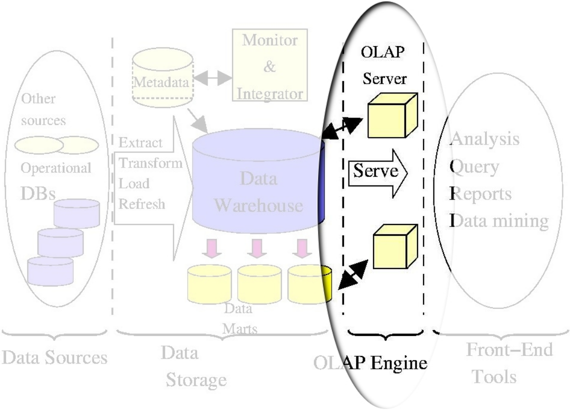
Datawarehouse and OLAP

OLAP

Syllabus, materials, notes, etc.

See http://www.info.univ-tours.fr/˜marcel/dw.html

On-Line Analytical Processing



today

OLAP models and languages

industry standards

formal models

logical models and query languages

no commonly agreed formal logical model and query language

- standards for data and metadata exchange
- query languages
  - SQL extensions
  - MDX: the de facto standard?
- many research works

SQL Extensions

SQL extensions

- Microsoft MDX
- ANSI SQL 99

microsoft's MDX

cf. SQL Server on-line doc
http://msdn.microsoft.com/en-us/library/ms145506.aspx
(10/2009)

typical instruction

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

syntax: select-from-where?

clause	parameter	gives
SELECT	1 relation per axis	axes of resulting cross-tab
FROM	1 cube name	the queried cells
WHERE	1 tuple	the queried slice

syntax: typical functions

navigation	PARENT	a member's parent
	CHILDREN	a member's children
	MEMBERS	a level's members or a dimension's members
structuration	CROSSJOIN	dimension nesting
ranking	TOPCOUNT	first members

example

SalesCube with six dimensions:

- SalesPerson
- Geography (Countries > Regions > States > Cities)
- Quarters (Quarters > Months > Days)
- Years
- Measures (Sales, PercentChange, and BudgetedSales)
- Products (Category > Product)

example

```
SELECT    CROSSJOIN({[Venkatrao], [Netz]},
          {[USA_North].CHILDREN, [USA_South], [Japan]})
          ON COLUMNS,
          {[Qtr1].CHILDREN, [Qtr2], [Qtr3], [Qtr4].CHILDREN}
          ON ROWS
FROM      [SalesCube]
WHERE     ([Sales], [1991], [Products].[All])
```

{ } delimit sets, [ ] delimit terms,

. identifies terms

example

Sales for 1991, All Products

		Venkatrao				Netz			
		USA			Japan	USA			Japan
		USA_North		USA_South		USA_North		USA_South	
		Seattle	Boston			Seattle	Boston		
Qtr1	Jan	00	10	20	30	40	50	60	70
	Feb	01	11	21	31	41	51	61	71
	Mar	02	12	22	32	42	52	62	72
Qtr2		03	13	23	33	43	53	63	73
Qtr3		04	14	24	34	44	54	64	74
Qtr4	Oct	05	15	25	35	45	55	65	75
	Nov	06	16	26	36	46	56	66	76
	Dec	07	17	27	37	47	57	67	77

language closure?

semantics

```
SELECT   CROSSJOIN({[Venkatrao], [Netz]},
         {[USA_North].CHILDREN, [USA_South], [Japan]})
         ON COLUMNS,
         {[Qtr1].CHILDREN, [Qtr2], [Qtr3], [Qtr4].CHILDREN}
         ON ROWS
FROM     [SalesCube]
WHERE    ([Sales], [1991], [Products].[All])
```

1. SELECT, WHERE : cross product of queries over the dimension tables to compute the desired references
2. FROM : semi join with the data cube of the fact table to obtain the measures' value

formally

if C is an n-dimensional cube with

- dimension tables $D_1, \ldots, D_n$
- and fact table F of schema $F[D_1, \ldots, D_n, val]$

an MDX query q with k axes is a tuple $\langle q_1, \ldots, q_k, q_{k+1} \rangle$ where

- $\forall i, i' \neq i \in [1, k+1], sort(q_i) \neq sort(q_{i'})$
- $\forall i \in [1, k], q_i$ is a relational query $q_i^1(D_i^1) \times \ldots \times q_i^x(D_i^{x_i})$ where $\{D_i^1, \ldots, D_i^{x_i}\} \subseteq \{D_1, \ldots, D_n\}$ are the dimensions appearing on axis i
- q_{k+1} is a conjunctive query yielding exactly one tuple $q_{k+1}^1(D_{k+1}^1) \times \ldots \times q_{k+1}^y(D_{k+1}^y)$ where $\{D_{k+1}^1, \ldots, D_{k+1}^y\} \subseteq \{D_1, \ldots, D_n\}$ are the dimensions appearing in the WHERE clause

formally

Let $\{D_a^1,\ldots,D_a^z\} \subseteq \{D_1,\ldots,D_n\}$ be the dimension tables not appearing in the MDX query, then $q_a^i(D_a^i)$ extracts the 'all' member of dimension D_a^i

an MDX query q with k axes on an n-dimensional cube corresponds to the set of cells

$$q_1^1(D_1^1) \times \ldots \times q_{k+1}^y(D_{k+1}^y) \times q_a^1(D_a^1) \times \ldots \times q_a^z(D_a^z) \rtimes (\gamma_{D_1;agg(val)}(F) \cup \gamma_{D_1,D_2;agg(val)}(F) \cup \ldots \cup F)$$

where the $\{D_1^1,\ldots,D_a^z\} = \{D_1,\ldots,D_n\}$, and γ is the grouping/aggregation operator

example

Suppose the following star schema for SalesCube:

- SalesPerson[all_salesperson,name]
- Geography[all_geography,countries,regions,states,cities]
- Quarters[all_quarters,quarters,months,days]
- Years[all_years,years]
- Measures[name]
- Products[all_products, category, product]

sales[SalesPerson,Geography,Quarters,Years,Measures,Product,value]

example

$[Geography].MEMBERS$ translates

select distinct all_geography from Geography union
select distinct countries from Geography union
select distinct regions from Geography union
select distinct states from Geography union
select distinct cities from Geography ;

example

$\{[Qtr1].CHILDREN,[Qtr2],[Qtr3],[Qtr4].CHILDREN\}$
translates

select distinct months from Quarters where quarters='Qtr1' union
select distinct quarters from Quarters where quarters='Qtr2' union
select distinct quarters from Quarters where quarters='Qtr3' union
select distinct months from Quarters where quarters='Qtr4';

example

```
CROSSJOIN({[Venkatrao], [Netz]},
{[USA_North].CHILDREN, [USA_South], [Japan]})
```

translates

```
select name, T.location
from SalesPerson, (
select distinct states as location from Geography
where regions='USA_North'
union
select distinct 'USA_South' as location from Geography
union
select distinct 'Japan' as location from Geography
) T
where name='Netz' or name='Venkatrao'
```

example

WHERE ([Sales], [1991], [Products].[All]) translates

select measure, years, all_product
from Measures, Years, Products
where name='Sales' and years=1991 and all_products='All'

example

```
SELECT [Qtr1].CHILDREN ON ROWS
FROM SalesCube
WHERE ([Sales],[1991])
```

translates into the star join query over the fact table sales

```
select years,months,name,sum(value)
from Years, Quarters, Measures, sales
where quarters='Qtr1' and years=1991 and name='Sales'
and Years.years=sales.Years and Quarters.days=sales.Quarters and
Measures.name=sales.Measures
group by years,months,name
```

example

```
SELECT [Qtr1].CHILDREN ON ROWS
FROM SalesCube
WHERE ([Sales],[1991])
```

suppose now the data cube is stored in a table of schema salesDataCube[SalesPerson,Geography,Quarters,Years,Measures,Product,value]

then the MDX query translates

select value from salesDataCube natural right outer join (select years as Years, months as Quarters, name as Measures from Years, Quarters, Measures where quarters='Qtr1' and years=1991 and name='Sales') S where SalesPerson='All' and Geography='All' and Product='All';

calculated members / ad-hoc aggregates

```
WITH MEMBER   [Measures].[Special Discount] AS
              [Measures].[Sales] * 0.75
SELECT        [Measures].[Special Discount] ON COLUMNS,
              [Products].[Product].MEMBERS ON ROWS
FROM          [SalesCube]
```

```
WITH MEMBER   [Product].[All Products].[Drink].[Avg Drinks]
AS            'AVG([Product].[All Products].[Drink].Children,
              [Measures].[Sales])'
SELECT        { [Product].[All Products].[Drink].Children,
              [Product].[All Products].[Drink].[Avg Drinks] } ON COLUMNS,
              [USA].Children ON ROWS
FROM          [SalesCube]
WHERE         [Measures].[Sales]
```

ANSI SQL-99

adds OLAP features to SQL-92 :

- GROUPING SETS: extends GROUP BY
- CUBE, ROLLUP: particular cases of GROUPING SETS
- ranking: extends ORDER BY
- windowing: moving averages or sums

supported by Oracle, IBM DB2, SAS, partially by MySQL...

examples: consider the facts

```
SELECT jour, ville, SUM(ventes)
FROM c1
GROUP BY jour,ville
```

c_1	jour	ville	ventes
	$jour_1$	$ville_1$	v_{11}
	$jour_1$	$ville_2$	v_{12}
	$jour_2$	$ville_1$	v_{21}
	$\vdots$	$\vdots$	$\vdots$
	$jour_q$	$ville_p$	v_{qp}

CUBE

computes the UNION of GROUP BY for every subset of the set of attributes

```
SELECT      jour, ville, SUM(ventes)
FROM        c1
GROUP BY    CUBE(jour,ville)
```

yields the groupings

{(jour,ville),(jour),(ville),$\emptyset$ }

CUBE

jour	ville	ventes
$jour_1$	$ville_1$	v_{11}
$jour_1$	$ville_2$	v_{12}
$jour_1$	NULL	v_{1_ALL}
$jour_2$	$ville_1$	v_{21}
$\vdots$	$\vdots$	$\vdots$
NULL	$ville_1$	v_{ALL_1}
$\vdots$	$\vdots$	$\vdots$
NULL	$ville_p$	v_{ALL_p}
NULL	NULL	v_{ALL_ALL}

ROLLUP

computes the UNION of GROUP BY of each prefix of the set of attributes

```
SELECT      jour, ville, SUM(ventes)
FROM        c1
GROUP BY    ROLLUP(jour,ville)
```

yields the groupings

$$\{\texttt{(jour,ville)},\texttt{(jour)},\emptyset\}$$

ROLLUP

	jour	ville	ventes
	$jour_1$	$ville_1$	v_{11}
	$jour_1$	$ville_2$	v_{12}
	$jour_1$	NULL	v_{1_ALL}
	$jour_2$	$ville_1$	v_{21}
	$\vdots$	$\vdots$	$\vdots$
	NULL	NULL	v_{ALL_ALL}

ROLLUP

```
SELECT      jour, ville, SUM(ventes)
FROM        c1
GROUP BY    ROLLUP(jour), ROLLUP(ville)
```

yields the groupings
$\{(jour),\emptyset\} \times \{(ville),\emptyset\} \equiv \{(jour,ville),(jour),(ville),\emptyset\}$

ROLLUP with hierarchy

```
SELECT     jour, mois, années, SUM(ventes)
FROM       c1, dimension_time
WHERE      c1.jour=dimension_time.jour
GROUP BY   ROLLUP(années,mois,jour)
```

computes the aggregates for all levels of the time dimension:
jour → mois → année

GROUPING SETS

consider the facts

c_1	jour	ville	pièce	ventes
	$jour_1$	$ville_1$	$pièce_1$	v_{111}
	$jour_1$	$ville_2$	$pièce_1$	v_{121}
	$jour_2$	$ville_1$	$pièce_2$	v_{212}
	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	$jour_q$	$ville_p$	$pièce_r$	v_{qpr}

GROUPING SETS

multiple GROUP BY precising the desired UNION

attribute nesting allows to separate simple GROUP BY from UNION of GROUP BY

CUBE and ROLLUP are particular cases of GROUPING SETS

GROUPING SETS

GROUP BY GROUPING SETS ((jour, ville, pièce))	≡	GROUP BY jour, ville, pièce
GROUP BY GROUPING SETS (jour, ville, pièce)	≡	GROUP BY jour UNION GROUP BY ville UNION GROUP BY pièce
GROUP BY GROUPING SETS (jour,(ville,pièce))	≡	GROUP BY jour UNION GROUP BY ville, pièce

ranking

ranks an ORDER BY result

```
SELECT jour,ville,rank() OVER (ORDER BY sum(ventes) DESC)
FROM c1
```

"top-n" query

```
SELECT jour,ville,rank() OVER (ORDER BY sum(ventes) DESC) as rang
FROM c1
ORDER BY rang
FETCH FIRST 5 ROWS ONLY
```

windowing

cumulative or moving aggregates

```
SELECT ville,jour,avg(ventes) OVER (ORDER BY ville, jour ROWS
BETWEEN 1 PRECEDING AND 1 FOLLOWING)
FROM c1
```

Standards and API

Standards and API

- CWM
- XMLA
- java API

Common Warehouse Metamodel

www.omg.org/cwm

- standard for BI and DW metadata exchange
- based upon
  - UML, Unified Modeling Language, an OMG modeling standard
  - MOF, Meta Object Facility, an OMG metamodeling and metadata repository standard
  - XMI, XML Metadata Interchange, an OMG metadata interchange standard
- supported by IBM, SAS, Oracle, Hyperion, Pentaho (mondrian), ...

XML for analysis

www.xmla.org

- specification for a set of XML message interfaces
  - data access interaction between a client application and an analytical data provider
  - over the Internet
- based upon XML, MDX
- supported by Microsoft, Hyperion, SAP, SAS, Pentaho, ...

the late JOLAP API

J2EE API

- proposed by IBM, Sun, Hyperion, Oracle, Nokia, SAS, ...
- final draft september 2003
- approval june 2004
- nothing since then

JOLAP was not properly implemented by any vendors and has been quietly forgotten. [...] we do not expect JOLAP to be resurrected.

OLAP report, 2007

olap4j API

www.olap4j.org

- common java API for any OLAP server
- JDBC for OLAP (extension to JDBC)
- based upon XMLA, MDX, AJAX
- open source project, supported by Pentaho, ...

Formal models and languages

OLAP query languages

4 examples

- Gyssens and Lakshmanan, VLDB'97 (algebra)
- Agrawal, Gupta, Sarawagi, ICDE'97 (algebra)
- Hacid, Marcel et Rigotti, DOOD'97 (datalog)
- Vassiliadis, Skiadopoulos, CAISE'00 (algebra)

introductory remark

few logical/physical optimisations

expressiveness poorly characterised

close to the relational model

Gyssens and Lakshmanan's algebra : intuitions

cube =

content

set of relations

+ structure

member/measure status

11 operators exploitant la séparation contenu/structure

data model

content of an n-dimensional cube c of dimensions $d_1, \ldots, d_n$

- a set of n relations, $r_{d_1}, \ldots, r_{d_n}$
  - each tuple with an unique id
  - each tuple corresponds to a member
- a relation r_m
  - has each key attribute of $r_{d_1}, \ldots, r_{d_n}$
  - has more attributes for the measures

schema

schema of a cube $< D,R,par >$

- $D = \{d_1,\ldots,d_n\}$ a set of dimensions
- $R = \{A_1,\ldots,A_m\}$ a set of d'attributes
- $par : D \rightarrow 2^{\{A_1,\ldots,A_m\}}$ with
  - $\forall i,j = 1,\ldots,n, i \neq j, par(d_i) \cap par(d_j) = \emptyset$
  - $\cup_{d \in D} par(d) \subseteq R$
- $M = R - \cup_{1 \leq i \leq n} par(d_i)$

instance

instance of an n-dimensional cube of schema $< D,R,par >$:
$rd_1(T_{id},par(d_1)),\ldots,rd_n(T_{id},par(d_n)),$
$r_m(rd_1.T_{id},\ldots,rd_n.T_{id},M)$

- $\pi_{T_{id}}(r_{d_1}) \times \ldots \times \pi_{T_{id}}(r_{d_n}) = \pi_{r_{d_1}.T_{id},\ldots,r_{d_n}.T_{id}}(r_m)$
- $\forall i = 1,\ldots,n, T_{id}$ is a primary key of r_{d_i}
- $\forall i,j = 1,\ldots,n, i \neq j, \pi_{T_{id}}(r_{d_i}) \cap \pi_{T_{id}}(r_{d_j}) = \emptyset$

correspondance

for a cube τ of schema $S =< D,R,par >$, $rep(\tau)$

- relational representation of τ
- a relation r of schema R

for a relation r of schema R, $tabs_S(r)$

- cube representation of r
- a cube of schema $S =< D,R,par >$

contents of the cube dimensions

produit

T_{id}	pièces
p1	écrous
p2	clous
p3	vis

lieu

T_{id}	régions
l1	est
l2	sud
l3	nord
l4	ouest

temps

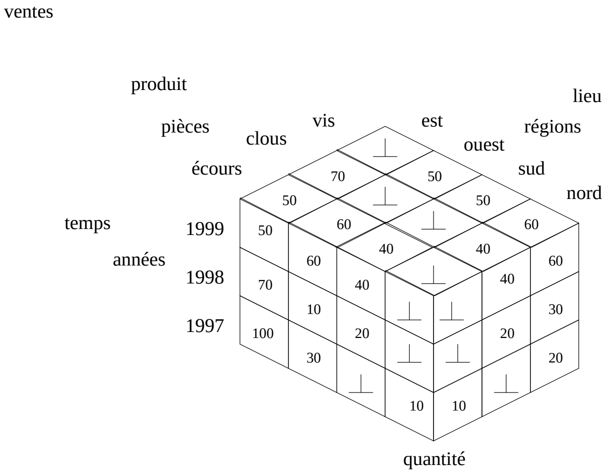
T_{id}	années
t1	1997
t2	1998
t3	1999

contents of the cube cells

r_m	$temps.T_{id}$	$produit.T_{id}$	$lieu.T_{id}$	quantité
	t1	p1	l1	60
	t1	p1	l2	40
	⋮	⋮	⋮	⋮
	t2	p2	l3	20
	⋮	⋮	⋮	⋮

structure

one possible representation of $r_{d_1}, \ldots, r_{d_n}, r_m$



set operations

7 operators for manipulating contents

τ_1 et τ_2 of schema $S_1 = < D_1, R_1, par_1 >$
τ of schema $S = < D, R, par >$
with $D_1 \cap D = \emptyset$ et $R_1 \cap R = \emptyset$

- unary operators (SPR) : $op(\tau) = tabs_S(op(rep(\tau)))$
- cross product (C) : $\tau_1 \times \tau = tabs_{S'}(rep(\tau_1) \times rep(\tau))$ with $S' = < D_1 \cup D, R_1 \cup R, par_1 \cup par >$
- binary (UID) : $\tau_1 op\ \tau_2 = tabs_{S_1}(rep(\tau_1)\ op\ rep(\tau_2))$

restructuration

2 operators on the cube's structure

τ a cube of schema $< D, R, par >$

- $unfold^d_X(\tau)$ is a cube of schema $< D \cup \{d\}, R, par' >$
  - $\forall d_i \in D, par'(d_i) = par(d_i)$
  - $par'(d) = X$
- $fold^d(\tau)$ is a cube of schema $< D - \{d\}, R, par' >$
  - $\forall d_i \in D - \{d\}, par'(d_i) = par(d_i)$

granularity

not part of the model
deals with only the contents

2 operators using external functions

- classification: for grouping tuples
- consolidation: for aggregating tuples

classification

τ a cube of schema $< D,R,par >$

$K(\tau,f) = tab_{S'}(K(rep(\tau),f))$ where

- $\{A_1,\ldots,A_k\} \subset R$
- $f : dom(f.A_1) \times \ldots \times dom(f.A_k) \rightarrow 2^{dom(A_1)\times\ldots\times dom(A_k)}$
- $S' =< D,R \cup \{f.A_1,\ldots,f.A_k\},par' >$
- $f.A_i \in par'(d) \iff A_i \in par(d)$

classification

$K(r,f)$ is a relation of schema

$(f.A_1,\ldots,f.A_k,A_1,\ldots,A_k,\ldots A_m)$

and of instance

$\{(a_1,\ldots a_k,a'_1,\ldots a'_k,\ldots a'_m)|$
$(a'_1,\ldots,a'_k) \in f(a_1,\ldots a_k) \wedge (a'_1,\ldots,a'_k,\ldots a'_m) \in r\}$

consolidation

τ is a cube of schema $< D,R,par >$

$A(\tau,g) = tab_{S'}(A(rep(\tau),g))$ where

- $\{A_1,\ldots,A_k\} \subset R = \{A_1,\ldots,A_m\}$
- B a new attribute $B \notin \{A_1,\ldots,A_k\}$
- $A_j, k+1 \leq j \leq m$ of same type than B
- $S' =< D,\{A_1,\ldots,A_k,B\},par >$

consolidation

$$g_{A_j \rightarrow B} : 2^{dom(A_{k+1}) \times \ldots \times dom(A_m)} \longrightarrow dom(B)$$

$A(r,g)$ is a relation of schema $\{A_1, \ldots, A_k, B\}$

and of instance

$$\{(a_1, \ldots, a_k, b) | b = g(\{(a_{k+1}, \ldots, a_m) | (a_1, \ldots, a_k, a_{k+1}, \ldots, a_m) \in r\})\}$$

example of a typical query formulation

$$unfold^{lignes}_{années,produits}(\; fold^{produits}(\; fold^{années}(\; \pi_{années,produits,quantité}(sud))))$$ where sud is defined by

$$\sigma_{régions=sud}(\; \pi_{années,régions,produits,quantité}(\; \sigma_{années=a+1 \wedge quantité>q \wedge régions=r \wedge produits=p}(\; \rho_{quantités,régions,produits,années \rightarrow q,r,p,a}(ventes) \times ventes)))$$

example of a typical query formulation

$A(K(ventes, f_{gliss}), g_{quantit\acute{e} \to moyenne})$, with

$$f_{gliss}(x) = \{x' \mid x' = x \vee x' = x + 1\}$$

$$g_{quantit\acute{e} \to moyenne}(S) = (1/ \mid S \mid) \sum_{(w,x,y,z) \in S}(z)$$

algebra of Agrawal, Gupta, Sarawagi

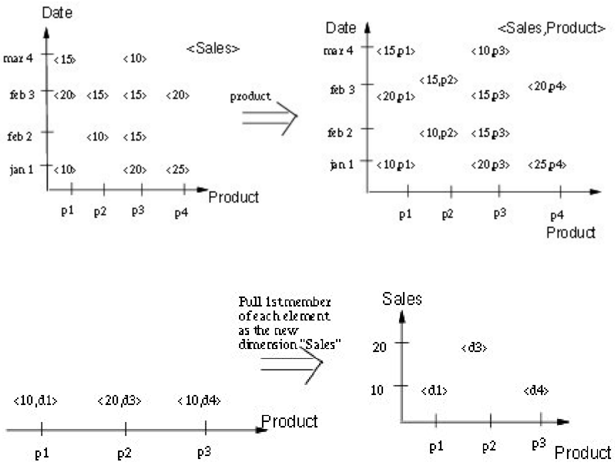
model: an n-dimensionsal cube C

- $D_{i,i \in [1,n]}$ dimensions of domain dom_{D_i}
- f a function from $dom_{D_1} \times \ldots \times dom_{D_n}$ to
    - 1
    - 0
    - a tuple with arity $< n$

operators

- push, pull
- projection, selection, join
- aggregation

push and pull



push

consider $C_1 = (D_1, \ldots, D_n, f_1)$ and $C_2 = (D_1, \ldots, D_n, f_2)$

$push(C_1, D_i) = C_2$ with

$f_2(d_1, \ldots, d_n) =$

- 0 if $f_1(d_1, \ldots, d_n) = 0$
- (d_i) if $f_1(d_1, \ldots, d_n) = 1$
- the tuple $(t_1, \ldots, t_m, d_i)$ if $f_1(d_1, \ldots, d_n) = (t_1, \ldots, t_m)$

pull

consider $C_1 = (D_1, \ldots, D_{n-1}, f_1)$ and $C_2 = (D_1, \ldots, D_n, f_2)$

the measures of C_1 cannot be 0 or 1

$pull(C_1, D_n, i) = C_2$ with

$f_2(d_1, \ldots, d_n) =$

- $(m_1, \ldots, m_{i-1}, m_{i+1}, \ldots, m_k)$ si $f_1(d_1, \ldots, d_{n-1}) = (m_1, \ldots, m_i, \ldots, m_k)$
- 0 otherwise

projection

consider $C_1 = (D_1, \ldots, D_n, f_1)$ with $|dom_{D_i}| = 1$

$remove(C_1, D_i) = C_2$ with

- $C_2 = (D_1, \ldots, D_{i-1}, D_{i+1}, \ldots, D_n, f_2)$
- $f_2(d_1, \ldots, d_{i-1}, d_{i+1}, \ldots, d_n) = f_1(d_1, \ldots, d_{i-1}, d_i, d_{i+1}, \ldots, d_n)$

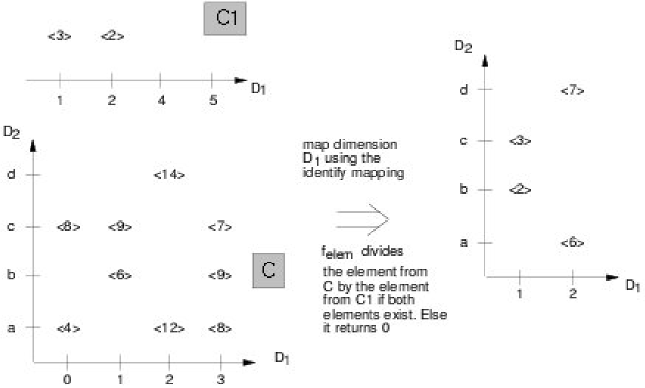
selection

consider $C_1 = (D_1, \ldots, D_n, f_1)$

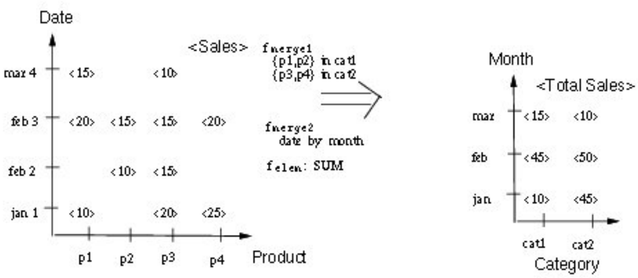
$\sigma_{D_i,P}(C_1) = C_2$ with

- $C_2 = (D_1, \ldots, D_n, f_2)$
- $dom'_{D_i} = P(dom_{D_i})$
- f_2 is the restriction of f_1 to $dom_{D_1} \times \ldots \times dom'_{D_i} \times \ldots \times dom_{D_n}$

join


C1
<3>
<2>
1
2
4
5
D1
D2
d
c
b
a
<14>
<8>
<9>
<7>
<6>
<9>
<4>
<12>
<8>
0
1
2
3
D1
C
map dimension D1 using the identify mapping
felem divides the element from C by the element from C1 if both elements exist. Else it returns 0
D2
d
<7>
c
<3>
b
<2>
a
<6>
1
2
D1

agrgegation



datalog : intuitions

une référence de cellule = un atome datalog

monovaluation	dépendances fonctionnelles
granularité	décrite par une relation spécifique (extension + intention)
flexibilité du schema	syntaxe d'ordre supérieur

modèle de données

nom atomique	écrous
nom composé	écrous . 1999
référence	ventes(écrous,1999,ouest)
cellule	ventes(écrous,1999,ouest) :⟨50⟩
cube	{ventes(écrous,1999,ouest) :⟨50⟩,
	⋮
	ventes(clous,1998,nord) :⟨20⟩}

monovaluation

$$\{\ldots,\ \text{ventes(écrous,1999,ouest)} :\langle 50\rangle,$$
$$\ldots,\ \text{ventes(écrous,1999,ouest)} :\langle 70\rangle,\ \ldots\}$$

est interdit, donc

$$a(b,c) : d \leftarrow \ .$$
$$a(b,c) : e \leftarrow \ .$$

est interdit aussi

termes du langage

$\mathcal{D}$ ensemble de noms atomiques, $\mathcal{V}$ ensemble de variables

$$
\begin{array}{lll}
atomicName & := & c \in \mathcal{D} \mid v \in \mathcal{V} \\
name & := & atomicName \mid name.name \\
contents & := & \langle name, \ldots, name \rangle \\
reference & := & name(name, \ldots, name) \\
cellAtom & := & reference : contents \\
atom & := & cellAtom \mid groupingAtom \\
literal & := & atom \mid aggregateSubgoal \\
body & := & literal, \ldots, literal \mid \epsilon \\
head & := & atom \\
rule & := & head \leftarrow body
\end{array}
$$

termes du langage : granularité

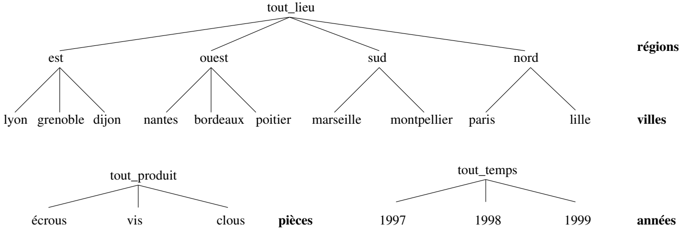
$\mathcal{AGG}$ ensemble d'opérateurs d'agrégat

$$\begin{array}{lcl} groupingAtom & := & in(atomicName, atomicName) \\ aggregateSubgoal & := & atomicName = f(reference) \\ literal & := & atom \mid aggregateSubgoal \end{array}$$

où $f \in \mathcal{AGG}$.

termes du langage : granularité

description des groupements
{in(vis,tout_produit), in(1997,tout_temps), ... in(lyon,est), in(est,tout_lieu)}



restructurations

split

$$\text{ventes.R(P,A)} : \langle Q \rangle \leftarrow \text{ventes(P,A,R)} : \langle Q \rangle .$$

restructurations

$$\text{ventes.R(P,A)} :\langle \text{Q} \rangle \xleftarrow{\textit{split}} \text{ventes(P,A,R)} :\langle \text{Q} \rangle.$$

$$\text{ventesNest(P.R, A)} :\langle \text{Q} \rangle \xleftarrow{\textit{nest}} \text{ventes(P,A,R)} :\langle \text{Q} \rangle.$$

restructurations

split

$$\text{ventes.R(P,A)} : \langle \text{Q} \rangle \leftarrow \text{ventes(P,A,R)} : \langle \text{Q} \rangle .$$

nest

$$\text{ventesNest(P.R, A)} : \langle \text{Q} \rangle \leftarrow \text{ventes(P,A,R)} : \langle \text{Q} \rangle .$$

push

$$\text{ventesPush(P,R)}:\langle A_1,Q_1,A_2,Q_2,A_3,Q_3 \rangle \leftarrow \text{ventes}(P,A_1,R):\langle Q_1 \rangle,$$

$$\text{ventes}(P,A_2,R):\langle Q_2 \rangle,$$

$$\text{ventes}(P,A_3,R):\langle Q_3 \rangle,$$

$$A_1 < A_2,$$

$$A_2 < A_3.$$

roll-up

$$20 = \text{sum}(\text{ventes}(\text{vis}, \text{tout\_temps}, \text{est}))$$

$$\{\text{ventes}(\text{vis}, 1998, \text{est}) : \langle 10 \rangle,\ \text{ventes}(\text{vis}, 1997, \text{est}) : \langle 10 \rangle\}$$

roll-up

$$20 = \text{sum}(\text{ventes}(\text{vis}, \text{tout\_temps}, \text{est}))$$

$$\{\text{ventes}(\text{vis}, 1998, \text{est}) : \langle 10 \rangle, \ \text{ventes}(\text{vis}, 1997, \text{est}) : \langle 10 \rangle\}$$

roll-up sur la dimension temps

$$\begin{aligned} \text{ventes}(P, \text{tout\_temps}, R) : \langle T \rangle] \leftarrow \ & T = \text{sum}(\text{ventes}(P, \text{tout\_temps}, R)), \\ & \text{in}(P, \text{tout\_produit}), \\ & \text{in}(R, \text{tout\_lieu}). \end{aligned}$$

ordre des références

J une interpretation à partir d'un ensemble de faits initial I.

$\forall x,y \in \mathcal{D}, in_J(x,y) \iff in(x,y) \in J$

$\forall rf = n(n_1,\ldots,n_p), rf' = n(n'_1,\ldots,n'_p),\ rf <_J rf' \iff$

- $rf \neq rf'$
- $\forall i \in [1,\ldots,p]$,
  - $in_J(n_i,n'_i)$
  - ou $n_i = n'_i$

sémantique des sous-buts d'agrégats

$B = k = f(n(n_1, \ldots, n_p))$ un sous-but d'agrégat

$$detailRef_I^J(B) = \{ref(A) \mid A \in I, ref(A) <_J ref(B)\}$$
$$detailCont_I^J(B) = \{\!\{ k \mid A \in I, ref(A) \in detailRef_I^J(B), k = val(A)\}\!\}$$

B satisfait si

- $detailRef_I^J(B) \neq \emptyset$
- $f(detailCont_I^J(B))$ est défini
- $f(detailCont_I^J(B)) = k$

caractéristiques

cadre formel proche du cadre datalog classique

- sémantique déclarative (type théorie des modèles)
- sémantique opérationnelle (type pppf) équivalente

progammes traduisibles en datalog

example de spécification de requête typique

$\text{R}(\text{A}_2.\text{P},\text{ventes}){:}\langle \text{Q}_2 \rangle \leftarrow \text{ventes}(\text{P},\text{A}_2,\text{R}){:}\langle \text{Q}_2 \rangle,$

$\text{ventes}(\text{P},\text{A}_1,\text{R}){:}\langle \text{Q}_1 \rangle,$

$\text{Q}_2 > \text{Q}_1,$

$\text{A}_2 \text{ is } \text{A}_1 + 1.$

example de spécification de requête typique

$$\text{resultat(A.A',P,R)}:\langle \text{M} \rangle \leftarrow \text{ventes(P,A,R)}:\langle \text{S1} \rangle,$$

$$\text{A' is A} + 1,$$

$$\text{ventes(P,A',R)}:\langle \text{S2} \rangle,$$

$$\text{M is (S1} + \text{S2)}/2.$$

example de spécification de requête typique

$\text{ventes}(P,A_2,R):\langle Q_2\rangle \leftarrow \text{ventes}(P,A_1,R):\langle Q_1\rangle$,

A_2 is $A_1 + 1$,

Q_2 is $Q_1 + Q_1/10$,

$A_2 \leq 2002$,

$A_1 \geq 1999$.

{ventes(clous,2000,est):⟨77⟩, ventes(clous,2000,nord):⟨44⟩, ventes(vis,2000,sud):⟨55⟩, ventes(clous,2001,est):⟨84.7⟩, ventes(clous,2001,nord):⟨48.4⟩, . . . }

Groupements ad-hoc

responsables	est	ouest	sud	nord
vis	john	mike	bob	mike
clous	bob	john	bob	mike
écrous	john	bob	john	bob

in(P,N) $\leftarrow$ responsables(P,R) :$\langle N \rangle$.
in(R,N) $\leftarrow$ responsables(P,R) :$\langle N \rangle$.

Groupements ad-hoc

$$\text{ventesResp(R,P,A,N)} : \langle Q \rangle \leftarrow \text{ventes(P,A,R)}:\langle Q \rangle, \text{responsables(P,R)} : \langle N \rangle.$$

$$\text{résultat99(total,N)} : \langle S \rangle \leftarrow S = \text{sum(ventesResp(N,N,1999,N))}, \text{responsables(\_,\_)} : \langle N \rangle.$$

résultat99	john	bob	mike
total	140	120	150

Opérateur “data cube”

in(jour_1,mois_1), in(jour_2,mois_1), ... in(mois_1,année_1),. . . ,
in(vendeur_1,ville_1),. . . , in(ville_1,pays_1),. . .

niveau(X) $\leftarrow$ in(X,Y)
niveau(Y) $\leftarrow$ in(X,Y)

c(T,C):$\langle$S$\rangle$ $\leftarrow$ S=sum(c(T,C)),niveau(T),niveau(C).

Vassiliadis and Skiadopoulos's algebra

- extends former proposals
- a cube is a view over an underlying data set
- a dimension is a lattice
- the history of performed selections is kept for subsequent optimisations

model

data model

- data sets
- dimensions
- cubes

algebraic operators

- navigation (roll-up and drill-down)
- selection
- split measure (projection for measures)

data set

Day	Title	Salesman	Store	Sales
6-Feb-97	Symposium	Netz	Paris	7
18-Feb-97	Karamazof brothers	Netz	Seattle	5
11-May-97	Ace of Spades	Netz	LosAngeles	20
3-Sep-97	Zarathustra	Netz	Nagasaki	50
3-Sep-97	Report to El Greco	Netz	Nagasaki	30
1-Jul-97	Ace of Spades	Venk	Athens	13
1-Jul-97	Piece of Mind	Venk	Athens	34
⋮				

global data source *DS*

dimensions and hierarchies

a dimension D is a lattice $D = (L, <)$ with

- L a set of levels $L = (L_1, \ldots, L_n, ALL)$
- $<$ a partial order over L such that, $\forall i \in [1,n], L_1 < L_i < ALL$
- a family of functions $anc_{L_1}^{L_2}$ that associates each element of $dom(L_1)$ with one element of $dom(L_2)$

example

dimension $Time = (L, <)$ with

- $L = \{$Day,Month,Year,ALL $\}$
- Day $<$ Month $<$ Year $<$ ALL
- $dom(Day) = \{18\ Feb\ 97,...\}$
- $dom(Month) = \{Feb\ 97,...\}$
- $dom(Year) = \{97,...\}$
- $dom(ALL) = \{all\}$
- anc_{Day}^{Month}, anc_{Month}^{Year}, anc_{Year}^{ALL},

and for example: $anc_{Day}^{Month}(18\ Feb\ 97) = Feb\ 97$, $anc_{Month}^{Year}(Feb\ 97) = 97$, $anc_{Year}^{ALL}(97) = all$

cube

a cube c of schema $[L_1, \ldots, L_n, M_1, \ldots, M_m]$ is
$(DS_0, \varphi, [L_1, \ldots, L_n, M_1, \ldots, M_m], [agg_1(M_1^0), \ldots, agg_m(M_m^0)])$

where

- DS_0 is a source data set of schema $[L_1^0, \ldots, L_n^0, M_1^0, \ldots, M_k^0], k > m$
- φ is a detailed selection condition
- the L_i are levels
- $M_1, \ldots, M_m$ are aggregated measures
- les agg_i are aggregated functions

example

let $sales\_97\_by\_Store$ be a cube of schema $[Year, ALL, ALL, Store, result]$

$(DS, Year = 97, [Year, ALL, ALL, Store, result], [sum(Sales)])$

with

- DS is the data source of schema $[Day, Title, Salesman, Store, Sales]$
- $Year = 97$ is a selection condition on DS
- $result$ corresponds to $sum(Sales)$

algebra

let c^a be the initial cube

$$c^a = (DS_0, \varphi^a, [L_1^a, \ldots, L_n^a, M_1^a, \ldots, M_m^a], [agg_1^a(M_1^0), \ldots, agg_m^a(M_m^0)])$$

navigation

$$navi(c^a, [L_1, \ldots, L_n, M_1, \ldots, M_m], agg_1, \ldots, agg_m) =$$
$$(DS_0, \varphi^a, [L_1, \ldots, L_n, M_1, \ldots, M_m], [agg_1(M_1^0), \ldots, agg_m(M_m^0)])$$

selection

$$\sigma_\varphi(c^a) =$$
$$(DS_0, \varphi^a \wedge \varphi, [L_1^a, \ldots, L_n^a, M_1^a, \ldots, M_m^a], [agg_1^a(M_1^0), \ldots, agg_m^a(M_m^0)])$$

split measure

$$\pi_{M_m}(c^a) =$$
$$(DS_0, \varphi^a, [L_1^a, \ldots, L_n^a, M_1^a, \ldots, M_{m-1}^a], [agg_1^a(M_1^0), \ldots, agg_m^a(M_{m-1}^0)])$$

example

let $sales\_97\_by\_store =$
$(DS, Year = 97, [Year, ALL, ALL, Store, result], [sum(Sales)])$

example

let $sales\_97\_by\_store =$
$(DS, Year = 97,[Year,ALL,ALL,Store,result],[sum(Sales)])$

$navi(sales\_97\_by\_store,[Year,ALL,ALL,ALL,result\_year],sum) =$

$(DS, Year = 97,[Year,ALL,ALL,ALL,result\_year],[sum(Sales)])$

example

let $sales\_97\_by\_store =$
$(DS, Year = 97, [Year, ALL, ALL, Store, result], [sum(Sales)])$

$navi(sales\_97\_by\_store, [Year, ALL, ALL, ALL, result\_year], sum) =$

$(DS, Year = 97, [Year, ALL, ALL, ALL, result\_year], [sum(Sales)])$

$\sigma_{Store=Paris}(sales\_97\_by\_store) =$

$(DS, Year = 97 \wedge Store = Paris, [Year, ALL, ALL, Store, result], [sum(Sales)])$

computation of a cube

1. apply the selection condition on source data
   each occurrence of L in φ is replaced with $anc_{L^0}^{L}(L^0)$
2. replace the values of the levels for the tuples of the result with their respective ancestor values at the levels of the schema of the cube
3. group them into a single value for each measure, through the application of the appropriate aggregate function

OLAP analysis

the cube c_2 is obtained from the cube c_1

- can c_2 be computed from c_1 ?
- can c_1's tuples be used?

variation of the view subsumption problem

are there new problems due to hierarchies?

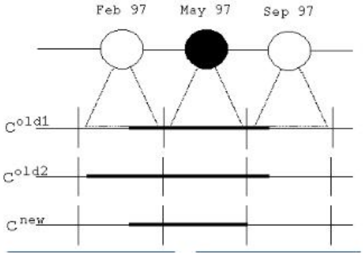
OLAP analysis

c_2 must

- be defined on the same dimensions as c_1, at a greater or equal level
- be defined on the same measures of DS_0 and with the same aggregation functions
- have a more restrictive selection condition over the same tuples as DS_0

example

$$c^i = [DS, \varphi_i, [Month, ALL, ALL, ALL, Sales], sum(sales)]$$



$$\varphi_{old1} = 18\text{-}Feb\text{-}97 \leq Day \leq 3\text{-}Sep\text{-}97 \wedge Salesman = Netz$$
$$\varphi_{old2} = 6\text{-}Feb\text{-}97 \leq Day \leq 3\text{-}Sep\text{-}97 \wedge Salesman = Netz$$
$$\varphi_{new} = 18\text{-}Feb\text{-}97 \leq Day \leq 31\text{-}May\text{-}97 \wedge Salesman = Netz$$

test of the selection condition

partition the values w.r.t. c_2's schema

for each partition

- test if there exists a partition for c_1
- if yes
  - test if the 2 partitions comply

before concluding...

note that we have only talk about query languages

now, what is an OLAP analysis?

short answer: a sequence of OLAP queries

is there more?

conclusion

So far: we know quite a lot now

Next: what's next?