Data Warehousing and OLAP

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Knowledge objectives

1. Explain the different requirements, characteristics, kinds of users and tools of a decisional DB, compared with an operational one
2. Give the definition and four characteristics of a “data warehouse”
3. Justify the usefulness of multidimensional analysis against operational databases and spreadsheets
4. Define OLAP (On-Line Analytical Processing)
5. Describe a data cube
6. Describe the most typical multidimensional operations over the cube
7. Explain the meaning of a star shape multidimensional schema
8. Enumerate the GROUPING SET clauses in SQL’99
9. Understand the different meaning of the NULL value

Understanding objectives

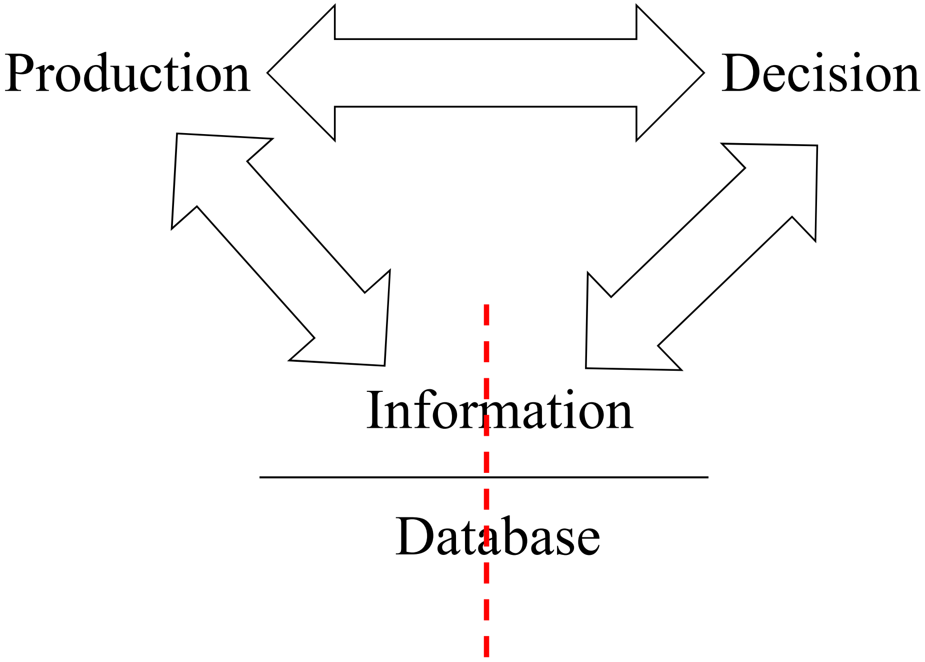
1. Translate a multidimensional UML diagram into a relational star-join schema

Application objectives

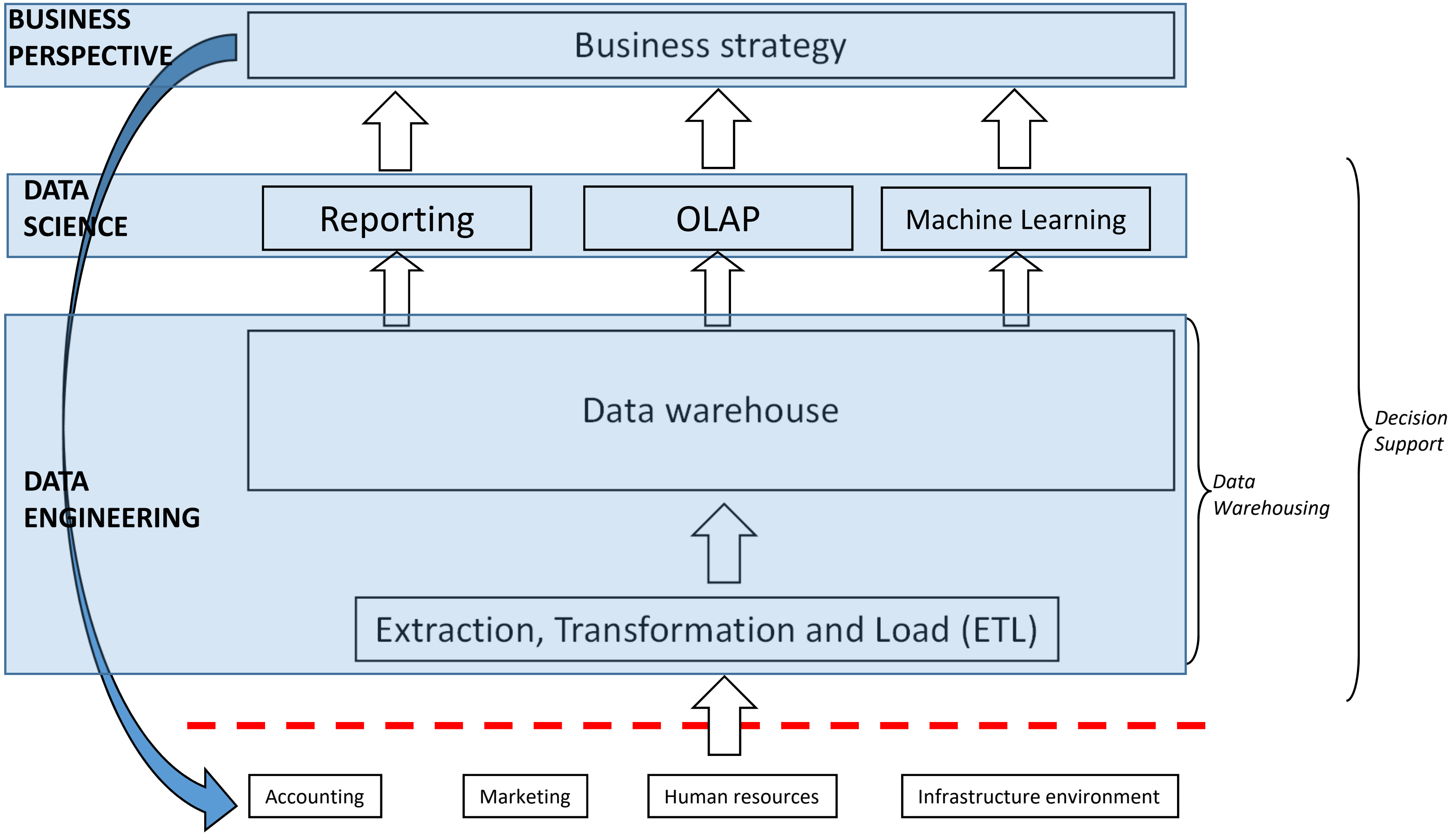
1. Given an operational conceptual diagram, define a star schema corresponding to part of its contents
2. Given a star schema and a requirement, write the corresponding star-join SQL query
3. Give the shortest query (using SQL'99 syntax) retrieving the necessary data to fill a statistical table

Operational vs Decisional

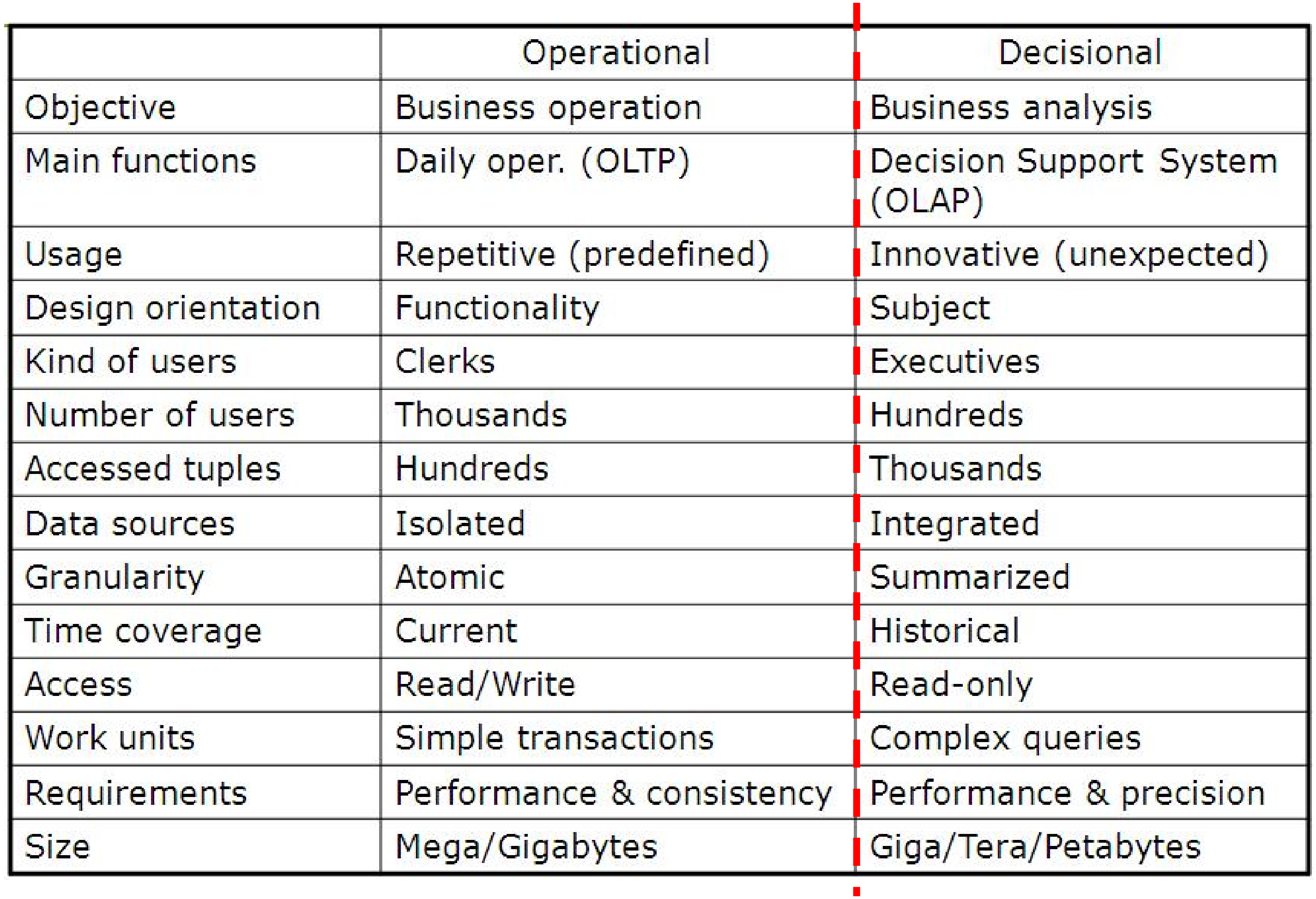
Organization subsystems



Business Intelligence (BI) Cycle



Comparison



	Operational	Decisional
Objective	Business operation	Business analysis
Main functions	Daily oper. (OLTP)	Decision Support System (OLAP)
Usage	Repetitive (predefined)	Innovative (unexpected)
Design orientation	Functionality	Subject
Kind of users	Clerks	Executives
Number of users	Thousands	Hundreds
Accessed tuples	Hundreds	Thousands
Data sources	Isolated	Integrated
Granularity	Atomic	Summarized
Time coverage	Current	Historical
Access	Read/Write	Read-only
Work units	Simple transactions	Complex queries
Requirements	Performance & consistency	Performance & precision
Size	Mega/Gigabytes	Giga/Tera/Petabytes

Data Warehouse definition

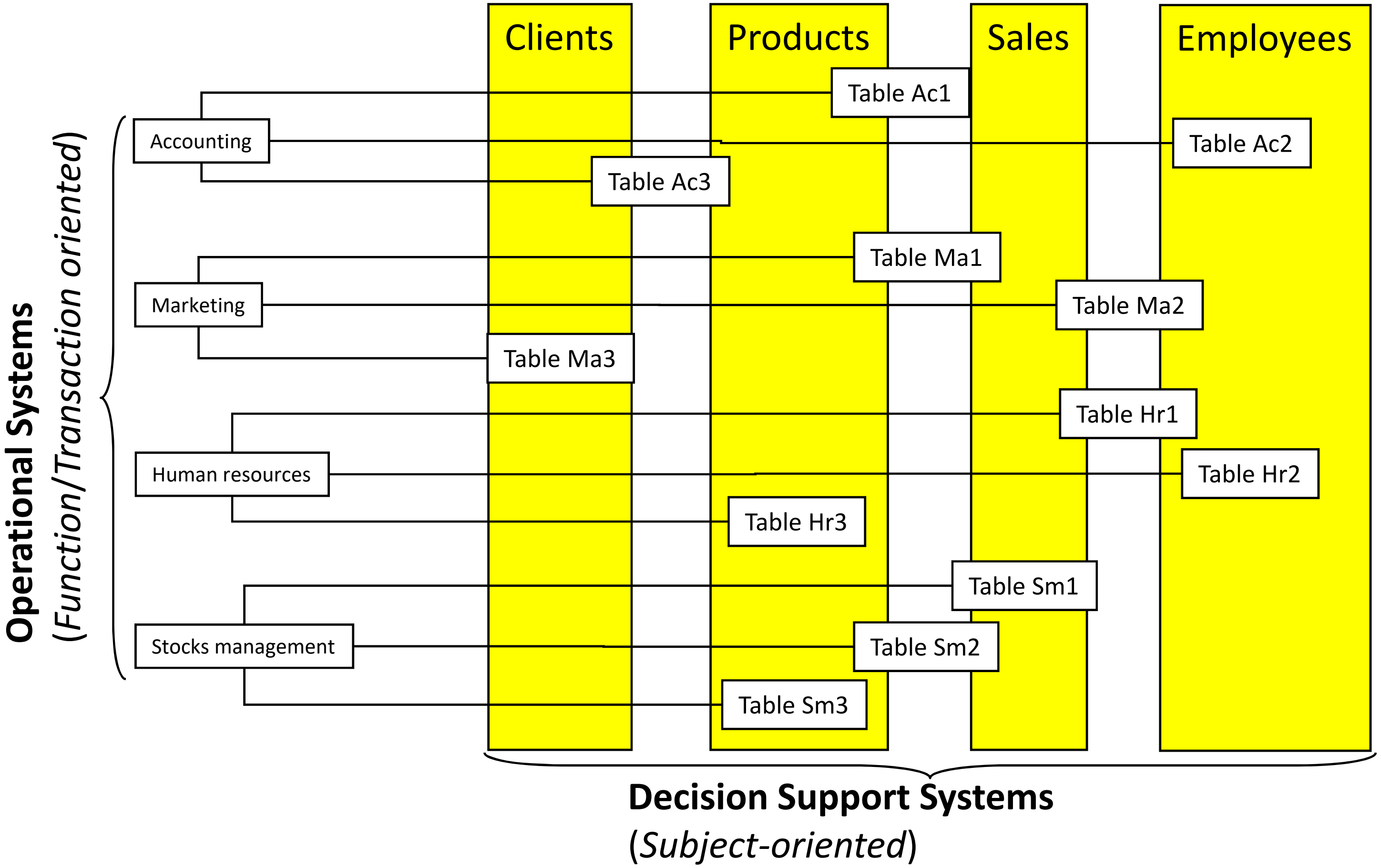
Integrated
Subject-Oriented
Non-volatile
Time-variant

Definition

"A Data Warehouse is a integrated, subject-oriented, time-variant, nonvolatile collection of data in support of management's decision-making process."

W. Inmon, 1992

Subject-oriented



Time variant and nonvolatile

Operational

Name	Salary
Jordi	1200E

Time variant and nonvolatile

Time variant (Valid Time)

Name	Salary	VT
Jordi	1000E	Jan
Jordi	1100E	Mar
Jordi	1200E	Jul

Operational

Name	Salary
Jordi	1200E

Time variant and nonvolatile

Time variant
(Valid Time)

Name	Salary	VT
Jordi	1000E	Jan
Jordi	1100E	Mar
Jordi	1200E	Jul

Operational

Name	Salary
Jordi	1200E

Nonvolatile
(Transaction Time)

Name	Salary	TT
Jordi	1000E	Jan
Jordi	900E	Mar
Jordi	1100E	Apr
Jordi	1200E	Sep

Time variant and nonvolatile

Time variant (Valid Time)

Name	Salary	VT
Jordi	1000E	Jan
Jordi	1100E	Mar
Jordi	1200E	Jul

Operational

Name	Salary
Jordi	1200E

Nonvolatile (Transaction Time)

Name	Salary	TT
Jordi	1000E	Jan
Jordi	900E	Mar
Jordi	1100E	Apr
Jordi	1200E	Sep

Data warehouse

Name	Salary	TT	VT
Jordi	1000E	Jan	Jan
Jordi	1000E	Mar	Jan
	900E		Mar
Jordi	1000E	Apr	Jan
	1100E		Mar
Jordi	1000E	Sep	Jan
	1100E		Mar
	1200E		Jul

Data warehousing

*“Data Warehous*ing *is a process, not a product, for assembling and managing data from various sources for the purpose of gaining a single, detailed view of part or all of a business.”*

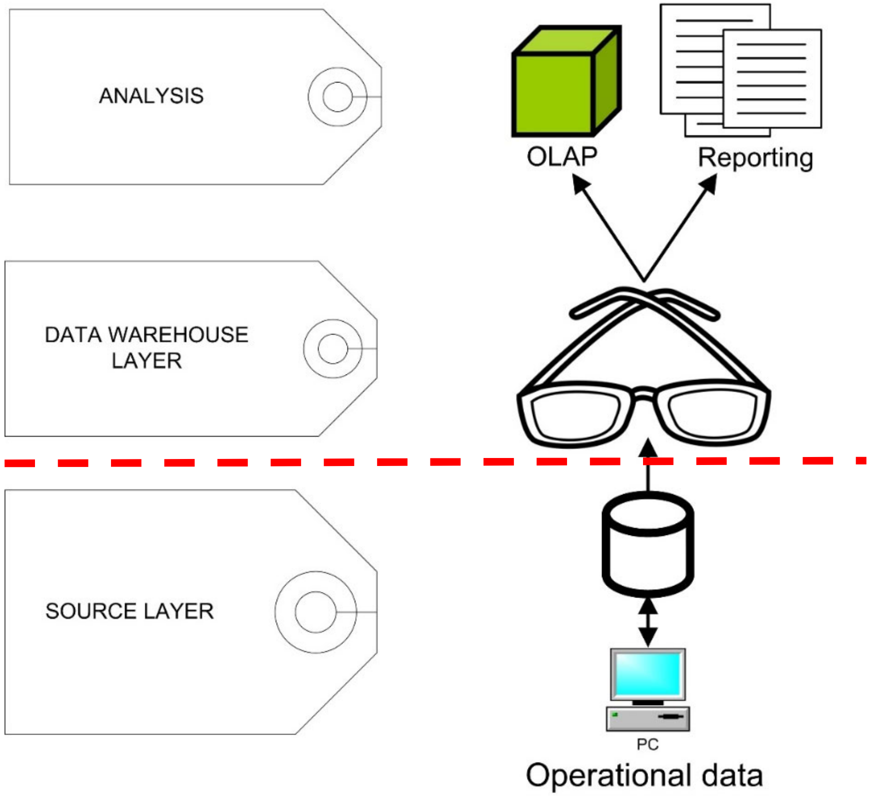
S. Gardner, 1998

Data Warehousing architectures

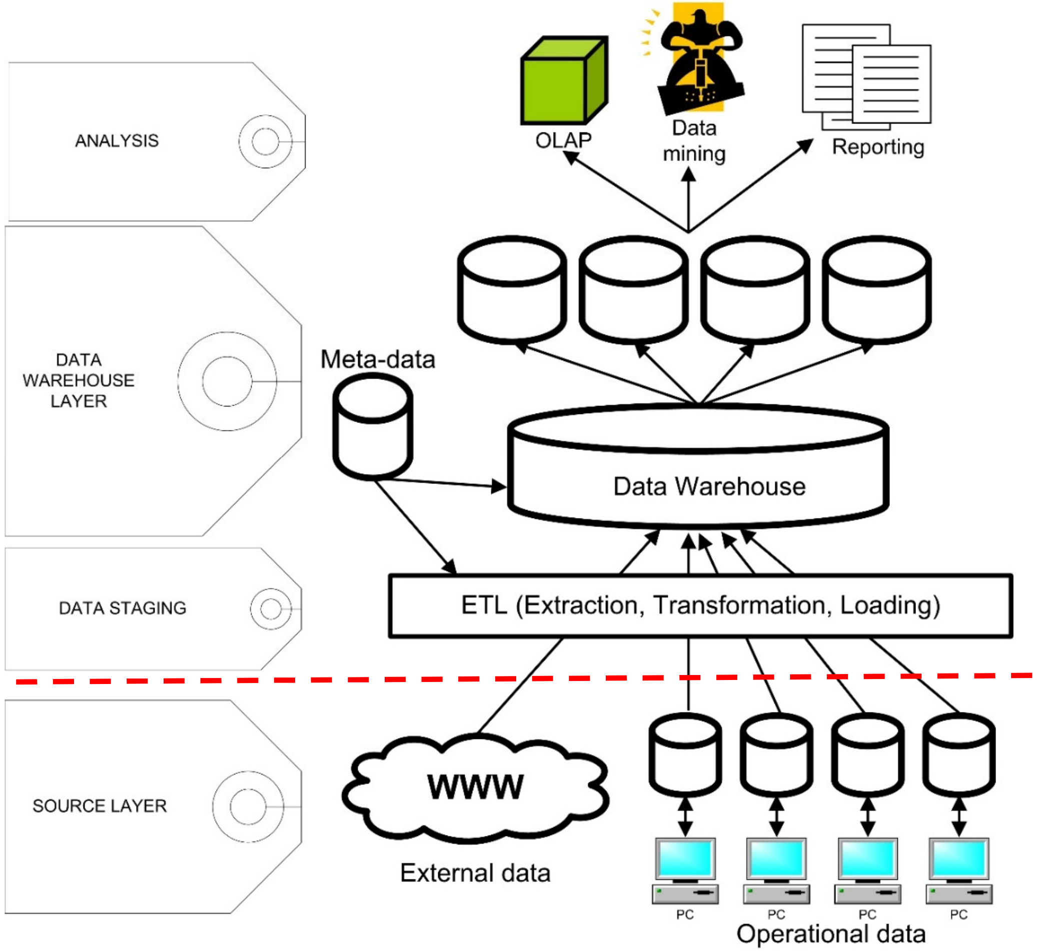
Single-layer

Two-layer

Single-layer architecture

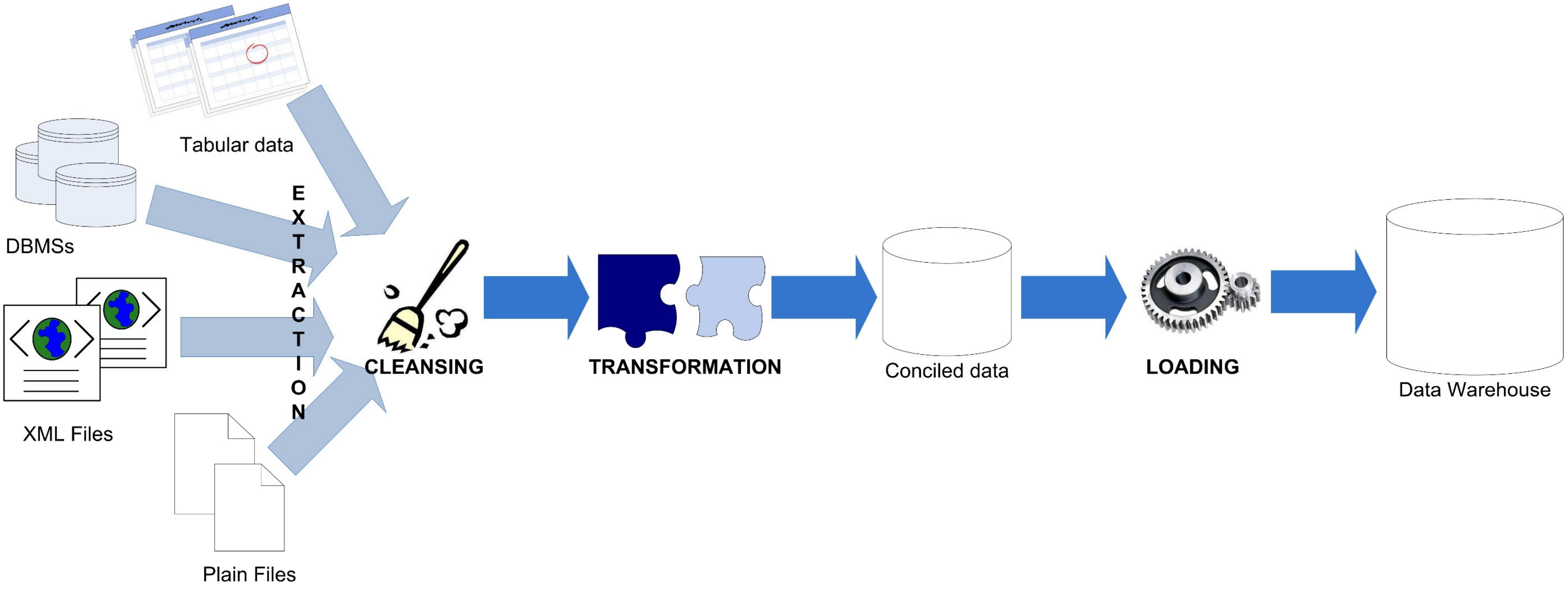


Two-layer architecture



UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

Extraction, Transformation and Load



Differences between OLTP and DW

- Decisional and not operational
  - Subject oriented
  - Huge amount of information
    - Integrate several data sources
    - Contain several versions (of both: data and schemas)
  - Can be composed by several storage systems
- Operational and not decisional
  - Redundancy is not allowed
  - Constant data actualization
  - Transactions are used (concurrency control)

OLAP definition

Spreadsheet (I)

	A	B	C	D	E	F	G	H	…
1									
2		x							
3		y							
4		z							
5									
6									
…									

SUM(B2:B4) → SUM(B2; G3; F6)

Spreadsheets (II)

- Absence of metadata
  - Rows and columns without associated meaning
  - Difficult query and interpretations

- Limited amount of data
  - MSExcel (65,000*256=16,000,000 cells)

- The position limits operations

- Aggregation hierarchies are not managed

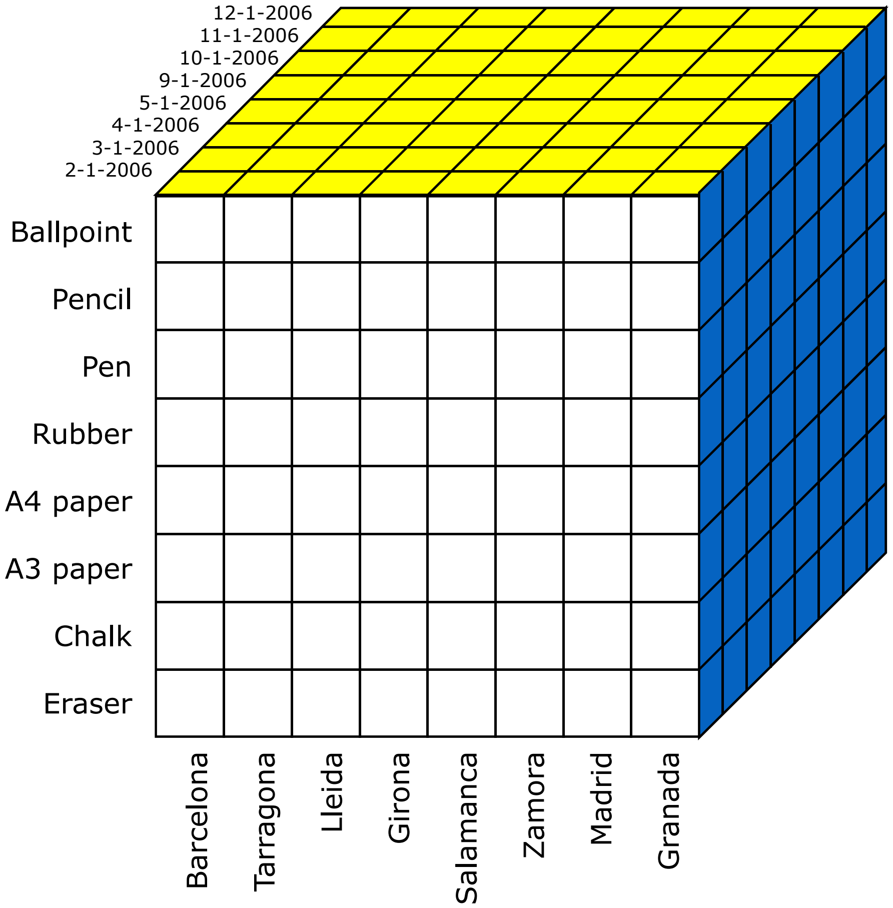
FASMI test

- Fast
- Analysis
- of Shared
- Multidimensional
- Information

Nigel Pendse, 1995

Spreadsheet	-> FAMSI
Operational DB	-> SIFAM

Cube



12-1-2006
11-1-2006
10-1-2006
9-1-2006
5-1-2006
4-1-2006
3-1-2006
2-1-2006

Ballpoint
Pencil
Pen
Rubber
A4 paper
A3 paper
Chalk
Eraser

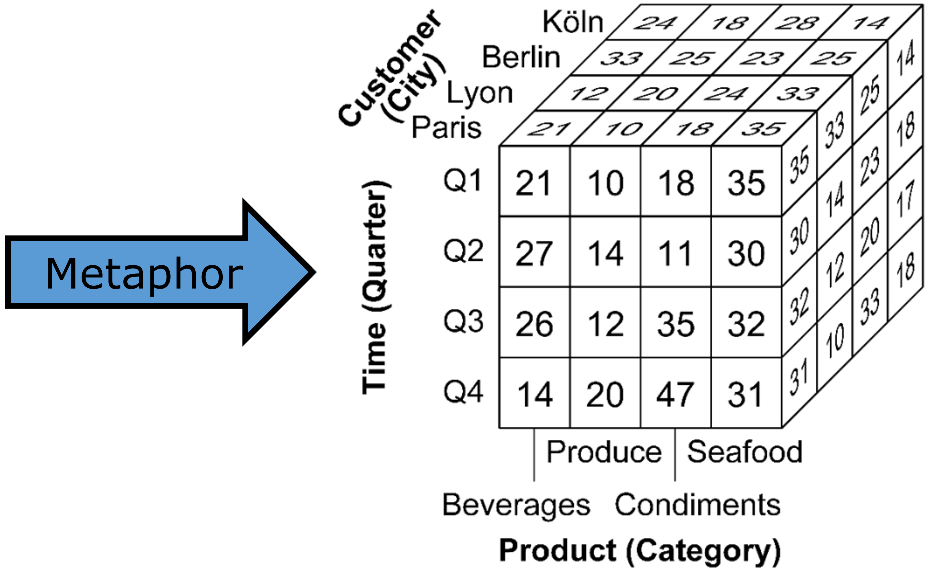
Barcelona
Tarragona
Lleida
Girona
Salamanca
Zamora
Madrid
Granada

OLAP operations

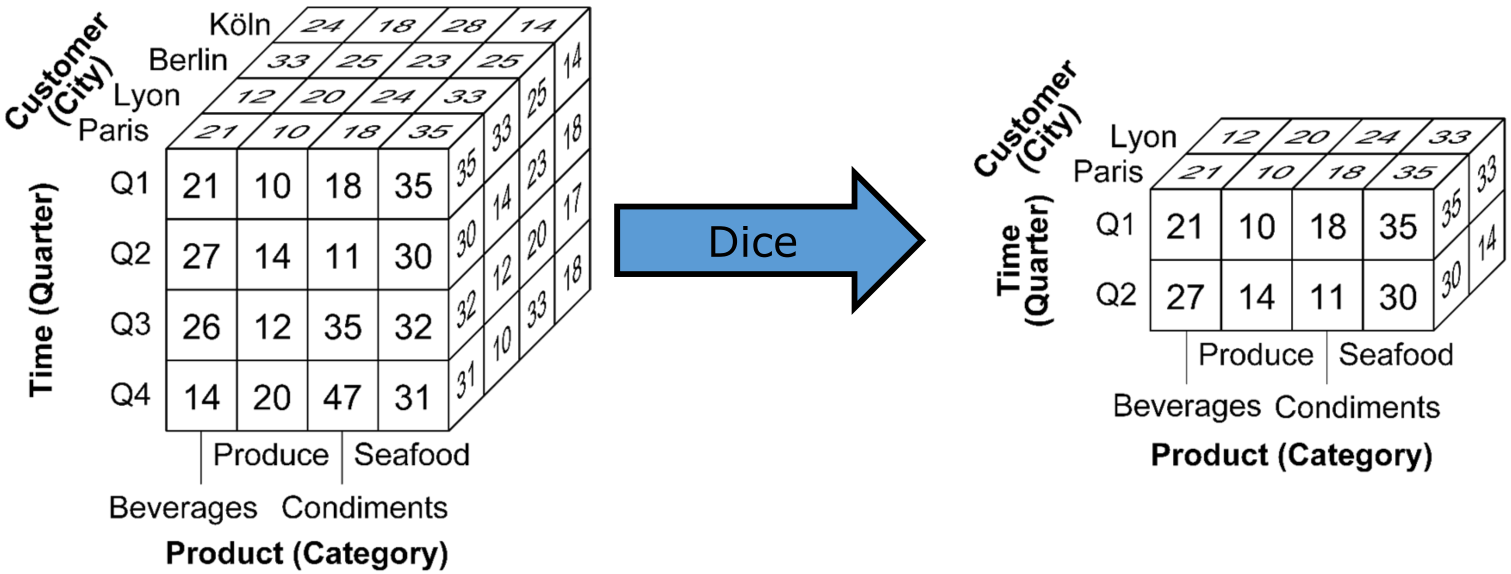
Cube view

Statistical/Cross-tab table

		Q1	Q2	Q3	Q4
Paris	Beverages	21	27	26	14
	Produce	10	14	12	20
	Condiments	18	11	35	47
	Seafood	35	30	32	31
Lyon	Beverages	12			
	Produce	20			
	Condiments	24			
	Seafood	33	14	12	10
Berlin	Beverages	33			
	Produce	25			
	Condiments	23			
	Seafood	25	23	20	33
Köln	Beverages	24			
	Produce	18			
	Condiments	28			
	Seafood	14	18	17	18

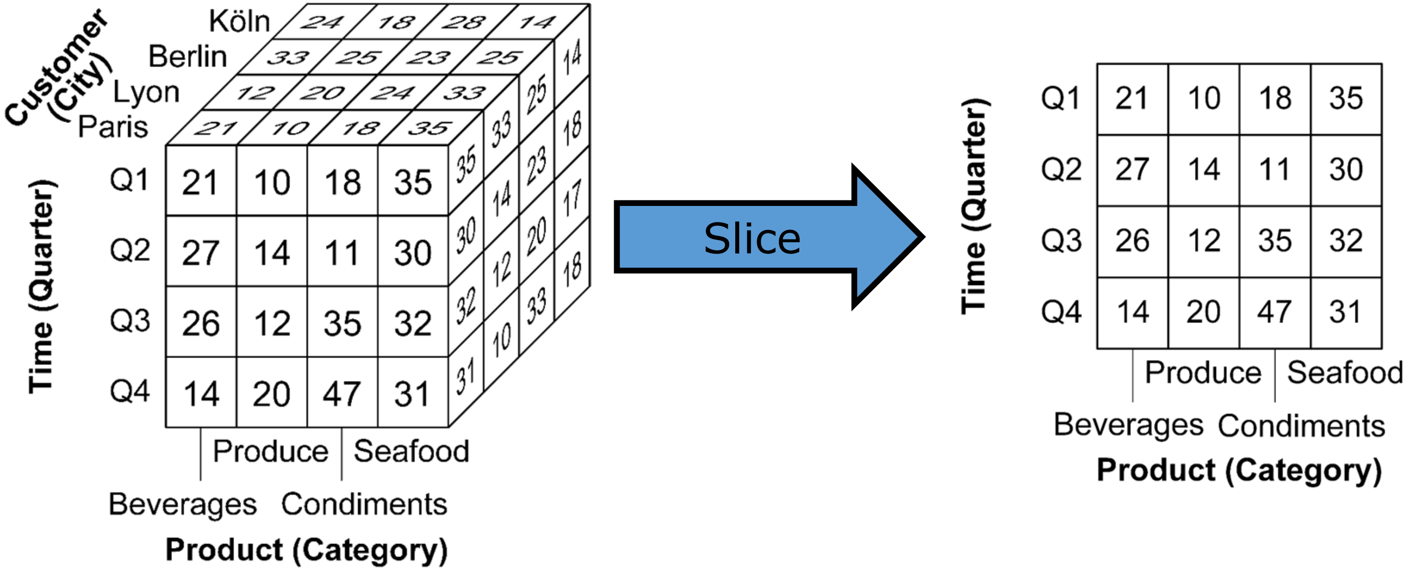


Dice



A. Vaisman & E.Zimanyi

Slice



Time (Quarter)	Beverages	Produce	Condiments	Seafood
Q1	21	10	18	35
Q2	27	14	11	30
Q3	26	12	35	32
Q4	14	20	47	31

Customer (City)	Beverages	Produce	Condiments	Seafood
Köln	24	18	28	14
Berlin	33	25	23	25
Lyon	12	20	24	33
Paris	21	10	18	35

Time (Quarter)	Paris	Lyon	Berlin	Köln
Q1	35	33	25	14
Q2	30	14	23	18
Q3	32	12	20	17
Q4	31	10	33	18

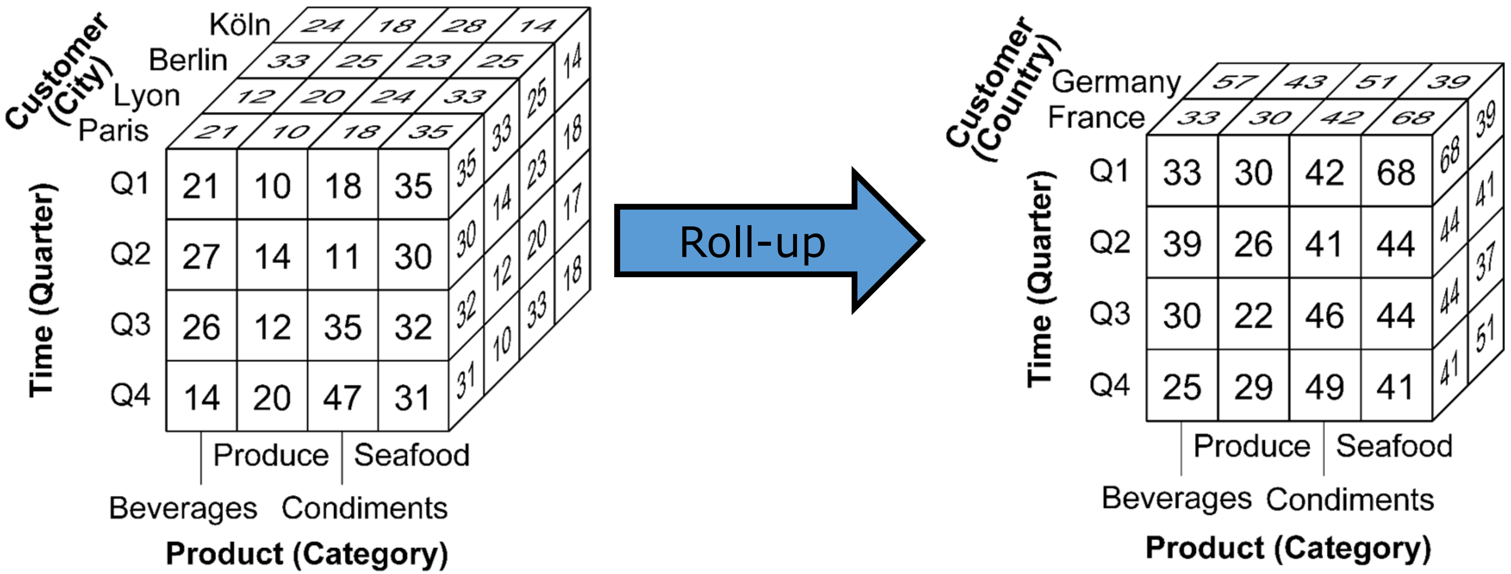
Slice

Time (Quarter)	Beverages	Produce	Condiments	Seafood
Q1	21	10	18	35
Q2	27	14	11	30
Q3	26	12	35	32
Q4	14	20	47	31

Product (Category)

A. Vaisman & E.Zimanyi

Roll-up (I)



	Beverages	Produce	Condiments	Seafood
Köln	24	18	28	14
Berlin	33	25	23	25
Lyon	12	20	24	33
Paris	21	10	18	35

	Paris	Lyon	Berlin	Köln
Q1	35	33	25	14
Q2	30	14	23	18
Q3	32	12	20	17
Q4	31	10	33	18

	Beverages	Produce	Condiments	Seafood
Q1	21	10	18	35
Q2	27	14	11	30
Q3	26	12	35	32
Q4	14	20	47	31

Roll-up

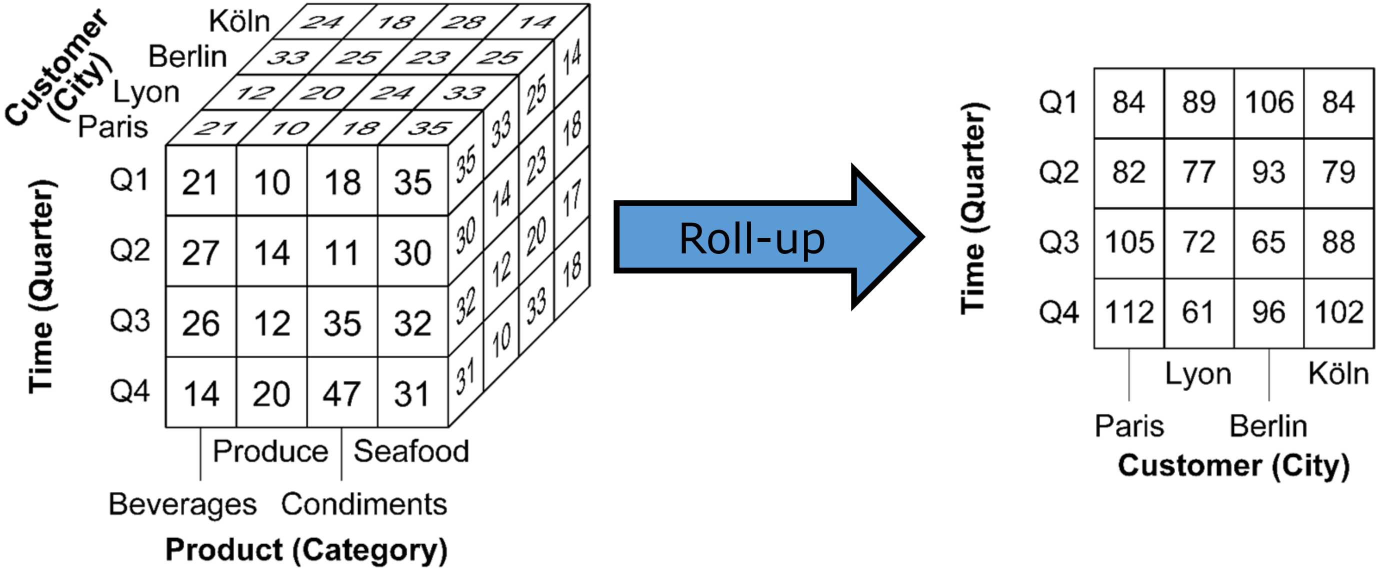
	Beverages	Produce	Condiments	Seafood
Germany	57	43	51	39
France	33	30	42	68

	France	Germany
Q1	68	39
Q2	44	41
Q3	44	37
Q4	41	51

	Beverages	Produce	Condiments	Seafood
Q1	33	30	42	68
Q2	39	26	41	44
Q3	30	22	46	44
Q4	25	29	49	41

A. Vaisman & E.Zimanyi

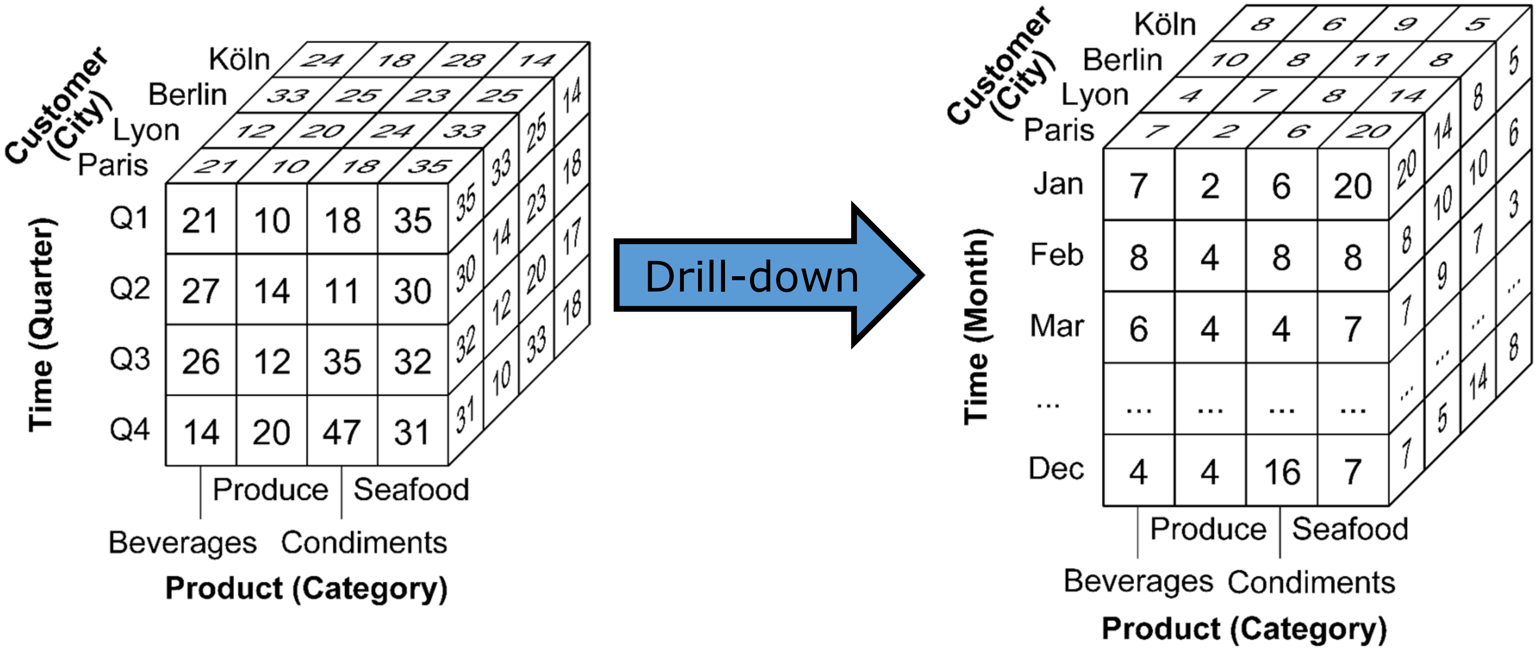
Roll-up (II)



Time (Quarter)	Paris	Lyon	Berlin	Köln
Q1	84	89	106	84
Q2	82	77	93	79
Q3	105	72	65	88
Q4	112	61	96	102

A. Vaisman & E.Zimanyi

Drill-down



	Beverages	Produce	Condiments	Seafood
Q1	21	10	18	35
Q2	27	14	11	30
Q3	26	12	35	32
Q4	14	20	47	31

	Beverages	Produce	Condiments	Seafood
Jan	7	2	6	20
Feb	8	4	8	8
Mar	6	4	4	7
...	...	...	...	...
Dec	4	4	16	7

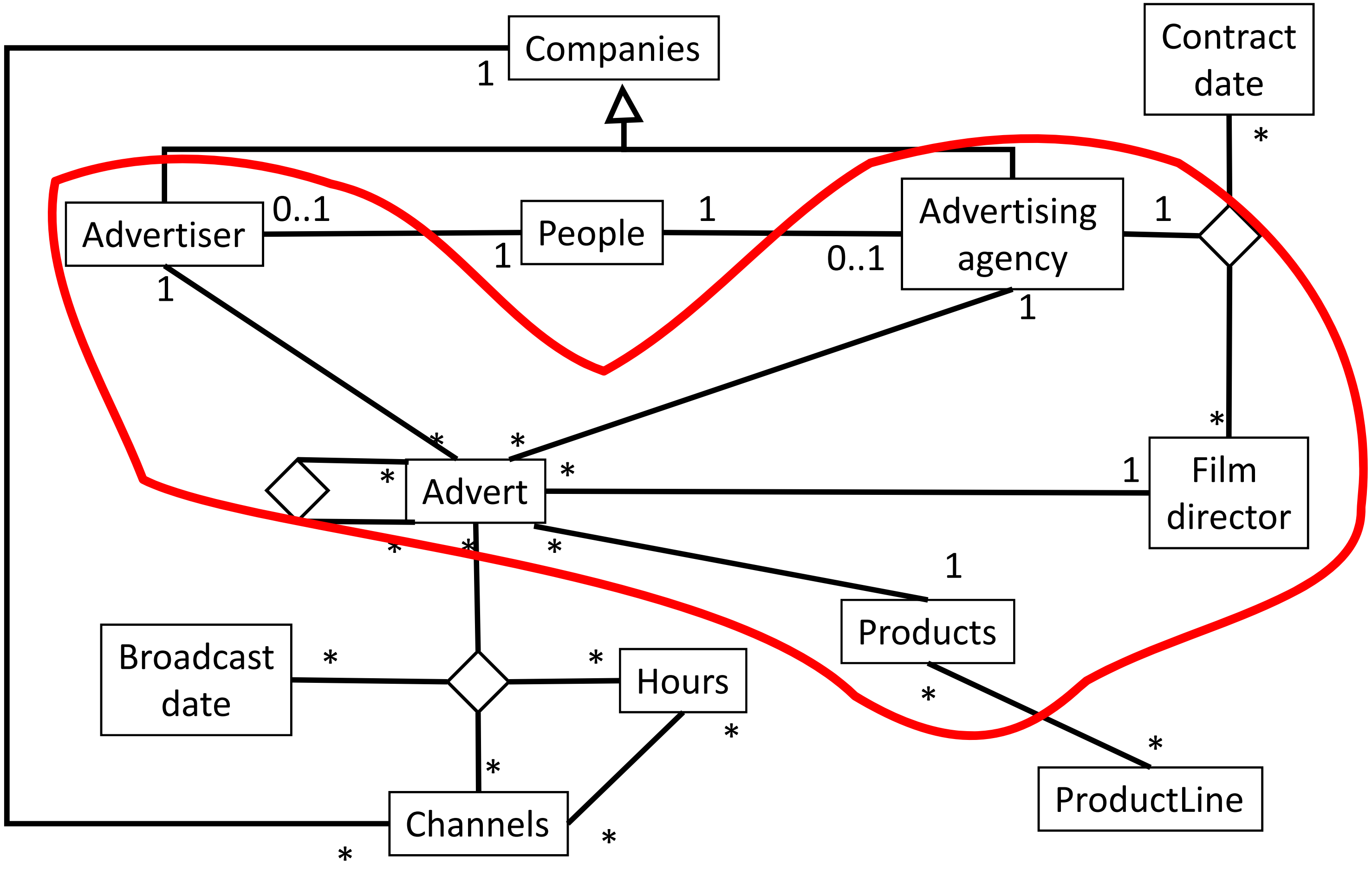
A. Vaisman & E.Zimanyi

Conceptual multidimensional schema

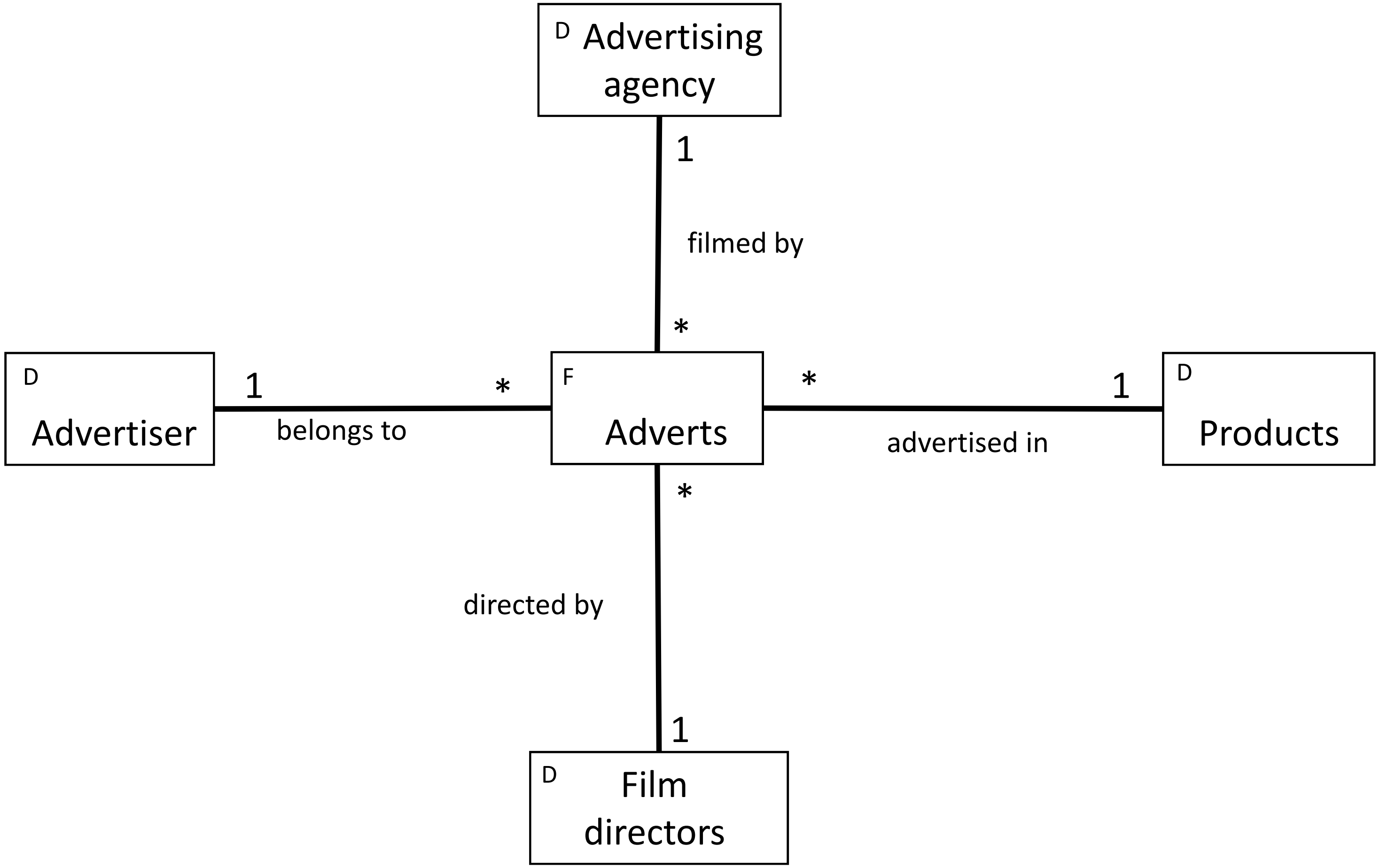
Characteristics of transactional modeling

Advantages	Disadvantages
Reduces the amount of redundant data	Degrades response time in front of queries
Eliminates the need to modify many records because of one modification, very efficient if data change very often	Ease to make mistakes if the user is not an expert in computer science

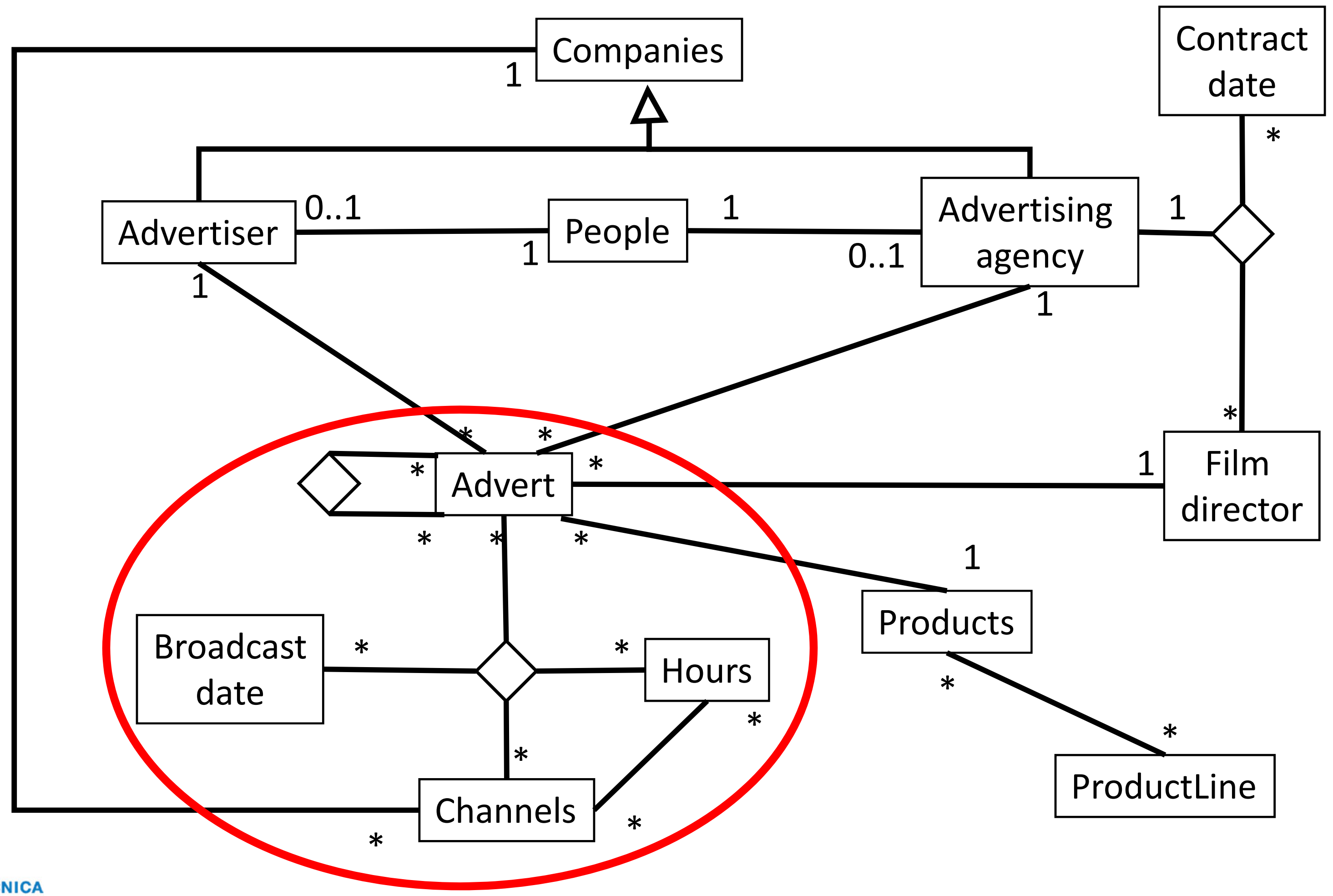
Example of transactional modeling (UML)



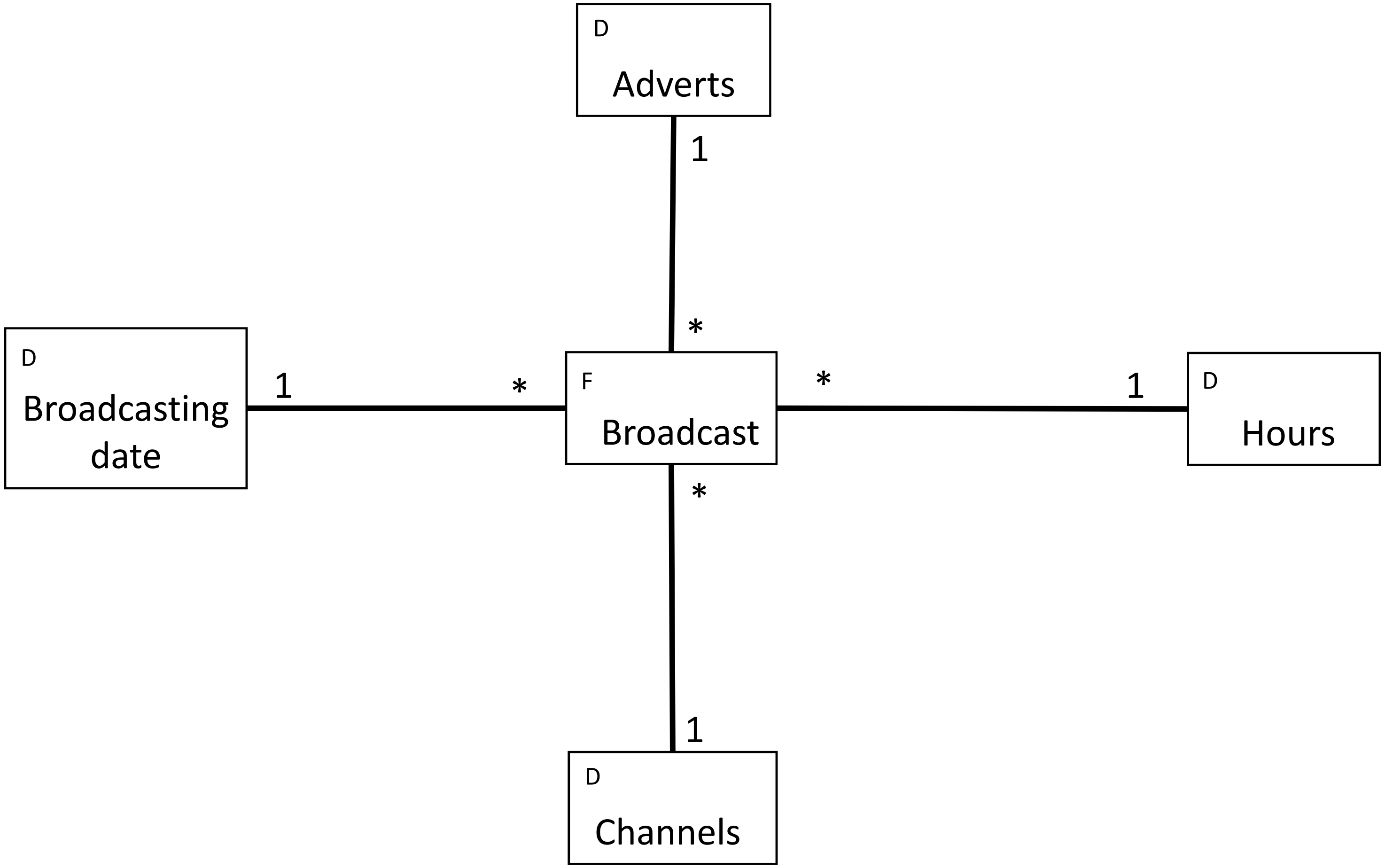
Star schema (I)



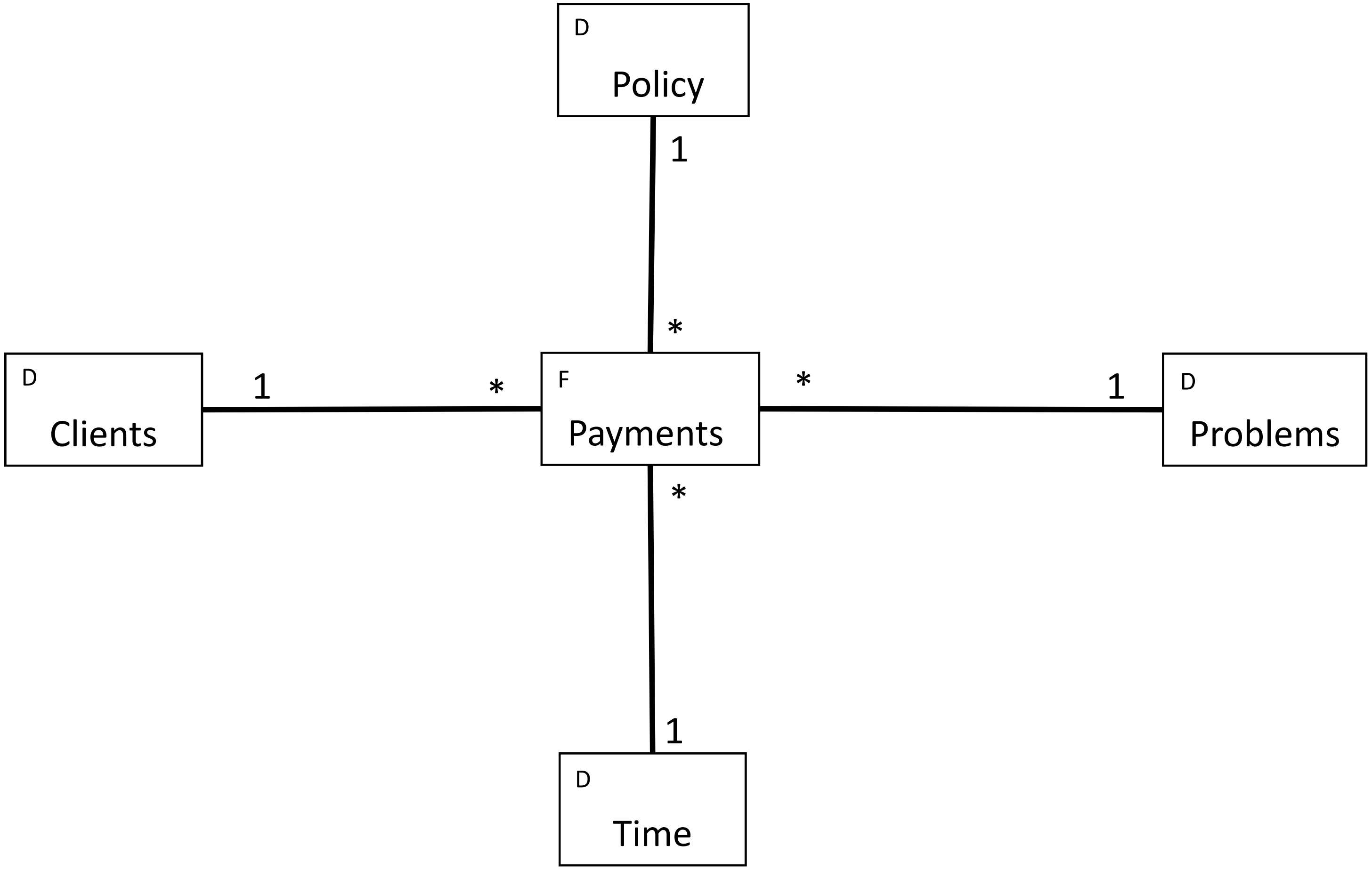
Example of transactional modeling (UML)



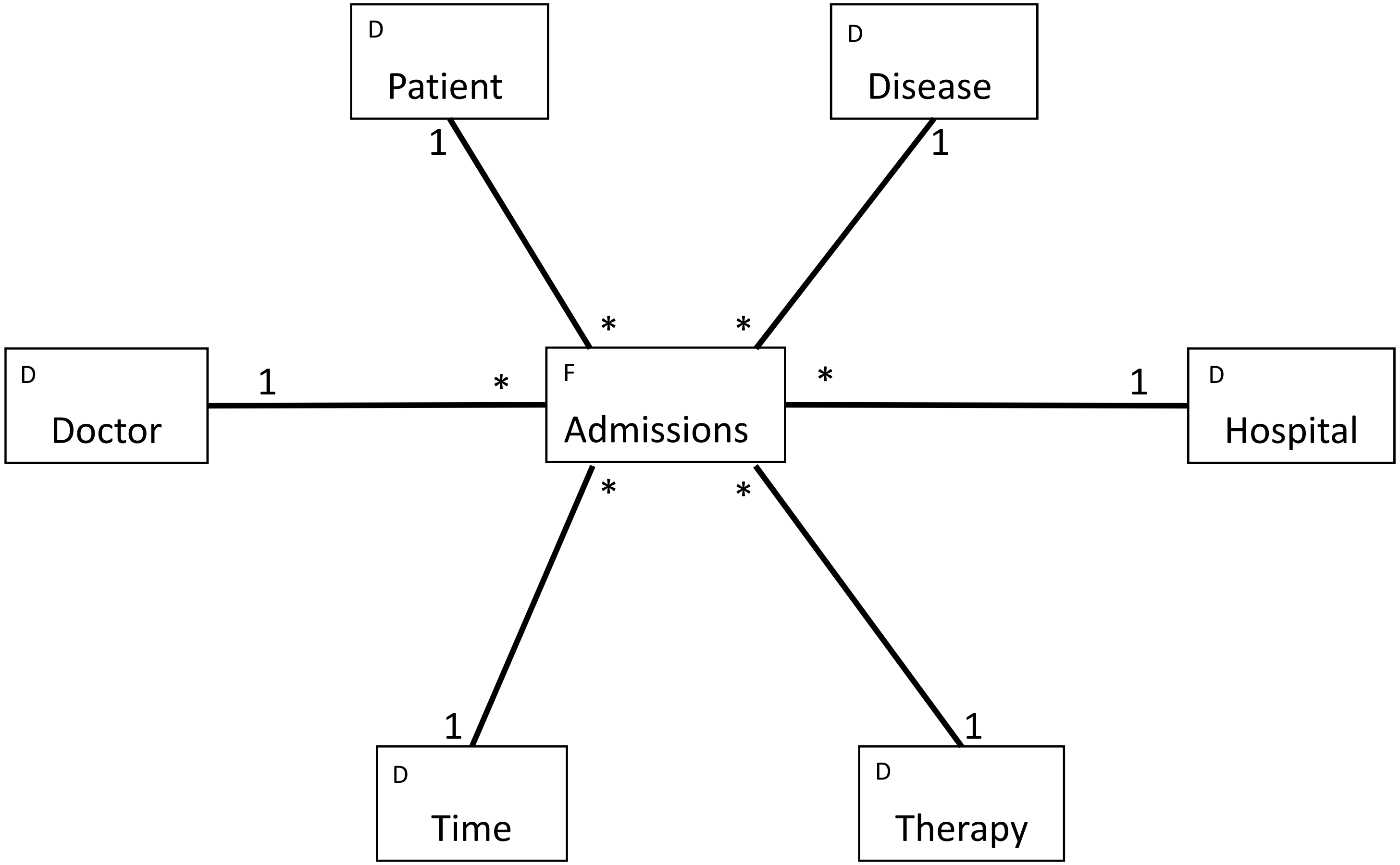
Star schema (II)



Star schema (III)

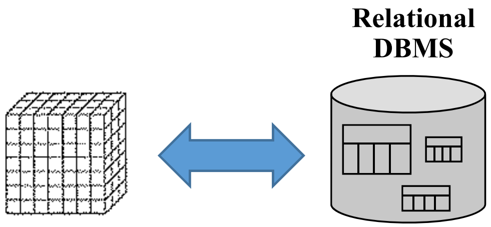


Star schema (IV)



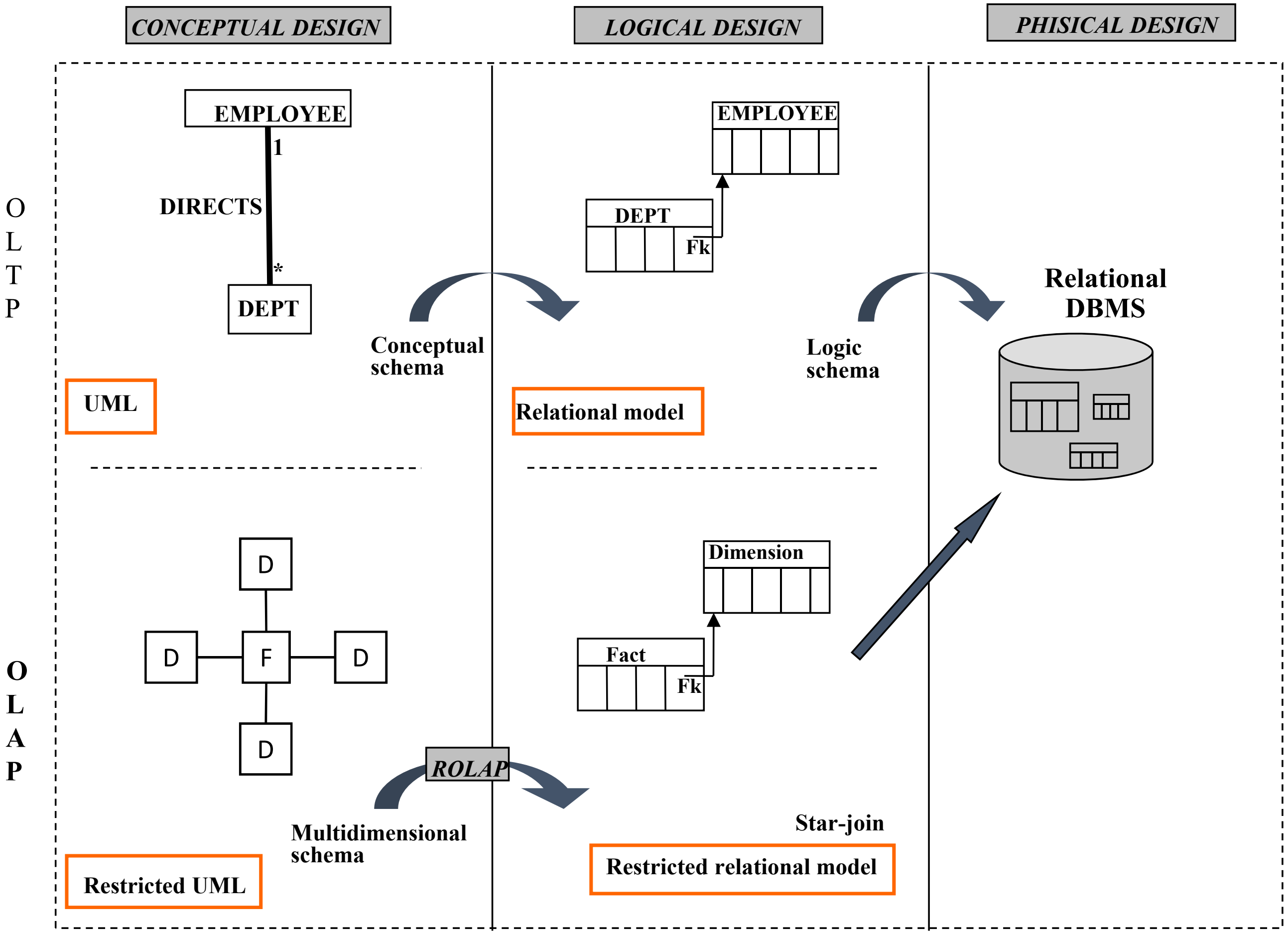
Logical Multidimensional schema

ROLAP: Characteristics

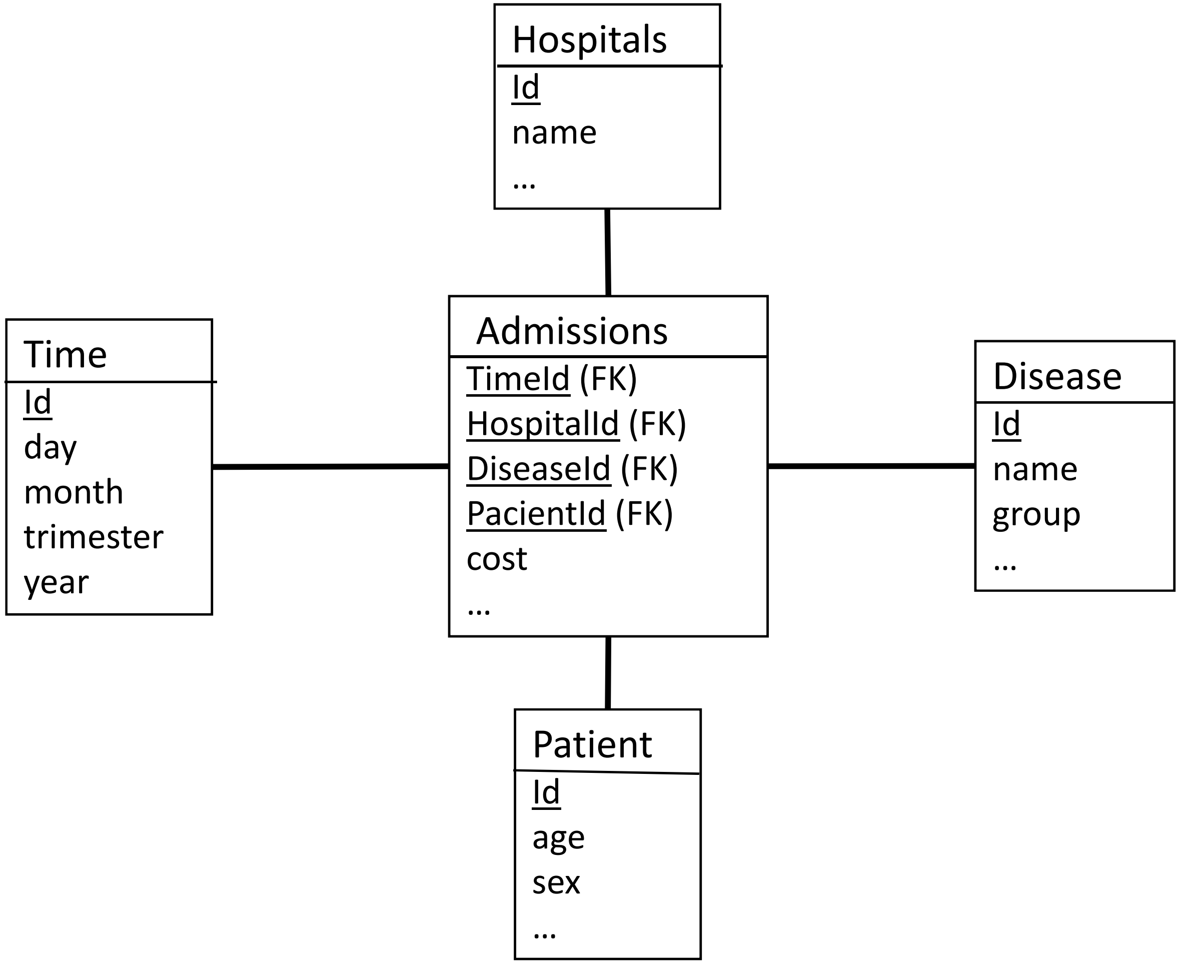


- Relational DBMS with multidimensional views
    - Two levels: Storage and Translation
- Use standard SQL
    - Easy to obtain
    - Independent of the DBMS
- Performance issues
    - Relational DBMS conceived for OLTP (later improved/adapted)
        - Conceptual OLAP operations are missing
    - Generates too many joins
        - Uses specific techniques

Comparison of design steps



ROLAP: Star-join schema



Hospitals
Id
name
…

Time
Id
day
month
trimester
year

Admissions
TimeId (FK)
HospitalId (FK)
DiseaseId (FK)
PacientId (FK)
cost
…

Disease
Id
name
group
…

Patient
Id
age
sex
…

ROLAP: Cube-Query

SELECT d_1.attr, …, d_n.attr, F(f.Measure$_1$), …

FROM Fact f, Dimension$_1$ d_1, …, Dimension$_n$ d_n

WHERE f.key$_1$ = d_1.ID AND … AND f.key$_n$= d_n.ID AND <slice-dice condition>

GROUP BY d_1.attr, …, d_n.attr

ORDER BY d_1.attr, …, d_n.attr

```
SELECT h.name AS Hospital , t.month AS Month, AVG(cost)
FROM Admissions a, Hospitals h, Time t
WHERE a.HospitalId=h.Id AND a.TimeId=t.id AND t.year=2024
GROUP BY h.name, t.month
ORDER BY h.name, t.month
```

ROLAP: Results table

Hospital	Month	Average Cost
Duran i Reinals	January’24	3300
Duran i Reinals	February’24	4500
Duran i Reinals	…	…
Bellvitge	January’24	180
Bellvitge	February’24	300
Bellvitge	…	…

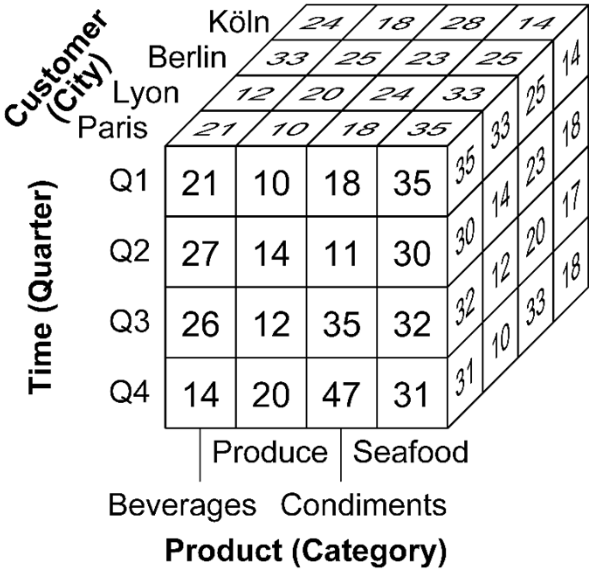
SQL Extension

GROUPING SETS

ROLLUP

CUBE

Cross-tab view



Statistical/Cross-tab table

		Q1	Q2	Q3	Q4
Paris	Beverages	21	27	26	14
	Produce	10	14	12	20
	Condiments	18	11	35	47
	Seafood	35	30	32	31
Lyon	Beverages	12			
	Produce	20			
	Condiments	24			
	Seafood	33	14	12	10
Berlin	Beverages	33			
	Produce	25			
	Condiments	23			
	Seafood	25	23	20	33
Köln	Beverages	24			
	Produce	18			
	Condiments	28			
	Seafood	14	18	17	18

Marginal values (I)

		Q1	Q2	Q3	Q4
Paris	Beverages	21	27	26	14
	Produce	10	14	12	20
	Condiments	18	11	35	47
	Seafood	35	30	32	31
Lyon	Beverages	12			
	Produce	20			
	Condiments	24			
	Seafood	33	14	12	10
Berlin	Beverages	33			
	Produce	25			
	Condiments	23			
	Seafood	25	23	20	33
Köln	Beverages	24			
	Produce	18			
	Condiments	28			
	Seafood	14	18	17	18

		Q1	Q2	Q3	Q4	**Total**
Paris	Beverages	21	27	26	14	**88**
	Produce	10	14	12	20	**56**
	Condiments	18	11	35	47	**111**
	Seafood	35	30	32	31	**128**
	Total	**84**	**82**	**105**	**112**	**383**
Lyon	Beverages	12				
	Produce	20				
	Condiments	24				
	Seafood	33	14	12	10	**69**
	Total	**89**				
Berlin	Beverages	33				
	Produce	25				
	Condiments	23				
	Seafood	25	23	20	33	**101**
	Total	**106**				
Köln	Beverages	24				
	Produce	18				
	Condiments	28				
	Seafood	14	18	17	18	**67**
	Total	**84**				
Total		**363**				

Marginal values (II)

Sales	Catalonia		
	January23	February23	All
Ballpoint	275827	290918	566745
Rubber	784172	918012	1702184
All	1059999	1208930	2268929

SQL-92: Multidimensional query

Sales	Catalonia		
	January23	February23	All
Ballpoint	275827	290918	566745
Rubber	784172	918012	1702184
All	1059999	1208930	2268929

```
SELECT d1.itemName, d2.region, d3.month, SUM(f.items)
FROM Sales f, Product d1, Place d2, Time d3
WHERE f.IDProduct=d1.ID AND f.IDPlace=d2.ID
          AND f.IDTime=d3.ID
          AND d1.itemName IN (‘Ballpoint’,’Rubber’)
          AND d2.region=’Catalonia’
          AND d3.month IN (’January23’,’February23’)
GROUP BY d1.itemName, d2.region, d3.month
UNION
SELECT d1.itemName, d2.region, ‘All’, SUM(f.items)
FROM Sales f, Product d1, Place d2, Time d3
WHERE f.IDProduct=d1.ID AND f.IDPlace=d2.ID
          AND f.IDTime=d3.ID
          AND d1.itemName IN (‘Ballpoint’,’Rubber’)
          AND d2.region=’Catalonia’
          AND d3.month IN (’January23’,’February23’)
GROUP BY d1.itemName, d2.region
UNION
SELECT ‘All’, d2.region, d3.month, SUM(f.items)
FROM Sales f, Product d1, Place d2, Time d3
WHERE f.IDProduct=d1.ID AND f.IDPlace=d2.ID
          AND f.IDTime=d3.ID
          AND d1.itemName IN (‘Ballpoint’,’Rubber’)
          AND d2.region=’Catalonia’
          AND d3.month IN (’January23’,’February23’)
GROUP BY d2.region, d3.month
UNION
SELECT ‘All’, d2.region, ‘All’ SUM(f.items)
FROM Sales f, Product d1, Place d2, Time d3
WHERE f.IDProduct=d1.ID AND f.IDPlace=d2.ID
          AND f.IDTime=d3.ID
          AND d1.itemName IN (‘Ballpoint’,’Rubber’)
          AND d2.region=’Catalonia’
          AND d3.month IN (’January23’,’February23’)
GROUP BY d2.region
ORDER BY d1.itemName, d2.region, d3.month;
```

SQL-99: GROUPING SETS (I)

```
SELECT d1.itemName, d2.region, d3.month, SUM(f.items)
FROM Sales f, Product d1, Place d2, Time d3
WHERE f.IDProduct=d1.ID AND f.IDPlace=d2.ID
          AND f.IDTime=d3.ID
          AND d1.itemName IN (‘Ballpoint’,’Rubber’)
          AND d2.region=’Catalonia’
          AND d3.month IN (‘January23’,’February23’)
GROUP BY GROUPING SETS ((d1.itemName, d2.region, d3.month),
                         (d1.itemName, d2.region),
                         (d2.region, d3.month)
                         (d2.region))
ORDER BY d1.itemName, d2.region, d3.month;
```

Sales	Catalonia		
	January23	February23	All
Ballpoint	275827	290918	566745
Rubber	784172	918012	1702184
All	1059999	1208930	2268929

https://www.postgresql.org/docs/current/queries-table-expressions.html#QUERIES-GROUPING-SETS

SQL-99: GROUPING SETS (II)

```
SELECT d1.itemName, d2.region, d3.month, SUM(f.items)
FROM Sales f, Product d1, Place d2, Time d3
WHERE f.IDProduct=d1.ID AND f.IDPlace=d2.ID
          AND f.IDTime=d3.ID
          AND d1.itemName IN (‘Ballpoint’,’Rubber’)
          AND d2.region=’Catalonia’
          AND d3.month IN (‘January23’,’February23’)
GROUP BY GROUPING SETS ((d1.itemName, d2.region, d3.month),
                        (d1.itemName, d2.region),
                        (d2.region, d3.month)
                        (d2.region))
ORDER BY d1.itemName, d2.region, d3.month;
```

itemName	region	month	items
Ballpoint	Catalonia	January23	275827
Ballpoint	Catalonia	February23	290918
Ballpoint	Catalonia	NULL	566745
Rubber	Catalonia	January23	784172
Rubber	Catalonia	February23	918012
Rubber	Catalonia	NULL	1702184
NULL	Catalonia	January23	1059999
NULL	Catalonia	February23	1208930
NULL	Catalonia	NULL	2268929

https://www.postgresql.org/docs/current/queries-table-expressions.html#QUERIES-GROUPING-SETS

SQL-99: Meaning of null value

```
SELECT
    CASE WHEN GROUPING(d1.itemName)=1 THEN ‘AllBallpoint&Rubber’
        ELSE d1.itemName,
    d2.region,
    CASE WHEN GROUPING(d3.month)=1 THEN ‘AllJanuary&February’
        ELSE d3.month,
    SUM(f.items)
FROM Sales f, Product d1, Place d2, Time d3
…
```

itemName	region	month	items
Ballpoint	Catalonia	January23	275827
Ballpoint	Catalonia	February23	290918
Ballpoint	Catalonia	AllJanuary&February	566745
Rubber	Catalonia	January23	784172
Rubber	Catalonia	February23	918012
Rubber	Catalonia	AllJanuary&February	1702184
AllBallpoint&Rubber	Catalonia	January23	1059999
AllBallpoint&Rubber	Catalonia	February23	1208930
AllBallpoint&Rubber	Catalonia	AllJanuary&February	2268929

https://www.postgresql.org/docs/current/functions-aggregate.html#FUNCTIONS-GROUPING-TABLE

SQL-99: ROLLUP (I)

GROUP BY ROLLUP (a_1,..,a_n)

equivalent to

```
GROUP BY GROUPING SETS ( (a1,.., an-1,an),
                         (a1,..,an-1),
                              …
                         (a1),
                         ())
```

https://www.postgresql.org/docs/current/queries-table-expressions.html#QUERIES-GROUPING-SETS

SQL-99: ROLLUP (II)

```
SELECT d1.itemName, d2.region, d3.month, SUM(f.items)
FROM Sales f, Product d1, Place d2, Time d3
WHERE f.IDProduct=d1.ID AND f.IDPlace=d2.ID
      AND f.IDTime=d3.ID
      AND d1.itemName IN (‘Ballpoint’,’Rubber’)
      AND d2.region=’Catalonia’
      AND d3.month IN (‘January23’,’February23’)
GROUP BY ROLLUP(d2.region, d1.itemName, d3.month)
ORDER BY d2.region, d3.month, d1.itemName;
```

itemName	region	month	items
Ballpoint	Catalonia	January23	275827
Rubber	Catalonia	January23	784172
Ballpoint	Catalonia	February23	290918
Rubber	Catalonia	February23	918012
Ballpoint	Catalonia	NULL	566745
Rubber	Catalonia	NULL	1702184
NULL	Catalonia	NULL	2268929
NULL	NULL	NULL	2268929

https://www.postgresql.org/docs/current/queries-table-expressions.html#QUERIES-GROUPING-SETS

SQL-99: ROLLUP (III)

```
SELECT d1.itemName, d2.region, d3.month, SUM(f.items)
FROM Sales f, Product d1, Place d2, Time d3
WHERE f.IDProduct=d1.ID AND f.IDPlace=d2.ID
      AND f.IDTime=d3.ID
      AND d1.itemName IN (‘Ballpoint’,’Rubber’)
      AND d2.region=’Catalonia’
      AND d3.month IN (‘January23’,’February23’)
GROUP BY d2.region, ROLLUP(d1.itemName, d3.month)
ORDER BY d2.region, d3.month, d1.itemName;
```

itemName	region	Month	items
Ballpoint	Catalonia	January23	275827
Rubber	Catalonia	January23	784172
Ballpoint	Catalonia	February23	290918
Rubber	Catalonia	February23	918012
Ballpoint	Catalonia	NULL	566745
Rubber	Catalonia	NULL	1702184
NULL	Catalonia	NULL	2268929

https://www.postgresql.org/docs/current/queries-table-expressions.html#QUERIES-GROUPING-SETS

SQL-99: CUBE (I)

```
SELECT d1.itemName, d2.region, d3.month, SUM(f.items)
FROM Sales f, Product d1, Place d2, Time d3
WHERE f.IDProduct=d1.ID AND f.IDLugar=d2.ID
      AND f.IDTime=d3.ID
      AND d1.itemName IN (‘Ballpoint’,’Rubber’)
      AND d2.region=’Catalonia’
      AND d3.month IN (’January23’,’February23’)
GROUP BY d2.region, CUBE(d1.itemName, d3.month)
ORDER BY d1.itemName, d2.region, d3.month;
```

itemName	region	month	items
Ballpoint	Catalonia	January23	275827
Ballpoint	Catalonia	February23	290918
Ballpoint	Catalonia	NULL	566745
Rubber	Catalonia	January23	784172
Rubber	Catalonia	February23	918012
Rubber	Catalonia	NULL	1702184
NULL	Catalonia	January23	1059999
NULL	Catalonia	February23	1208930
NULL	Catalonia	NULL	2268929

https://www.postgresql.org/docs/current/queries-table-expressions.html#QUERIES-GROUPING-SETS

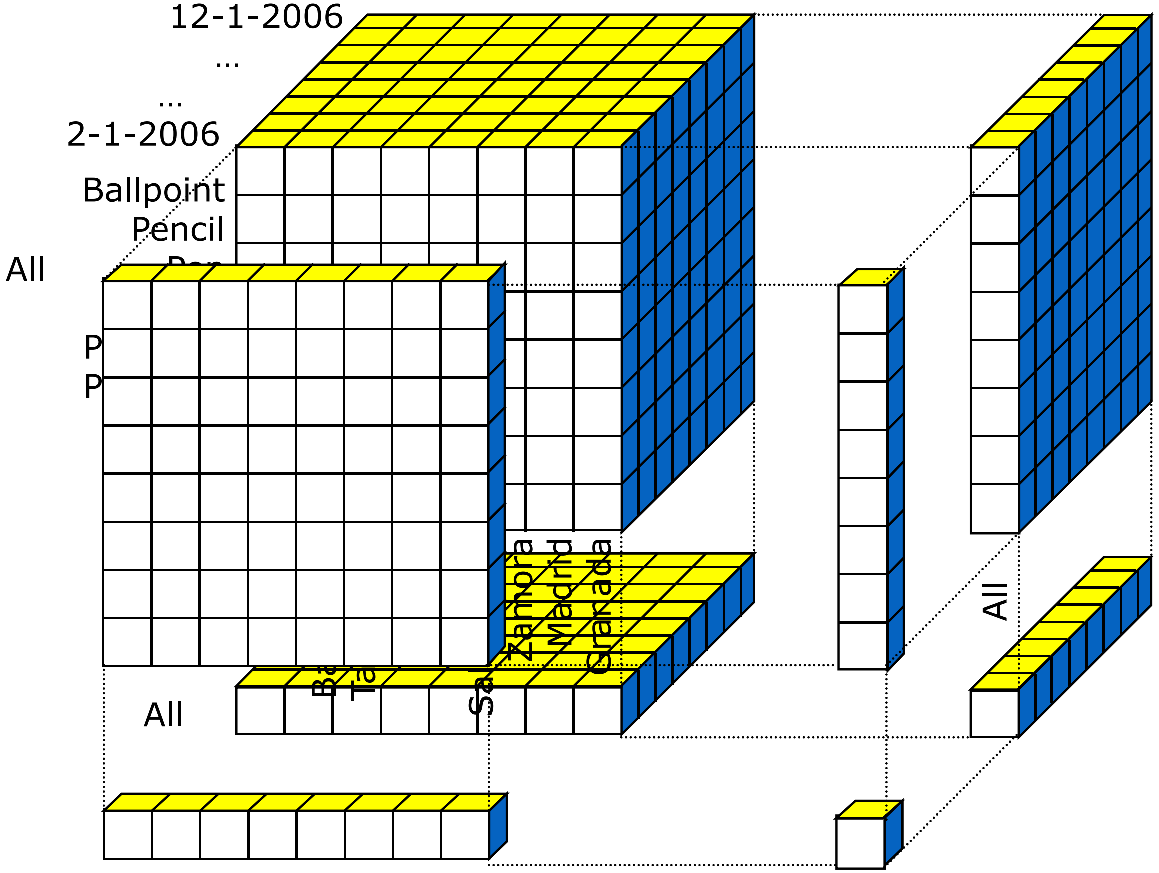
SQL-99: CUBE (II)

```
  GROUP BY CUBE (a,b)
equivalent to
  GROUP BY GROUPING SETS ( (a,b),
                           (a),
                           (b),
                           ());
```

```
  GROUP BY CUBE (a,b,c)
equivalent to
  GROUP BY GROUPING SETS ( (a,b,c),
                           (a,b),
                           (a,c),
                           (b,c),
                           (a),
                           (b),
                           (c),
                           ());
```

https://www.postgresql.org/docs/current/queries-table-expressions.html#QUERIES-GROUPING-SETS

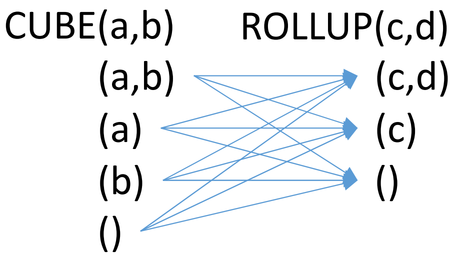
SQL-99: CUBE (III)



https://www.postgresql.org/docs/current/queries-table-expressions.html#QUERIES-GROUPING-SETS

SQL-99: Combinations

```
  GROUP BY CUBE(a,b), ROLLUP(c,d)
equivalent to
  GROUP BY GROUPING SETS ( (a,b,c,d),
                           (a,b,c),
                           (a,b),
                           (a,c,d),
                           (a,c),
                           (a),
                           (b,c,d),
                           (b,c),
                           (b),
                           (c,d),
                           (c),
                           ())
```



CUBE(a,b)

(a,b)

(a)

(b)

()

ROLLUP(c,d)

(c,d)

(c)

()

https://www.postgresql.org/docs/current/queries-table-expressions.html#QUERIES-GROUPING-SETS

Closing

Summary

- Comparison between decisional and operational environments
- Data Warehouse
  - Definition
  - Architectures
    - Single-layer
    - Two-layers
- OLAP definition
  - Cube
- Multidimensional schemas
  - Star
  - Snowflake
  - Galaxy or Constellation
- ROLAP
  - Star-join schema
- Cube-Query
  - SQL-99
    - GROUPING SETS
    - ROLLUP
    - CUBE

Bibliography

- E. F. Codd, S. B. Codd and C. T. Salley. *Providing OLAP to user-analysts: An IT mandate*. Technical report, E. F. Codd & Associates, 1993
- W. H. Inmon, C. Imhoff and R. Sousa. *Corporate Information Factory*. John Wiley & Sons, 1998
- R. Kimball, L. Reeves, M. Ross and W. Thornthwaite. *The Data Warehouse lifecycle toolkit*. John Wiley & Sons, 1998
- E. Malinowski and E. Zimányi. Advanced Data Warehouse Design. Springer, 2008
- W. H. Inmon, D. Strauss and G. Neushloss. *DW2.0*. Morgan Kaufmann, 2008
- A. Abelló and O. Romero. *On-Line Analytical Processing*. In Encyclopedia of Database Systems. Springer, 2009
- M. Golfarelli and S. Rizzi. *Data Warehouse Design*. McGrau-Hill, 2009
- BI-verdict. www.bi-verdict.com, June 2010
- A. Vaisman and E. Zimányi. Data Warehouse Systems. Springer, 2014