Universität Stuttgart

IPVS Institut für Parallele und Verteilte Systeme

Data-Warehouse-, Data-Mining- und OLAP-Technologien

Online Analytic Processing

Bernhard Mitschang
Universität Stuttgart

Winter Term 2017/2018

Overview

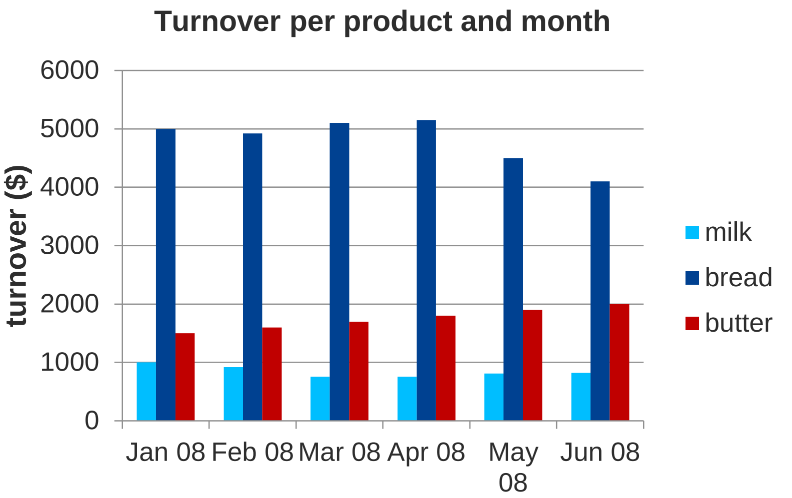
- OLAP
  - Introduction
  - Operations
  - Characteristics
- Storage of OLAP cubes
  - Relational vs. Multidimensional
  - Multidimensional Arrays
  - Sparse Cubes
  - Multidimensional Query Language
- Architecture
  - MOLAP, ROLAP, HOLAP

OnLine Analytic Processing (OLAP)

- Technologies and tools that support (ad-hoc) analysis of multi-dimensionally aggregated data
- Individual analysis is supported, i.e., the user is not restricted to available standard reports/analysis
- Graphical user interface is available for analysis specification
- Knowledge of a query language or programming language is not required
- Result information is given graphically and made available for incorporation into other applications
- Users: Analysts, Manager, “knowledge worker”
- Typical Analysis scenarios
  - Multi-dimensional views, e.g. turnover per product group and month
  - Comparisons, e.g. turnover in Q4 compared to that of Q3
  - Ranking, e.g. top 10 product in a certain group ranked by turnover

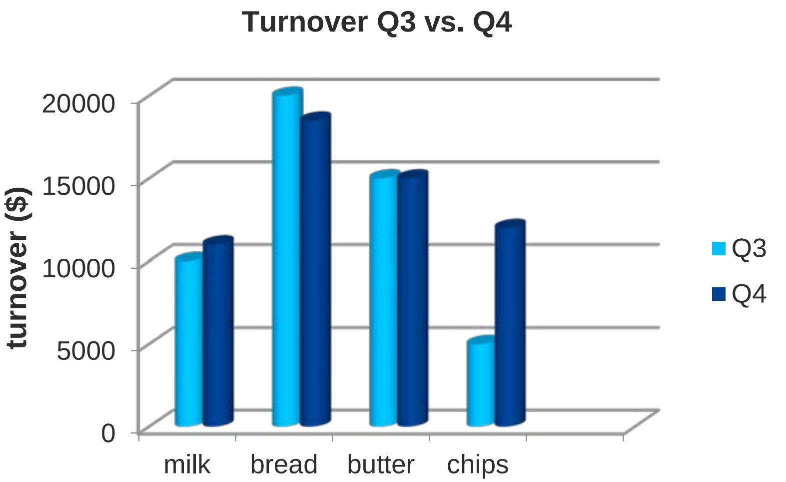
OLAP

- Defining OLAP reports/Analysis
  - select facts
  - select dimensions
  - define filters
  - define presentation



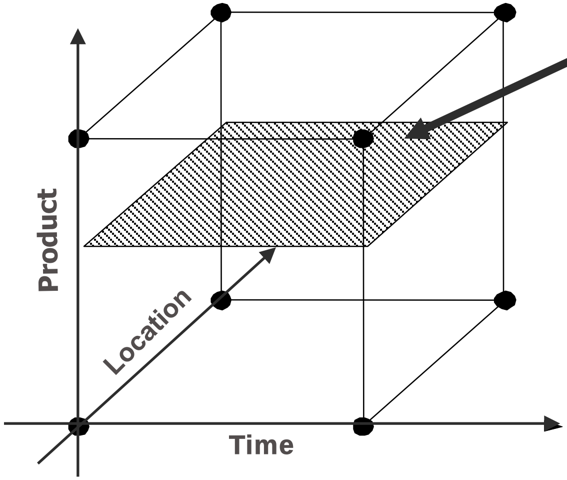
Top 10 fruit and vegetables

Rank	Produkt	Turnover ($)
1	potatoes	210000
2	carrots	205000
3	celery	190000
3	tomatoes	190000
5	kiwi fruit	150000
6	strawberry	145000
7	spinach	142000
8	zucchini	95500
9	lettuce	94000
10	blackberry	92000



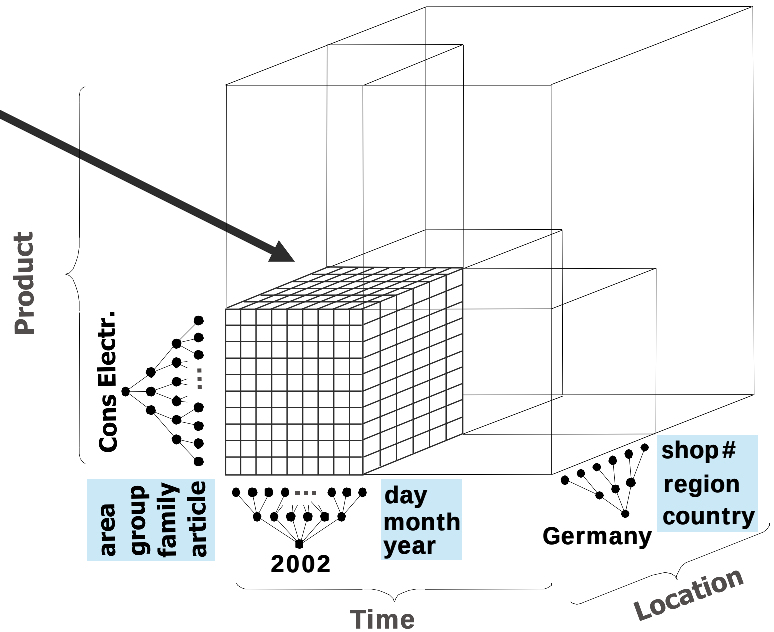
Data Warehouse Design

Multidimensional Model

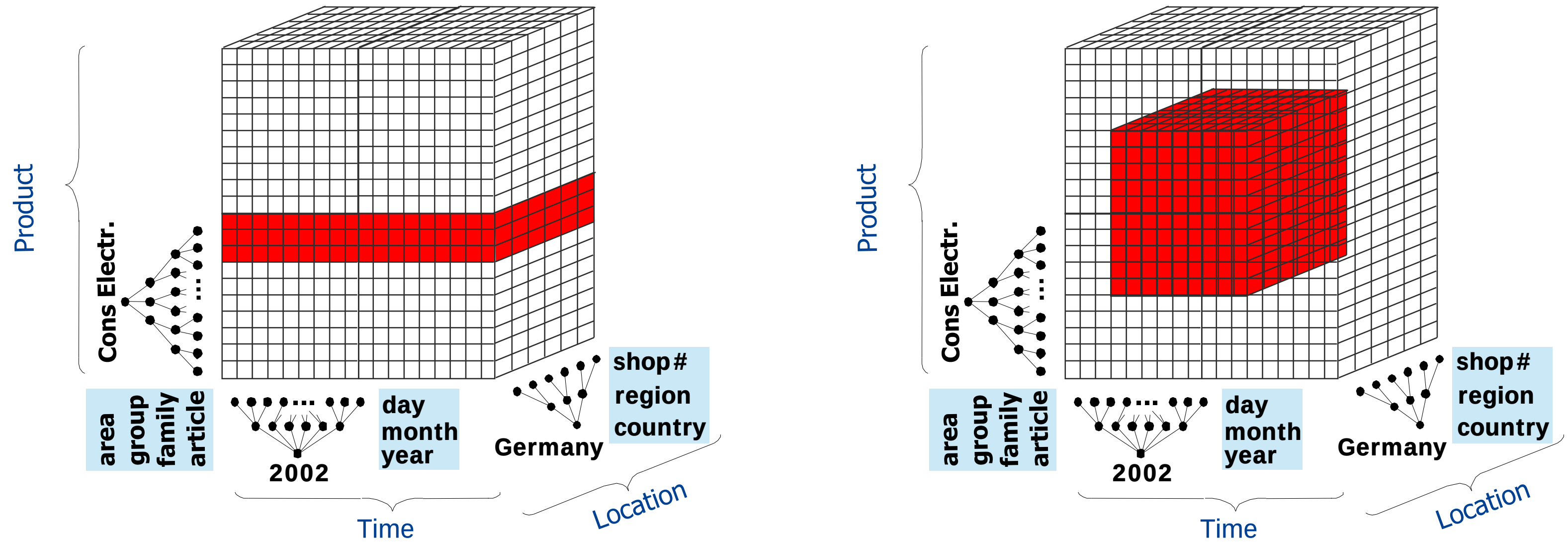


Fact Data (sales)

„Cube“ Metaphor

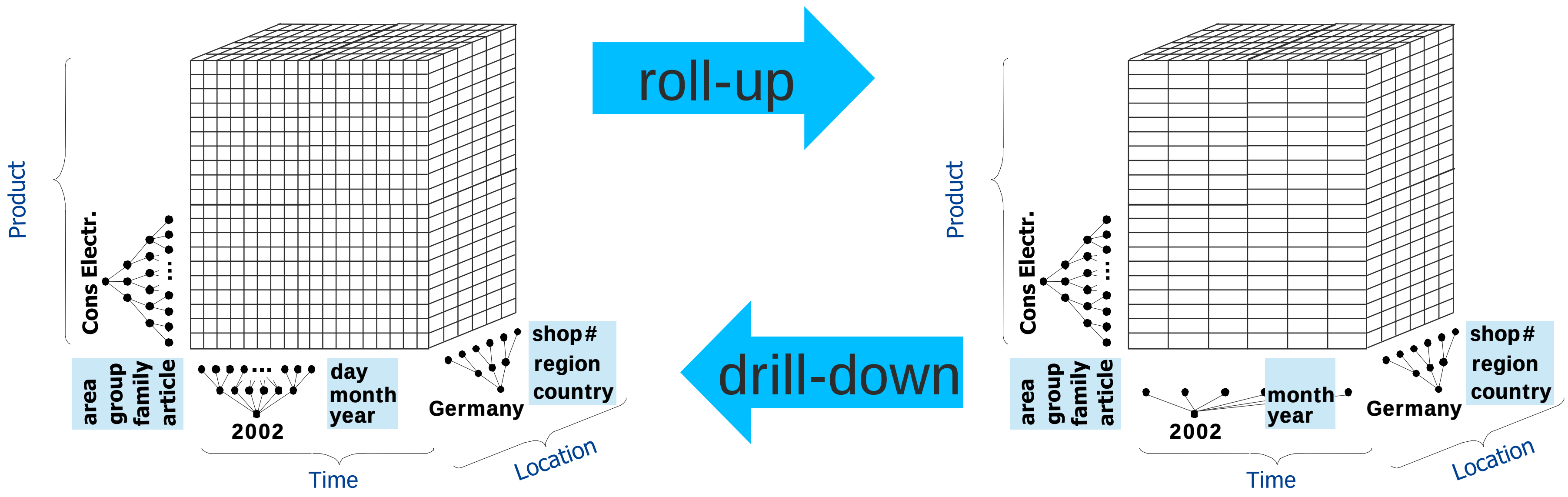


Slice and Dice



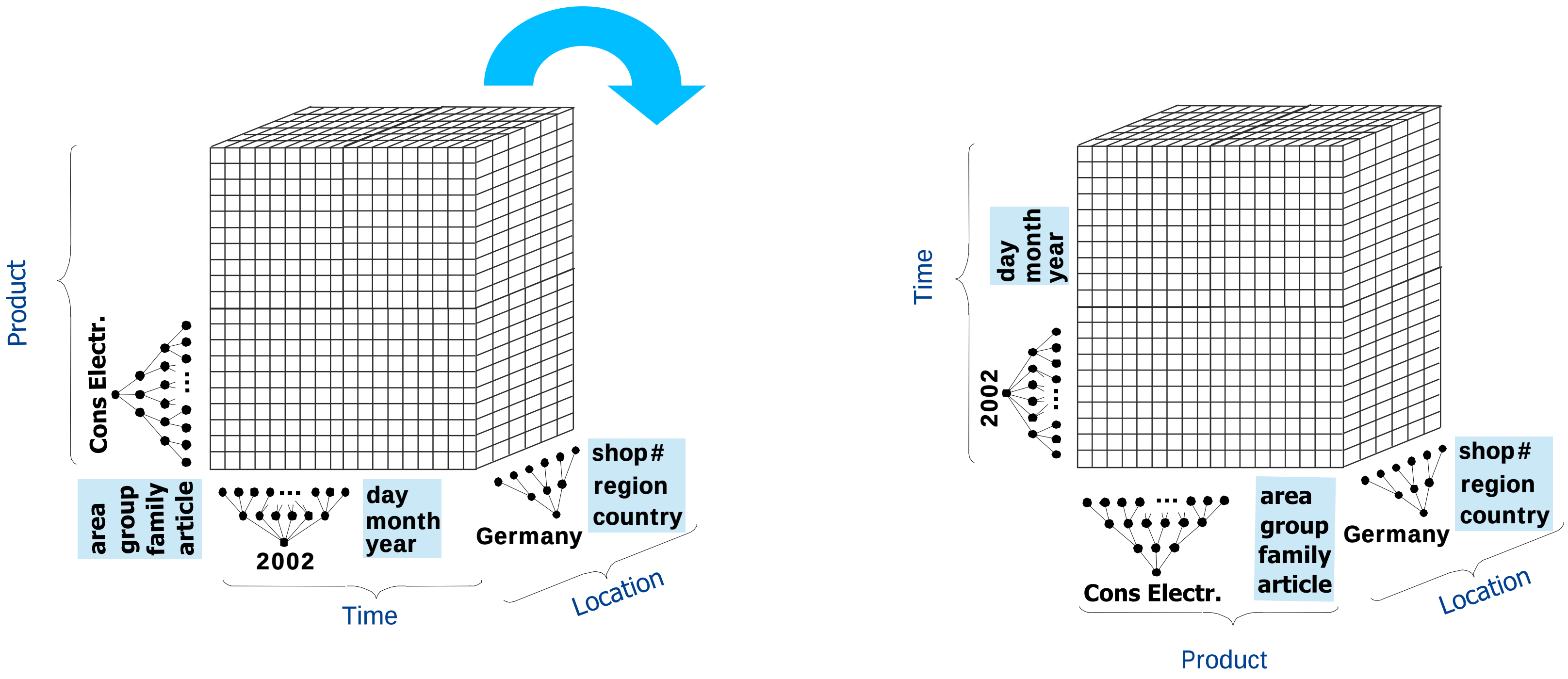
- Slice
  - restrict one dimension to a range of values
- Dice
  - restrict several dimensions to a range of values
  - results in a sub-cube
- Example: Analysis of a certain product family

Roll-up and Drill-down



- 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

Pivot and Rotate



- Pivot
  - reorient the cube
  - visualization
  - 3D to series of 2D planes

OLAP Operations

Overview

- Typical OLAP operations (explained in a general manner)
  - 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)

OLAP Product Evaluation Rules

Basic Features

R1: multi-dimensional conceptual view
R10: intuitive data manipulation
R3: accessibility
N: batch extraction vs. interpretive
N: OLAP analysis models
R5: client-server architecture
R2: transparency
R8: multi-user support

Reporting Features

R11: flexible reporting
R4: consistent reporting performance
R7: dynamic sparse matrix handling

Dimension Control

R6: generic dimensionality
R12: unlimited dimensions and aggregation levels
R9: unrestricted cross-dimensional operations

Special Features

N: treatment of non-normalized data
N: storing OLAP results: keeping them separate from source data
N: extraction of missing values
N: treatment of missing values

R1 - R12: original rules
N: additional rules

FASMI Test

FAST	• deliver most responses within about five seconds • simplest analysis taking no more than one second • very few taking more than 20 seconds
ANALYSIS	• cope with any business logic and statistical analysis that is relevant for applications and users • allow users to define new ad-hoc calculations without programming
SHARED	• confidentiality • concurrent update locking if multiple write access is needed
MULTIDIMENSIONAL	• multidimensional conceptual view of data • support for hierarchies and multiple hierarchies
INFORMATION	• handle huge amounts of input data

Source: Nigel Pendse: What is OLAP? www.olapreport.com, 2004

Overview

- OLAP
  - Introduction
  - Operations
  - Characteristics
- Storage of OLAP cubes
  - Relational vs. Multidimensional
  - Multidimensional Arrays
  - Sparse Cubes
  - Multidimensional Query Language
- Architecture
  - MOLAP, ROLAP, HOLAP

Relational Storage of OLAP Cubes

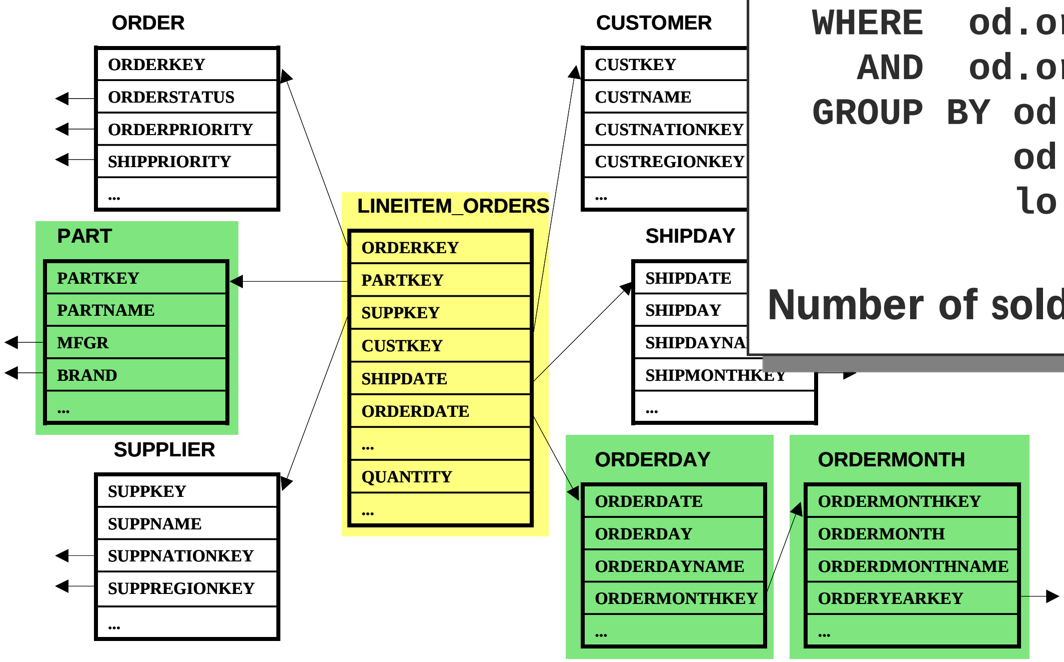
- Mapping the cube view to a star- or snowflake-schema
- Information requests of the users have to be mapped to the relational schema (see 'sequence of typical star queries')
- Result tables have to be mapped to the cube structure before they are presented to the user

Sequence of typical star queries (1)

```
INSERT INTO A1 (orderyearkey, ordermonthkey,
                partkey, sumquantity)

  SELECT od.orderyearkey, od.ordermonthkey,
         lo.partkey, SUM(lo.quantity)
  FROM   lineitem_orders lo, orderday od
  WHERE  od.orderdate = lo.orderdate
    AND  od.ordermonthkey IN (199401,199402)
  GROUP BY od.orderyearkey,
           od.ordermonthkey,
           lo.partkey;
```

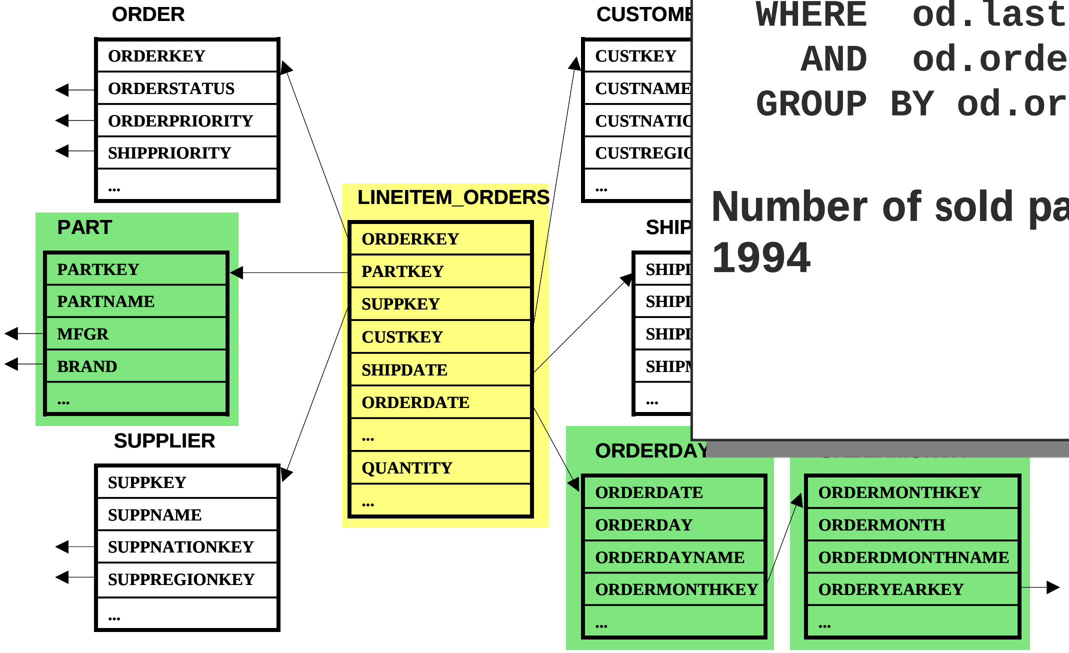
Number of sold parts in January and February 1994



Information request

Which are the top products whose number of sold pieces in the months chosen by the user compared to the respective month ago has increased most?

Sequence of typical star queries (2)




```
INSERT INTO A2 (ordermonthkey, partkey,
                sumquantity)

  SELECT od.ordermonthkey, lo.partkey,
         SUM(lo.quantity)
  FROM   lineitem_orders lo, orderday od
  WHERE  od.lastmonthdate = lo.orderdate
    AND  od.ordermonthkey IN (199401, 199402)
  GROUP BY od.ordermonthkey, lo.partkey;
```

Number of sold parts in December 1993 and January 1994

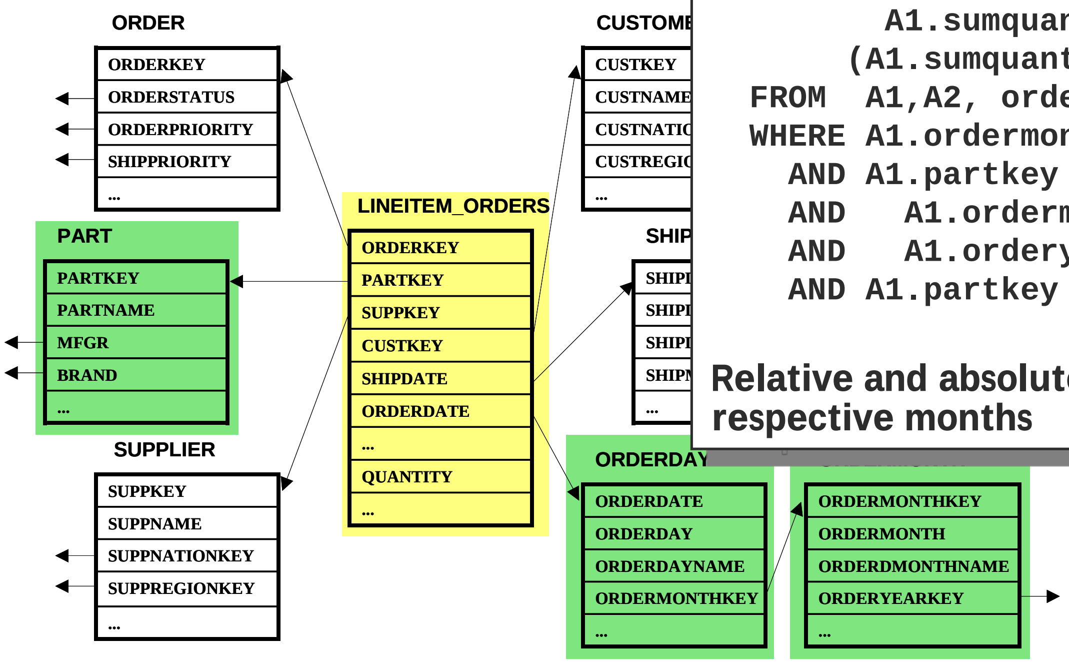
Information request

Which are the top products whose number of sold pieces in the months chosen by the user compared to the respective month ago has increased most?

Sequence of typical star queries (3)

```
INSERT INTO A3 (ordermonthkey, ordermonthname,
                orderyearkey, orderyear, partkey,
                partname, sumquantity, lmsumquantity,
                incrquantity, incrquantity2)
  SELECT om.ordermonthkey, om.ordermonthname,
         oy.orderyearkey, oy.orderyear, pa.partkey,
         pa.partname, A1.sumquantity, A2.sumquantity,
         A1.sumquantity - A2.sumquantity,
        (A1.sumquantity -A2.sumquantity)/A2.sumquantity
  FROM  A1,A2, ordermonth om, orderyear oy, part pa
  WHERE A1.ordermonthkey = A2.ordermonthkey
    AND A1.partkey = A2.partkey
    AND   A1.ordermonthkey = om.ordermonthkey
    AND   A1.orderyearkey = oy.orderyearkey
    AND A1.partkey = pa.partkey;
```

Relative and absolute increase of sold parts compared to respective months



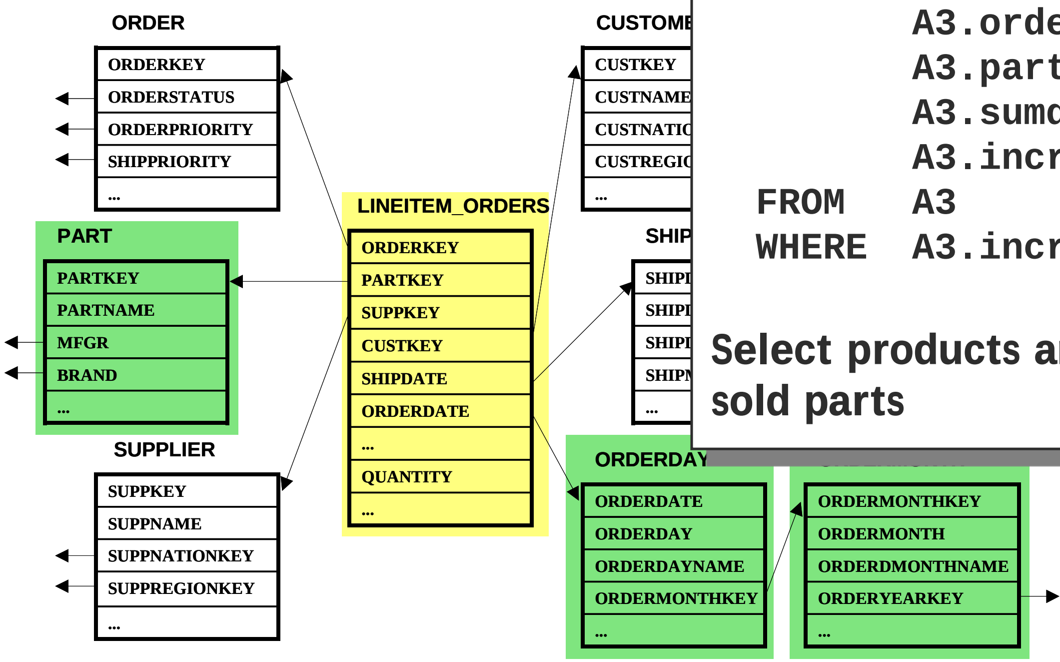
Information request

Which are the top products whose number of sold pieces in the months chosen by the user compared to the respective month ago has increased most?

Sequence of typical star queries (4)

```
INSERT INTO A4 (ordermonthkey, ordermonthname,
                orderyearkey, orderyear,
                partkey, partname,
                sumquantity, lmsumquantity,
                incrquantity, incrquantity2)
  SELECT A3.ordermonthkey, A3.ordermonthname,
         A3.orderyearkey,  A3.orderyear,
         A3.partkey,       A3.partname,
         A3.sumquantity,   A3.lmsumquantity,
         A3.incrquantity,  A3.incrquantity2
  FROM   A3
  WHERE  A3.incrquantity2 >= 98;
```

Select products and months with highest increase of sold parts



Information request

Which are the top products whose number of sold pieces in the months chosen by the user compared to the respective month ago has increased most?

Multidimensional Storage of OLAP Cubes

- Allows to directly store the cells of a data cube in a n-dimensional array
- Avoids mapping between cube view and relational schema
- May result in sparse cubes
- Multidimensional query language needed

Multidimensional Database Systems

- Allow to directly store the cells of a data cube in a n-dimensional array

	single cube	many cubes
single measure per cube		• relevant dimensionality for each measure
multiple measures per cube	• sparse dimensions likely	• direct mapping of the conceptual model

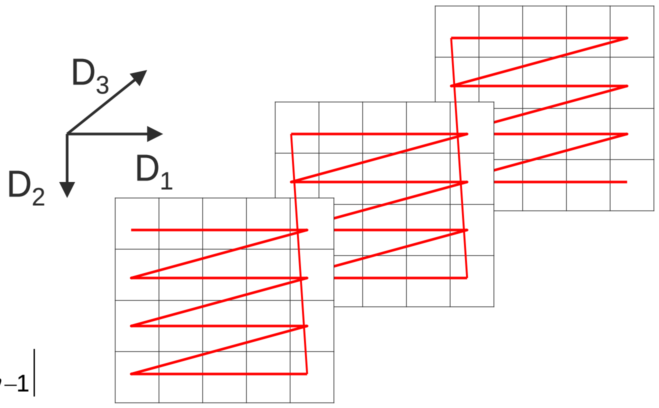
- Many proprietary implementations of storage structure
  - similar to common index structures

Multidimensional Arrays

- Dimensions $D_1, \ldots, D_n$
- Data cube with $|D_1| * |D_2| * \ldots * |D_n|$ cells
- Index of cell $(x_1, x_2, \ldots, x_n)$

$$= x_1 + (x_2 - 1) \cdot |D_1| + (x_3 - 1) \cdot |D_1| \cdot |D_2| + \ldots + (x_n - 1) \cdot |D_1| \cdot \ldots \cdot |D_{n-1}|$$

$$= 1 + \sum_{i=1}^{n} (x_i - 1) \cdot \prod_{j=1}^{i-1} |D_i|$$

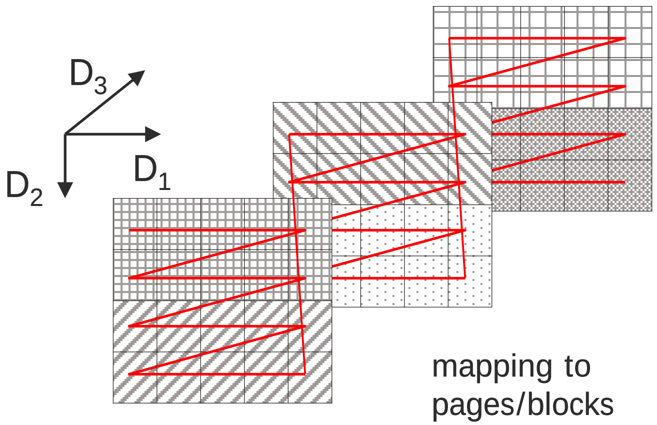


- Example
  - Dimension 1: Product
  - Dimension 2: Month
  - Which cell stores data for product C in April 2005?

A (1)	B (2)	C (3)	D (4)	E (5)	
1	2	3	4	5	Jan 05 (1)
6	7	8	9	10	Feb 05 (2)
11	12	13	14	15	Mar 05 (3)
16	17	18	19	20	Apr 05 (4)

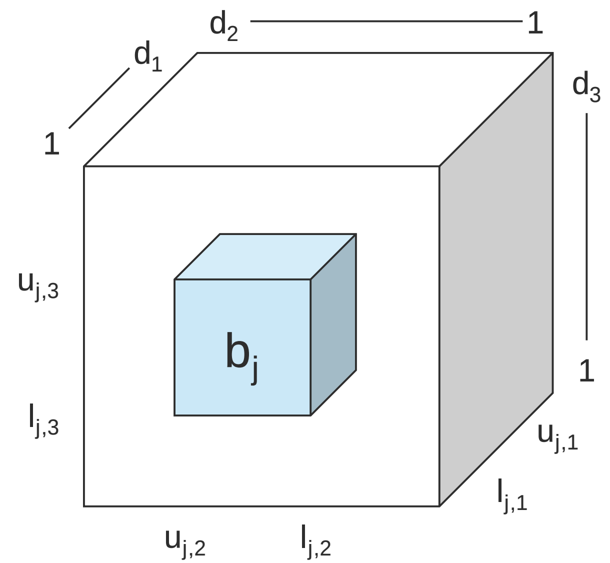
Query Processing in Multidimensional Arrays

- Query processing
  - determine index of cells
  - read pages/blocks for these cells into main memory
- Query performance depends on the number of pages to be read
- Example
  - How many blocks need to be read to get all cells on product A?
  - How many blocks need to be read to get all cells for February 2005?
- Order of dimensions is significant for query performance

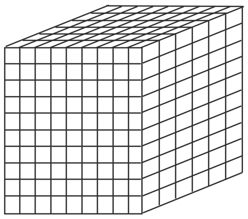


	A (1)	B (2)	C (3)	D (4)	E (5)
Jan 05 (1)	1	2	3	4	5
Feb 05 (2)	6	7	8	9	10
Mar 05 (3)	11	12	13	14	15
Apr 05 (4)	16	17	18	19	20

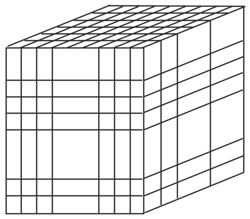
Multidimensional Partitioning

- Dimensions $D_1, ..., D_n$
- Dimension values $1 ... d_i$ for each dimension D_i.
- Partition $b_1, ..., b_m$ as
  $b_1 = [l_{1,1}:u_{1,1}, ..., l_{1,n}:u_{1,n}]$
  ...
  $b_m = [l_{m,1}:u_{m,1}, ..., l_{m,n}:u_{m,n}]$
- 

- regular partitioning
  same value range in dimension D_i for each partition b_j.



- irregular partitioning
  partition-specific value ranges



Multidimensional Partitioning

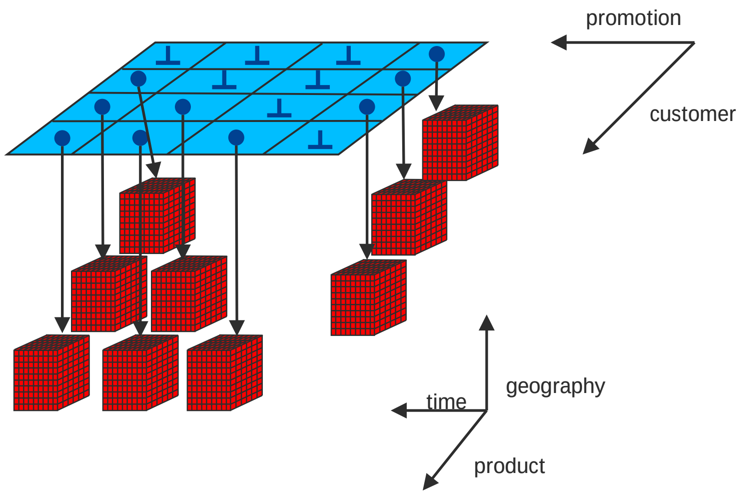
- Automatic partitioning
  - system automatically defines the partitioning
  - goals
    - identify sparse dimensions
    - efficient query processing
- Partitioning based on dimension semantics
  - e.g. partitioning according to time series
- user-defined partitioning
  - explicit specification based on
    - value ranges
    - dimensions
- Storage of partitions
  - relational: coordinates of cells are stored as primary key in a table
  - array: cells are stored in an array (as shown before)

Sparse Cubes

- A cube may contain empty cells
- Density of a cube

$$= \frac{\text{number of defined cells}}{\text{number of all cells}}$$

- N-dimensional array is efficient for dense cubes



- Sparse cubes need further optimizations
  - don't store empty pages/blocks
  - multidimensional partitioning + two storage levels
- Two storage levels
  - first level
    - index structure for sparse dimensions
    - index structures like B-trees, Grid, Hashing
  - second level
    - n-dim. array for dense dimensions
    - compressed arrays

Multidimensional Query Language

- Query language that includes specific features for multidimensional data
  - access to cubes
  - access to dimensions
  - aggregation of measure
  - restrictions on dimensions
  - selection of subcubes
  - set of functions for the manipulation of data
- No standard available
- Most tools provide queries based on the information users requested by means of a graphical user interface

- Example: MDX (MultiDimensional EXpression)
  - published in 1998
  - part of Microsofts OLE DB for OLAP
  - OLE DB provides COM interfaces for access to various data sources
  - supports the definition and manipulation of multidimensional objects and data (DML and DDL statements)

MDX

- Basic syntax

```
SELECT     [<axis_specification>
           [, <axis_specification>...]]
FROM       [<cube_specification>]
[WHERE     [<slicer_specification>]]


<axis_specification> ::= <set> ON <axis_name>
<axis_name> ::= COLUMNS | ROWS | PAGES | SECTIONS | CHAPTERS | AXIS(<index>)
```

- SELECT clause
  - determines the axis dimensions of an MDX SELECT statement
- FROM clause
  - determines which multidimensional data source is to be used when extracting data to populate the result set
- WHERE clause
  - determines which dimension or member to use as a slicer dimension
  - slicer dimension = dimension that is not assigned to an axis
  - restricts the extracting of data to a specific dimension or member

MDX: Examples

```
SELECT  { [Measures].[Unit Sales], [Measures].[Store Sales] } ON COLUMNS,
        { [Time].[1997], [Time].[1998] } ON ROWS
FROM    Sales
WHERE   ( [Store].[USA].[CA] )
```

- Specifies that
  - two measures should be presented in columns
  - values for two years should be presented in rows
  - only stores in CA should be included

- WHERE clauses

tuple

```
WHERE   ( [Route].[All], [Time].[1st half] )
WHERE   { ([Time].[1st half], [Route].[nonground]),
          ([Time].[1st half], [Route].[ground])
```

set

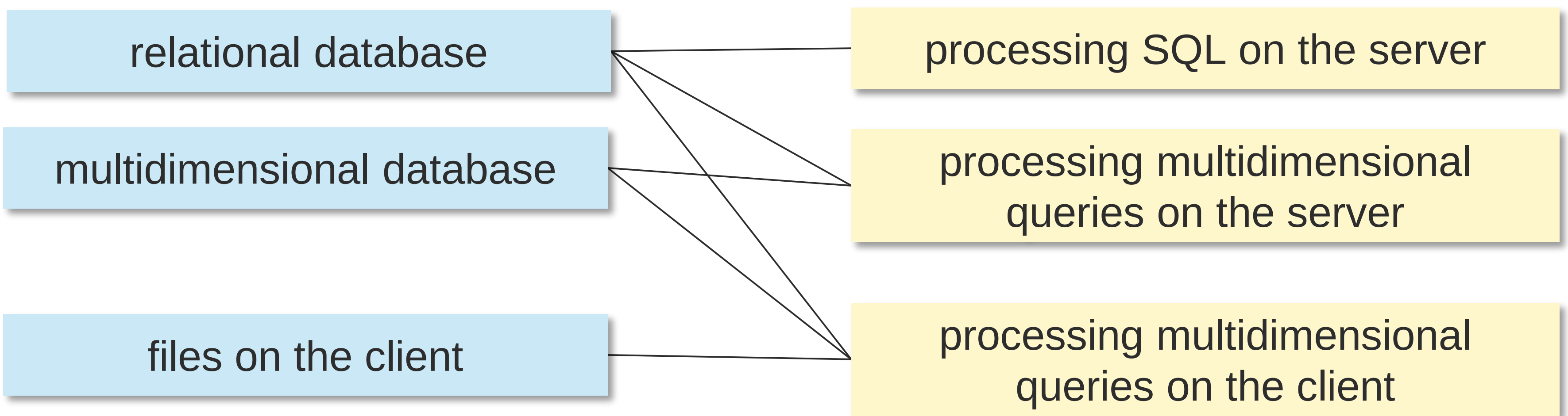
- tuple: uniquely identifies a section in the cube (subcube)
- if multiple tuples are specified (set) result cells in every tuple along the set will be aggregated

Overview

- OLAP
  - Introduction
  - Operations
  - Characteristics
- Storage of OLAP cubes
  - Relational vs. Multidimensional
  - Multidimensional Arrays
  - Sparse Cubes
  - Multidimensional Query Language
- Architecture
  - MOLAP, ROLAP, HOLAP

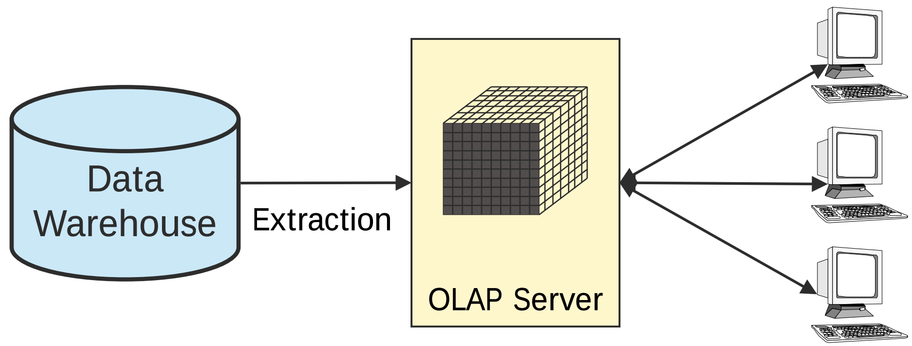
Architecture

- Different options based on
  - storage of OLAP data
  - processing of OLAP data

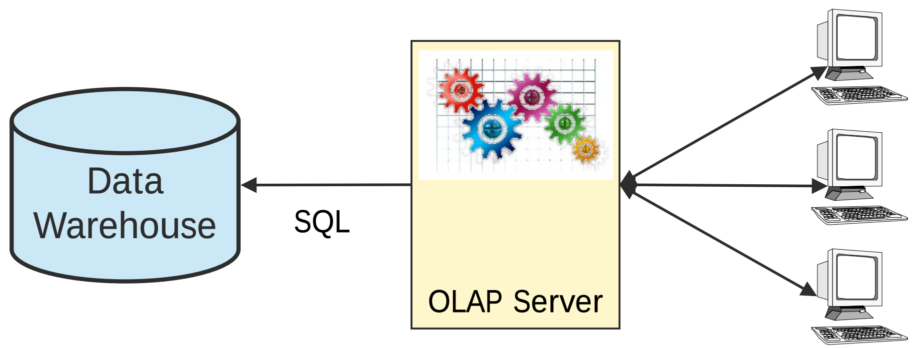


MOLAP, ROLAP, HOLAP

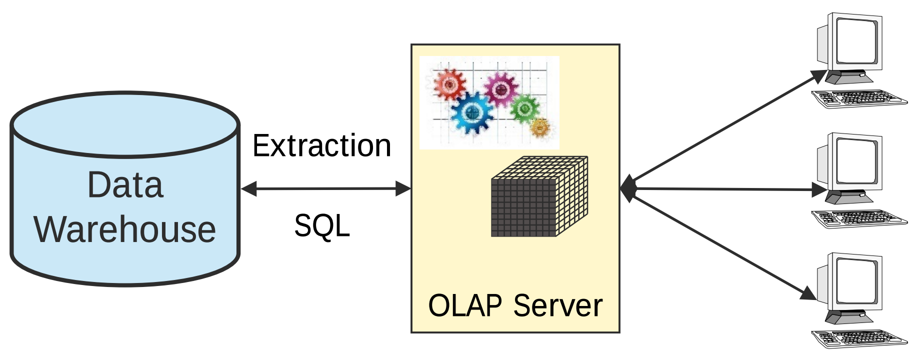
- MOLAP
  - data resides in a multidimensional DBMS
  - multidimensional engine (OLAP server) provides access



- ROLAP
  - data resides in a relational DBMS
  - OLAP server provides SQL queries



- HOLAP
  - detailed data resides in a relational DBMS
  - aggregated data resides in a multidimensional DBMS



Architecture
Comparison

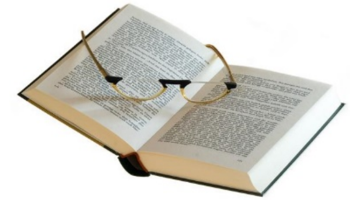
	MOLAP	ROLAP	HOLAP
Pros	• short response time • efficient storage structure	• mature relational technology • no limits on volumes of data	• short response time for aggregated data • efficient storage structure for aggregated data • no limits on volumes of data
Cons	• limited performance for large volumes of data • large volumes of data on OLAP server (detailed and aggregated) • preprocessing to provide OLAP cubes	• increased response time	• increased response time for detailed data • administration

Gartner Magic Quadrant for BI & Analytics

Summary

- OLAP
  - Technologies and tools that support (ad-hoc) analysis of multi-dimensionally aggregated data
- Basic Operations
  - Slice and Dice, Roll-up and Drill-down, Pivot
- Main characteristics of OLAP
  - Fast, Analysis, Shared, Multidimensional, Information
- Storage options
  - relational database system
  - multidimensional db (n-dimensional arrays, m-dim. query language)
- Architectural options
  - ROLAP, MOLAP, HOLAP

Papers



[CCS93] E. Codd, S. Codd, C. Salley: Providing OLAP (On-Line Analytical Processing) to User Analysts: An IT Mandate. White Paper, Arbor Software Cooperation, 1993.