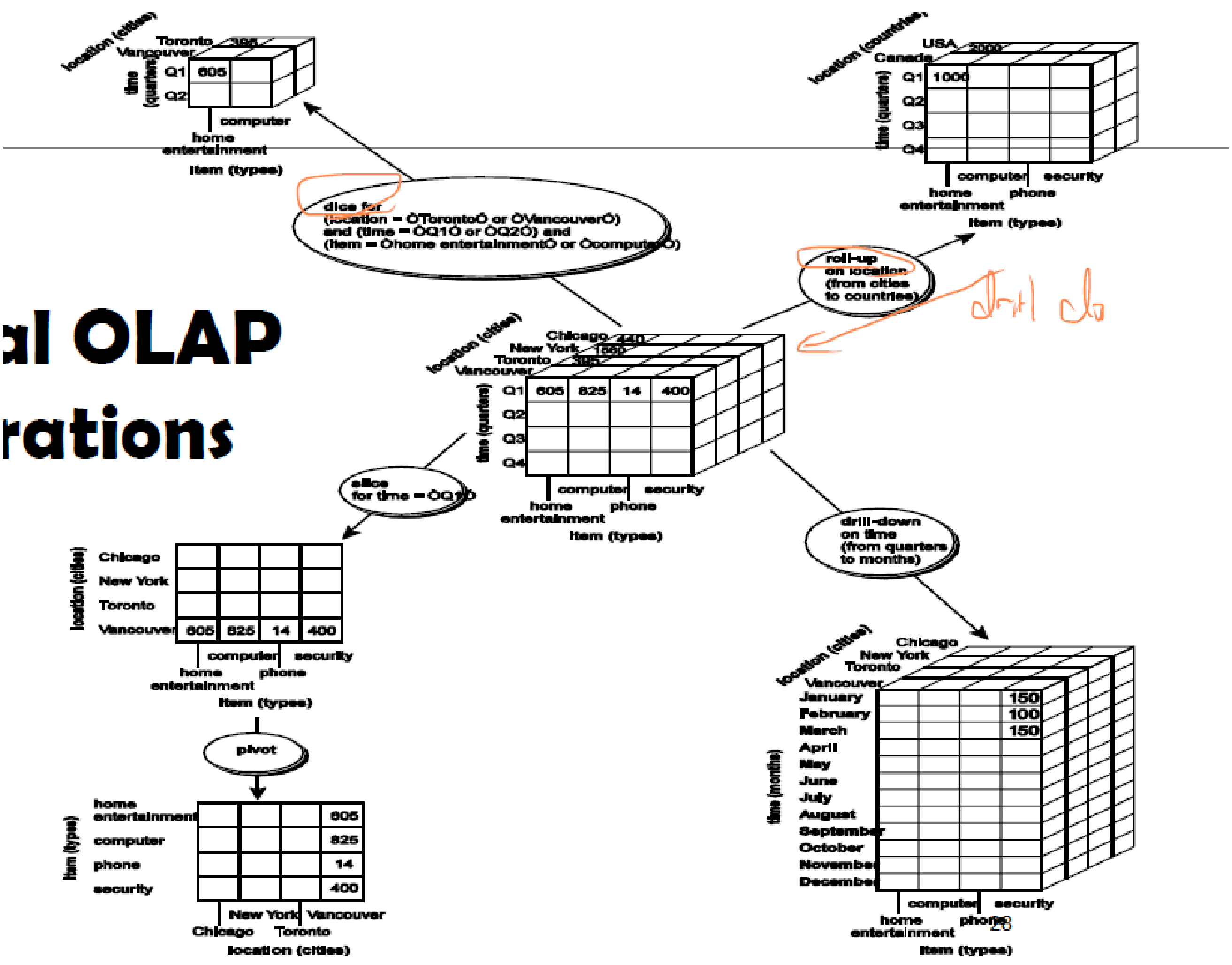

#### 2) Star schema, snowflake schema, fact constellations (P17-20)

- Star schema: A fact table in the middle connected to a set of dimension tables
- Snowflake schema: A refinement of star schema where some dimensional hierarchy is normalized into a set of smaller dimension tables, forming a shape similar to snowflake
- Fact constellations: Multiple fact tables share dimension tables, viewed as a collection of stars, therefore called galaxy schema or fact constellation

- Star Schema: 有一个中心的fact table连接着很多dimension tables
- Snowflake Schema： 连接的dimension tables有hierarchy
- Fact Constellation/ Galaxy Schema / Fact Constellation： 有多个fact table

**3) OLAP operations: drilling, rolling, slicing, dicing and pivoting**

- Roll up (drill-up): summarize data
  - *by climbing up hierarchy or by dimension reduction*
- Drill down (roll down): reverse of roll-up
  - *from higher level summary to lower level summary or detailed data, or introducing new dimensions*
- Slice and dice: *project and select*
- Pivot (rotate):
  - *reorient the cube, visualization, 3D to series of 2D planes*
- Other operations
  - *Drill across: involving (across) more than one fact table*
  - *Drill through: through the bottom level of the cube to its back-end relational tables (using SQL)*




- Roll Up/ Drill Up: summary并且降维
- Drill Down/ Roll Down： 增加维度
- Slice and Dice： slice 切片就是固定某个维度的值，dice是类似从多个维度挑选几个值
- Pivot / Rotate
- Drill Across
- Drill Through

**4) Measures: Distributive, Algebraic, Holistic**

## Data Cube Measures: Three Categories

- **Distributive**: if the result derived by applying the function to *n* aggregate values is the same as that derived by applying the function on all the data without partitioning
  分开算之后的结果一样
  - E.g., count(), sum(), min(), max()
- **Algebraic**: if it can be computed by an algebraic function with *M* arguments (where *M* is a bounded integer), each of which is obtained by applying a distributive aggregate function
  分开算出来后和一起的不一样
  - avg(x) = sum(x) / count(x)
  - Is min_N() an algebraic measure? std ? $SD = \sqrt{\frac{1}{N}\sum(x_i-\mu)^2}$
- **Holistic**: if there is no constant bound on the storage size needed to describe a subaggregate.
  need everything — BAD !
  - E.g., median(), mode(), rank()

**2. Data Warehouse: Architecture, Design and Usage**

**1) Multi-tiered architecture**

**2) Business analysis design framework**

**3) Information processing, analytical processing, data mining**

**3. Implementation: Efficient computation of data cubes**

**1) Partial Materialization vs. Full Materialization vs. No Materialization**

## Efficient Data Cube Computation

- Data cube can be viewed as a lattice of cuboids
  - The bottom-most cuboid is the base cuboid
  - The top-most cuboid (apex) contains only one cell
  - How many cuboids in an n-dimensional cube with L levels?

(3+1)(4+1)(5+1)

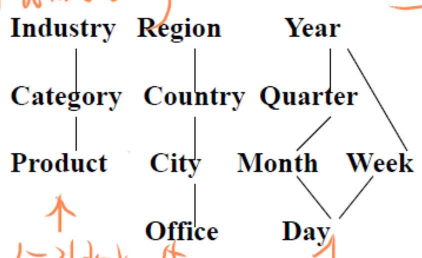
Why this formula?

$$T = \prod_{i=1}^{n} (L_i + 1)$$

level, everything

- Materialization of data cube
  - **Full materialization**: Materialize every (cuboid)
  - **No materialization**: Materialize none (cuboid)
  - **Partial materialization**: Materialize some cuboids
    - Which cuboids to materialize?
      - Selection based on size, sharing, access frequency, etc.

3 dimension




L=3 levels L=4 levels L=6 levels

**2) Indexing OALP data: Bitmap index and join index**

**3) OLAP query processing**

4) OLAP servers: Relational OLAP( ROLAP), Multidimensional OLAP(MOLAP), Hybrid OLAP(HOLAP)

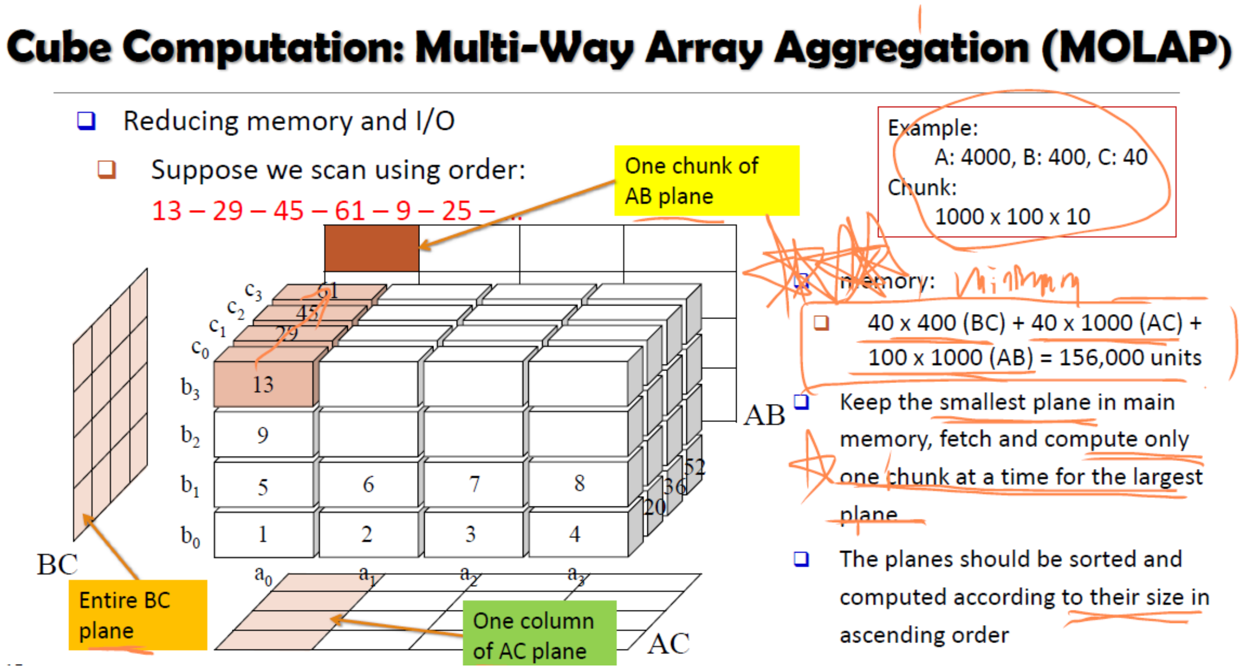
# Chap 5 Data Cube Technology

## 1. Data Cube Computation: Preliminary Concepts

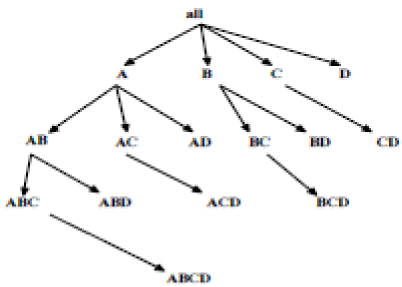
- Base Cell vs Aggregate Cell
- Ancestor Cell vs Descendant Cell
- Parent Cell vs Child Cell
- Full Cube vs Iceberg Cube
- **Close Cube & Close Cell**
- Cube Shell

## 2. Data Cube Computation Methods

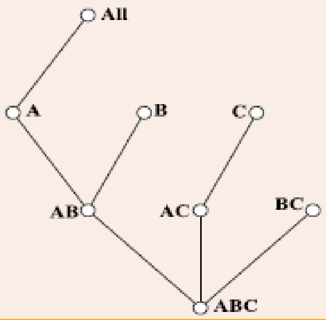
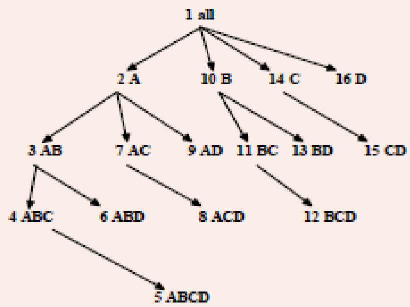
- MultiWay Array Aggregation —— small number of dimensions




- BUC —— large number of dimensions



## MultiWay VS BUC

| | multiway | BUC |
|---|---|---|
| Input format | Multi-dimensional array | Relational database 考什么时候选哪种 |
| Good for | Full cube | Iceberg cube |
| Key idea | Simultaneously Aggregation | Partition and sort |
| Calculation direction |  |  |





- High-Dimensional OLAP with Shell-Fragments

## Shell Fragment Cubes: Ideas

- Generalize the **1-D** inverted indices to **multi-dimensional** ones in the data cube sense
- Compute all cuboids for data cubes ABC and DE while retaining the inverted indices
  - Ex. shell fragment cube ABC contains 7 cuboids:
    - A, B, C; AB, AC, BC; ABC
- This completes the offline computation

| Attribute Value | TID List | List Size |
|---|---|---|
| a1 | 1 2 3 | 3 |
| a2 | 4 5 | 2 |
| b1 | 1 4 5 | 3 |
| b2 | 2 3 | 2 |
| c1 | 1 2 3 4 5 | 5 |
| d1 | 1 3 4 5 | 4 |
| d2 | 2 | 1 |
| e1 | 1 2 | 2 |
| e2 | 3 4 | 2 |
| e3 | 5 | 1 |

Shell-fragment AB

| Cell | Intersection | TID List | List Size |
|---|---|---|---|
| a1 b1 | 1 2 3 ∩ 1 4 5 | 1 | 1 |
| a1 b2 | 1 2 3 ∩ 2 3 | 2 3 | 2 |
| a2 b1 | 4 5 ∩ 1 4 5 | 4 5 | 2 |
| a2 b2 | 4 5 ∩ 2 3 | ф | 0 |

- ID_Measure Table
  - If measures other than count are present, store in *ID_measure* table separate from the shell fragments

| tid | count | sum |
|---|---|---|
| 1 | 5 | 70 |
| 2 | 3 | 10 |
| 3 | 8 | 20 |
| 4 | 5 | 40 |
| 5 | 2 | 30 |

### 3. Multidimensional Data Analysis in Cube Space

- Multi-feature Cubes
- Discover-Driven Exploration of Data Cubes

# Chap 6 Mining Frequent Patterns, Association and Correlations

## 1. Basic Concepts

#### 1) Pattern Discovery

#### 2) Basic Concepts: Frequent Patterns and Association Rules

#### 3) Compressed Representation: Closed Patterns and Max-Patterns

## 2. Efficient Pattern Mining Methods:

#### 1) The Downward Closure Property of Frequent Patterns

#### 2) The Apriori Algorithm

$C_k$: Candidate itemset of size k

$F_k$ : Frequent itemset of size k

K := 1;
$F_k$ := {frequent items}; // frequent 1-itemset
**While** ($F_k$ != ∅) **do {** // when $F_k$ is non-empty
$C_{k+1}$ := candidates generated from $F_k$; // candidate generation
Derive $F_{k+1}$ by counting candidates in $C_{k+1}$ with respect to *TDB* at minsup;
k := k + 1
**}**
**return** $\cup_k F_k$ // return $F_k$ generated at each level

### The Apriori Algorithm—An Example

从小的开始，然后不断增加set

minsup = 2

Database TDB

| Tid | Items |
|---|---|
| 10 | A, C, D |
| 20 | B, C, E |
| 30 | A, B, C, E |
| 40 | B, E |

$C_1$ (1st scan)

| Itemset | sup |
|---|---|
| {A} | 2 |
| {B} | 3 |
| {C} | 3 |
| ~~{D}~~ | ~~1~~ |
| {E} | 3 |

$F_1$ (frequent pattern)

| Itemset | sup |
|---|---|
| {A} | 2 |
| {B} | 3 |
| {C} | 3 |
| {E} | 3 |

$C_2$

| Itemset |
|---|
| {A, B} |
| {A, C} |
| {A, E} |
| {B, C} |
| {B, E} |
| {C, E} |

$C_2$ (2nd scan)

| Itemset | sup |
|---|---|
| ~~{A, B}~~ | ~~1~~ |
| {A, C} | 2 |
| {A, E} | 1 |
| {B, C} | 2 |
| [illegible], E} | 3 |
| [illegible], E} | 2 |

只有C是close pattern，不会影响F的数量

$F_2$

[A, C]是max pattern

| Itemset | sup |
|---|---|
| {A, C} | 2 |
| {B, C} | 2 |
| {B, E} | 3 |
| {C, E} | 2 |

不是max pattern

$C_3$

| Itemset |
|---|
| {B, C, E} |

3rd scan

$F_3$

| Itemset | sup |
|---|---|
| {B, C, E} | 2 |

max pattern

max pattern可以减少storage，但是会减少信息

**3) Extensions or Improvements of Apriori**

**4) Mining Frequent Patterns by Exploring Vertical Data Format**

**5) FP Growth A Frequent Pattern-Growth Approach**

# Example: Construct FP-tree from a Transaction DB

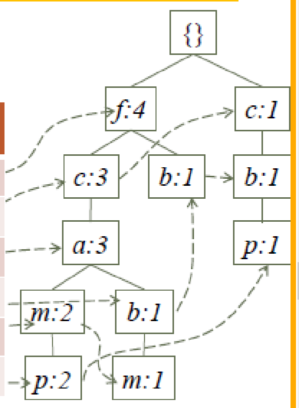
| TID | Items in the Transaction | Ordered, frequent itemlist |
|---|---|---|
| 100 | {f, a, c, d, g, i, m, p} | f, c, a, m, p |
| 200 | {a, b, c, f, l, m, o} | f, c, a, b, m |
| 300 | {b, f, h, j, o, w} | f, b |
| 400 | {b, c, k, s, p} | c, b, p |
| 500 | {a, f, c, e, l, p, m, n} | f, c, a, m, p |

4. For each transaction, insert the ordered frequent itemlist into an FP-tree, with shared sub-branches merged, counts accumulated

After inserting all the frequent itemlists

Header Table

| Item | Frequency | header |
|---|---|---|
| f | 4 | |
| c | 4 | |
| a | 3 | |
| b | 3 | |
| m | 3 | |
| p | 3 | |




memory会减少

**6) Mining Closed Patterns**

## 3. Pattern Evaluation

**1) Interestingness Measures in Pattern Mining: Lift**

- Measure of dependent/correlated events: **lift**

(400/1000) / (600/1000 * 750/1000)

$$lift(B,C) = \frac{c(B \rightarrow C)}{s(C)} = \frac{s(B \cup C)}{s(B) \times s(C)}$$

Lift is more telling than s & c

| | B | ¬B | $\Sigma_{row}$ |
|---|---|---|---|
| C | 400 | 350 | 750 |
| ¬C | 200 | 50 | 250 |
| $\Sigma_{col.}$ | 600 | 400 | 1000 |

- Lift(B, C) may tell how B and C are correlated
  - Lift(B, C) = 1: B and C are independent
  - > 1: positively correlated
  - < 1: negatively correlated
- For our example,

$$lift(B,C) = \frac{400/1000}{600/1000 \times 750/1000} = 0.89$$

$$lift(B,\neg C) = \frac{200/1000}{600/1000 \times 250/1000} = 1.33$$

- Thus, B and C are negatively correlated since lift(B, C) < 1;
  - B and ¬C are positively correlated since lift(B, ¬C) > 1

- Another measure to test correlated events: $\chi^2$

$$\boxed{\chi^2 = \sum \frac{(Observed - Expected)^2}{Expected}}$$

| | B | ¬B | $\Sigma_{row}$ |
|---|---|---|---|
| C | 400 (450) | 350 (300) | 750 |
| ¬C | 200 (150) | 50 (100) | 250 |
| $\Sigma_{col}$ | 600 | 400 | 1000 |

Expected value

Observed value

- For the table on the right,

$$\chi^2 = \frac{(400-450)^2}{450} + \frac{(350-300)^2}{300} + \frac{(200-150)^2}{150} + \frac{(50-100)^2}{100} = 55.56$$

- By consulting a table of critical values of the $\chi^2$ distribution, one can conclude that the chance for B and C to be independent is very low (< 0.01)
- $\chi^2$-test shows B and C are negatively correlated since the expected value is 450 but the observed is only 400
- Thus, $\chi^2$ is also more telling than the support-confidence framework

**2) Interestingness Measures: Lift and X^2**

**3) Null-Invariant Measures**

# Imbalance Ratio with Kulczynski Measure

- IR (Imbalance Ratio): measure the imbalance of two itemsets A and B in rule implications:

$$\boxed{IR(A,B) = \frac{|s(A)-s(B)|}{s(A)+s(B)-s(A\cup B)}}$$

**4) Comparison of Interestingness Measures**